FLINT 2001–
NEW DIRECTIONS IN ENHANCING THE POWER OF INTERNET

PROCEEDINGS OF THE 2001 BISC INTERNATIONAL WORKSHOP ON FUZZY LOGIC AND THE INTERNET

by

Masoud Nikravesh and Ben Azvine,
BISC–BTexact Technology

Memorandum No. UCB/ERL M01/28

8 August 2001

FLINT 2001–
NEW DIRECTIONS IN ENHANCING THE POWER OF INTERNET

PROCEEDINGS OF THE 2001 BISC INTERNATIONAL WORKSHOP ON FUZZY LOGIC AND THE INTERNET

by

Masoud Nikravesh and Ben Azvine,
BISC–BTexact Technology

Memorandum No. UCB/ERL M01/28

8 August 2001

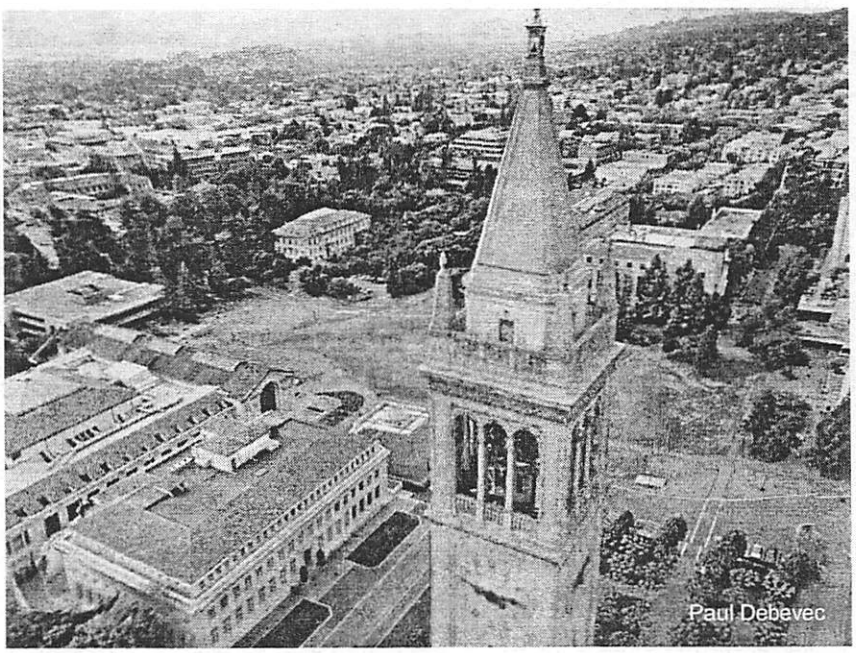
ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

FLINT 2001

NEW DIRECTIONS IN ENHANCING THE POWER OF THE INTERNET

August 14-17, 2001
Hewlett-Packard Auditorium (306 Soda Hall)
Computer Science Division
Electrical and Computer Sciences Department
University of California-Berkeley



Cal BTexact TECHNOLOGIES

Sponsored by BISC Program, UC Berkeley ILP, and BTexact Technologies

Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720-1776
Tel: (510) 643-4522 and 4519, FAX: 510-642-5775
Email: zadeh@cs.berkeley.edu
Email: nikravesh@cs.berkeley.edu
URL: http://www-bisc.cs.berkeley.edu/

Sponsored by BISC Program, UC Berkeley ILP, and BTexact Technologies

Foreword

The 2001 BISC International Workshop on issues related to the Internet will be held in the Computer Sciences Division of the University of California, Berkeley, from August 14-18. UC Berkeley is a preeminent research institution in the proximity of Silicon Valley. At the dawn of the new millennium, we can expect dramatic increase in the use of intelligent systems in the Internet applications, since we have to deal with an increasing amount of data, that is mainly unstructured and designed for human access. Therefore, it is usually hard to extract relevant information automatically. These aspects will be reflected in the subjects treated at this 2001 BISC International Workshop. The main purpose of the Workshop is to draw the attention of the research community as well as the Internet community to the fundamental importance of specific Internet-related problems. This issue is critically significant about problems that center on search and deduction in large, unstructured knowledge bases. The BISC Special Interest Group on Fuzzy Logic and the Internet (BISC-SIG-FLINT) invites researchers from around the world who are interested in advancing the frontiers of the Internet by use of intelligent techniques and soft computing methods. The Workshop will provide a unique opportunity for the academic and corporate communities to address new challenges, share solutions, and discuss research directions for the future.

FLINT ORGANIZERS

BISC Program – U.C. Berkeley

Letter from Lotfi A. Zadeh

To: Members of the BISC Group
From: L. A. Zadeh

Subject: Fuzzy Logic as the "Brainware" of the Internet

Unlike classical logic, fuzzy logic is concerned, in the main, with modes of reasoning which are approximate rather than exact. In Internet, almost everything, especially in the realm of search, is approximate in nature. Putting these two facts together, an intriguing thought merges; in time, fuzzy logic may replace classical logic as what may be called the brain-ware of the Internet. Actually, to a far greater extent than many of us, including myself had realized, fuzzy logic is already in use in search engines. To get an idea, try the entry +fuzzy + search +engine (+ is conjunction) in excite. But this may be just beginning of a steep ascent in the use of fuzzy logic in the Internet. In this connection, Eli Sanchez brought to my attention the following quote from Tim Barnes-Lee (Father of WWW); "... Toleration of inconsistency can only be done by fuzzy systems. We need a semantic web which will provide guarantees, and about which one can reason with logic. A fuzzy system might be good for finding a proof ..."

In my view, among the many ways in which fuzzy logic may be employed, there are two that stand out in importance. The first is search. Another, and less obvious, is deduction in an unstructured and imprecise environment.

Existing search engines have zero deductive capability. To convince yourself that this is the case, try the query "How many computer science graduates were produced by European universities in 1999?" To add a deductive capability to a search engine, the use of fuzzy logic is not an option - it is a necessity.

To promote the use of fuzzy Logic in the Internet, a Special Interest Group on Fuzzy Logic in the Internet, with Dr. Masoud Nikravesh as Chair, has been formed. Please contact Dr. Nikravesh (nikravesh@cs.berkeley.edu) if you are interested in joining this SIG or posting your views and comments to the BISC Group.

Warm regards to all,

Lotfi Zadeh
December 2000

Letter from Masoud Nikravesh

Dear BISC Members,

One of the problems that Internet users are facing today is to find the desired information correctly and effectively in an environment that the available information, the repositories of information, indexing, and tools are all dynamic. Even though some tools were developed for a dynamic environment, they are suffering from "too much" or " too little" information retrieval. Some tools return too few resources and some tool returns too many resources.

For example, World Wide Web search engines have become the most heavily- used online services, with millions of searches performed each day. Their popularity is due, in part, to their ease of use. While searches may retrieve thousands of hits, finding relevant partial matches might be a problem. Already explosive amount of users on the Internet is estimated over 200 million. The estimated user of wireless devices is estimated 1 billion within 2003 and 95 % of all wireless devices will be Internet enabled within 2005.

The main goals of the Interest group are:

1. Carry out R & D in Soft computing and Internet
2. Provide forum for debate and discussion on Soft computing and Internet
3. Organize and contribute to conferences and short courses on Soft computing and Internet
4. Provide repository and archive for Soft computing and Internet related R & D
5. Attract soft computing and Internet researchers worldwide to collaborate with the interest group
6. Help technology transfer and commercialization of Soft computing and Internet applications
7. Create a think-tank to analyze impact of Internet on society

The focus of BISC Special Interest group on Fuzzy Logic and the Internet will be and are not limited to:

1. Search Engines and Web Crawlers
2. Agent Technology (i.e., Web-Based Collaborative and Distributed Agents)
3. Adaptive and Evolutionary techniques for dynamic environment (i.e. Evolutionary search engine and text retrieval, Dynamic learning and adaptation of the Web Databases, etc)
4. Fuzzy Queries in Multimedia Database Systems
5. Query Based on User Profile
6. Information Retrievals
7. Summary of Documents
8. Information Fusion Such as Medical Records, Research Papers, News, etc
9. File and Folder Organizer
10. Data Management for Mobile Applications and eBusiness Mobile Solutions over the Web

11. Matching People, Interests, Products, etc
12. Association Rule Mining for Terms-Documents and Text Mining
13. E-mail Notification
14. Web-Based Calendar Manager
15. Web-Based Telephony
16. Web-Based Call Center
17. Workgroup Messages
18. E-Mail and Web-Mail
19. Web-Based Personal Info
20. Internet related issues such as Information overload and load balancing, Wireless Internet, E-coding and D-coding (Encryption), Security such as Web security and Wireless/Embedded Web Security, Web-based Fraud detection and prediction, Recognition, issues related to E-commerce and E-bussiness, etc.

Best Regards,
Masoud Nikravesh

Berkeley Initiative in Soft Computing (BISC)
CS Division, Department of EECS
University of California,
Berkeley, California
Email: nikravesh@eecs.berkeley.edu
URL: http://www.cs.berkeley.edu/~nikraves/
Tel: (510) 643-4522
Fax: (510) 642-5775

Organizing Committee

Honorary Chairs: Lotfi A. Zadeh (USA)

General Chairs: Nader Azarmi (UK)
Masoud Nikravesh (USA)
Ronald R. Yager (USA)

Local Chairs: Mori Anvari (BISC-UC-Berkeley, USA)
Dae-Young Choi (BISC-UC-Berkeley, USA)
Marcin Detyniecki (BISC-UC-Berkeley, USA)
Virginia Johnson (BISC-LLNL, USA)
Andreas Nürnberger (BISC-UC-Berkeley, USA)

Program Committee: H. Adeli (USA)
Ben Azvine (UK)
Hamid Berenji (USA)
Michael R. Berthold (USA)
Antonio Di Nola (Italy)
Ronald Fagin (USA)
Marti Hearst (USA)
Mohammad Jamshidi (USA)
Janusz Kacprzyk (Poland)
T. Y. Lin (USA)
Vincenzo Loia (Italy)
John A. Meech (Canada)
Bouchon-Meunier (France)
Detlef Nauck (UK)
Charles Ortiz (USA)
Gabriella Pasi (Italy)
Fred Petry (USA)
James Shanahan (France)
Vincent S.M. Tseng (Taiwan)
I. Burhan Türksen (Canada)
John Yen (USA)

International Advisory Committee

H. Adeli
Klaus-Peter Adlassnig
Rafik Aliev
Fuad Aliew
Matthew Anderson
Mori Anvari
Nader Azarmi
Ben Azvine
James Baldwin
Hamid Berenji
Michael Berthold
Piero Bonissone
Patrick Bosc
Bernadette Bouchon-Meunier
Rita de Caluwe
Peter Chen
Tony Cowden
Fabio Crestani
Ernesto Damiani
Miguel Delgado
Marcin Detyniecki
Didier Dubois
Ronald Fagin
Madjid Fathi
Mario Fedrizzi
T. Fukuda
Erol Gelenbe
J. Goguen
F. Gomide
Peter Hajek
Larry Hall
Rainer Hampel
Yutaka Hata
Hans Hellendoorn
Babak Hodjat
Moe Jamshidi
Bob Jannarone
Robert John
Ingmari Jonson
Janusz Kacprzyk
Abraham Kandel
James Keller
Nikolas Kasabov
Etienne E. Kerre
O. Kaynak
T.Y. Lin
Peter Klement
Donald Kraft
Rudolf Kruse
Vladik Kreinovich
Jose Luis Verdegay
Masao Makaidono
Abe Mamdani
Udi Manber
Ramon Lopes De Mantaras
John A. Meech
Jerry Mendel
S. Mitra
Allen Moshfegh
M. Mizumoto
Constantin Negotia
Masoud Nikravesh
Vesa Niskanen
Antonio Di Nola
Hung T. Nguyen
Vilem Novak
Andreas Nürnberger
Rainer Palm
Borne Pierre
Irina Perfilieva
Rita Ribeiro
Enrique H. Ruspini
Charlie Ortiz
Sankar Pal
Gabriella Pasi
Witold Pedrycz
F. Petry
Henri Prade
Gero Presser
Anca Ralescu
Gian Guido Rizzotto
Bernard Reusch
Alexander Ryjov
Elie Sanchez
Jimi Shanahan
Michael Smith
Tom Sudkamp
Harold Szu
Tomohiro Takagi
Hideyuki Takagi
E. Trillas
I. Burhan Türksen
Mihaela Ulieru
Aynur Unal
T. Vamos
Ronald Yager
Takeshi Yamakava
John Yen
Lotfi A Zadeh
Hans-Jürgen Zimmermann

Table of Contents

Presentations of WEDNESDAY, AUGUST 15:

Database Querying, Ontology, Content Management and Matching Technologies

Presentations of THURSDAY, AUGUST 16:

Recognition Technology, Data Mining, Summarization, Information Aggregation and Fusion

Presentations of FRIDAY, AUGUST 17:

E-Commerce, Intelligent Agents, Customization and Personalization

Panel Discussions

Presentations of Tuesday, August 14

Search Engines and Retrieving Information

Searching World Wide Web

C. Lee Giles
Pennsylvania State University
Email: giles@ist.psu.edu

Research and Teaching Interests

Giles is an expert in Web analysis, search engines, and intelligent information processing. Formerly a senior research scientist with NEC Research Institute, Princeton, N.J. and now a consulting scientist there, he is well known for coauthoring recent papers published in the prestigious journals Science and Nature that showed that the Web is not only larger than most people thought, but that search engines index only a small portion of it. This work generated press coverage in over 100 news organizations world wide. He has been quoted in the New York Times, the Wall Street Journal, the Washington Post, Red Herring, BBC, AP, and other new services.

He also helped spearhead the development of various Web search techniques and tools, such as the metasearch engine, Inquirus (inquirus.com), which dramatically improves the effectiveness and precision of Web searches, and Researchindex (researchindex.com), an autonomous citation-indexing tool which is the world's largest resource of papers in computer science, encompassing over 250,000 publications.

Giles has taught basic and applied courses in intelligent information processing systems, e-world and eCommerce, and the Internet and World Wide Web.

Using Dynamic Metadata to Improve Search User Interfaces

Marti Hearst
SIMS, UC Berkeley
http://www.sims.berkeley.edu/~hearst
E-mail: hearst@sims.berkeley.edu

Abstract

The current state of web search is most successful at directing users to appropriate web sites. Once at the site, the user has a choice of following hyperlinks or using site search. The former is often too restrictive and the latter is often too permissive, resulting in a disorganized and unhelpful results list. One solution is to develop specialized search interfaces that explictly support the types of tasks users of the site are concerned about. One way to support this solution is to dynamically present appropriate metadata that organizes the search results and suggests what to look at next, as a personalized intermixing of search and hypertext. Thus, rather than focusing on smarter algorithms to be used 'under the hood,' our research group advocates the study and development of better search user interfaces. The goal of our research project, Flamenco, is to develop a general methodology for specifying task-oriented search interfaces across a wide variety of domains and tasks. We suggest that rich, faceted metadata be used in a flexible manner to give users information about where to go next, and to have these suggestions and hints reflect the users' individual tasks. We are conducting usability studies testing various methods of presenting metadata-based search results in a systematic way, to help shed real insight about how do design effective site-level search interfaces.

For more information, see
http://bailando.sims.berkeley.edu/flamenco.html

Searching and Smushing on the Semantic Web - Challenges for Soft Computing

T. P. Martin[1]
Artificial Intelligence Group
University of Bristol, BS8 1TR, UK
Trevor.Martin@bristol.ac.uk

Abstract

The World Wide Web is an astonishing information repository, founded on the simple principles that any web resource can link to any other web resource and that as little as possible should be centrally regulated and imposed. In practice, access to the right information on the web is hampered by the sheer volume of data. Tim Berners-Lee, widely acknowledged as the "father of the web" has pointed out that this is mainly due to the data being *machine-readable* but not *machine-understandable*, and has proposed the Semantic Web. This allows relational knowledge to be encoded in web pages enabling machines to use inference rules in retrieving and manipulating data. In turn, this will reduce the quantity of irrelevant data retrieved and increase the usefulness of the web.

The need for web pages to include knowledge representation presents a tremendous opportunity for fuzzy researchers. In this paper we outline some knowledge representation issues involved and focus on the process of fuzzy matching within graph structures. In particular, we highlight a process known as "smushing" which allows syntactically different nodes to be combined, so that information drawn from multiple sources may be fused.

1 Introduction

Large quantities of data are collected and stored in computer-based systems. In itself, the data is useless unless it can be queried, retrieved and explored in order to realise its value. Questions of data storage and retrieval are fundamental to the information revolution giving rise to the field of intelligent information which involves

- sophisticated mechanisms for searching, extracting and presenting information
- metadata techniques in which a rich array of 'data about the data' is stored
- the ability to configure data to different users with different needs and
- the ability to make inferences from data when appropriate.

The Semantic Web [10] offers a tremendous opportunity for fuzzy researchers to demonstrate the power of soft inference in making useful, human-understandable, deductions from the semi-structured and sometimes contradictory information available on the web. This paper outlines existing techniques for retrieval and representation, and points to some areas in which fuzzy methods could profitably be applied.

2 Information Retrieval and Searching

Classical techniques for extracting data depend on the form of the stored data. At one extreme is the relational database with complete and well-known data, a rigidly specified schema and a formal, highly structured querying mechanism. In return for this rigidity and adherence to structure, guarantees can be made about the semantics, integrity and consistency of the data (e.g. [23,28]). Data extraction, conversion, transformation, and integration are well-understood, with clear theoretical underpinnings.

At the opposite extreme, we have unstructured data which is largely free of formal semantics and generally requires human interpretation to make sense of the data. The process of extracting and searching for data within free text documents has given rise to the field of information retrieval - see [2] or [29] for a good introduction. Searching is generally on the basis of weighted keywords [25], which can be generated automatically from a text corpus. Variations of this model include probabilistic retrieval, fuzzification of the vector model, neural nets, Bayesian nets, etc.

In the middle of the spectrum, overlapping with both categories, we have semi-structured data such as classified directories, product catalogues, help systems, and much of the World Wide Web. Semi-structured data contains compulsory components (e.g. business name and telephone number in a classified directory) and optional, less-structured components (e.g. description of products and services). Substantial change must be made to the relational approach to accommodate semi-structured data - e.g. [24] or [20].

[1] Currently BT Senior Research Fellow, Intelligent Systems Research Lab, BTExact, Adastral Park Ipswich IP5 3RE, UK

More work has been carried out on extending information retrieval concepts to semi-structured data, particularly with the proliferation of the world wide web. Typically, the structured elements are ignored and keyword-based free-text search is used.

2.1 Extensions

In a soft computing context, it is most appropriate to consider fuzzy extensions to these approaches first. It is interesting to note that, although the real world does not map naturally to the crisp and artificial categories required by the relational model, there are few, if any, large scale commercial applications of fuzzy databases. The fundamental difficulty in extending the relational framework with fuzzy set theory is providing an unambiguous (and preferably objective) definition of what is meant by the membership degrees. Too often, this problem is side-stepped by appeal to the subjective opinion of a user, or as a context-dependent parameter whose interpretation is decided by the system designer. A concise statement of this position is [11] :

> "[membership functions should be] adjusted for maximum utility in a given situation".

Dubios and Prade [17] acknowledge the lack of semantics for membership functions, identifying similarity, preference and uncertainty as three possible interpretations. They go on to claim that this is an advantage, noting the parallel with different interpretations of probability (frequency, subjective, axiomatic) whilst acknowledging a fundamental difference in that the different models of probability all have "thought experiments" which dictate how degrees of probability can be obtained.

Baldwin's mass assignment theory [3] relates probabilistic and fuzzy uncertainty via a voting model interpretation which allows membership degrees to be obtained. This gives a clear semantics to the membership values, and should be adopted widely. To promote fuzzy technology as part of the semantic web, it is essential that the semantics of membership functions are easily and widely understood.

Other soft computing methods have been applied to the problem of information retrieval. Given a sufficiently large set of examples, neural net and machine learning approaches have been used to derive an association or rule between query terms and the keywords in documents judged to be relevant [1].

Additional refinement can be achieved by clustering documents. The most obvious (and time consuming) approach to clustering is to use manual classification of documents. For example, Yahoo (www.yahoo.com) uses a hierarchical labelling to assign each web page to a particular category. Documents can be retrieved by navigation through the category hierarchy as well as by keyword search.

An interesting approach to clustering is provided by [22] in which the vector representation of a document is based on trigram frequencies. A standard Kohonen net approach reduces the dimensionality of the document vectors, yielding a 2 or 3 dimensional map. A query can be classified in a similar manner and retrieved documents should be close to it on the map. This approach is computationally intensive, and difficult to update incrementally. However, it can be argued that Kohonen clusters are based on concepts rather than simple keywords, a claim that can also be made for some of the more advanced search-engine products such as Autonomy (www.kenjin.com)

An alternative to pure keyword search, exploited by Google, is to use the link structure of the web [13]. Any web page with a large number of links pointing to it is assumed to be important, and is allocated a high rating. The score is propagated by assuming that any "important" page will contain links to other important pages; thus links to a particular page are weighted by the score calculated for the source of the link. By iterating this process, a *PageRank* is calculated for each page. Formally, if we treat the web as a graph with pages as vertices and hyperlinks as edges, then the *PageRank* corresponds to the principal eigenvector of the normalised adjacency matrix.

2.2 Performance Evaluation

A relational database has no flexibility when evaluating query results - if the system does not return *exactly* the set of correct answers, it is logically inconsistent. Information retrieval, on the other hand, recognises that the set of relevant answers is often inherently ill-defined and that there is a degree of dependence on the query. *Precision* and *recall*, (respectively the proportion of retrieved documents that are relevant and the proportion of relevant documents that are retrieved) are frequently used to assess system performance. These measures need to be viewed with some caution - we can obtain maximum recall by retrieving all documents, and retrieving no documents at all has infinite precision. Also it is difficult to apply these measures reliably to the question of web retrieval, since the set of all "relevant" documents is frequently not known.

3 Metadata

There is a widely-recognised problem with keyword-based searches which often return hundreds or thousands of documents - the "relevant" documents may be in the first ten or twenty returned, but it is likely that a lot of relevant documents are missed. The scale of this problem is increasing daily as the number of web pages increases.

To save time and space, many search engines do not extract keywords from the entire page, but use HTML tags to focus on important data. Searching for text marked up with *<title>* and heading *<h1>*, *<h2>*, ... tags is assumed to give sufficient information to index a page. In addition, the optional use of *<meta>* tags enables a list of keywords, author, creation date etc to be specified for use in indexing.

Most of this structure is concerned with *how* to display the document rather than bearing any relation to its content. Although there are guidelines on the use of *<meta>* tags, there is a danger that an unscrupulous web author could promote a site by misleading use of such tags.

A possible solution is "push" technologies, in which a simple user profile is used to screen out unwanted material and deliver only relevant information to the user. (see [14] or www.infogate.com for example). This requires mechanisms for determining the content of a page, representing the interests of a user, and judging how well the two match for any given page.

3.1 XML and RDF

In order to deliver "relevant" information, we require knowledge about the content of a page, i.e. metadata. XML was created as a method of marking up documents and conveying the semantics - in the same way as HTML specifies the display properties of different pieces of text, XML can be used to convey a hierarchical relationship of data values. XML allows new tags to be defined, with a BNF grammar to determine the precise combinations of tags that are permitted. XML is becoming a de facto standard for electronic data interchange. An example is shown below.

```
<p>
Department of Engineering Mathematics
<br>University of Bristol<br>
  Queens Building<br>
  University Walk<br>
  Bristol BS8 1TR <br>
  UK </p>
<p>Tel. +44 117 928 8200</p>
<p>Fax. +44 117 925 1154</p>
```

```
<ADDRESS>
   <DEPT> Engineering Mathematics </DEPT>
   <ORG> University of Bristol </ORG>
   <BUILDING> Queens Building </BUILDING>
   <STREET> University Walk </STREET>
   <CITY> Bristol </CITY>
   <POSTCODE> BS8 1TR </POSTCODE>
   <COUNTRY> UK </COUNTRY>
   <PHONE> +44 117 928 8200 </PHONE >
   <FAX> +44 117 925 1154 </FAX>
</ADDRESS>
```

On its own, XML is not necessarily any more informative than HTML. The tags are arbitrary and can be defined by any creator of web pages, so that to a computer there is little difference between * * ... * * and *<address>* ... *</address>*.

XML is an extendible *syntax* for document description, whereas RDF defines an extendible structure for expressing the *semantics* of document description [16,30] by defining resources in terms of property values. The Dublin Core metadata standard defines properties to describe web resources, including creator, title, format, as well as vaguer properties such as relation, description, rights.

An example (from http:// dublincore.org / documents / 2001 / 04 / 12 / usageguide / generic.shtml) shows the use of some properties:

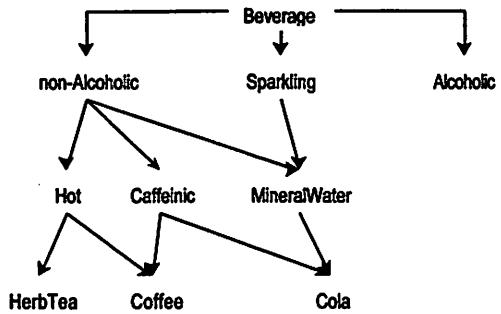
```
Title="Candle in the Wind"
Subject="Diana, Princess of Wales"
Date="1997"
Creator="John, Elton"
Type="sound"
Description="Tribute to a dead princess"
Relation="IsVersionOf Elton John's 1976
        song Candle in the Wind"
```

It is impossible for a single, centrally defined resource to encompass all possible metadata schemas. The extensible nature of RDF means that different metadata schemas can be established by different groups - if a definition is available, the semantics of the data can be accessed by any interested party. This is in accordance with the task of the working group on RDF "to specify semantics for data based on XML in a standardised interoperable manner". Metadata from different sources will have some common elements and some unique elements; it is also possible that different sources may use different structures to refer to essentially the same information.

Additional difficulties may arise where property values free text - an author's description of the content may not accurately reflect all information contained in the document.

3.2 Inference and Ontologies

RDF is a simple knowledge representation system, and can be used in reasoning processes. The AI community has worked hard on applying formal logic to inference, including first/higher order logics, etc. (see www.semanticweb.org/inference.html). Many approaches assume a centralised definition for knowledge representation, although work has been carried out in portability of ontologies. (see e.g. [19])
In practice it is easy to become enmeshed in esoteric philosophical questions of exact definitions, precise limits on expressive power, etc. In part, this arises from the insistence on crisp classification even when a very limited domain is considered. For example, the following fragment of a classification hierarchy is taken from [27] (see also http:// users.bestweb.net



/~sowa/ontology/toplevel.htm, also [18])

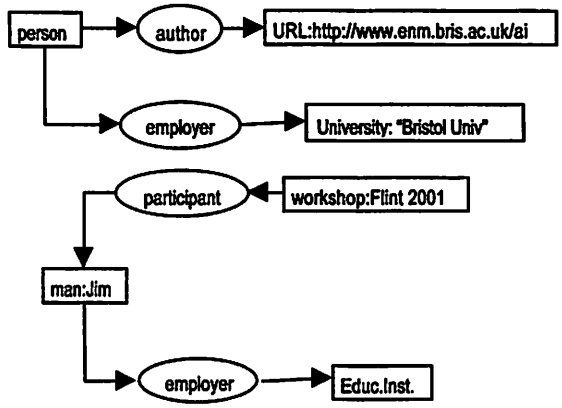
Clearly this should not be interpreted as a crisp hierarchy. Different blends of coffee contain caffeine to a greater or lesser degree; adding leaf tea to the hierarchy would introduce a category that contained caffeine but to a lower degree than coffee, etc.
The Fril++ approach (below) allows us to build hierarchies involving partial memberships.

4 Opportunities for Soft Computing

At first sight, the semantic web is tailor made for the techniques of soft computing. The aim is to bridge the gap between machine-readable "hard logic" and human-understandable natural language (possibly in some restricted form). None of the usual logical requirements can be guaranteed - there is no centrally defined format for data, no guarantee of truth for assertions made, no guarantee of consistency, etc.
However there are dangers in proposing fuzzy logic as the solution to all problems on the semantic web. What is needed is not just a simple fuzzification of numerical terms - although this may be a useful aid in some search problems, given a commonly understood definition of membership. The real need is for fuzzification and matching of concepts. It is relatively easy to search within a database for a restaurant which is "near" to one's current location, or which has a menu on which "most" items are "reasonably priced". It is much more difficult to search for a restaurant with a "high quality wine list" or which has "a good range of spicy vegetarian main courses".
Additionally, we must remember the available computing power and the need for practical implementations of systems. In order to prove the usefulness of fuzzy logic, examples are needed which are more efficient using fuzzy or which can't be done at all without fuzzy It is essential that fuzzy extensions demonstrate performance improvements over "classical" techniques.

4.1 Knowledge Representation using Conceptual Graphs and Fril++

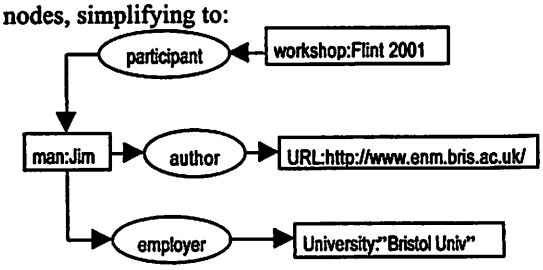
Conceptual graphs [26] are a powerful and general form of graphical knowledge representation[2]. Their similarity to RDF has been noted by several researchers, including Berners-Lee [9].
A conceptual graph is a finite, connected bipartite graph representing a proposition. Two examples are below, corresponding to the propositions "a person employed by the University of Bristol is author of the web page http://www.enm.bris.ac.uk/ai" and "a man called Jim, employed by an educational institute, is a participant at the FLINT2001 workshop".



Rectangular nodes are *concept* nodes, and consist of a *type* e.g. person, man, university.... and an optional referent, labelling an individual conforming to that type. Types correspond to RDF classes. Elliptical nodes denote *relations* between concepts.

[2] A conceptual graph toolkit implementing these operations is available in Fril from ftp://xian.enm.bris.ac.uk/pub/CGPackage

Types are partially ordered, via a hierarchy enabling inheritance of properties and generalisation and specialisation of graphs. New graphs are constructed by formation rules *copy*, *restrict*, *join* and *simplify*.
From the graphs defined above, we restrict the concept [person] to [man], (*man* is a subtype of *person*), then restrict the referent to *Jim*, which conforms to the type *man*. Similarly, restrict *EducInst* to *University:Bristol*, and join the graph on these nodes, simplifying to:



These operations preserve meaningfulness; starting from false graphs we can only derive false graphs.
In addition to concept and relation nodes, a third category - actors - implements computed relations. We also note that a node of type *Proposition* can contain a conceptual graph as referent.

4.2 Uncertainty in Conceptual Graphs

Baldwin and Morton [8] extended the conceptual graph framework to allow fuzzy referents. They partitioned the type lattice into mutually exclusive subtypes : entity, attribute, information, event, and state, and defined functions on the first three enabling fuzzy information to be represented. For example, an entity exhibits perceptual fuzziness, i.e. a compatibility between a referent and a type. Linguistic fuzziness arises in an attribute such as height or colour, with referents such as *tall* and *pale-red*. Finally, propositional fuzziness allows a fuzzy truth value to be associated with a graph which expresses a proposition. These require modified graph operations such as join, project. [31] extended the framework to allow fuzzy relations. [21] has applied fuzzy conceptual graphs to machine learning.
Conceptual graphs also have powerful mechanisms for handling incomplete and default information.

- A *schema* is a graph which represents an example of the usage of a given concept.
- A *prototype* gives central tendencies of a concept with generic concepts restricted to default values.
- A *canonical graph* for a type states necessary constraints on the use of that type.
- A *type definition* gives necessary and sufficient conditions for a type. High level concepts may be decomposed into graphs of primitive types.

4.3 Fril++

Fril++ [4,5] is an extended Fril [6,7] which allows an uncertain class hierarchy to be defined. Much analysis and design is founded on the assumption that a real-world problem can be modelled using crisp sets of objects. For example, an employee is definitely a member of the class (set) *person*, and not a member of the class *job*. The decomposition may not always be crisp - e.g the class *job* can be partitioned into managerial, technical, manual, etc. sub-classes. These are not crisply defined categories - e.g. a research team leader may fall into both managerial and technical classes. This is an obvious case for a fuzzy approach, since we have a number of instances (elements) which belong to the category to a greater or lesser extent. There are other, more obvious fuzzifications of the object oriented model, allowing uncertainty in data values and the propagation of uncertainty through local and inter-object computations.
Fuzzy classes can be of considerable use in conceptual graph modelling. Memberships do not need to be numbers, nor even explicitly stored or evaluated for any particular instance. An instance may be a close match to one class when one subset of its properties is considered; it may be close to another when another subset is considered. The programmer has a set of properties in mind when deciding on the class hierarchy, but it is by no means obvious that every inheritance must take class memberships into account.

4.4 Fusion of concepts - "smushing"

Almost inevitably, there are will be small inconsistencies, both within RDF data from one source and between RDF data derived from different sources. To take a very simple example, we may have a document with author details

```
name="Martin, Trevor"
```

and another with

```
name="Trevor Martin"
```

It is highly likely that these represent the same person (particularly if other attributes such as institution, email address, etc are the same). A method of approximate matching is needed, enabling data from the different sources to be combined (or "inconsistencies" within data to be identified).
More complex problems may arise where different representations have been used - e.g. the structures

```
workshop
  title = FLINT 2001 - 2001 BISC Int'l
Workshop on Fuzzy Logic and the Internet
  url=bisc7.eecs.berkeley.edu/BISC/flint2001
  location = University of California,
        Berkeley
  date = August 14-18, 2001
```

```
conference
   location = California
   start = 14 August 2001
   duration = 4 days
   title = Fuzzy Logic and the Internet
   participants = { ....etc }
```

refer to the same event, but differ in almost all respects. The "smushing" problem [12] involves the correct combination of disparate knowledge sources. Damiani [15] has made initial inroads from an XML perspective using weighted digraphs. There are many levels at which this graph matching problem can be tackled. At the simplest, we have a syntactic match - the names "Martin, Trevor" and "Trevor Martin" require a simple re-ordering of words to become identical. Syntactic closeness may be misleading in same cases. For example, the UK postcodes IP1 1AB and IG1 1AB differ only by 1 character but designate very different places, whereas IP1 1AB and IP5 2NP are syntactically very different (2 common characters) but are relatively close - syntactic similarity is not necessarily a useful indicator of semantic similarity.

At a higher level, we are faced with computing the match between structures such as the *workshop* and *conference* examples above. This requires context-dependent procedures attached to the graphs themselves. Fril++ and the CG toolkit is a software system with the required features to implement this.

5 Summary

The vision of a semantic web includes many aspects which require fuzzy knowledge representation and reasoning. There are opportunities for soft computing - the needs are for a common, understandable interpretation of membership functions, and efficient, portable, implemented systems which solve real semantic web problems. These include:

- the mismatch between crisp hierarchical structures and the "fuzzier" real world in which objects may have partial membership in classes
- notions of fuzzy equality in data, and semantic equivalence of syntactically different structures
- robustness vs missing, partial and incorrect data

Knowledge representation by means of RDF maps easily into the conceptual graph framework and Fril++, and future work will address these issues.

6 References

1. Austin, J. and Lees, K., *A search engine based on neural correlation matrix memories.* Neurocomputing, 2000. 35, 55-72.
2. Baeza-Yates, R. and Ribeiro-Neto, B., *Modern Information Retrieval.* 1999, Harlow, UK: Addison Wesley.
3. Baldwin, J.F., *The Management of Fuzzy and Probabilistic Uncertainties for Knowledge Based Systems,* in *Encyclopedia of AI,* S.A. Shapiro, Editor. 1992, John Wiley. p. 528-537.
4. Baldwin, J.F., *et al. Implementing Fril++ for Uncertain Object-Oriented Logic Programming.* Proc*IPMU 2000,* 496-503.
5. Baldwin, J.F. and Martin, T.P. *Fuzzy Classes in Object-Oriented Logic Programming.* in *FUZZ-IEEE-96.* p 1358-1364
6. Baldwin, J.F., Martin, T.P. and Pilsworth, B.W., *FRIL Manual (Version 4.0).* 1988, Fril Systems Ltd, Bristol Business Centre, Maggs House, Queens Road, Bristol, BS8 1QX, UK.
7. Baldwin, J.F., Martin, T.P. and Pilsworth, B.W., *FRIL - Fuzzy and Evidential Reasoning in AI.* 1995, RSP (Wiley).
8. Baldwin, J.F. and Morton, S.K., *Conceptual Graphs and Fuzzy Qualifiers in Natural Language Interfaces.*1985, Univ Bristol
9. Berners-Lee, T., *Conceptual Graphs and the Semantic Web,* http://www.w3.org/DesignIssues/CG, 2001.
10. Berners-Lee, T., Hendler, J. and Lassila, O., *The Semantic Web,* in *Scientific American.* 2001. p. 28-37.
11. Bezdek, J.C, *Fuzzy Models.* IEEE TransFuzzSyst,1993 1 1-5.
12. Brickley, *RDFWeb notebook: aggregation strategies,* http://rdfweb.org/2001/01/design/smush.html, 2001.
13. Brin, S. and Page, L. *The anatomy of a large-scale hypertextual Web search engine.* in *International world wide web conference.* p 107-118, 1998. Brisbane; Australia: Elsevier.
14. Case, S.J., *et al., Enhancing e-Communities with Agent-Based Systems.* IEEE Computer, 2001. 33(7): p. 64.
15. Damiani, E., Tanca, L. and Fontana, F.A., *Fuzzy XML Queries via Context-based Choice of Aggregations.* Kybernetika - Praha-, 2000. 36(6): p. 635-656.
16. DublinCoreGroup, *Dublin Core Metadata Initiative,* http://dublincore.org/, 2001.
17. Dubois, D. and Prade, H., *The three semantics of fuzzy sets.* Fuzzy Sets and Systems, 1997. 90: p. 141-150.
18. Erdmann, M., *Formal Concept Analysis to Learn from Sisyphus-III Material,* http://www.aifb.uni-karlsruhe.de/~mer/Pubs/Sisy_FCA/, 2000.
19. Fensel, D., *et al., OIL: An Ontology Infrastructure for the Semantic Web.* IEEE Intell Syst +Applications, 2001. 16 p. 38-45.
20. Goldman, R., McHugh, J. and Widom, J., *Lore: A Database Management System for XML.* Dr Dobbs Journal, 2000 25 76-80.
21. Ho, K.H.L. *Learning Fuzzy Concepts by Example with Fuzzy Conceptual Graphs.* in *1st Australian Conceptual Structures Workshop.* 1994. Armidale, Australia.
22. Kaski, S., *et al., WEBSOM - Self-organizing maps of document collections.* Neurocomputing, 1998. **21;** 101-117.
23. Maier, D., *Theory of Relational Databases.* 1983: Pitman.
24. McHugh, J., *et al., Lore: A Database Management System for Semistructured Data.* Sigmod Record, 1997. 26(3): p. 54-66.
25. Salton, G. and McGill, M.J., *Introduction to Modern Information Retrieval.* 1983, New York: McGraw Hill.
26. Sowa, J.F., *Conceptual Structures.* 1984: Addison Wesley.
27. Sowa, J.F., *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* 1999: Brooks Cole
28. Ullman, J.D., *Principles of Database and Knowledge-Base Systems Parts 1 and 2.* 1988: Computer Science Press.
29. van Rijsbergen, C.J., *Information Retrieval.* 2nd ed. 1979, London, UK: Butterworths.
30. W3C, *W3C Semantic Web Activity: Resource Description Framework,* http://www.w3.org/RDF/, 2001.
31. Wuwongse, V. and Manzano, M., *Fuzzy Conceptual Graphs,* in *Conceptual Graphs for Knowledge Representation,* G.W. Mineau, B. Moulin, and J.F. Sowa, Editors. 1993, Springer (LNAI 699). p. 430-449.

Fuzzy Reinforcement Learning and the Internet with Applications in Power Management of Wireless Networks

Hamid Berenji
Intelligent Inference Systems Corp
Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

In this talk, I will discuss the strength of Fuzzy Reinforcement Learning (FRL) for learning in continuous input/output domains. I will then introduce a newly developed Actor Critic based Reinforcement Learning Algorithm called ACFRL that trains a critic and an actor simultaneously. I will then describe an application of ACFRL in the power management of wireless networks where power usage can be optimized based on the interference and the congestion on the network. Lastly, I will describe the potentials of FRL in text mining and internet search engines. This work is a joint effort with David Vengerov of Stanford University.

Fuzzy logic e-motion

Elie Sanchez
Laboratoire d'Informatique Médicale[1] and Neurinfo/N&F Institute
[1]Faculté de Médecine, 13385 Marseille Cedex5, France
elie.sanchez@medecine.univ-mrs.fr

Abstract
Telerobotics over the World Wide Web has gained considerable interest in recent years, but Internet-based fuzzy telerobotics is still in its infancy. This paper presents telerobots in general and dicusses fuzzy logic contributions to the field.

1. Introduction

Internet dates back to ... : whatever date is given, it is interesting to note that in 1889, Jules Verne envisioned Internet in *The Day of an American Journalist in the Year 2889*. In this novel, he wrote: "Here 1,500 reporters, in their respective places, facing an equal number of telephones, are communicating to the subscribers the news of the world as gathered during the night ... besides his telephone, each reporter has in front of him a set of *commutators*, which enable him to communicate with any desired *telephotic* line. Thus the subscribers not only *hear* the news but *see* the occurrences ... furthermore, the hearers are free to listen only to what interests them." In addition, news from "reporters in the astronomical department -- a department still in the embryonic stage, but which will yet play an important part in journalism" is sent from other planets: "We have *phototelegrams* from Mercury, Venus, and Mars," which finally could prefigure the *Interplanetary System Wide Web*, a current project of Vinton Cerf at NASA's JPL: the *InterPlaNet*.

Several researchers have recognized the importance of the emotions in thinking and in intelligent behavior. Logical models have been proposed to model human perception, intelligence and thinking. *Logic* is *cold* and *emotion* is *warm*, the two terms *Logic* and *emotion* appear as antinomic. But the picture changes, when associating *fuzzy logic* with *emotion*, which corresponds to the title of this paper, when pronounced. In the written version it is *e-motion*, for electronic-motion, just like in e-commerce, e-doctor or e-mail, i.e. terms that are invading our everyday (e-)life. And *motion* is for (mobile) robots.

Advanced technologies of information and communication are changing our everyday life. Our society is permeated by telecommunications. Telephones, cellular phones and especially Internet have increased teleconnectivity, offering the possibility to exchange text, images, sound and video.

A telerobot is a robot that can extend a human operator's manipulating capacity to a remote location. That human supervises it through a computer intermediary. Hardware of robots and of telerobots is not much different, but robots require less human involvement for instruction and guidance than telerobots. Moreover, even an out of control telerobot must safely interact with persons, with other telerobots or with fixed devices: safety must be a primary concern in designing the system.

Internet-based fuzzy telerobotics is an ubiquitous technology for humanistic systems, i.e. man-machine communication systems in which humans play a major role. It is important, and necessary, to involve humans in telerobotic control: robots don't have the capability to perform intelligent tasks.

Telerobots were developed in the fifties to facilitate action at a distance. An excellent review of research developments in telerobotics and teleoperation is given in [20] by T. Sheridan and Yahoo's classical site [7] contains a good sample of machines interfaced to Internet. In the general spirit of this paper, telerobots are supposed to be controlled via Internet. Internet is adding a new dimension by allowing users on the Web, so worldwide, to have remote access to the robot.

Telerobots are used when the robot environment is not easily accessible or hostile, to extend the capabilities of the operator (cf. Mars Pathfinder mission [22], undersea robotics [19] or nuclear power plants). In the case of unique or costly resources for example, is it natural to consider the accessibility of applications via Internet. It is then necessary to use or develop new tools to interface interactively these resources with Internet, making them available via a Web browser.

And of course, feedback from the remote system is also expected.

A *fuzzy telerobot* executes tasks on the basis of information received from the human operator, plus it has its own capability for *fuzzy reasoning*. To be complete, in Internet-based fuzzy reasoning, the system should also include *fuzzy (Java) applets*.

The first Internet telerobots date back to the mid-nineties and the introduction of fuzzy logic techniques in this domain is still in its infancy. In the following section will be presented, on the basis of an example, some basic components of (Internet-)telerobots.

2. A telerobot

A telerobot is a robot that receives instructions from a remote location, from a human operator, trained or not, depending on applications. The telerobot is controlled via Internet through a computer running a browser. The robot is connected, wired or wireless, to a computer hosting a Web server.

Man-machine interaction is an important factor. With regard to *vision*, feedback from cameras is very useful. The system must be capable of allowing the Internet users to switch from the robot view (what it sees) to a bird's eye camera view of the scene, in order to observe the movements of the robot, as if one were in the robot's physical location. The alternative is to have both views in two different windows of the operator's screen. In addition to vision, *audio* communication can add useful facilities to control the robot.

A "Minimalist Telerobotic Installation on the Internet" is presented in [17]. It is an interactive online lightbox system, accessible to anyone with a graphical browser. The project uses motors and light switches and returns live images resulting from user specified combinations of devices (see: http://www.dislocation.net/)

Figure 1 illustrates an example of a mobile telerobot controlled via Internet, a new collaboration project of Neurinfo/N&F Institute and ESIM, at Marseille. We have adapted the client-server model inherited from TCP/IP Internet programming. Here follows a description of the system's components.

- An operator station: a computer with Internet connection and browser.
- From this computer an Internet user/operator sends instructions to a telerobot, a mobile robot here, via (fuzzy) applets (cf. section 3), to guide it for a specific task.
- The applets allow control of:
  - a video camera located on top of the robot's operating room, more generally in its physical environment; control is for focusing, zooming or orientation
  - the telerobot itself: moving instructions for the drive base, and later on, a robot arm (or simply a pointer) mounted on the base
- A Web server in the robot's operating room (instructions from the operator, feedback from the robot and its environment, etc.). This computer is connected from serial/parallel ports to the video camera and a wireless networking router.
- A robot with a mobile base, micro-controllers (we use 68HC11's) or PC with wireless networking, sensors, stepper motors (to turn the camera and to run the mobile base), wireless equipment, on-board camera, batteries, custom electronics to control motors, sensors, etc.
- Depending on applications, in a further step, the robot will possibly be equipped with a microphone, speakers, LCD screen.

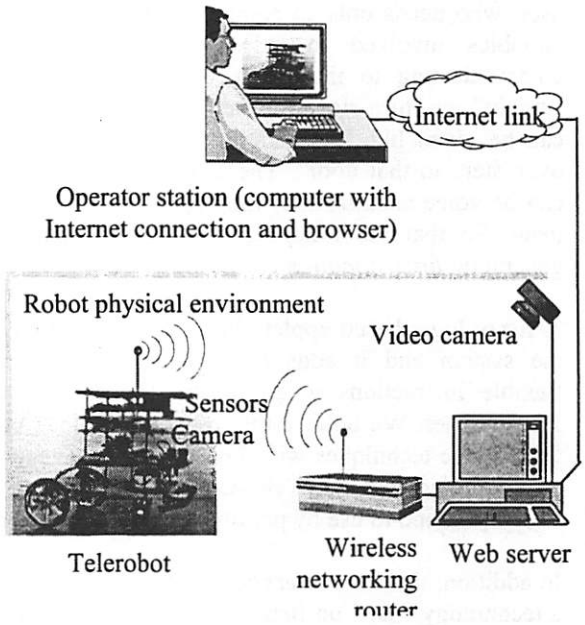


Figure 1: System overview for a telerobot.

3. Fuzzy logic in Internet-based robotics

Fuzzy logic has been successfully used in many industrial applications, especially for control, in devices ranging from consumer products to cars, trains, planes or space modules.

With regard to the field of *Internet-Based Robotics*, fuzzy logic can be applied in both *Internet* and *Robotics* constituent parts.

In Internet applications, Sun Microsystems' Java is considered as the most widely adopted programming language. Java allows effortless portability just about everywhere, from the smallest devices to supercomputers. Applets are Java programs written to be executed by a Web browser. They are often downloaded from a remote server and run in a controlled environment.

It is then natural to consider fuzzy logic applets, or *fuzzy applets* for short, i.e. Java modules implementing *fuzzy logic propositions*, in the form of *fuzzy rules*, that are instructions given by the operator to the telerobot. Note that fuzzy applets are not necessarily built for fuzzy logic control, they can also be used to perform smart data base searches, using fuzzy logic concepts to make intelligent choices from imprecise queries.

Of course, fuzzy logic has to be transparent to the user, who needs only to communicate with linguistic variables involved in rules. So that instead of communicating to the system instructions like "go straight", or "turn right" or "left", natural instructions can be given like "turn slightly to the right" or "go over there to that door". These linguistic instructions can be voice commands if the system is equipped for them. So that the fuzzy applet will be capable of generating *fuzzy e-motion* on the Web !

A fuzzy logic based applet allows soft interactions in the system and it adds *intelligence* to it, through flexible instructions given to the telerobot by the Internet user. We believe that a good combination of fuzzy logic techniques with Java applets can generate smart applications: the Web interface has to be easy to understand and to use by possibly unskilled operators.

In addition, the *fuzzy telerobot* will be controlled with a technology based on fuzzy logic. This part is now a classical chapter in control theory, as shown in the literature and illustrated by so many running industrial applications. In our prototype and from the experience gained at the N&F Institute (cf. for ex. Peugeot S.A. automotive applications [1] or computerized vehicles for handicapped persons [18]), a vehicle can avoid, or stop for, obstacles, reaching a goal (cf. odometry, ceiling marks), maintaining a safe distance between vehicles, following persons or corridors, parking in a reserved place, etc.

4. Internet-based robotics applications

The first Internet camera was set up in 1991 to monitor the status of a coffeepot [8], and since then, applications have been flourishing. The first Internet telerobot was developed in 1994 by Ken Goldberg and his group [5,6]: it was the Mercury project. A digital camera and air jet were mounted on a robot arm so that a user on the Internet could view and excavate for artifacts in a sandbox located at the University of Southern California. Then, Ken Taylor in Australia, introduced a remotely controlled six-axis telerobot on the Internet [4]. Later on, still in 1994, Richard Wallace demonstrated a telerobotic camera and Mark Cox put up a system to control, via Internet, a robotic telescope. Then, Eric Paulos and John Canny developed a Mechanical Gaze [15], a telerobotic system where users could control a camera's viewpoint and image resolution to observe museum artifacts in the vicinity of the robot.

Most of the applications have been developed at universities. In addition to the early developments in 1994 mentioned above, below is a sample of applications (some of them at the end, are merely suggestions). It does not claim to be complete, but to be representative of work in the field. Applications are listed to suggest or inspire new fuzzy logic-based e-applications.

- Australia's telerobot, to pick and place children's building blocks [24].
- Telegarden, to plant, water, and monitor the progress of living plants [9]. It is this application that suggested the title of K. Goldberg's book [6].
- Remotely Operated Telescope [10].
- Mars Pathfinder Mission [2,22].
- Airborne and Terrestrial untethered telerobots [16]: helium-filled blimp airborne telerobots called "space browsers", then terrestrial surface-cruisers.
- Xavier [11,23]: a robot that navigates corridors in a lab.
- Communications Robot [12]: a web server fitted with video conferencing equipment mounted on a mobile robot base.
- RemoteBot.net [14], for visual remote control of a small Khepera mobile robot.
- Model Railroad [13] (interesting application, although strictly speaking not involving a robot.)
- Museum visits [3].

. Visits to exciting cities or scenic natural parks and exploring the desert, the poles or even other planets [21].
. Home or kindergarten surveillance
. Elderly or handicapped persons assistance
. Wild animal watching
. Prehistoric cave exploration, when public visits can cause damage or when access is hazardous
. Participation in auctions
. Prison/Police/Military surveillance.

5. Discussion and conclusion

In addition to interpretation and treatment of traditional measurements, it is important for future research, to take into account perceptions, related to different senses, as proposed in L.A. Zadeh's computational theory [25].

Now, some difficulties might temporarily restrain the development of Internet-based technological applications:

- bandwidth limitations: many users participate over low bandwidth connections such as modems, but video is very demanding in terms of network bandwidth consumption and processor use. The update rate of the transmitted video images should be as fluid as possible to provide a good feeling of reality. Data compression, optimal choice of resolution or of focussing are characteristics to be considered.

- communication propagation delays: time lag is not a new issue for telerobotics. Space relevant rover technology has focused on the time lag issue for years because of the amount of time it takes for signals to get from the Earth to off world locations and back again. The Web can involve long time delays for which no upper bound can be guaranteed. Hence a telerobot system requires a high degree of autonomy and a robust installation.

Network delays can cause the user to feel detached from the telerobot: with the current Internet generation, real-time control over the Internet is still a problem. But these obstacles, while significant, are not real barriers and the coming new Internet generation facilities will help to surmount them. Also, applications can be sought for which the above points are not major issues, with respect to the utility of the application.

In conclusion, following the existing telerobotics applications and the evolution of Internet, we believe that Internet-based fuzzy telerobotics is a new and promising future domain that will evolve towards more humanistic applications.

6. references

[1] J.P. Aurrand-Lions, L. Fournier, M. de Saint Blancard and E. Sanchez, "Application of Fuzzy Control for ISIS Vehicule Braking", Int. Conf. on Fuzzy and Neural Systems, and Vehicle Applications'91, univ. of Tokyo, Japan (1991).

[2] P.G. Backes, K.S. Tso and G.K. Tharp, "Mars Pathfinder Mission Internet-Based Operations Using WITS", Proc. of the IEEE Int. Conf. on Robotics & Automation, Leuven, Belgium (1998).

[3] W. Burgard et al, "The Interactive Museum Tour-Guide Robot", Proc. Of the 15th National Conference on Artificial Intelligence (1998).

[4] B. Dalton and K. Taylor, "A framework for Internet Robotics", Workshop on Web Robots, IEEE/RSJ Int. Conf. On Robots and Systems "IROS" (1998).

[5] K. Goldberg (Ed.), M. Mascha, S. Gentner, J. Rossman, N. Rothenberg, C. Sutter and J. Wiegley, "Beyond the Web: Manipulating the Real World", *Computer Networks and ISDN Systems Journal* 28, No.1 (1995).

[6] K. Goldberg (Ed.), "The Robot in the Garden. Telerobotics and Telepistemology in the Age of the Internet", The MIT Press (2000).

[7] http://dir.yahoo.com/Computers_and_Internet/Interesting_Devices_Connected_to_the_Net/

[8] http://www.cl.cam.ac.uk/coffee/coffee.html

[9] http://telegarden.aec.at/

[10] http://www.deepspace.ucsb.edu/rot.htm

[11] http://www.cs.cmu.edu/People/Xavier/

[12] http://freedom.artm.ulst.ac.uk/~antonh/research/CR/CommsRobot.shtml

[13] http://lyc-arago.scola.ac-paris.fr/home-page-800.htm

[14] O. Michel, P. Saucy and F. Mondada, "Khep-On The Web: An Experimental Demonstrator in Telerobotics and Virtual Reality", Virtual Reality and Multimedia Conference, IEEE Computer Society Press, Switzerland (1997). See: http://www.remotebot.net

[15] E. Paulos and J. Canny, "Delivering real reality to the world wide web via telerobotics", in IEEE Int. Conf. On Robotics and Automation (1996).

[16] E. Paulos and J. Canny, ProP: Personal Roving Presence", ACM SIGCHI (1998), In: http://www.prop.org/prop/papers.html

[17] E. Paulos and K. Golberg (Eds.), "Current Challenges in Internet Robotics", Full-Day Workshop, IEEE Int. Conf. on Robotics and Automation, Detroit, Michigan (1999). In: http://www.cs.berkeley.edu/~paulos/papers/icra 99

[18] E. Sanchez, Ph. Pierre and J.P. Aurrand-Lions, "VITAUR: Computerized Mobile Vehicle base on Fuzzy Logic Reasoning, for Integration and Autonomy of Disabled Persons", 5th Int. Symp. ORIA'94: "From Telepresence to Virtual Reality", Marseille, 349-356 (1994).

[19] C. Sayers, "Remote Control Robotics", Springer (1999).

[20] T.B. Sheridan, "Telerobotics, Automation and Human Supervisory Control", MIT Press (1992).

[21] T.B. Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis", IEEE Trans. Robotics and Automation, 9(5), 592-606 (1993).

[22] D. Shirley and J. Matijevic, "Mars Pathfinder micro-rover", Autonomous Robots, 2(4), 281-289 (1995).

[23] R. Simmons, "Where in the world is Xavier, the robot ?", Robotics and Machine Perception, Special Issue: Net-worked Robotics, 5(1), 5-9 (1996).

[24] K. Taylor and J. Trevelyan, "Australia's Telerobot On The Web", 26th Int. Symp. On Industrial Robots, Singapore (1995). See: http://telerobot.mech.uwa.edu.au/

[25] L.A. Zadeh, "A New Direction in AI - Toward a Computational Theory of Perceptions", AAAI Magazine, 73-84, Spring 2001.

Intelligent Collection Environment for an Interpretation System

William J Maurer
DSP Labs, WJM Inc.
Livermore, California, USA
maurer@dsplabs.com

Abstract

An Intelligent Collection Environment for a data interpretation system is described. The environment accepts two inputs: A data model and a number between 0.0 and 1.0. The data model is as simple as a single word or as complex as a multi-level/multi-dimensional model. The number between 0.0 and 1.0 is a control knob to indicate the user's desire to allow loose matching of the data (things are ambiguous and unknown) versus strict matching of the data (things are precise and known). The environment produces a set of possible interpretations, a set of requirements to further strengthen or to differentiate a particular subset of the possible interpretation from the others, a set of inconsistencies, and a logic map that graphically shows the lines of reasoning used to derive the above output.

The environment is comprised of a knowledge editor, model explorer, expertise server, and the World Wide Web. The *Knowledge Editor* is used by a subject matter expert to define Linguistic Types, Term Sets, detailed explanations, and dynamically created URI's, and to create rule bases using a straight forward hyper matrix representation. The *Model Explorer* allows rapid construction and browsing of multi-level models. A multi-level model is a model whose elements may also be models themselves. The *Expertise Server* is an inference engine used to interpret the data submitted. It incorporates a semantic network knowledge representation, an assumption based truth maintenance system, and a fuzzy logic calculus. It can be extended by employing any classifier (e.g. statistical/neural networks) of complex data types. The *World Wide Web* is an unstructured data space accessed by the URI's supplied as part of the output of the environment.

By recognizing the input data model as a query, the environment serves as a deductive search engine.

Applications include (but are not limited to) interpretation of geophysical phenomena, a navigation aid for very large web sites, monitoring of computer or sensor networks, customer support, trouble shooting, and searching complex digital libraries (e.g. genome libraries).

1. Introduction

In recent years, much effort has gone into studying the problem of computer interpretation of interacting data from multiple sources or sensors [1]. A number of prototype (automated or semi-automated) systems have been developed. Due to the combinatorial explosion of possible data/interpretations, an automated interpretation system prohibits a single algorithm. Automated statistical/neural networks/expert systems have been partially successful. These systems have been beneficial as a good first order filter to classify the data as having obvious or non-obvious interpretations. A limitation of these systems is that when dealing with non-obvious interpretations, the system needs to be integrated better with the user problem solving methods.

Data must be gathered and analyzed in a timely manner to improve the chances of interpretation and not overwhelm or waste the system resources (e.g. battery power). In support of the requirements for accurate and efficient data collection and analysis, the system must be able to draw upon diverse (and potentially limited) data sources. In order to do this thoughtfully, the capability to rank the relative worth of data must be integrated into the system. The value of data in proving or refuting a particular hypothesis and the potential cost in resources to obtain the data must be balanced by the system.

We propose an "Intelligent Collection Environment for an Interpretation System". The environment allows the user to tailor the use of whatever data sources are available at the time of operation. If a data source is unavailable or is not properly configured for a hypothesis, then the system must make do with the data at hand. The system achieves these adaptive data collection and evaluation objectives through the use of "soft computing" techniques. The basic premise underlying soft computing is "Exploit the tolerance for imprecision,

uncertainty and partial truth to achieve tractability, robustness, low solution cost, and better rapport with reality".

The problem falls into a class of problems we define as being semi-structured. By semi-structured we mean problems where human judgment is essential but can be improved by using automation tools. In this approach, we show that non-obvious interpretations require a high degree of interactive analysis and adaptation of methods. In addition to interactive analysis, a general capability of analysis of data and transformations on many scales allows the user to exercise problem-solving methods that are appropriate for the issue at hand. Rapid visualization and direct manipulation of the results allows the user to explore the data space and develop interpretations adaptively. The modern web browser used to access the World Wide Web has proved to be an effective tool for the exploration of unstructured data spaces.

2. Semi-structured problems: structure and approach

The difference between a problem that is structured and one that is unstructured helps to determine where automation tools are appropriate. A solution to a problem that can be programmed is one for which clear rules and a computer program can be defined. The complex process of production scheduling is best dealt with by a linear programming model. There are very definite rules relating inputs to outputs and production to costs, the problem possesses a fundamental deep underlying structure, too complex for the human mind to easily grasp in its detailed entirety but easy for a computer to resolve. At the same time there are many unstructured problems, an extreme example of one is a romance novel. No amount of formal analysis can solve the dilemma.

An unstructured problem does not permit the programming of a solution. The objectives, trade-offs , relevant information, and methodology for analysis cannot be predetermined. Some problems are unstructured simply because of a lack of knowledge or an unwillingness to explore the problem in depth. The degree of potential structure in a problem predefines the procedures, types of computation and analysis, and the information to be used. In a highly unstructured problem, the user must rely on personal judgement, often especially in identifying just what the problem is. There are many differences in the design of a system to support unstructured problems as compared with structured ones. Most obviously, the activities of the user are more central for a system to support unstructured problems. The user initiates and controls the problem solving process and sequence, and uses judgment, personalized objectives, and interpretations to guide the choice of solution. In structured problems, the system will be designed and most of the effort of building the system will be put into the development of routines and sequences of analysis designed to GIVE answers.

3. Browsing in a semi-structured problem domain

A semi-structured problem domain often requires a search for a solution to be found. The solution to be found lies somewhere in the data-space. When the data-space is highly structured, the solution can be retrieved directly by performing some computation. When the data-space is unstructured, the appropriate mechanism for retrieval of a solution is browsing. Browsing is exploratory searching which assumes little knowledge about the structure of the domain being searched [2]. Browsing is available in two different styles.

Navigation is an iterative process in which a user examines the neighborhood of a solution, picks a solution from this neighborhood, examines its neighborhood, and so on.

Probing is a mode in which solutions computed are either a hit (i.e. acceptable) or a miss. When the solution is a miss it can be viewed as being an over-qualification of a solution. In this mode every miss initiates a set of retractions that attempt to broaden the scope of the solution.

Imagine a customer in a store searching for a particular item. The most efficient method to locate this item is to consult a directory and then access the correct shelf. If the customer cannot describe or does not know the item he is looking for, or if the store is not organized in any meaningful way then the customer must apply a *browsing* search technique. Often browsing is done by strolling down the isles, adjusting direction and speed according to the items encountered and the proximity to the desired item. It may also involve hit-and-miss attempts where a customer goes directly to a shelf where he hopes the desired item will be found.

Navigation is analogous to strolling along the aisles of a store. On the other hand, a user who attempts to compute solutions without sufficient familiarity with

the data-space is like the store browser who makes a hit-and-miss attempt by going directly to a shelf in the store. This is analogous to probing. What characterizes probing is that it will fail frequently. What most customers do is use a combination of probing and navigation. The customer may go to a shelf to use as a starting point for navigation.

3.1 Probing using analysis tools

The operations that may be applied to the model objects are *domain independent* and *domain dependent* in nature.

Domain independent operations are used for analysis in many problem domains. These operations don't compute a hit or miss solution as a probe is defined above to. Instead these operations allow the user to compute features which allow the data space to be segmented into regions of interest. Examples of these types of operations are signal processing algorithms and statistical algorithms.

Domain dependent operations are used for analysis in a single domain. These operations do compute a hit or miss solution as a probe is defined above to. These operations allow the user to either use the solution as a starting point for further navigation or to confirm the validity of a solution obtained through navigation. Examples of these types of operations are:

- Model-free estimators such as neural networks that have been trained using features from a particular domain
    - Model-based classifiers such as Bayesian classifiers that have been developed using statistics gathered
    - Rule-based operators used as interpretation tools to allow the user to incorporate non-deterministic feature interpretation operations
    - fuzzy logic operators are used to compute multiple features with a measure of certainty and/or precision.

4.Results

The current environment prototype consists of an interactive model construction and understanding tool for multi level models called Analyst Assistant (AA). The tool, consisting of the World Wide Web, a Java GUI, and a Lisp based expertise server, takes full advantage of the modern web browser and integrates traditional domain independent analysis methods with intelligent domain specific tools for the exploration and analysis of semi-structured problems.

The GUI interface consists of a Knowledge Editor and Model Explorer. The Knowledge Editor is used to define Linguistic Types, Term Sets, detailed explanations, and dynamically created URI's, and to create rule bases using a straight forward hyper matrix representation The Model Explorer allows rapid construction and browsing of multi-level models. By a multi-level model we mean a model whose elements may also be models themselves. One of the features of the Model Explorer is that it can expand a model element. This feature allows the user to probe and recognize meaningful features in the model elements. Models of sub-models and different granularities can be rapidly selected with the mouse interface and analyzed.

The inference engine used to interpret the model incorporates:

- Semantic Network Knowledge Representation [3]
- Assumption-based Truth Maintenance System [4]
- Fuzzy Logic Calculus [5]

Each fact in the Semantic Network has a fuzzy grade that represents how well the value of the fact represents itself as a member of a fuzzy set. A global user-defined grade called the CROSSOVER-POINT is set to a number between 0.0 and 1.0. The CROSSOVER-POINT represents the transition from false to true and is used in pattern matching as follows:

- BELIEVED facts are grade >= CROSSOVER-POINT
- DISBELIEVED facts are grade < CROSSOVER-POINT
- KNOWN facts are grade >= 0
- UNKNOWN facts do not exist

By adjusting the CROSSOVER-POINT, you are allowed to instruct the inference engine to loose match your model with the rule set used to interpret your model (CROSSOVER-POINT closer to 0) or to strict match your model with the rule set used to

interpret your model (CROSSOVER-POINT closer to 1). Usually you will want to start your interpretation of a model in a loose matching mode (things are ambiguous and unknown). As you gain experience with your model and the real world equivalent you will want to interpret your model in a strict matching mode (things are precise and known).

The inference engine interprets your model by applying a rule set that has been previously defined using the Knowledge Editor. The input to the inference engine is:

- the model data
- the names of the rule sets previously defined using the Knowledge Editor
- the value of the CROSSOVER-POINT

Given the input described above, the output interpretation of the
model is an html page full of hyper links and consists of:

- a set of possible interpretations of the data,
- additional data required to strengthen an interpretation or
  differentiate an interpretation from others in the set of possible interpretations, with data ranked according to value,
  - inconsistencies between an interpretation and the data,
  - a logic map that graphically shows the lines of reasoning being used to derive the above output,

In summary, the tools in the system help the user to perform flexible analysis, generate a tentative set of interpretations with further data requirements, and explore the effects of interpretations at different levels of abstraction.

5. References

[1] William J. Maurer, Farid U. Dowla, "Seismic Event Interpretation Using Fuzzy Logic and Neural Networks", Lawrence Livermore National Laboratory, Livermore CA., UCRL-ID- 116130, January 1994.

[2] Motro, A., Browsing in a Loosely Structured Database, SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984, ACM Press, pp. 197-207

[3] Norman, D., Explorations in Cognition, 1975, WH Freeman and Company, pp. 35-59

[4] Johnson, R. R., T. W. Canales, D. L. Lager, C. L. Mason, and R. M. Searfus. (1987). Interpreting Signals with an Assumption-Based Truth Maintenance System. Proc. SPIE, vol. 786, pp. 332-337.

[5] Zadeh, L., The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Information Sciences, 1975 vol. 9, pp. 199-249.

User Profiles and Fuzzy Logic in Web Retrieval

M.J. Martín-Bautista
Dept. of Computer Science and
Artificial Intelligence, Granada University
Granada 18071, Spain
mbautis@decsai.ugr.es

M.A. Vila
Dept. of Computer Science and
Artificial Intelligence, Granada University
Granada 18071, Spain
vila@decsai.ugr.es

D.H. Kraft
Dept. of Computer Science,
Louisiana State University
Baton Rouge, LA 70803-4020, USA
kraft@bit.csc.lsu.edu

J. Chen
Dept. of Computer Science,
Louisiana State University
Baton Rouge, LA 70803-4020, USA
jianhua@bit.csc.lsu.edu

Abstract

We present an approach to utilize user profiles in order to expand queries in information retrieval processes and extract knowledge about groups of interests. A classical user profile is a group of terms extracted for the set of relevant documents for a certain user. A fuzzy representation of terms based on linguistic qualifiers in user profiles is presented in this work. Additional information can be added to the profile related to the user in order to personalize and customize the retrieval process and the web site. Fuzzy clustering of these profiles could be a valid tool for rule construction and fuzzy inference in order to modify queries and extract knowledge from profiles with marketing purposes within a web framework.

1. Introduction

With the appearance of new technologies and the web, the number of non-professional users is increasing everyday. The flexibility and adaptability become key features of web sites to fulfill users' needs. For this purpose, soft computing techniques such as fuzzy logic, genetic algorithms, rough sets and neural networks have been applied [8], [17], [12], [14], [3].

The knowledge about users is stored in profiles. In traditional Information Retrieval Systems (IRS), these profiles contain keywords reflecting the user preferences (simple profiles). This information can be completed with user personal data (extended profiles), as is suggested in [7].

In a web framework, the knowledge about the navigational behavior of the user is stored as additional data in extended profiles. This information can be used for customizing and personalizing of the web sites, with special application in commercial ones.

In this paper, an initial model for constructing extended profiles using fuzzy logic and computing with words is presented. In the following section, the use of fuzzy logic and in traditional models is reviewed. In third and fourth sections, the concept of simple profiles and their extension to include knowledge about the navigational behavior of the users is defined. Customization and personalization of the web sites based on these extended profiles are discussed in the fifth section. Some clustering processes of these profiles and the rule extraction for business users are included in the last sections. Finally, some concluding remarks and future trend lines are proposed.

2. Fuzzy Logic in IRS (Information Retrieval Systems): Computing with Words

Traditional models for IRS construction can be found in the literature [7]; including the vector space model, the probabilistic model, and the Boolean model. In the Boolean model, both the indexing terms and the query terms are evaluated by absence/presence values. Extensions of this and the other models come from the inclusion of Boolean logic and numerical weights on the terms. However, a lack of flexibility in representing the documents has been the main reason for the use of fuzzy

logic. An extended Boolean model with fuzzy logic is presented in [2], where a fuzzy IRS is defined by a function F as:

$$F : DxT \rightarrow [0,1]$$
$$\forall d \in D, t \in T \quad F(d,t) = \mu_d(t)$$

where $D = \{d_1, ..., d_m\}$ is a set of documents and $T = \{t_1, ..., t_n\}$ is a set of terms. The fuzzy extension affects not only documents and term representation, but also the retrieval processes and measures.

It is possible [1] to consider the use of soft computing, i.e., linguistic qualifiers for computing with words, to help retrieval. This in the past has aided the indexing and the querying processes, as well as the matching of a document to a query. However, it can also be employed by the user to provide relevance feedback information. Moreover, mechanisms to evaluate the retrieval system based on the weak orderings of the documents by both the retrieval system and of the user can be considered.

3. Simple User Profiles and Fuzzy Logic

We study the application of fuzzy logic and soft computing to the processes related to user profile construction. In traditional retrieval systems, user profiles consist of a set of terms and a weight to indicate the strength of the user's interest the topic(s) related to that term. These profiles are denominated simple profiles. Terms in these profiles can be extracted from both previous queries and index terms in relevant documents retrieved in response to those queries for the user in question. When we are dealing with a fuzzy IRS, it seems reasonable that each term in a profile has associated with it a fuzzy value signifying the strength of user interest in the topic(s) represented by that term [1]. Given a set of user profiles $Z = \{z_1, ..., z_s\}$, with s the number of profiles and where $z = \{t'_1, ..., t'_c\}$, $t'_i \in T$, $1 \leq i \leq c$, we can define a function

$$G : ZxT \rightarrow [0,1]$$
$$\forall z \in Z, t \in T \quad G(z,t) = \mu_z(t)$$

This function is analogous to the indexing function F, but for user profiles.

3.1. User Profiles in an IRS

The main application of user profiles in retrieval comes from the use of profile terms as query terms, perhaps in an expanded query. As terms in the profile represent user preferences, potentially relevant documents can be retrieved by using a profile as a query, or as an extension to an original query to the system. With the appearance of new technologies and the web, new aspects of user profiles must be considered.

First, user profiles can be used for modifying the query. Some additional knowledge concerning the level of user interest in topics, beyond the terms related to user preferences in queries, must be considered in order to modify the query and thereby improve the retrieval process. In [9] a new approach is given within a fuzzy framework. With the help of a fuzzy clustering process for documents, rules can be constructed relating terms and weights, so that queries can be modified *via* fuzzy inference.

Second, user profiles can be used for filtering and ranking documents. The knowledge of user preferences stored in these profiles is used to filter documents and show the user a subset with the most adequate web sites, ordered decreasingly by grade of relevance. In [11], an intelligent agent for constructing adaptive user profiles using genetic algorithms and fuzzy logic is presented. An alternative approach for on-line news articles using neural networks can be found in [15]

4. Extended User Profiles in the Web

In traditional IRS, extended profiles consist of additional knowledge to terms related to a user's interest. In the web framework, the navigational behavior of users, information related to users as persons, and not just related to their preferences can be collected. In this last case, the knowledge stored in extended profiles after the retrieval process can be used for two purposes.

On the one hand, let consider as an example a health web site set up for commercial purposes. The site can offer to a user some information related to health with the agenda of suggesting some products related to a user's health interests based on the user query. If the user is identified in some way (e.g., nickname or IP address), the health product providers could extract user preferences and personalize the web site according to the user's preferences. Perhaps this could also lead to the introduction of publicizing within the web pages.

On the other hand, when a set of profiles is available, the company can use this information for purposes of marketing. As the number of profiles increases exponentially with time, a clustering process to group them by areas of interest would be needed. A set of rules could be generated in order to inference the relation among

users and terms. A marketing expert can use this information, perhaps via data mining, to connect to social groups based on information extracted from web pages meta-data and user behavior. This information could also be used to customize and identify a non-identified user with a social group, by assigning a general profile related to a preferences shown by the user while web surfing through the web site.

4.1. Registered and Unregistered Users

The main inconvenience of handling of user profiles in the web is the lack of knowledge about user identity. Consider as an example a health web site set up for commercial purposes. Such a site would offer a user access to information related to health. Of course, the site's owners might have an agenda of suggesting to the users that there are some products related to their health interests based on the user queries. If different users connect in web sessions from the same IP address, the profile for each one must be different. Thus, to distinguish which user is surfing, additional information must be provided from the user side, such as a nickname. The availability of this information generates two different situations:

- **Unregistered users:** User profiles can also be used to *personalize* and identify a non-identified user with a social group, by assigning a general profile related to a preferences shown by the user while is surfing through the web site. Personalization is a tool to attract potential customers, so non-identified users with good experiences in the web site could register next time they connect to the site.
- **Registered users:** If a user is identified in some way, the web site can be *customized* according to the user's preferences. From the business point of view, a user has visited the web site before and has registered. The system keeps track of the user from previous visits along with the user profile, so it can use this information to enhance and personalize the web site to maximize user satisfaction with 'her/his web site.' Perhaps this could also lead to the introduction of publicizing the web pages.

5. Extended User Profiles and Fuzzy Logic

Most of the concepts to handle in extended profiles are fuzzy themselves: the links the user follows, the time (dwell) a web page is visible on a user's browser, the country from which is connected, and so forth can give us an idea about the user, how patient the user is, the user's age, the user's language, and so forth. We can at best approximate the age, or the level of language, or express by a linguistic qualifier how patient is the user. These features about users could be modeled by different set of labels, for instance {very low, low, regular, high, very high} to determine the level of the user's ability to communicate in a given language. As these variables can be precise or imprecise, fuzzy techniques manage data with different granularity must be considered [18].

These variables can be determined by combining two different streams of information: *web log files* and *clickstream*. A *web log file* is a group of data from a web server related to the connection, e.g., host, identity and authentication of the user, and the first request. This information can be completed with behavior surfing data, that is, a group of tracks describing when and where is the user clicking at any time. These complementary data are together called a *clickstream*. Each time a user connects to a web site, a new *session* starts. Each click on an URL, image, or general link in a user session represents an entry in the web log file. A session is closed when the elapsed time between two clicks is higher than a pre-fixed threshold [13]. We call the file with the union of web log files information and clickstream information *metadata* file.

5.1. Definition of Extended Profiles

In our model, an extended profile $e_i = (V_i, L_i, K_i, z_i')$ $e_i \in E$, $1 \leq i \leq s$ is formed by several components corresponding to behavioral variables, identification variables, clickstream variables, and the session simple profile, respectively. The obtaining of these components is detailed in the following:

- **Demographic variables:** This set of b variables $V_i = (v_{i1}, v_{i2}, ..., v_{ib})$ is related to demographic and/or social aspects of the user, including the user's age range, the educational level, language skills, and so forth. Most of these variables are imprecise; some can be obtained directly from the user, while others can be deduced from the demographic classes obtained from fuzzy clustering processes of the extended profiles.
- **Identification variables:** This set of c variables $L_i = (l_{i1}, l_{i2}, ..., l_{ic})$ is the set of variables from the web log file about the user identification, such as the host (domain or IP address), user agent (name and version of the browser), and so forth.

These are directly obtainable from the server, and are part of the metadata file corresponding to a certain session user.

- **Clickstream variables:** $K_i = (k_{i1}, k_{i2}, ..., k_{ir})$ represents the set of weights associated to each available $page_j$, $1 \leq j \leq r$, expressed in base of the elapsed time in that page. If $page_j$ is not visited, the value k_{ij} is 0. These variables are extracted from the clickstream data in the metadata file in the same session.

- **Session simple profile:** This component is the result of an aggregation of the profiles associated to each user session $z'_i = \{t'_{i1}, ..., t'_{ic}\}$, $z'_i \in Z$, $t'_{ij} \in T$, $1 \leq j \leq c$ by such aggregation operators as, for instance, the minimum.

It must be take into account that these terms can have a fuzzy value related to their presence in the pages visited by the user, but these values do not represent importance for the user since feedback is not available.

5.2. Fuzzy Clustering of Extended Profiles

Behavioral variables represent the navigational behavior of users in multiple sessions (connection to a web site), which is extracted from the web log files. Similar behavioral variables are expected from general demographic and/or social groups classified by age, profession, gender, and so forth. The imprecision in determining the boundaries between the demographic classes leads us to employ fuzzy clustering methods. We consider a general method for clustering [4], since additional measures based on fuzzy sets to establish a ranking among the different possible partitions are provided.

Let be $C_1, ...C_n$ the demographic classes obtained from the fuzzy clustering. Since each class comes from a group of sessions, and each session has associated a group of session simple profiles, these profiles must be aggregated in order to get a unique profile representative for the class. This *aggregate profile* is a fuzzy set of terms related to the topics in which users in that class are interested.

5.3. Mining Extended Profiles

In our model, fuzzy logic also has an important role in this phase of profile information post-processing. We can describe a demographic group by a class with certain navigational behaviors, favorite web pages, and a simple profile describing the class's preferences. To this end, once the fuzzy clusters of extended profiles have been obtained, classification rules connecting classes to demographic groups should be extracted. Different methods to extract TSK fuzzy classification rules are compared in [5].

Relations between classes and their aggregate profiles also need to be determined. On the one hand, if we have a simple profile available but we know nothing about the user identity or behavior, we could connect that profile to the prototype of the class with an aggregate profile that is more similar to the simple profile. Therefore, a similarity measure for profiles is needed. A survey of some similarity measures can be found in [19]. On the other, direct relations among terms in the aggregate profiles and the classes could also be established from a probabilistic point of view by classifying the aggregate profile terms into the demographic classes. A method to solve this problem can be found in [16].

6. Business Users

The knowledge in the extended profiles can also be used for marketing issues. A marketing organization can interpret and mine the knowledge stored by the system in the user profiles, and use it for marketing purposes related to electronic commerce. A new kind of users must be considered: business users.

Business users can use the information in the extended profiles, perhaps via data mining, from the set of rules extracted from the clustering of profiles by areas of interest. Once the rules are obtained, business users can identify social groups based on the behavioral components of the rules. However, the lack of understanding between mining analysts, who speaks a different technical language, and business users, as well as the difficulty of understand the generated rules by business users, lead to the developing of software products for data mining. Research in this area is perhaps most interesting from the business point of view, since companies can make their marketing organizations aware of which areas of interest exist among their customers. An approach for this purpose is given in [10], where the authors present a knowledge discovery assistant for helping business users with mining issues of data extracted from customer groups of interest.

7. Conclusions

We have presented an initial model to construct user profiles, based on some prior research. We consider the use of soft computing, i.e., computing with words,

and fuzzy logic to construct and incorporate user profiles for purposes of web retrieval. Simple profiles in traditional retrieval systems are extended to include behavioral components of the navigators. The clustering and mining of these profiles allow business users in marketing organizations to extract knowledge about customer groups of interest for commercial purposes, i.e., e-commerce.

References

[1] F. Berzal, H.L. Larsen, M.J. Martín-Bautista and M.A. Vila, "Computing with Words in Information Retrieval". *Proc. of IFSA/NAFIPS International Conference*, Vancouver, Canada, July 2001 (to appear).

[2] D.A. Buell and D.H. Kraft, "A Model for a Weighted Retrieval System". *Journal of the American Society for Information Science*, Vol. 32(3), 1981, pp 211-216.

[3] H. Chen, "Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms". *Journal of the American Society for Information Science*, Vol. 46(3), 1995, pp 194-216.

[4] M. Delgado, A.F. Gómez-Skarmeta, M.A. Vila, "On the Use of Hierarchical Clustering in Fuzzy Modeling". *International Journal of Approximate Reasoning*, Vol. 14, 1996, pp 237-257.

[5] A.F. Gómez-Skarmeta, M. Delgado, M.A. Vila, "About the use of fuzzy clustering techniques for fuzzy model identification". *Fuzzy Sets and Systems*, Vol. 106, 1999, pp 179-188.

[6] R. Kimball and R. Merz, *The Data Webhouse Toolkit*, John Wiley and Sons, 2000, USA.

[7] R.R. Korfhage, *Information Storage and Retrieval*, John Wiley and Sons, 1997, New York, USA.

[8] D.H. Kraft, G. Bordogna, G. Pasi, "Fuzzy Set Techniques in Information Retrieval". In Dubois, D. and Prade H (eds.) *Handbook of Fuzzy Sets (vol. 3): Approximate Reasoning and Information Systems*. Kluwer Academic Publishers, The Netherlands, 1999, pp 469-510.

[9] D.H. Kraft and J. Chen, "Integrating and Extending Fuzzy Clustering and Inferencing to Improve Text Retrieval Performance" In *Flexible Query Answering Systems: Recent Advances, Proceedings of the Fourth International Conference on Flexible Query Answering Systems, FQAS'2000*, October 25-28, 2000, Warsaw, Poland. Physica-Verlag, Heidelberg, Germany, 2000, pp 386-395.

[10] O. Hogl, H. Stoyan, M. Müller, "The Knowledge Discovery Assistant: Making Data Mining Available for Business Users". *Proc. 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Dallas, Texas, 2000.

[11] M.J. Martín-Bautista, M.A. Vila, H.L. Larsen, "A fuzzy genetic algorithm approach to an adaptive information retrieval agent". *Journal of the American Society for Information Science*, Vol. 50(9), 1999, pp 760-771.

[12] M.J. Martín-Bautista, M.A. Vila , D. Sánchez, H.L. Larsen, "Intelligent filtering with genetic algorithms and fuzzy logic". In Bouchon-Meunier, B., Gutiérrez-Ríos, J., Magdalena, L., and Yager, R.R. (eds.) *Technologies for Constructing Intelligent Systems*, Springer-Verlag, (in press).

[13] O. Nasraoui, H. Frigui, R. Krishnapuram, A. Joshi, "Extracting web user profiles using relational competitive fuzzy clustering". *International Journal on Artificial Intelligence Tools*, Vol. 9(4), 2000, pp 509-526.

[14] P. Srinivasan, M.E. Ruiz, D.H. Kraft, J. Chen, "Vocabulary mining for information retrieval: rough sets and fuzzy sets". *Information Processing and Management*, Vol. 37, 2001, pp 15-38.

[15] A. Tan, C. Teo, "Learning User Profiles for Personalized Information Dissemination". *Proceedings of the International Joint Conference on Neural Networks*, Alaska, 1998, pp 183-188.

[16] M.A. Vila, M. Delgado, A.F. Gómez-Skarmeta, "Pattern Recognition with Evidential Knowledge". *International Journal of Intelligent Systems*, Vol. 14, 1999, pp 145-164.

[17] J. Yang, R.R. Korfhage, "Query Optimization in Information Retrieval using Genetic Algorithms". *Proc. of the Fifth International Conference of Genetic Algorithms*, Urbana-Champaign, Illinois, 1993, pp 603-611.

[18] L.A. Zadeh, "The Concept of a Linguistic Variable and its Application to Approximate Reasoning I, II and III". *Information Sciences*, Vol. 8, 1975, pp 199-251; Vol. 9, 1975, pp 43-80.

[19] R. Zwick, E. Carlstein, D.V. Budescu, "Measures of Similarity Among Fuzzy Concepts: A Comparative Analysis". *International Journal of Approximate Reasoning,* Vol. 1, 1987, pp 221-242.

Search Engines: Key to Knowledge Acquisition

Mori Anvari
EECS Department
UC Berkeley
Berkeley CA 94720
anvari@eecs.berkeley.edu

Abstract. There are many common complaints about the existing search engines:

1. Too many documents are retrieved in response to queries.
2. Most of the retrieved documents are not sufficiently relevant in relation to the inquirer's intent.
3. The outcome of a query is often "an ocean of unedited data", which makes it impractical and time-consuming to "separate the wheat from the chaff".
4. As of this writing, search engines do not handle synonyms, antonyms, and polysems (same word having distinct meanings).

Several search engines allow the user to make refinements to queries. However, most users would normally key-in a word or a phrase without bothering to learn the refinement features of various search engines. This includes the use of AND, OR, NOT, NEAR, nested searches, wildcards, and stopwords. In recent months, several search engines have claimed to use fuzzy logic as a part of their search algorithms. What these search engines use is a form of lexical approximation, which reflects either the spelling of a word or its distance in the document from some other word. For instance in Altavista, NEAR means that two words are situated within 10 words of each other in the document. LYCOS defines NEAR to mean within 25 words.

1. Case Studies

Using Google, we searched the web for documents about "Mongol genocide". It resulted in 1,420 documents. The search took 0.05 seconds. Using excite.com, the same query returned 21,470 documents. We then used excite.com's precision search. This time, we specified that the phrase Mongol genocide must appear in the title and the terms jewish, Cambodia, and holocaust must not be included in the title. The outcome of the query comprised three documents, none of which had the phrase Mongol genocide in the title or the body of the document. One of the documents contained a large number of exchanged email messages which had absolutely nothing to do with the search .

We then made the query 'missile defense' on Google. It returned ½ a million documents. The same query on Excite.com returned 363,905 documents. There is more literature available concerning the workings of Google than any other search engine. We, therefore, devote the next section to how Google works.

2. Google: An Overview

Google runs on a unique combination of advanced hardware and software. The speed you experience can be attributed in part to the efficiency of the search algorithm and partly to the thousands of low cost PC's that have been networked together to create a superfast search engine.

The heart of the software is PageRank(TM), a system for ranking web pages developed by Google's founders, Larry Page and Sergey Brin at Stanford University. And while dozens of engineers are working to improve every aspect of Google on a daily basis, PageRank continues to provide the basis for all of web search tools. PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links that a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important."

Important, high-quality sites receive a higher PageRank, which Google remembers each time it conducts a search. Of course, important pages mean nothing to you if they don't match your query. So, Google combines PageRank with sophisticated text-matching techniques to find pages that are both important and relevant to your search. Google goes far beyond the number of times a term appears on a page and examines all aspects of the page's content (and the content of the pages linking to it) to determine if it's a good match for your query.

In an article (see the first item in References) entitled "The Anatomy of a Large-Scale Hypertextual Web Search Engine", the fonders of Google, Sergey Brin and Lawrence Page describe the workings of Google in sufficient detail. In what follows, we will mention the highlights of the article.

Google, as a prototype of a large-scale search engine, makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/. To construct a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago. Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. The article addresses the question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to deal effectively with uncontrolled hypertext

collections where anyone can publish anything they want.

The web creates new challenges for information retrieval. The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the art of web research. People are likely to surf the web using its link graph, often starting with high quality human maintained indices such as Yahoo! or with search engines. Human maintained lists cover popular topics effectively but are subjective, expensive to build and maintain, slow to improve, and cannot cover all esoteric topics. Automated search engines that rely on keyword matching usually return too many low quality matches. To make matters worse, some advertisers attempt to gain people's attention by taking measures meant to mislead automated search engines.

3. Google's Internals

The Google search engine has two important features that help it produce high precision results. First, it makes use of the link structure of the Web to calculate a quality ranking for each web page. This ranking is called PageRank which is described below. Second, Google utilizes link to improve search results.

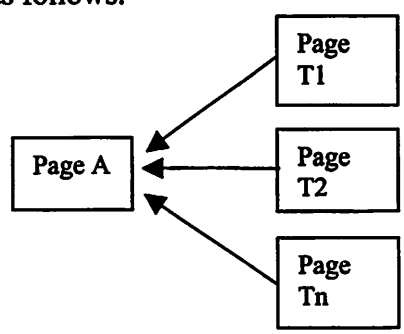
PageRank: Bringing Order to the Web

The citation (link) graph of the web is an important resource that has largely gone unused in existing web search engines. There are maps containing as many as 518 million of these hyperlinks, a significant sample of the total. These maps allow rapid calculation of a web page's "PageRank", an objective measure of its citation importance that corresponds well with people's subjective idea of importance. Because of this correspondence, PageRank is an excellent way to prioritize the results of web keyword searches. For most popular subjects, a simple text matching search that is restricted to web page titles performs admirably when PageRank prioritizes the results (demo available at google.stanford.edu). For the type of full text searches in the main Google system, PageRank also helps a great deal.

Academic citation literature has been applied to the web, largely by counting citations or backlinks to a given page. This gives some approximation of a page's importance or quality. PageRank extends this idea by not counting links from all pages equally, and by normalizing by the number of links on a page. PageRank is defined as follows:

We assume page A has pages T1...Tn, which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. Usually d is

to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:



$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

Note that the PageRanks form a probability distribution over web pages, so the sum of PageRanks of all web pages will add up to one.

PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web. Also, a PageRank for 26 million web pages can be computed in a few hours on a medium size workstation. There are many other details which are beyond the scope of this paper. PageRank can be thought of as a model of user behavior. We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page. The probability that the random surfer visits a page is its PageRank. And, the d damping factor is the probability at each page the "random surfer" will get bored and request another random page. One important variation is to only add the damping factor d to a single page, or a group of pages. This allows for personalization and can make it nearly impossible to deliberately mislead the system in order to get a higher ranking. Another intuitive justification is that a page can have a high PageRank if there are many pages that point to it, or if there are some pages that point to it and have a high PageRank. Intuitively, pages that are well cited from many places around the web are worth looking at. Also, pages that have perhaps only one citation from something like the Yahoo! homepage are also generally worth looking at. If a page was not high quality, or was a broken link, it is quite likely that Yahoo's homepage would not link to it. PageRank handles both these cases and everything in between by recursively propagating weights through the link structure of the web.

4. AI and Search Engines

Most users of search engines work in a few fairly well-defined knowledge domains, such as nutrition, real-estate law, or chip technology. We propose to design and implement a prototype of an intelligent front-end to enhance the usability and functionality of existing search engines. It will be user-centered, in the sense that users can refine their search based on their perception of relevance. We anticipate a few different approaches to attain a reliable and scalable solution. One approach is to construct an ontology (or meta-thesaurus) for a specific knowledge domain and restrict the search to that knowledge domain. One of the values of ontology lies in the fact that in a web search we can access not only the documents that are normally returned in response to keywords, but also the ones that are ontologically related to the keywords. Another value is that different people often use different terms for the same object or concept. Ontology can accommodate such different users by encapsulating related concepts and object referents. A user can narrow the search by a relevancy hierarchy that is derived from the ontology. We plan to generate a knowledge model that would incorporate knowledge representation schemes that are currently available in the industry such as RDF, DAML, and DAML + OIL. We envision incorporating inference capability in the search engine. Also, as XML is widely used for support of interoperability and information sharing, the ontology needs to work through XML interfaces. Since ontologies continue to evolve, a versioning system is critical for ontologies. An ap-

plication is based on a particular version, hence the application must know on what version it is based. We believe that the incorporation of inference capability will radically improve performance of a search engine. Besides the bivalent logic, there is a myriad of useful logics, e.g. temporal logic and modal logic, that are commonly used in reasoning with common sense knowledge. As a part of this initiative, we plan to explore the level of usefulness of these logics in relation to ontology and search engines.

References

[Das 01] Aseem Das et al: Industrial Strengh Ontology Management, SWWS'01, Stanford University

[Anvari 01] Mori Anvari, Intelligent Information Systems, Research and Software Development funded by British Telecom.

[Voorhes 94] Best of the Web 1994 -- Navigators http://botw.org/1994/awards/navigators.html and E. M. Voorhees. Full text at: http://trec.nist.gov/

[Witten 94] Ian H Witten, Alistair Moffat, and Timothy C. Bell. Managing Gigabytes: Compressing and Indexing Documents and Images. New York: Van Nostrand Reinhold, 1994.

[Weiss 96] Ron Weiss, Bienvenido Velez, Mark A. Sheldon, Chanathip Manprempre, Peter Szilagyi, Andrzej Duda, and David K. Gifford. HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering. Proceedings of the 7th ACM Conference on Hypertext. New York, 1996.

Presentations of Wednesday, August 15

Database Querying, Ontology, Content Management and Matching Technologies

FUZZY LOGIC AND THE INTERNET (FLINT)

Masoud Nikravesh

BISC Program,
EECS Department-CS Division
University of California
Berkeley, CA 94720

Background

Humans have a remarkable capability (perception) to perform a wide variety of physical and mental tasks without any measurements or computations. Familiar examples of such tasks are: playing golf, assessing wine, recognizing distorted speech, and summarizing a story. The question is whether a special type information retrieval processing strategy can be designed that build in perception.

Here is what Tim Bernres-Lee (Father of WWW) would like to add (Transcript of Tim Berners-Lee's talk to the LCS 35th Anniversary celebrations, Cambridge Massachusetts, 1999/April/14):

"How well are we doing? Are we doing human communication through shared knowledge? Let's look through the document side. On this side the languages are natural language. They're people talking to people. So the language is you just can't analyze them very well. And this is the big problem on the net for a lot of people, is the problem for my mother and your mother and our kids. They go out to search engines and they ask a question and the search engine gives these stupid answers. It has read a large proportion of the pages on the entire Web (which is of course amazing) but it doesn't understand any of them — and it tries to answer the question on that basis. Obviously you get pretty unpredictable results. However, the wonderful thing is that when people communicate in this way, this kind of fuzzy way, people can solve problems intuitively. When people browse across the Web and see something expressed in natural language, they think, "Aha!" and suddenly solve a totally unrelated problem due to the incredible ability that the human brain has to spot a pattern totally out of context by a huge amount of parallel processing."

Retrieving relevant information is a crucial component of cased-based reasoning systems for Internet applications such as search engines. The task is to use user-defined queries to retrieve useful information according to certain measures. Even though techniques exist for locating exact matches, finding relevant partial matches might be a problem. It may not be also easy to specify query requests precisely and completely - resulting in a situation known as a fuzzy-querying. It is usually not a problem for small domains, but for large repositories such as World Wide Web, a request specification becomes a bottleneck. Thus, a flexible retrieval algorithm is required, allowing for imprecise or fuzzy query specification or search.

Here is what Professor Lotfi A. Zadeh would like to add:

"Unlike classical logic, fuzzy logic is concerned, in the main, with modes of reasoning which are approximate rather than exact. In Internet, almost everything, especially in the realm of search, is approximate in nature. Putting these two facts together, an intriguing thought merges; in time, fuzzy logic may replace classical logic as what may be called the brainware of the Internet.

...

In my view, among the many ways in which fuzzy logic may be employed, there are two that stand out in importance. The first is search. Another, and less obvious, is deduction in an unstructured and imprecise environment. Existing search engines have zero deductive capability. ...

To add a deductive capability to a search engine, the use of fuzzy logic is not an option - it is a necessity."

Here is what Tim Bernres-Lee (Father of WWW) would like to add:

"The FOPC inference model is extremely intolerant of inconsistency [i.e. P(x) & NOT (P(X)) -> Q], the semantic web has to tolerate many kinds of inconsistency. Toleration of inconsistency can only be done by fuzzy systems. We need a semantic web which will provide guarantees, and about which one can reason with logic. (A fuzzy system might be good for finding a proof -- but then it should be able to go back and justify each deduction logically to produce a proof in the unifying HOL language, which anyone can check) Any real SW system will work not by believing anything it reads on the web but by checking the source of any information. (I wish people would learn to do this on the Web as it is!). So in fact, a rule will allow a system to infer things only from statements of a particular

form signed by particular keys. Within such a system, an inconsistency is a serious problem, not something to work around. If my bank says my bank balance is $100 and my computer says it is $200, then we need to figure out the problem. Same with launching missiles, IMHO. The semantic web model is that a URI dereferences to a document which parses to a directed labeled graph of statements. The statements can have URIs as parameters, so they can may statements about documents and about other statements. So you can express trust and reason about it, and limit your information to trusted consistent data."

Approach

The theory which is being developed for the special types of information processing and information retrieval put forth in this project is focused on the development of what is referred to as the *computational theory of perception* (CTP) -- a theory which comprises a conceptual framework and a methodology for computing and reasoning with perceptions. The base for CTP is the methodology of computing with words (CW). In CW, the objects of computation are words and propositions drawn from a natural language
The main problem with conventional vector space representation of term-document vectors are that 1) there is no real theoretical basis for the assumption of a term and document space and 2) terms and documents are not really orthogonal dimensions. These techniques are used more for visualization and most similarity measures work about the same regardless of model. In addition, terms are not independent of all other terms. With regards to probabilistic models, important indicators of relevance may not be term -- though terms only are usually used. Regarding Boolean model, complex query syntax is often misunderstood and problems of null output and Information overload exist. One solution to these problems is to use extended Boolean model or fuzzy logic. In this case, one can add a fuzzy quantifier to each term or concept. In addition, one can interpret the AND as fuzzy-MIN and OR as fuzzy-MAX functions. Alternatively, one can add agents in the user interface and assign certain tasks to them or use machine learning to learn user behavior or preferences to improve performance. This technique is useful when past behavior is a useful predictor of the future and wide variety of behaviors amongst users exist.

Perception-Based Information Processing for Internet: One of the problems that Internet users are facing today is to find the desired information correctly and effectively in an environment that the available information, the repositories of information, indexing, and tools are all dynamic. Even though some tools were developed for a dynamic environment, they are suffering from "too much" or " too little" information retrieval. Some tools return too few resources and some tool returns too many resources ***(Figures 1.).***

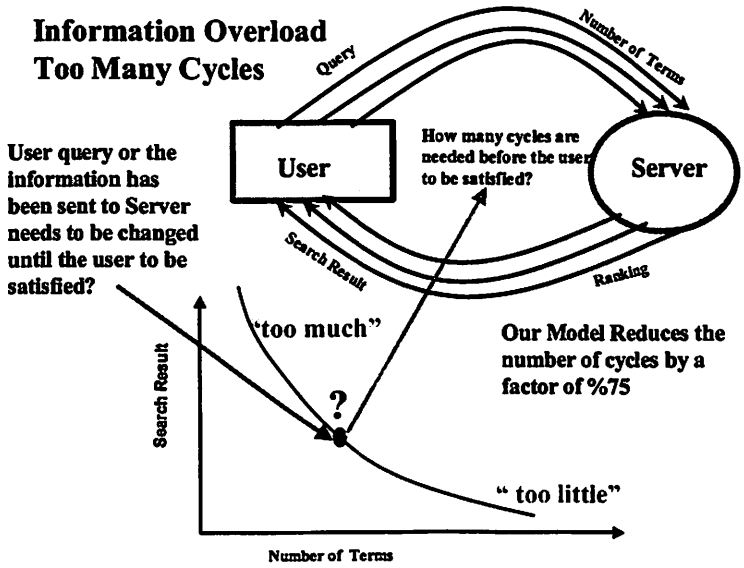


Figure 1. Information overload

In addition, the user's perception, which is one of the most important key features, is oftentimes ignored. For example, consider the word " football". The perception of an American differs from the perception of a European who understands football to mean "Soccer." Therefore, if the search engine knows something about the user and its perception, it might be able to better refine the users results. For this example, there is no need to eliminate American football pages for those in the UK looking for real football information, since this information inclusively exists in user's profile. Search Engines also often return a large list of irrelevant search results due to the ambiguity of search query terms. To solve this problem one can use the following approaches 1) from Users Side/ Client Side by selecting a very specific (unique) term and 2) from Systems Sides/Server by offering alternate query terms for users to refine the query terms (Figures 2 and 3).

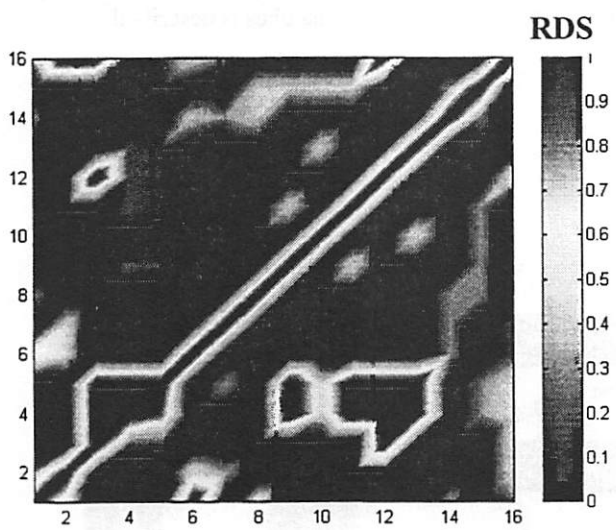


Figure 2. Terms Similarity; Automated Ontology Generation and Automated Indexing
The ontology is automatically constructed from text document collection and can be used for Query Refinement.

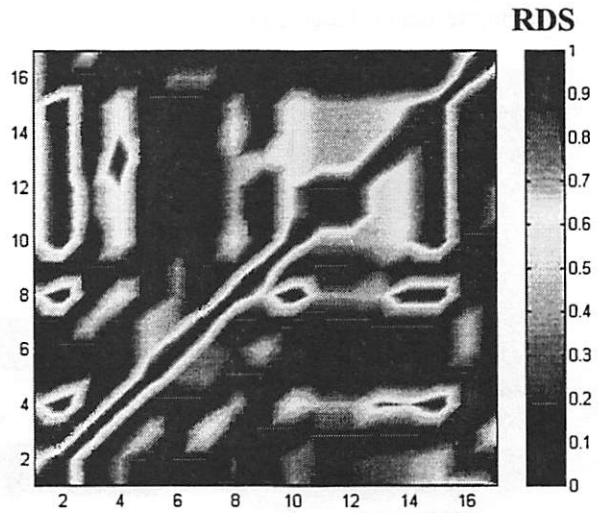


Figure 3. Documents Similarity; Search Personalization-User Profiling
Often time it is hard to find the "right" term and even in some cases the term does not exist.
The User Profile is automatically constructed from text document collection and can be used for Query Refinement and provide suggestions and for ranking the information based on pre-existence user profile.

The main goal of this project is to develop perception- based information processes and retrieval system for the internet based on user profile with capability of exchanging and updating the rules dynamically and ***"do what I mean, not as I say"*** and using programming with ***"human common sense capability"***. ***Figure 4.*** shows the structure of search engine and retrieval technique and the problem related to perception and areas that soft computing can be used as a mean for improvement.

Perception Based Search Engine: World Wide Web search engines have become the most heavily-used online services, with millions of searches performed each day. Their popularity is due, in part, to their ease of use. Already explosive amount of users on the Internet is estimated over 200 million (Table 1). The estimated user of wireless devices is estimated 1 billion within 2003 and 95 % of all wireless devices will be Internet enabled within 2005.

Jan 1998: 30 Millions web hosts
Jan 1999: 44 Millions web hosts
Jan 2000: 70 Millions web hosts
Feb 2000: +72 Millions web hosts

Dec 1997: 320 Millions
Feb 1999: 800 Millions
March 2000: +1,720 Millions

The number of pages available on the Internet almost doubles every year

Table 1. Internet and Rate of Changes

Courtois and Berry (Martin P. Courtois and Michael W. Berry, ONLINE, May 1999-Copyright © Online Inc.) published a very interesting paper "Results Ranking in Web Search Engines". In their work for each search, the following topics were selected: credit card fraud, quantity theory of money, liberation tigers, evolutionary psychology, french and indian war, classical greek philosophy, beowulf criticism, abstract expressionism, tilt up concrete, latent semantic indexing, fm synthesis, pyloric stenosis, and the first 20 and 100 items were downloaded using the search engine. Three criteria 1) All Terms, 2) Proximity, and 3) Location were used as a major for testing the relevancy ranking. Following Table shows the summary of their results. Many search engines support Boolean operators, field searching, and other advanced techniques. While searches may retrieve thousands of hits, finding relevant partial matches might be a problem. In this project (Fuzzy Logic and the Internet; FLINT), a new fuzzy clustering technique based on perception for automatic information retrieval with partial matches is described.

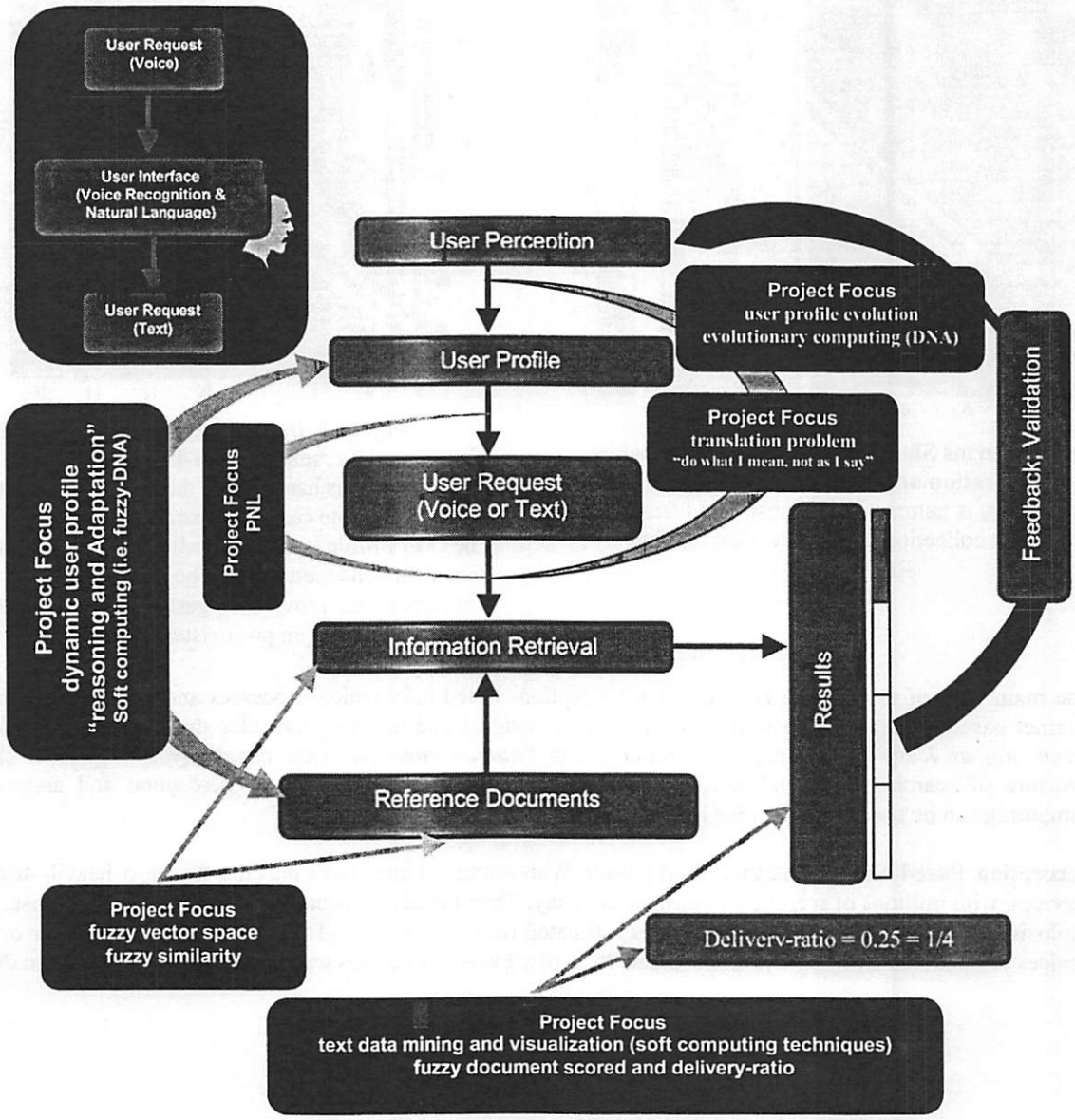


Figure 4. Structure of search engine and retrieval technique and the problem related to perception and areas that soft computing can be used as a mean for improvement.

Table 2. Results Ranking in Web Search Engines

Criteria	All Terms		Proximity		Location	
First 20 and 100 items	100 hits	20 hits	100 hits	20 hits	100 hits	20 hits
ALTAVISTA	31.0%	12.9%	11.1%	7.7%	40.5%	10.4%
EXCITE	18.6%	5.0%	28.2%	23.5%	77.3%	53.2%
HOTBOT	19.5%	12.3%	40.1%	23.5%	61.7%	28.8%
INFOSEEK	23.4%	16.0%	14.5%	9.5%	78.6%	50.1%
LYCOS	8.4%	5.4%	48.7%	26.3%	69.2%	31.9%

Table 3. Examples of Fuzzy Web Search Engines

Search Engine	Simple Form	Search Logic				Fuzzy Logic in any form	Term Weighting	Sorted Output	Ranked output	Find Like
		Boolean	Proximity	Nesting	Truncation					
Excite!	X	X	X	X		X	X	X	X	X
AltaVista	X	X	X	X	X			X	X	
HotBot		X	X	X		X	X		X	
Infoseek	X	X	X	X	X	X	X	X	X	
Lycos	X				X	X*			X	
Open Text		X	X	X					X	(X)
Web Crawler	X	X	X	X		X	X	X	X	
Yahoo	X	X	X	X			X			
Google	X	X	*	*	*	X	*	*	*	*
Northern Light Power	X	X	*	*	*	X	*	*	*	*
Fast Search Advanced	X	X	*	*	*	X	*	*	*	*

* No information provided

Potential Applications:

1. Search Engines and Web Crawlers
2. Agent Technology (i.e., Web-Based Collaborative and Distributed Agents)
3. Adaptive and Evolutionary techniques for dynamic environment (i.e. Evolutionary search engine and text retrieval, Dynamic learning and adaptation of the Web Databases, etc)
4. Fuzzy Queries in Multimedia Database Systems
5. Query Based on User Profile
6. Information Retrievals
7. Summary of Documents
8. Information Fusion Such as Medical Records, Research Papers, News, etc
9. Files and Folder Organizer

10. Data Management for Mobile Applications and eBusiness Mobile Solutions over the Web
11. Matching People, Interests, Products, etc
12. Association Rule Mining for Terms-Documents and Text Mining
13. E-mail Notification
14. Web-Based Calendar Manager
15. Web-Based Telephony
16. Web-Based Call Centre
17. Workgroup Messages
18. E-Mail and Web-Mail
19. Web-Based Personal Info
20. Internet related issues such as Information overload and load balancing, Wireless Internet-coding and D-coding (Encryption), Security such as Web security and Wireless/Embedded Web Security, Web-based Fraud detection and prediction, Recognition, issues related to E-commerce and E-bussiness, etc.

References

1. L.A. Zadeh, "The role of fuzzy logic and soft computing in the conception, design and deployment of intelligent systems", *BT technol J,* 14 (4) 32-36 (1996).
2. L.A. Zadeh, "From computing with numbers to computing with words to computation with Perceptions--A paradigm shift", *Berkeley project.* The ERL research summary for 2000, chapter 10: Artificial intelligence. University of California, Berkeley.
3. M. NikRavesh, F. Aminzadeh, L. A. Zadeh, " Perception-based cooperative robots", *Frontiers in robotics and intelligent machines, complex information processing, and computational nanoscience, Proposal for continued basic research,* CESAR-Oak Ridge National Laboratory-DOE, 47-50 (2000).
4. M. Nikravesh, Lotfi A. Zadeh, "Perception based information processing and retrieval application to user profiling", Berkeley-BT project, 2000-2004.
5. L. A. Zadeh, " Toward a perception-based theory of probabilistic reasoning with imprecise probabilities, to be appear in *The Journal of Statistical Planning and Inference.*
6. L. A. Zadeh, " A new direction in AI – Toward a computational theory of perceptions", to be appear in *The Journal of Artificial Intelligence.*

Fuzzy Detection of Network Intrusions

Seyed A. Shahrestani

School of Computing and Information Technology
University of Western Sydney
Penrith Campus, Locked Bag 1797
PENRITH SOUTH DC NSW 1797
AUSTRALIA
seyed@ieee.org

Abstract
The networking has brought about limitless possibilities and opportunities along with increased risks and chances of malicious intrusions. It is therefore, critical to have the security mechanisms that are able to prevent unauthorized access to system resources and data. However, complete prevention of security breaches does not appear to be practical. Intrusion detection can be regarded as an alternative, or as a compromise to this situation. Several techniques for detecting intrusions have been studied by many researchers. This paper discusses why intrusion detection systems are needed and presents the main techniques that these systems utilize. In any of these techniques, the need for exploiting the tolerance for imprecision and uncertainty to achieve robustness and low solution costs is evident. It can be noted that, this is in fact, the guiding principle of soft computing and more particularly fuzzy logic introduced by Zadeh. This work reports the preliminary steps taken in the study of the implications and advantages of using fuzzy logic for handling intrusion detection through approximate reasoning and approximate matching.

1. Introduction

In general, an intrusion attempt is defined as the potential possibility of a deliberate unauthorized attempt to access or manipulate information, or render a system unreliable or unusable. An intrusion detection system (IDS) is a tool that attempts to perform intrusion detection. While the complexities of host computers are already making intrusion detection a difficult task, the increasing prevalence of distributed networked-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection [1].

All Internet-based and intranet-based computer systems are vulnerable to intrusions and abuse by both legitimate users (who abuse their authorities) and unauthorized individuals. With the rapidly increasing dependence of businesses and government agencies on their computer networks, protecting these systems from intrusions is critical. These are the facets of the problem: the personal computer and the Internet have become indispensable parts of everyday lives, while they are exceedingly vulnerable to even simple attacks. The vulnerability of some of these systems stems from the simple fact that they were never intended for a massive interconnection.

An IDS assumes that an intruder's behavior will be noticeably different from that of a legitimate user. It also assumes that many unauthorized actions are detectable. Two major approaches for detecting computer security intrusions in real time are misuse detection and anomaly detection. Misuse detection attempts to detect known attacks against computer systems. Anomaly detection uses knowledge of users' normal behavior to detect attempted attacks. The primary advantage of anomaly detection over misuse detection methods is the ability to detect novel and unknown intrusion. These approaches are further discussed in the next section. Section 3 gives an overview of an intrusion detection approach that is inspired by the principles of natural immune systems. A discussion on research needs of the field demonstrating the importance of fuzzy intrusion detection is presented in Section 4. The last section gives the concluding remarks.

2. Intrusion Models and Detection Algorithms

Typically, IDSs employ statistical anomaly and rule-based misuse models in order to detect intrusions. The detection in statistical anomaly model is based on the profile of normal user's behavior. It will statistically analyse the parameters of the user's current session and compares them to the user's normal behavior. Any significant deviation between the two is regarded as a suspicious session. As the main aim of this approach is to catch sessions that are not normal, it is also referred to as an 'anomaly'

detection model. The second model is dependent on a rule-base of techniques that are known to be used by attackers to penetrate. Comparing the parameters of the user's session with this rule-base carries out the actual act of intrusion detection. This model is sometimes referred to as a misuse detection model, as it essentially looks for patterns of misuse- patterns known to cause security problem [2]. This section takes a closer look at these models.

2.1. Statistical Anomaly Detection

Statistical anomaly detection systems initiate the detection of the security breaches by analysing the audit-log data for abnormal user and system behavior. These systems assume that such abnormal behavior is indicative of an attack being carried out. An anomaly detection system will therefore attempt to recognizing the occurrence of 'out of the ordinary' events. For implementation purposes, the first step is concerned with building a statistical base for intrusion detection that contains profiles of normal user and system behavior. Based on that, these systems can then adaptively expand their statistical base by learning user and system behavior. This model of intrusion detection is essentially based on Pattern recognition approaches, i.e. the ability to perceive structure in some data.

2.2. Recognition of Intrusive Patterns

To carry out the pattern recognition act, the raw input data is pre-processed to form a pattern. A pattern is an extract of information regarding various characteristics or features of an object, state of a system, etc. Patterns either implicitly or explicitly contain names and values of features, and if they exist, relationships among features. The entire act of recognition can be carried out in two steps. In the first step a particular manifestation of an object is described in terms of suitably selected features. The second step, which is much easier than the first one, is to define and implement an unambiguous mapping of these features into class-membership space.

Patterns whose feature values are real numbers (continuous or discrete) can be viewed as vectors in n-dimensional space, where n is the number of features in each pattern. With this representation, each pattern corresponds to a point in the n-dimensional *metric* feature space. In such a space, distance between two points, Euclidean distance being one example, indicates similarities (or differences) of the corresponding two patterns. Generally speaking, the key problem is reduction of the dimensionality of the feature vector (and space). Partitioning the feature space then carries out the actual decision making act (classification) by any of the many available methods; e.g. maximum likelihood, K-nearest neighbors, decision surfaces and discriminate functions. This approach to pattern recognition is generally considered as *statistical* (or decision theoretic).

More specifically for intrusion detection purposes, the statistical analysis detects variation in a user's behavior by looking for significant changes in the session in comparison to user's behavior profiles or patterns. The profiles consist of the individual behavior in previous sessions and serve as a means for representing the expected behavior. Obviously, the information content of the patterns that make up the profiles need to be dynamically updated. For intrusion detection purposes, various types of subjects may need to be considered and monitored. These may include users, groups, remote hosts, and overall target systems. Monitoring groups enables the detection system to single out an individual whose behavior significantly deviates from the overall average group behavior. Detection of system wide deviations in behavior that are not connected to a single user may be achieved by monitoring the target system. For instance, a large deviation in the number of system wide login attempts may be related to an intrusion.

To determine whether the behavior is normal or not, it is characterized in terms of some of its key features. The key features are then applied to individual sessions. While the features employed within different intrusion detection systems may vary substantially, they may be categorised as either a continuous or a discrete feature. A continuous feature is a function of some quantifiable aspect of the behavior such that during the course of the session its value varies continuously. Connection time is an example of this type of feature. This is in contrast to a discrete feature that will necessarily belong to a set of finite values. An example of such a feature is the set of terminal location. For each subject, the maintained profile is a collection of the subject's normal expected behavior during a session described in terms of suitably selected features.

The classification process to determine whether the behavior is anomalous or not is based on statistical evaluations of the patterns stored as profiles specific for each subject. Each session is described by a pattern (usually represented as a vector of real

numbers) consisting of the values of the features pre-selected for intrusion detection. The pattern corresponds to the same type of features recorded in the profiles. With the arrival of each audit record, the relevant profiles are solicited and their contents (the patterns they contain) are compared with the pattern (vector) of intrusion detection features. If the point defined by the session vector in the n-dimensional space is far enough from the points corresponding to the vectors stored in the profiles, then the audit record is considered to be anomalous. It can be noted that while the classification is based on the overall pattern of usage (the vector), highly significant deviations of the value of a single feature can also result in the behavior being considered as anomalous.

2.3. Important Characteristics

To be useful, the intrusion detection system must maximize the true positive rate and minimize the false positive rate. In most cases (but not all), achieving a very low false positive rate (i.e. a low percentage of normal use classified incorrectly as anomalous) is considered more crucial. This can be achieved by changing the threshold of the distance metric that is used for classifying the session vector. By raising this threshold, the false positive rate will be reduced while this will also lower the true positive rate (i.e. fewer events are considered abnormal).

To increase speed and to reduce misclassification error, particularly when the number of classes (e.g. the number of users) is large or not known, some suggestions have been made for grouping of classes. For example, patterns can be mapped into a generalized indicator vector, on the basis of their similarities. This vector is then used in conjunction with a standard search tree method for identification purposes. Another method first computes a similarity measure- based on distance metric- between each pattern and every other pattern and merges close samples with each other. Yet another proposed method is to find a pattern prototype (a typical example of certain classes) and use that for establishing the category of a new pattern before comparing it with other exemplars of that category to recover its specific identity (see [3] for details).

All of the methods described so far start the grouping process by trying to identify similarities between classes and their representing patterns. Obviously, patterns representing the same class of objects should have some features and feature values in common, while patterns describing members of a different class should have different values for some or all of these features. In other words, objects are classified as members of a particular class if they possess some distinctive features, which make them distinguished from other objects present in the universe of objects. Consequently, one may start the process of grouping of classes on the basis of their evident differences. That is, collect objects (e.g. users), which have some evident differences from all other objects (classes) into one group. Some efficient algorithms developed for finding the necessary and sufficient conditions that describe class membership, based on this approach are described in [3].

2.4. Rule-based Misuse Detection

Obviously, attempting to detect intrusions on the basis of deviations from expected behaviors of individual users has some difficulties. For some users, it is difficult to establish a normal pattern of behavior. Therefor, it will be easy for a masquerader to go undetected as well. Alternatively, the rule-based detection systems are based on the understanding that most known network attacks can be characterized by a sequence of events. For implementation purposes, high-level system state changes or audit-log events during the attacks are used for building the models that form the rule bases. In Rule-based misuse detection model, the IDS will monitor system logs for possible matches with known attack profiles [2]. Rule-based systems generate very few false alarms, as they monitor for known attack patterns.

There is another situation for which statistical anomaly detection may not be able to detect intrusions. This is related to the case when legitimate users abuse their privileges. That is, such abuses are normal behavior for these users and are consequently undetectable through statistical approaches. For both of these cases, it may be possible to defend the system by enforcing rules that describe suspicious patterns of behavior. These types of rules must be independent of the behavior of an individual user or their deviations from past behavior patterns. These rules are based on the knowledge of past intrusions and known deficiencies of the system security. In some sense, these rules define a minimum standard of conduct for users on the host system. They attempt to define what can be regarded as the proper behavior that its breaches will be detected. Most current approaches to detecting intrusions utilize some form of rule-based analysis. Expert systems are probably the most common form of rule-based intrusion

detection approaches; they have been in use for several years [4].

Knowledge Based Systems (KBS) are modular structures in which the knowledge is separate from the inference procedure. Knowledge may be utilized in many forms, e.g. collection of facts, heuristic, common sense, etc. When the knowledge is acquired from (and represents) some particular domain expert, the system is considered an expert system. In many cases, production rules or specification of the conditions that must be satisfied for the rule to become applicable represents knowledge. Also included are the provisions of what should be done in case a rule is activated.

The areas of KBS, expert systems, and their application to intrusion detection have been and still are a very active research area. Among the very important aspects of the KBSs, are their knowledge bases and their establishment. This area and related subjects may be considered as a field by itself, referred to as 'knowledge engineering'. Knowledge engineering is the process of converting human knowledge into forms suitable for machines, e.g. rules in expert systems. Some examples of an interdisciplinary approach based for knowledge engineering in computer security systems are described in [5].

For successful intrusion detection, the rule-based subsystem contains knowledge about known system vulnerability, attack scenarios, and other information about suspicious behavior. The rules are independent from the past behavior of the users. With each user gaining access and becoming active, the system generates audit records that in turn are evaluated by the rule-based subsystem. This can result in an anomaly report for users whose activity results in suspicious ratings exceeding a pre-defined threshold value.

Clearly, this type of intrusion detection is limited in the sense that it is not capable of detecting attacks that the system designer does not know about. To benefit from the advantages of both approaches, most intrusion detection systems utilize a hybrid approach, implementing a rule-based component in parallel with statistical anomaly detection. While in general, the inferences made by the two approaches are independent or loosely coupled. The two subsystems share the same audit records with different internal processing approaches. There are arguments and ongoing research in tightening the two together in the hope of achieving a reduced false-positive rate of anomaly detection and eliminating the possibility of multiple alarms [6].

3. Immunology Based Intrusion Detection

This section gives a brief overview of an interesting and somehow different approach to intrusion detection. The design objective for this approach is related to building computer immune systems as inspired by anomaly detection mechanisms in natural immune systems. The analogy between computer security problems and biological processes was suggested as early as 1987, when the term 'computer virus' was introduced [1]. But it took some years for the connection between immune systems and computer security to be eventually introduced [7]. This view of computer security can also be of great value for implementing other intrusion detection approaches. This type of intrusion detection has been expanded into a distributed, local, and tunable anomaly detection method [8].

In the immune system, the intrusion detection problem is viewed as a problem of distinguishing self (e.g. legitimate users and authorized actions) from nonself (e.g. intruders). To solve this problem, Detectors that match anything not belonging to self are generated. The method relies on a large enough set of random detectors that are eventually capable of detecting all nonself objects. While these systems show several similarities with more traditional IDSs, they are more autonomous. Such systems present many desirable characteristics [8]. In particular, it needs to be noted that the detection carried out by the immune system is approximate; the match between antigen (foreign protein) and receptor (surface of the specialized cells in the immune system) need not be exact. This will allow each receptor to bind to a range of similar antigens and vice versa. Based on the cited literature, these concepts and ideas are further discussed in the remainder of this section.

4. Fuzzy Intrusion Detection

Hybrid systems that are claimed to combine the advantages of both statistical and rule-based algorithms, while partially eliminating the shortcomings of each one, are also devised. In general, such systems will use the rule-based approach for detection of previously encountered intrusions and statistical anomaly detection

algorithms for checking new types of attacks. An example of this general approach is based on utilization of neural networks that are trained to model the user and system behavior, while the anomaly detection consists of the statistical likelihood analysis of system calls [9]. Another approach is based on state transition analysis [10]. It attempts to model penetrations as a series of state changes that lead from an initial secure state to a target compromised state. A case based reasoning approach to intrusion detection, which alleviates some of the difficulties in acquiring and representing the knowledge is presented in [11]. A data-mining framework for adaptively building intrusion detection models is described in [12]. It utilizes auditing programs to extract an extensive set of features that describe each network connection or host session, and applies data mining approaches to learn rules that accurately capture the behavior of intrusions and normal activities.

For any type of the intrusion detection algorithm, some points need to be further considered. In rule-based (expert) systems administrators or security experts must regularly update the rule base to account for newly discovered attacks [2]. There are some concerns about any system that relies heavily on human operators (or experts) for knowledge elicitation. Some of the crucial ones are:

- Humans, in the course of decision making and reaching a conclusion, might use variables that are not readily measurable or quantifiable.
- Humans might articulate non-significant features. This, among other reasons, can lead to the establishment of inconsistent (from one expert to another) rule bases. Also, the system will be slower than what it should be as some of the rules that make up the knowledge base are of secondary importance.
- Broadly speaking, experts' knowledge is necessarily neither complete nor precise.

For these reasons, it is highly desirable to have systems and algorithms that acquire knowledge from experiential evidence automatically.

The statistical-anomaly detection algorithm will report 'significant' deviations of a behavior from the profile representing the user's normal behavior. While the significant usually refers to a threshold set by the system security officer, in practice it can be difficult to determine the amount that a behavior must deviate from a profile to be considered a possible attack. In the case of distributed anomaly detection based on the mechanisms in natural immune system, it is in fact advantageous to be able to carry out approximate detection.

In any of these algorithms, the need for exploiting the tolerance for imprecision and uncertainty to achieve robustness and low solution costs is evident. This is in fact, the guiding principle of soft computing and more particularly fuzzy logic [13]. The subject of fuzzy logic is the representation of imprecise descriptions and uncertainties in a logical manner. Many IDSs are mainly dependent on knowledge bases or input/output descriptions of the operation, rather than on deterministic models. Inadequacies in the knowledge base, insufficiency or unreliability of data on the particular object under consideration, or stochastic relations between propositions may lead to uncertainty. Uncertainty refers to any state of affair or process that is not completely determined. In rule-based and expert systems, lack of consensus among experts can also be considered as uncertainty. Also, humans (administrators, security experts...) prefer to think and reason qualitatively, which leads to imprecise descriptions, models, and required actions.

Zadeh introduced the calculus of fuzzy logic as a means for representing imprecise propositions (in a natural language) as non-crisp, fuzzy constraints on a variable [14]. This is 'vagueness': a clear but not precise meaning. That is to say, fuzzy logic started to cover vagueness, but turned out to be useful for dealing with both vagueness and uncertainty. The use of fuzzy reasoning in expert systems is naturally justifiable, as imprecise language is the characteristic of much expert knowledge. In crisp logic, propositions are either true or false, while in fuzzy logic different modes of qualifications are considered.

There seems to be an urgent need for further work on exploring the ways that artificial intelligence techniques can make the intrusion detection systems more efficient. More specifically, intelligent approaches that learn and automatically update user and system profiles need to be investigated. More research to study the implications and advantages of using fuzzy logic for approximate reasoning and handling intrusion detection through approximate matching are definitely required. Additionally, the capabilities of fuzzy logic in using the linguistic variables and fuzzy rules for analysing and

summarizing the audit log data need to be investigated.

5. Concluding Remarks

Any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource is defined as an intrusion. Many intrusion detection systems base their operations on analysis of operating system audit trail data. Intrusions can be categorised into two main classes: misuse intrusions and anomaly intrusions. As misuse intrusions follow well-defined patterns they can be detected by performing pattern matching on audit-trail information. Anomalous intrusions are detected by observing significant deviations from normal behavior. Anomaly detection is also performed using other mechanisms, such as neural networks, machine learning classification techniques, and approaches that mimic biological immune systems. Anomalous intrusions are harder to detect, mainly because are no fixed patterns of intrusion. So, for this type of intrusion detection fuzzy approaches are more suitable. A system that combines human-like capabilities in handling imprecision and adaptive pattern recognition with the alertness of a computer program can be highly advantageous. This is an area that demands further study and research work.

References

[1] M. Wilikens, " RAID'98: Recent advances in intrusion detection- workshop report," 1998. Available at http://www.zurich.ibm.com/pub/Other/RAID/

[2] B. Mukherjee, L. Heberlein, and K. Levitt, "Network intrusion detection," *IEEE Network*, pp. 26 – 41, May-June 1994.

[3] S. Shahrestani, H. Yee, and J. Ypsilantis, "Adaptive recognition by specialized grouping of classes," in *Proc. 4th IEEE Conference on Control Applications*, Albany, New York, 1995, pp. 637-642.

[4] D. Denning, "An intrusion-detection model," *IEEE Trans. Software Engineering*, Vol. 13, No. 2, pp. 222, February 1988.

[5] R. Venkatesan and S. Bhattacharya, "Threat-adaptive security policy," *Proc. IEEE International Performance, Computing, and Communications Conference*, 1997. pp. 525 – 531.

[6] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, and P. Neumann, " A real-time intrusion detection system (IDES)," Technical Report, 1992, available at: http://www.sdl.sri.com/nides/reports/9sri.pdf.

[7] J. Kephart, "A biologically inspired immune system for computers," in *Artificial Life: Proc. International Workshop on the Synthesis and Simulation of Living Systems*, R. Brooks and P. Maes, Eds. Cambridge, MA: MIT Press, 1994.

[8] P. D' haeseleer, S. Forrest, and P. Helman, "A distributed approach to anomaly detection," 1997, Available at http://www.cs.unm.edu/~forrest/papers.html.

[9] D. Endler, "Intrusion detection. Applying machine learning to Solaris audit data," *Proc. 14th Computer Security Applications Conference*, 1998, pp. 268 – 279.

[10] K. Ilgun, R. Kemmerer, and P. Porras, "State transition analysis: a rule-based intrusion detection approach," *IEEE Trans. Software Engineering*, Vol. 21, pp. 181 – 199, March 1995.

[11] M. Esmaili, B. Balachandran, R. Safavi-Naini, and J. Pieprzyk, "Case-based reasoning for intrusion detection," *Proc. 12th Computer Security Applications Conference*, 1996, pp. 214 – 223.

[12] L. Wenke, S. Stolfo, and K. Mok, "A data mining framework for building intrusion detection models," *Proc. IEEE Symposium on Security and Privacy*, 1999, pp. 120 – 132.

[13] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, Vol. 11, no. 6, pp. 48-56, Nov. 1994.

[14] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.

Fuzzy Bayesian Nets for User Modelling, Message Filtering and Data Mining

J. F. Baldwin

Engineering Mathematics Department

University of Bristol

Email: jim.baldwin@bristol.ac.uk

Summary

In this paper we will discuss the basic theory of Fuzzy Bayesian Nets and their application to user modelling, message filtering and data mining. We can now capture and store large amounts of data that we would like to transform into useful knowledge. We also expect to receive far too many messages to be able to handle without some form of intelligent filtering which acts on our behalf. Our Society feeds on information with data banks, web sites, Internet wireless etc. We require a more intelligent way of handling all this data and methods for transforming it into useful knowledge. Summarising data, finding appropriate rules to discover models, sending only relevant information to any individual, modelling users are all important tasks for our modern day computer systems. We want to use the available data, not be swamped with it.

Introduction

Various machine learning techniques such as decision trees, neural nets and Bayesian Nets, [2, 3], have been successful in the general field of data mining. A database is given and used to find rules, determine a neural connectionist model or provide the required probabilities for a Bayesian net to answer queries that cannot be answered directly. Finding the optimal architectures or form of rules is more difficult. A decision tree can be derived to fly an airplane for a given flight path from a database of states and events taken every time a pilot makes a decision. To find a decision tree to fly more generally for any flight plan cannot be done. There are limitations to all these machine-learning methods. Which ever model is used it should not be too complex and must provide good generalisation. If it is over complex by over fitting the data then errors in classifications or predictions will be made. Good generalisation means sensible interpolation between the cases in the database to match the desired case. It is easier to treat discrete variables than continuous ones.

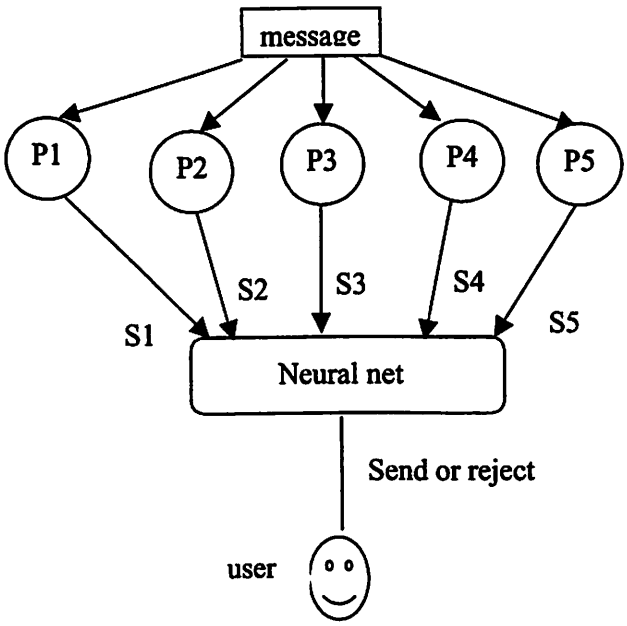
We will let continuous variables take fuzzy values rather than a continuum of values. Thus, a variable representing the height of a person will be allowed to take values such as

short, average and tall rather than a range of values. The semantics of short, average, tall will be given by a fuzzy set interpretation. All machine learning methods and algorithms can be modified to allow variables to take these fuzzy values. These modifications will be illustrated in this paper. This provides an example of computing with words. The advantage of using fuzzy words as compared to crisp sets is that better interpolation is provided and this results in more simple models. For example, to classify points belonging to the upper triangle of a square as apposed to those in the lower triangle requires only 2 overlapping fuzzy sets on each of the axes of the square to give 97% accuracy. To obtain the same accuracy in the case of crisp sets we would require 16 divisions of each axis. We would expect the modifications required to move from crisp to fuzzy sets to not be too complex. Certainly, we would not expect a totally different mathematics. For prediction problems, the variable whose value is to be predicted is allowed to take a discrete set of fuzzy values. The prediction from our machine learning method will be a probability distribution over these fuzzy sets. A method of defuzzification will be described which converts this to a single point value.

For filtering messages we will use prototypes representing different types of people for given contexts. Each prototype will output a support for receiving the message. Each user will have a neural net to input these supports and provide a command to send the message to the user or reject it. The prototype can be modelled with fuzzy rules, a fuzzy decision tree, a fuzzy Bayesian net or a fuzzy conceptual graph. In this paper we will only consider Bayesian nets and decision trees.

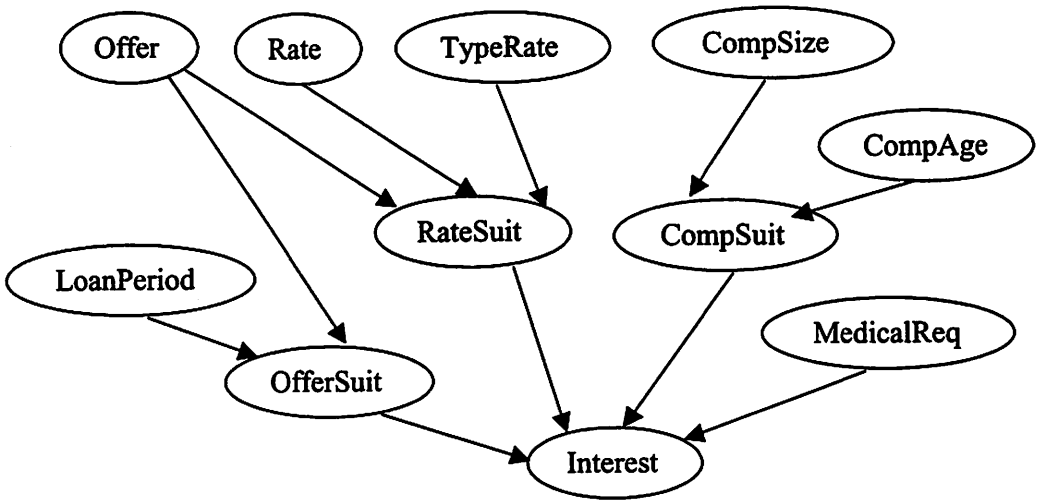
Prototypes for Message Passing and other Applications

As discussed above a message is sent to prototypes representing prototypical persons and a support of interest is output from each prototype. These are sent to a user's neural net that decides to either send or reject the message. We can illustrate this diagrammatically as



The representation for the prototypes, P1, ..., P5 could be fuzzy rules, fuzzy conceptual graphs, fuzzy decision trees or fuzzy Bayesian nets. We will choose for this paper fuzzy Bayesian nets.

A fuzzy Bayesian net is like a Bayesian net, [4], but the variables can take fuzzy values. An example of such a net is



The values of the variables are
Offer: {mortgage, personal loan, car loan, car insurance, holiday insurance, credit card, payment protection, charge card, home insurance}

TypeRate : {fixed, variable}
Rate : {good, fair, bad}
CompSize : {large, medium, small}
CompAge : {old, medium, young}
LoanPeriod : {long, medium, short}
RateSuit : {very, fairly, a little}
CompSuit : {very, fairly, a little}
OfferSuit : {good, average, bad}
MedicalReq : {yes, no}
Interest : {high, medium, low}

The meaning of the variables is obvious for this financial application. Some of the values are fuzzy sets such as old, long, very, good, high etc.

Prior probability distributions are given for those nodes without parents and conditional probabilities Pr(Node | Parents) for those nodes with parents. These probabilities define a type of person – risk taking for example. Each prototype would have a different set of

conditional probabilities. They could, of course, also have different architectures with different nodes. These probabilities can be determined from appropriate databases. The entries in these databases would more likely be precise values and we would need to convert this database to one where the attributes could take the fuzzy values given above. We will discuss this later.

A message might be as follows. The Robert Building Society is offering long term mortgages at 4% fixed rate. From this message we can extract the some variable instantiations. The Company size and age can be deduced from published details of the Robert Company. There is missing information about the variable medical required. The rate of loan, Company size and age will be given as numbers and we must convert these to probability distributions over the word values of the variables. The words are represented by fuzzy sets and mass assignment theory is used to obtain these distributions as discussed later.

Normally we would compile the net without instantiations and then update this prior representation using the variable instantiations to obtain a final probability distribution over the values of the Interest variable. A clique tree is first formed and a message-passing algorithm used to do the compiling and updating. This method must be modified to take account of the fact that some of the instantiations are probability distributions. Bayesian updating assumes precise values for the evidence and does not handle the situation in which distributions are given. We will later discuss the philosophy of this modification but we will not give all the details of the modification to the message-passing algorithm.

For now we can assume that we use the modified algorithm to give us a final probability distribution over the values of the Interest variable. Suppose this is

$$\text{high} : p1, \text{medium} : p2, \text{low} : p3 \qquad \text{where } p1 + p2 + p3 = 1$$

The values high, medium and low are fuzzy sets and the mass assignment theory can give expected values of these fuzzy sets as the expected values of the least prejudiced distributions associated with the fuzzy sets. Let these be $\mu 1$, $\mu 2$ and $\mu 3$ respectively. The Interest variable then takes the defuzzified value s where

$$s = p1\mu 1 + p2\mu 2 + p3\mu 3$$

Each prototype delivers a support in a similar manner. The neural net for a given user is trained on an example set.

This is a toy example to illustrate the general approach. More generally contexts would be defined and the net prototypes given for each defined contexts. Message interpretation would use some form of semantic analysis to extract the relevant variable instantiations.

The approach can be used for many different types of personalisation examples. A web page could be personalised for a given user to give relevant and interesting information. Books and films could be chosen to interest the user. Internet wireless will require approaches like this to prevent deadlock in communication traffic.

The Bayesian Modification Philosophy

Classical Bayesian updating cannot handle updating with distributions. We require a modification and we base the idea of choosing the updated distribution to minimise the relative entropy of this with the prior. Applied to the problem of updating the prior of AB, namely

ab : p1, a¬b : p2, ¬ab : p3, ¬a¬b : p4

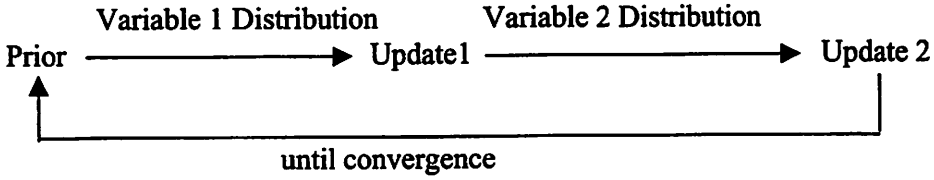
with the distribution for A of

a : q1, ¬a : q2 = 1-q1

we use

		q1	q2	
		a	¬a	update
prior	AB			
p1	ab	p1q1 / N1	0	p1q1 / N1
p2	a¬b	p2q1 / N1	0	p2q1 / N1
p3	¬ab	0	p3q2 / N2	p3q2 / N2
p4	¬a¬b	0	p4q2 / N2	p4q2 / N2
		N1 = p1+p2	N2 = p3+p4	

For the case in which we update with several variable evidences we perform an update, one at a time and repeat with the final update becoming the new prior until we obtain convergence. This is depicted diagrammatically as



The clique message-passing algorithm of Bayesian Nets is modified to effectively be equivalent to this modified updating. The modification does not add significant complexity to the algorithm.

A Little Mass Assignment Theory

We suppose the variable X can take values in {1, 2, 3, 4, 5, 6}. Let us suppose that we wish the values of X to be the mutually exclusive partition {even and odd}. If we were told that X is even then we would say that the distribution over the integer set is

Pr(2 | even) = Pr(4 | even) = Pr(6 | even) = 1/3

This assumes an equally likely prior.

If we replace the partition with the fuzzy partition {small, medium, large} where

small = 1 / 1 + 2 / 0.7 + 3 / 0.3

medium = 2 / 0.3 + 3 / 0.7 + 4 / 1 + 5 / 0.2

large = 5 / 0.8 + 6 / 1

and are told that X is **small** we should be able to derive a distribution over X assuming an equally likely prior. This we call the least prejudiced distribution.

The mass assignment for the fuzzy set **small** is

$$MA_{\mathbf{small}} = \{1\} : 0.3, \{1, 2\} : 0.4, \{1, 2, 3\} : 0.3$$

Assuming the prior distribution of X is that each integer is equally likely then the least prejudiced distribution is obtained by allocating a mass associated with a set of values equally among those values. This gives

$$LPD_{\mathbf{small}} = 1 : 0.3 + 0.2 + 0.1 = 0.6,\ 2 : 0.2 + 0.1 = 0.3,\ 3 : 0.1$$

If we had a different prior then this would be used to allocate the masses.

The mass assignment, [1], is a random set. We give it a different name to emphasise that the mass assignment theory is different to both random set and Dempster Shafer theories.

This definition of LPD can be extended to the case of continuous fuzzy sets and the expected value of the LPD is the mean μ of the fuzzy set that we used for defuzzification.

Suppose that we are told that X is **about_2** where

$$\mathbf{about\_2} = 1 / 0.3 + 2 / 1 + 3 / 0.3$$

then we are interested to determine Pr(**small** | **about_2**), Pr(**medium** | **about_2**) and Pr(**large** | **about_2**) giving the distribution over our partition. . The calculation of Pr(**f** | **g**) where **f** and **g** are fuzzy sets is called point value semantic unification.

The mass assignment for **about_2** is given by

$$MA_{\mathbf{about\_2}} = \{2\} : 0.7, \{1, 2, 3\} : 0.3$$

so that the LPD of **about_2** is

$$LPD_{\mathbf{about\_2}} = 1 : 0.1,\ 2 : 0.7 + 0.1 = 0.8, 3 : 0.1$$

$LPD_{\mathbf{about\_2}}(x) = Pr(x \mid \mathbf{about\_2})$

Thus

$Pr(\mathbf{small} \mid \mathbf{about\_2})$

$= 0.3Pr(\{1 \mid \mathbf{about\_2}) + 0.4\ Pr(\{1, 2\} \mid \mathbf{about\_2}) + 0.3\ Pr(\{1, 2, 3\} \mid \mathbf{about\_2})$

$= 0.3\ 0.1 + 0.4(0.1 + 0.8) + 0.3(0.1 + 0.8 + 0.1) = 0.69$

Suppose we are given a database with attributes X, Y and Z that can take values in [0, 10], [5, 12] and {true, false} respectively. Suppose further that we partition X and Y into the fuzzy partitions {**fi**} and {**gi**} respectively.

A new range of values for attributes in a database can be a set of words where each word is defined by a fuzzy set. The original database attribute values are transformed into distributions over these words to form a reduced database. This reduced database can be used to obtain decision trees using ID3, the required probabilities for Bayesian analysis and other applications. The entries in the original database can be precise, vague, fuzzy or unknown

Let one line in the database be X = x, Y = y and Z = true, then this can be rewritten as the lines
X = **fi**, Y = **gj**, Z = true with probability Pr(**fi** | x) Pr(**gj** | y) for all i, j
where x and y can be point values, intervals or fuzzy sets.

We can repeat this for all lines of the database and calculate the joint probability distribution for XYZ from which the Bayesian Net conditional probabilities can be calculated.

A similar approach can be used to determine decision trees from databases for classification and prediction.

Conclusions

The methods given here for the management of uncertainty, machine learning and modelling provides a unified and consistent approach to computing with words eliminating the ad hoc nature usually encountered. For example, one method of defuzzification suggests itself rather than the numerous ad hoc methods used in the literature.

In this paper we have described a Bayesian Net approach to user modelling and personalisation. We extended the theory of Bayesian Nets to allow node variables to take fuzzy values. These word values were represented by fuzzy sets. The updating message algorithm was extended to allow probability distribution instantiations of the variables. This was necessary since instantiated values from messages were translated into distributions over words.

The approach was illustrated using a message traffic filtering application. A similar approach could be used for many other applications such as personalising web pages to give users relevant information, providing TV programs, books and films of interest, intelligent filtering of emails. Advertisers could use the approach to select who should receive their adverts.

Fuzzy sets were used to provide better generalisation and the use of continuous variables.

A similar approach has been used for data mining and other machine learning applications.

References

1. Baldwin, J. F., Martin, T. P. and Pilsworth, B. W. (1995). "FRIL - Fuzzy and Evidential Reasoning in AI", Research Studies Press (John Wiley).

2. Baldwin, J. F. (1996). "Knowledge from Data using Fril and Fuzzy Methods" in Fuzzy Logic in AI, Ed. J. F. Baldwin, John Wiley. 33-76.

3. Baldwin, J. F. and Martin, T. P. (1997). "Basic Concepts of a Fuzzy Logic Data Browser with Applications" in Software Agents and Soft Computing: Concepts and Applications, Ed. H. S. Nwana and N. Azarmi, Springer (LNAI 1198). 211-241.

4. Jenson F., (1996), An Introduction to Bayesian Networks Springer-Verlag.

Aggregation Methods for Intelligent Search

Ronald R. Yager
Iona College
New Rochelle, NY 10801
yager@panix.com

Abstract.
We describe a document retrieval language which enables user's to better represent their requirements with respect to the desired documents to be retrieve. This language allows for a specification of the interrelationship between the desired attributes using linguistic quantifiers. This framework also supports a hierarchical formulation of queries. These features allow for an increased expressiveness in the queries that be handled by a retrieval system.

1. Introduction

With the explosive growth of the internet the need to effectively retrieve documents[1] satisfying user requirements has emerged as one of the most important technological problems we are facing. At the heart of the current problem with retrieval systems is the ability to effectively express search requirements in a way that can be "understood" by the computer. Here we need a bridge to translate between the linguistic type expression favorite by humans and the formal representations needed for computer manipulation. Fuzzy logic and the related disciplines of approximate reasoning [1] and computing with words [2] have been primarily developed to provide such a bridge.

For the most part, the current retrieval paradigm involves a situation in which a document is "represented." Essentially this representation consists of a decomposition of a document into attributes on which the document can be scored. These attributes can be based upon ideas as simple as the appearance of a word or phase in the document or can involve complex processing of the document using notions like frequency of occurrence. When searching, a user expresses their requirements in terms of a subset of these primary attributes. The overall evaluation of a document by an aggregation of the scores of the specified attributes.

[1]Here we use the term document to generically indicate a broad spectrum of objects which include web–pages, reports, books, movie titles etc

The method of aggregation used can be seen to inherently reflect an expression of a desired interrelationship between the specified attributes. Thus the aggregation can be seen as a kind of *recomposition* of the document from its attributes. Typical examples of aggregation are the simple average and those based upon logical connections, *anding* and *oring*.

One way to improve retrieval systems is to provide a wide class of aggregation operations to enable the system to implement sophisticated interactions and thereby allow the user increased expressiveness in specifying their desires. In addition any extension of aggregation options would greatly benefit from a strong correspondence between formal methods of aggregation and natural language specification. One goal of the agenda of computing with words is to provide such a capability.

Here we shall describe a query language for evaluating user specifications which we call this language Hi-RET. This language will make considerable us of the Ordered Weighted Averaging (OWA) operators [3]. These OWA operators provide a large class of aggregation methods which have a natural correspondence between a linguistic specification and a formal method of aggregation. In addition the expressive capability of the query language will be enhanced by the use of a hierarchical structure to represent queries.

As will become clear this approach is extremely generic and can be used as an integral part of many different types search and retrieval systems.

2. OWA Aggregation Operators

Central to any document retrieval system is the need to aggregate scores, in order to provide a very general framework to implement aggregations, we shall use the OWA operators [3].

Definition: An Ordered Weighted Averaging, OWA, operator of dimension n is a mapping which has an associated weighting vector W in which $w_j \in [0,1]$ and

$\sum_{j=1}^{n} w_j = 1$ and where

$$F(a_1, a_2, ..., a_n) = \sum_{j=1}^{n} w_j b_j$$

with b_j being the j^{th} largest of the a_i.

By appropriately selecting the vector W we can implement various different types of aggregation.

We now consider a basic application of the OWA operator in a document retrieval system. Assume A_1, A_2,...., A_n is a collection of attributes of interest to a user of a retrieval system. For any document d, let $A_i(d) \in [0,1]$ indicate the degree to which document d satisfies the property associated with attribute A_i. Using the OWA operator, we can obtain an overall valuation of document d as

$$Val(d) = F_W(A_i(d), A_2(d), ..., A_n(d)).$$

A number of different approaches have been suggested for obtaining the weighting vector W to use in any given application [4]. For our purposes, that of developing a user friendly document retrieval system, we shall describe an approach based upon the idea of linguistic quantifiers.

The concept of linguistic quantifiers was originally introduced by Zadeh [5]. A proportional linguistic quantifier is a term corresponding to a proportion of objects. Examples of this are *most, at least half, all, about* $1/3$. Let Q be a proportional linguistic quantifier we can represent this as a fuzzy subset Q over $I = [0, 1]$ in which for any proportion $r \in I$, $Q(r)$ indicates the degree to which r satisfies the concept indicated by the quantifier Q.

In [3] Yager showed how we can use a linguistic quantifier to obtain a weighting vector W associated with an OWA aggregation. For our purposes we shall restrict ourselves to regularly increasing monotonic (RIM) quantifiers. A fuzzy subset $Q : I \rightarrow I$ is said to represent a RIM linguistic quantifier if: 1) $Q(0) = 0$, 2) $Q(1) = 1$ and 3) if $r_1 > r_2$ then $Q(r_1) \geq Q(r_2)$ (monotonic). These RIM quantifiers model the class in which an increase in proportion results in an increase in compatibility to the linguistic expression being modeled. Examples of these types of quantifiers are *at least one, all, at least α %, most, more than a few, some.*

Assume Q is a RIM quantifier. Then we can associate with Q an OWA weighting vector W such that for $j = 1$ to n

$$w_j = Q(\frac{j}{n}) - Q(\frac{j-1}{n})$$

Thus using this approach we obtain the weighting vector directly from the linguistic expression of the quantifier

Let us look at the situation for some prototypical quantifiers. The quantifier *for all* is shown in figure #1.

Figure #1. Linguistic quantifier "for all"

In this case we get that $w_j = 0$ for $j \neq n$, and $w_n = 1$, W $= W_*$ In this case we get as our aggregation the minimum of the aggregates. We also recall that the quantifier *for all* corresponds to the logical "anding" of all the arguments

In figure #2 we see the existential quantifier , *at least one*, this is the same as *not none.*

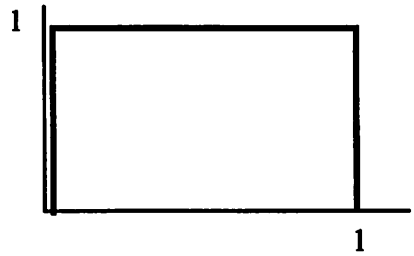


Figure #2. Linguistic quantifier "not none"

In this case $w_1 = 1$ and $w_j = 0$ for $j > 1$, $W = W^*$. This can be seen as inducing the maximum aggregation. It is recalled this quantifier corresponds to a logical *oring* of the arguments

Figure #3 is seen as corresponding to the quantifier *at least α.* For this quantifier $w_j = 1$ for j such that $\frac{j-i}{n} < \alpha \leq \frac{j}{n}$ and $w_i = 0$ for all other.

Another quantifier is one in which $Q(r) = r$ for $r \in [0, 1]$. For this quantifier we get $w_j = \frac{j}{n} - \frac{j-i}{n} = \frac{1}{n}$ for all j. This gives us the simple average. We shall denote this quantifier as *some.*

As discussed by Yager [6] one can consider parameterized families of quantifiers. For example consider the parameterized family $Q(r) = r^{\rho}$, where $\rho \in [0, \infty]$. Here if $\rho = 0$, we get the existential quantifier; when $\rho = \infty$, we get the quantifier *for all* and when $\rho = 1$, we are get the quantifier *some.* In addition for the

case in which $\rho = 2$, $Q(r) = r^2$, we get one possible interpretation of the quantifier *most*.

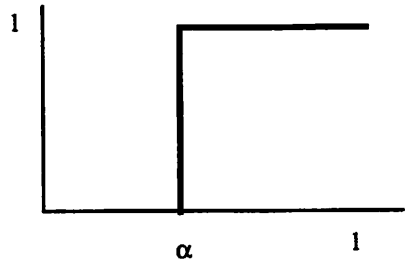


Figure #3. Linguistic quantifier "at least α"

We are now in a position to address the issue of obtaining the OWA weighting vector to be used in a search in a user friendly document retrieval system. In constructing such a user friendly system we shall make available to the user a vocabulary, $Q = \{Q_1, Q_2, ..., Q_q\}$, of linguist expressions, each corresponding to a linguistic quantifier. When posing a query, the user, after specifying a collection of attributes of interest (A_1, A_2, ..., A_n), will be prompted to also specify one of the linguistic quantifiers in Q as guiding the query formation. Transparent to the user is the association of each of the linguistic terms in Q, with a representative fuzzy subset, $Q_i \Leftrightarrow Q_i$, and the process of converting this fuzzy subset into an OWA weighting vector based on the formulation

$$w_j = Q_i(\frac{j}{n}) - Q_i(\frac{j-i}{n}).$$

One of the elements in Q should be designated as the default quantifier, this is the one that is to be used when no selection is specified by the user. Perhaps the most appropriate choice for this is the average quantifier $w_j = \frac{1}{n}$, which corresponds to the linguistic expression *some*.

The process of actually selecting the set Q while clearly of great importance is beyond our scope here and should benefit from some empirical research trying to match users perceptions and vocabulary with fuzzy sets. It should also deal the issue of finding a selection of terms to cover a spectrum wide enough to allow users to appropriately express their desires.

Based on the ideas so far presented here we can introduce the idea of a ***query module***: $<A_1, A_2, ... A_n: Q>$, consisting of a collection of attributes of interest and a linguistic quantifier indicating the proportion of the relevant attributes we desire. Implicit in this query module is the fact the the linguistic expression Q is essentially defining a weighting vector W for an OWA aggregation.

3. Introducing Importances In Queries

In the preceding we have indicated a query object to be a module consisting of a collection of attributes of interest and a quantifier Q indicating a mode of interaction between the attributes. Implicit in the preceding is the equal treatment of all desired attributes. Often a user may desire to ascribe different weights or importances to the different attributes. In the following we shall consider the introduction of importance weights into our procedure.

Let $\alpha_i \in [0, 1]$ be a value associated with an attribute indicating the importance associated with the attribute. We shall assume the larger α_i the more important and let $\alpha_i = 0$ stipulate zero importance. With the introduction of these weights we can now consider a more general query object, $<A_1, A_2, ..., A_n: M: Q>$. Here as before, the A_i are a collection of attributes and Q is a linguistic quantifier, however, here M is an n vector whose component $m_j = \alpha_j$, the importance associated with A_j.

Our goal now is to calculate the overall score Val(d) associated with a document d, we shall denote this

$$Val(d) = F_{Q/M}(A_1(d), A_2(d), ..., A_n(d))$$

Here $F_{Q/M}$ indicates an OWA operator. Our agenda here will be to first find an associated OWA weighting vector, W(d), based upon Q and M. Once having obtained this vector we calculate Val(d) by the usual OWA process

$$Val(d) = W(d)^T B(d) = \sum_{j=1}^{n} w_j(d)\, b_j(d)$$

Here $b_j(d)$ is denoting the j^{th} largest of the $A_i(d)$ and $w_j(d)$ is the j^{th} component of the associated OWA vector W(d)

We now describe the procedure [7] that shall be used to calculate the weighting vector, $w_j(d)$. The first step is to calculate the ordered argument vector B(d) such that $b_j(d)$ is the j^{th} largest of the $A_i(d)$. Furthermore, we shall let μ_j denote the importance weight associated with the attribute that has the j^{th} largest value. Thus if $A_5(d)$ is the largest of the $A_i(d)$, then $b_1(d) = A_5(d)$ and $u_1 = \alpha_5$. Our next step is to calculate the OWA weighting vector W(d). We obtain the associated weights as

$$w_j(d) = Q(\frac{S_j}{T}) - Q(\frac{S_{j-1}}{T})$$

where $S_j = \sum_{k=1}^{j} u_k$ and $T = S_n = \sum_{k=1}^{n} u_k$. Thus T is the sum of all the importances and S_j is the sum of the importances of the j^{th} most satisfied attributes. Once having obtained these weights we can then obtain the aggregated value by the usual method, $B^T W$. The following example will illustrate the use of this technique.

Example: We assume four criteria of interest to the user: A_1, A_2, A_3, A_4. Their importances are $u_1 = 1$, $u_2 = 0.6$, $u_3 = 0.5$ and $u_4 = 0.9$. From this we get $T = \sum_{k=1}^{4} u_k = 3$. We shall assume the quantifier guiding this aggregation is *most* which is defined by $Q(r) = r^2$. Consider document x whose satisfaction to each of the attributes is:

$$A_1(x) = 0.7,\ A_2(x) = 1,\ A_3(x) = 0.5,\ A_4(x) = 0.6$$

Our objective here is to obtain the valuation of the document with respect to this query structure. In this case the ordering of the criteria satisfactions gives us:

	b_j	u_j
A_2	1	0.6
A_1	0.7	1
A_4	0.6	0.9
A_3	0.5	0.5

Calculating the weights associated with x, which we denoted $w_j(x)$, we get

$$w_1(x) = Q(\tfrac{0.6}{3}) - Q(0) = 0.04$$

$$w_2(x) = Q(\tfrac{1.6}{3}) - Q(\tfrac{0.6}{3}) = 0.24$$

$$w_3(x) = Q(\tfrac{2.5}{3}) - Q(\tfrac{1.6}{3}) = 0.41$$

$$w_4(x) = Q(\tfrac{3}{3}) - Q(\tfrac{1.6}{3}) = 0.31$$

Using this we calculate Val(x)

$$Val(x) = \sum_{j=1}^{4} w_j(x)\, b_j$$

$$= (.04)(1) + (.24)(.7) + (.41)(.6) + (.31)(.5) = 0.609$$

More details with respect to the properties of this methodology can be found in [7], however here we shall point one special case. Consider the case of the quantifier *some*, $Q(r) = r$. For this quantifier $w_j = \frac{\alpha_j}{T}$ and hence $Val(d) = \frac{1}{T} \sum_{j=1}^{n} \alpha_j a_j$. This is simply the weighted average of the attributes.

4. Concepts and Hierarchies

In the preceding we have considered a retrieval system in which we have a collection of primary attributes A_i, i = 1 to n. These attributes are characterized by the fact that for any $d \in D$, we have available $A_i(d) \in [0,1]$, the values of attribute A_j are directly accessible. We shall now associate with our retrieval system slightly more general idea called a ***concept***. We define a concept as an object whose measure of satisfaction can be obtained for any document in D. It is clear that the attributes are examples of concepts, they are special concepts in that their values are directly accessible from the document base.

Consider now a query object of the type we have previously introduced. This is an object of the form $\langle A_1, A_2, ..., A_q: M: Q\rangle$. As we have indicated, the satisfaction of this object for any $d \in D$ can be obtained by our aggregation process. In the light of this observation we can consider this query object to be a concept, with

$$Con = \langle A_1, A_2, ..., A_q: M: Q\rangle$$

then

$$Con(d) = F_{Q/M}(A_1(d), A_2(d), ..., A_q(d)).$$

Thus a query object is a concept. A special concept is an individual attribute,

$$Con = \langle A_j : M: Q\rangle = A_j,$$

we shall call these atomic concepts. These atomic concepts require no Q or M, but just need A_j specification.

Let us look at the query object type concept in more detail. The basic components in these objects are the attributes, the A_j. However, from a formal point of view, the ability to evaluate the query objects-concept is based upon the fact that for each A_j, we have a value for any d, $A_j(d)$. As we have just indicated, a concept also has this property, for any d we can obtain a measure of its satisfaction. This observation allows us to extend our idea of query object-concept to allow for concepts whose evaluation depends upon other concepts. Thus we can consider concepts of the form

$$Con = \langle Con_1, Con_2, ..., Con_l: M: Q\rangle.$$

Here each of the Con_j are concepts used to determine the satisfaction of Con by an aggregation process where M determines the weight of each of the participating concepts and Q is the quantifier guiding the aggregation of the component concepts.

The introduction of concepts into the query objects results in a hierarchical structure for query formation.

Essentially, we unfold until we end up with queries made up of just attributes which we can directly evaluate. The following simple examples illustrate the structure.

Example: Consider here the query

$(A_1$ and A_2 and $A_3)$ or $(A_3$ and $A_4)$.

We can consider this as a concept

$<Con_1, Con_2 : M: Q>$.

Here Q is the existential quantifier and $M = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. In addition

$Con_1 = <A_1, A_2, A_3: M_1: Q_1>$

$Con_2 = <A_3, A_4 : M_2: Q_2>$

Here $Q_1 = Q_2 = all$ and $M_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ and $M_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

This query can be expressed in a hierarchical fashion as shown in figure #4.

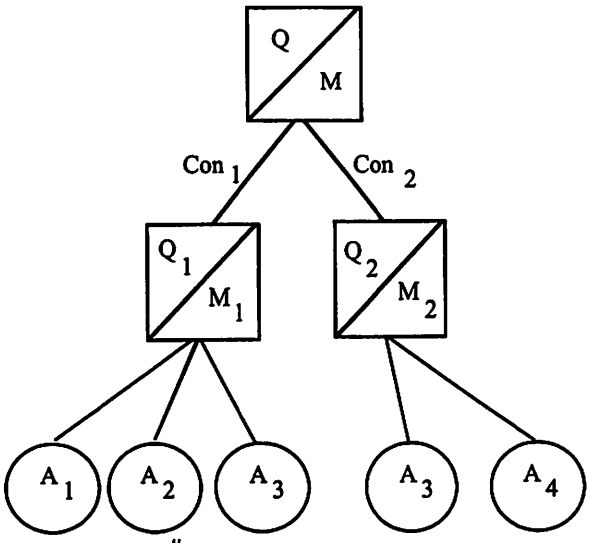


Figure #4. Hierarchical Formulation

5. Hierarchical Querying in Information Retrieval

Using the ideas discussed in the preceding we shall describe a hierarchical querying framework that can be used for information retrieval, we shall call this the Hierarchical Document Retrieval Language and use the acronym HI-RET. This language can be used to retrieve documents from the internet or intranet type environment or any other computer based environment.

Associated with any implementation of this language is a set $A = \{A_1, A_2, ..., A_n\}$ of atomic attributes, words or concepts. These atomic concepts are such that for any document d in D and any concept A_j in A we have directly available the value $A_j(d) \in [0,1]$, the satisfaction of attribute A_j by document d. This information can be stored in a database such that each record is a tuple consisting of the values $A_j(d)$ for j = 1 to n and the address of document d. Essentially each object can be viewed as a n vector whose components are the $A_j(d)$.

In addition to the attributes we also assume associated with any implementation of HI-RET is a vocabulary of linguistic quantifiers, $Q = \{Q_1, Q_2, ..., Q_q\}$ available to the searcher. Within this set of quantifiers we should surely have the quantifiers *all*, *any*. and *some*. One quantifier should be designated as the default quantifier. Perhaps a best choice for this is the quantifier *some*. Transparent to the user is a fuzzy subset Q_i on the unit interval associated with each linguistic quantifier Q_i. This fuzzy subset is used to generate the associated weights used in the aggregation.

A query to the document retrieval system is indicated by the user by the specification of a "concept" that the user desires satisfied. The user is asked to "define" this concept by expressing it in terms of a query object, $<C_1, C_2, C_n: M: Q>$, consisting of a group of components C_j, an importance weight associated with each of the components, M, and a quantifier, Q, expressing the imperative for aggregating the components. The specification of the importance weights as well as quantifier are optional. If the weights are not expressed, then by default they are assumed to have importance one, if the quantifier is not expressed, then the designated default quantifier is assumed. For each of the components of the query that are not an atomic object the searcher is asked to provide a definition. This process is continued until the complete hierarchy defining the query is formulated. It is noted that this hierarchy is a tree like structure in the leaves are atomic components. Figure #5 shows a prototypical example of such a query.

Once having obtained the HI-RET expansion of a query as in Figure#5, we can then use our aggregation methods to evaluate the query for each document. For example, in the case of figure #5 for document d we have

$Con_4(d) = F_{M4/Q4}(A_6(d), A_3(d))$

$Con_3(d) = F_{M3/Q3}(A_2(d), A_5(d), A_9(d))$

$Con_2(d) = F_{M_2/Q_2}(Con_4(d), A_8(d))$

$Con_1(d) = F_{M_1/Q_1}(A_7(d), Con_2(d), Con_3(d))$

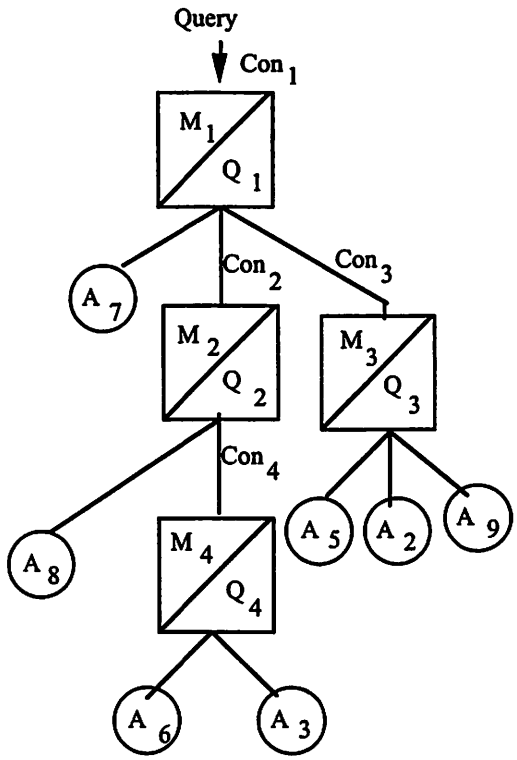


Figure #5. Prototypical Query in HI-RET

6. References

[1]. Yager, R. R., "Approximate reasoning as a basis for computing with words," in Computing with Words in Information/Intelligent Systems 1, edited by Zadeh, L. A. and Kacprzyk, J., Springer-Verlag: Heidelberg, 50-77, 1999.

[2]. Zadeh, L. A., "Fuzzy logic = computing with words," IEEE Transactions on Fuzzy Systems 4, 103-111, 1996.

[3]. Yager, R. R., "On ordered weighted averaging aggregation operators in multi-criteria decision making," IEEE Transactions on Systems, Man and Cybernetics 18, 183-190, 1988.

[4]. Yager, R. R. and Kacprzyk, J., The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer: Norwell, MA, 1997.

[5]. Zadeh, L. A., "A computational approach to fuzzy quantifiers in natural languages," Computing and Mathematics with Applications 9, 149-184, 1983.

[6]. Yager, R. R., "Families of OWA operators," Fuzzy Sets and Systems 59, 125-148, 1993.

[7]. Yager, R. R., "On the inclusion of importances in OWA aggregations," in The Ordered Weighted Averaging Operators: Theory and Applications, edited by Yager, R. R. and Kacprzyk, J., Kluwer Academic Publishers: Norwell, MA, 41-59, 1997.

Proposal of a Search Engine based on Conceptual Matching of Text Notes

Tomohiro TAKAGI, Masanori TAJIMA
Dept. of Computer Science, Meiji University.

Abstract

We propose a search engine which conceptually matches input keywords and web pages. The conceptual matching is realized by context-dependent keyword expansion using conceptual fuzzy sets.

First, we show the necessity and also the problems of applying fuzzy sets to information retrieval. Next, we introduce the usefulness of conceptual fuzzy sets in overcoming those problems, and propose the realization of conceptual fuzzy sets using Hopfield Networks. We also propose the architecture of the search engine which can execute conceptual matching dealing with context-dependent word ambiguity. Finally, we evaluate our proposed method through two simulations of retrieving actual web pages, and compare the proposed method with the ordinary TF-IDF method. We show that our method can correlate seemingly unrelated input keywords and produce matching Web pages, whereas the TF-IDF method cannot.

1. Introduction

Information retrieval in the Internet is generally done by using keyword matching, which requires that for words to match, they must be the same or synonyms. But essentially, not only the information that matches the keywords exactly, but also information related in meaning to the input keywords should be retrieved. The following reasons are why fuzzy sets are essential for information retrieval.

First, a fuzzy set is defined by enumerating its elements and the degree of membership of each element. It is useful for retrieving information which includes not only the keyword, but also elements of the fuzzy set labeled by the input keyword. For example, a search engine may use baseball, diving, skiing, etc., as kinds of sports, when a user inputs "sports" as the keyword.

Second, the same word can have various meanings. Several words are used concurrently in usual sentences, but each word has multiple possible meanings (region), so we suppose an appropriate context which suits all regions of meaning of all words. At the same time, the context determines the meaning of each word.

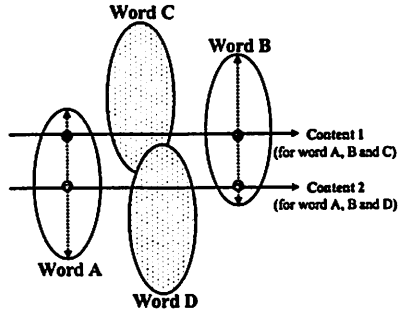


Fig.1. Meanings of words determined by a context.

For example, "sports" may mean "diving" or "sailing" when it is used with "marine," and may mean "baseball" or "basketball" when used with "TV programs." That is, each possibility distribution of meaning is considered as a fuzzy set itself. For information retrieval, keyword expansion that considers context is necessary, because simple expansion of a possible region causes a flood of words. For example, even if the user intends "marine sports," the set of expanded keywords includes unnecessary sports such as "baseball." However, an ordinary fuzzy set does not provide us the method to deal with context-dependent word ambiguity. To overcome this problem, we previously proposed using conceptual fuzzy sets (CFSs) [1]-[4] , which conform to Wittgenstein's concept, to represent the meanings of concepts.

In this paper, we propose a search engine which conceptually matches input keywords and Web pages. The conceptual matching is attained by context dependent keyword expansion using conceptual fuzzy sets. We describe the necessity of conceptual fuzzy sets for information retrieval in Section 2, and propose the use of conceptual fuzzy sets using Hopfield Networks in section 3. Section 4 proposes the search engine which can execute conceptual matching and deal with context-dependent word ambiguity. In Section 5, we show two simulations of retrieving actual Web pages comparing the proposed method with the ordinary TF-IDF method. In section 6, we will conclude the paper.

2. Fuzzy Sets and Context Dependent Word Ambiguity

2.1 Conceptual Fuzzy Sets [1]-[4]

Let's think about the meaning of "heavy." A person weighting 100kg would usually be considered heavy. But there is no clear boundary between "heavy" and "not heavy." Fuzzy sets are generally used to indicate these regions. That is, we have a problem of specificity.

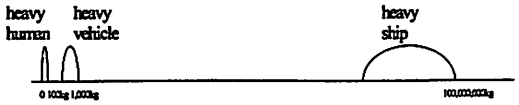


Fig. 2. The meaning of "heavy."

Let's think about it some more. For a vehicle, "heavy" might be several thousand kgs. For a ship, it might be more than ten thousand tons. Therefore, the item "heavy" being judged affects the vagueness of the meaning of "heavy" much more than the non-specificity of the amount when the item is already determined as shown in Fig. 2. Moreover, "work" can be heavy, "traffic" can be heavy, and "smoking" can be heavy. So the meaning of "heavy" changes with the context, which results in the vagueness of the meaning.

That is, the main cause of vagueness is ambiguity in the language, not specificity. Ordinary fuzzy set theory has not dealt with the context dependent meaning representation concerning language ambiguity. However, as we mentioned in the Introduction, a fuzzy set is defined by enumerating its elements and the degree of membership of each element, we can use it to express word ambiguity by enumerating all possible meanings of a word, then estimating the degrees of compatibilities between the word and the meanings. Fuzzy set theory should therefore deal with language ambiguity as the main cause of vagueness.

To overcome this problem, we previously proposed using conceptual fuzzy sets. Although several works have been published, we will explain CFSs for understanding the following section. According to Wittgenstein [5], the meaning of a concept can be represented by the totality of its uses. In this spirit, conceptual fuzzy sets, in which a concept is represented by the distribution of the activation concepts, are proposed.

The label of a fuzzy set represents the name of a concept, and a fuzzy set represents the meaning of the concept. Therefore, the shape of a fuzzy set is determined by the meaning of the label, which depends on the situation (Fig.3).

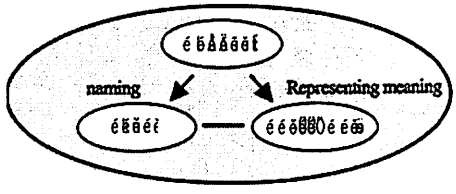


Fig.3. A fuzzy set as a meaning representation.

According to the theory of "meaning representation from use" proposed by Wittgenstein) the various meanings of a label (word) can be represented by other labels (words), and we can assign grades of activation showing the degree of compatibility between labels.

A CFS achieves this by using distributions of activations. A CFS is realized as an associative memory in which a node represents a concept and a link represents the strength of the relation between two (connected) concepts. The activation values agreeing with the grades of membership are determined through this associative memory. In a CFS, the meaning of a concept is represented by the distribution of the activation values of the other nodes. The distribution evolves from the activation of the node representing the concept of interest. The image of a CFS is shown in Fig. 4.

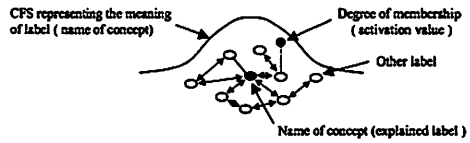


Fig. 4. A conceptual fuzzy set represented by associative memories.

2.2 CFS Representing a Composed Concept having Multiple Meanings Depending on the Situation

Because the distribution changes depending on which labels are activated as a result of the conditions the activations show a context-dependent meaning. When more than two labels are activated, a CFS is realized by the overlapping propagations of their activations. In CFS notation, operations and their controls are all realized by the distribution of the activations and their propagations in the associative memories.

We can say that the distribution determined by the activation of a label agrees with the region of thought corresponding to the word expressing its meaning. The distribution (meaning of a label), that is a figure of a fuzzy set, changes depending on considered aspects that reflect the context.

3. Creating of Conceptual Fuzzy Sets

Previously we used bidirectional associative memories (BAMs) [6],[7] to generate CFSs, because of the clarity of the constraints used for their utilization. In this paper, we use Hopfield Networks, whose output can be also used with a continuous value, to overcome the limitation of BAMs that are a layered neural network. We do so because in a correlative neural network, relationships between concepts may not be a layered structure.

The following shows how to construct CFSs using Hopfield Networks [8],[9].

Memorizing pieces of knowledge:

1. Classify piece of knowledge into several aspects. One piece becomes one pattern and one aspect corresponds to one Hopfield Network.
2. Each Hopfield network contains multiple patterns in the same aspect.

Generating CFSs:

1. Recollect patterns which include a node of interest in each Hopfield Network.
2. Sum all recollected patterns and normalize the activation values.

Figure 5 shows the image of memorized patterns and a generated CFS.

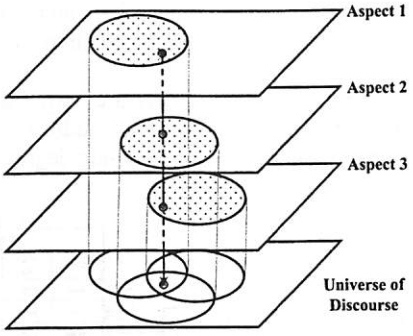


Fig. 5. Image of memorized patterns and a generated CFS.

Logic-based paradigms for knowledge representation use symbolic processing both for concept representation and inference. Their underlying assumption is that a concept can be defined precisely. This causes an exponential increase in the number of cases, because we have to memorize all cases according to every possible context. In contrast, knowledge representation using CFSs memorizes knowledge about generalizations instead of exact cases. The following is an example to compare the proposed knowledge representation with ordinary logic-based paradigms. It shows that context-dependent meanings are generated by activating several keywords.

Example:

Let's think about the meaning of "heavy" again. The subject may be a human, a cat, or an elephant. Moreover, the age of the subject may be a baby or an adult, which also influences the meaning of "heavy." Therefore, since the number of cases increases exponentially as:

(cat, human, elephant……) *
(baby, adult, ……) * ……,

it is impossible to know how heavy the subjects are in all cases. On the other hand, using CFSs, which create meaning by overlapping activations, number of cases to be memorized becomes:

(cat, human, elephant……) +
(baby, adult…….) +……..,

and increases linearly.

Let's generate CFSs in these contexts. Assume the universe of discourse is "weight," from 0-1000 kg. Aspects and memorized patterns are as follows.

	(Aspect)	(Memorized pattern)
kind		cat, human, elephant
age		baby, adult

<Step1> Memorize patterns such as those in Fig. 6 for each aspect. For example, the pattern "cat" shows that it memorizes its usual heavy weight within the activation range of [-1,1]. [-1,1] is the bi-polar expression of [0,1].

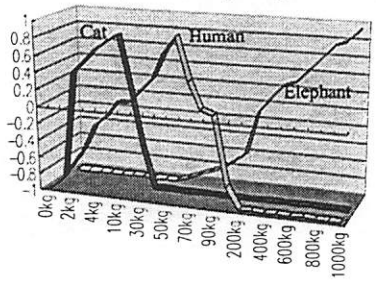


Fig. 6. Memorized patterns of "heavy cat," "heavy human," and "heavy elephant"

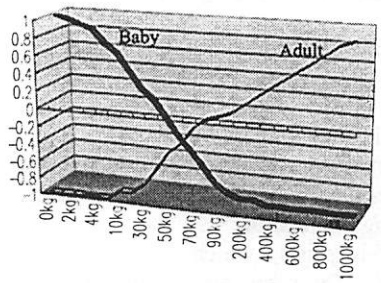


Fig.7. Memorized patterns of "heavy baby," and "heavy adult"

<Step2>When keywords are input, the input values of the neurons corresponding to these keywords are set as 1 and the input values of the other neurons are set as -1, and each Hopfield Network recollects the patterns.

<Step3> Finally, the activations of nodes in all aspects are summed up, and they are normalized in the range of [-1,1]. The normalized outputs become the final outputs result.

Figure 8 shows the ability of CFSs to generate context-dependent meanings of "heavy human" in the case of "adult" and "baby." We can recognize that both fuzzy sets have different shapes even when considering the same word "human." Figure 9 compares the difference between the case of "human" and "elephant." Here, [a + b] means that the activation is started from the concepts in nodes "a "(ex: adult) and "b" (ex: elephant).

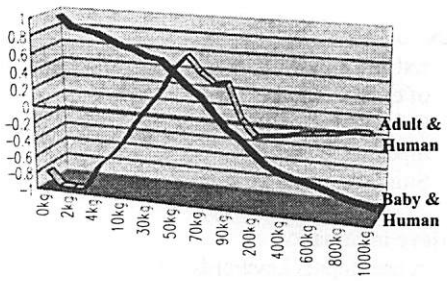


Fig. 8.Example of outputs "heavy human."

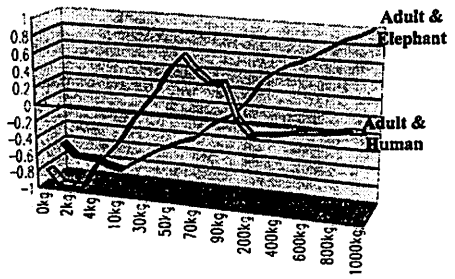


Fig. 9.Example of outputs "heavy adult."

4. Conceptual Matching in a Search Engine Using CFS

4.1 Scheme of Search Engine [10],[11]

Usually, search engines work as follows.

Index collecting of Web pages:
An indexer extracts words from Web pages, analyzes them, and stores the result as indexing information.
Retrieving information:
The Web pages, which include input keywords, are extracted. The pages are assigned priority and are sorted referencing the indexing information above.

As we mentioned in the Introduction, information retrieval is generally done by using keyword matching, which requires words to match and is different from conceptual matching.

4.2 Conceptual Matching

We propose a search engine system which conceptually matches input keywords and Web pages according to the following idea.

1. Expand input keywords to the elements of CFSs.
2. Evaluate the matching degree between the set of expanded keywords and the set of words included in each Web page.
3. Sort the Web pages and display them according to the matching degrees.

The following shows the process in the proposed search engine.

Index collecting of Web pages:
1. Extract nouns and adjectives, and count the frequency of each word.
2. Calculate an evaluation of each word using the TF-IDF method for each Web page.
3. Store the evaluation into a lexicon.

Retrieve information:
1. A user inputs keywords into a browser, which transfers the keywords to a CFS unit.
2. Propagation of activation occurs from input keywords in the CFS unit. The meanings of the keywords are represented in other expanded words using conceptual fuzzy sets, and the activation value of each word is stored into the lexicon.
3. Matching is executed in the following process for each Web page. Obtain the final evaluation of each word by multiplying the evaluation by the TF-IDF method and the activation value. Sum up the final evaluations of all words and attach the result to each Web page as a matching degree.
4. The matched Web pages are sorted according to the matching degrees, and their addresses are returned to the browser with their matching degrees.

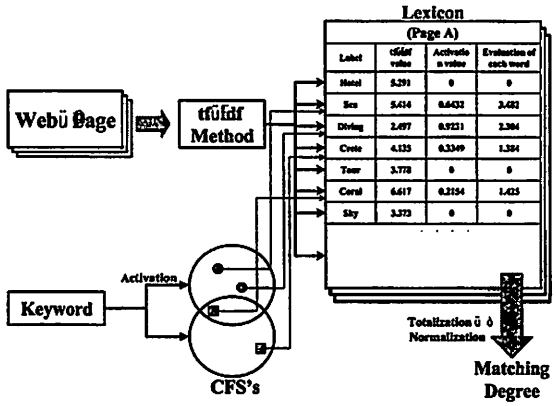


Fig.10. Scheme of the proposed search engine.

5. Simulations and Evaluations

Let's think about the case where we are searching for Web pages of places to visit using certain keywords, we indexed 200 actual Web pages, and compared the search result of the following two matching methods.

1. TF-IDF method
2. our proposed method

Evaluation 1:
If the CFS unit has knowledge in fuzzy sets about places, and if a user inputs "famous resort" as a keyword, relating name of places are added as expanded keywords with their activation degrees agreeing with membership degrees.

Famous resort = 0.95/gold coast + 0.95/the Cote d'Azur + 0.91/Fiji + ..

Table 1 shows the result when "famous resort" and "the Mediterranean Sea" are input as keywords. It consists of names of places and activation values, and shows the extended keywords generated by the activation of the above two keywords.

Table 1. Extended keywords.

Ranking	Word	Activation Value
é P	The Côte d'Azur	1.0000
2	The Mediterranean Sea	0.9773
3	Famous Resort	0.9187
é S	Crete	0.8482
5	Capri	0.6445
6	Anguilla	0.6445
7	Santorini	0.6445
8	Taormina	0.6445
9	Sicily	0.4802
10	Gold Coast	0.0748

Next, abstracts of the retrieved Web pages are listed. Note that, no Web pages were matched by the simple TF-IDF method starting with the keyword input of "famous resort and the Mediterranean Sea."

Taormina: Ranking 1, Matching degree 1.0

The greatest high-class resort on the island of Sicily. It is located 250 meters above sea level and is known as the "Mediterranean Queen." It has superb views of Mount Etna and the Ionian Sea.

Crete island: Ranking 2, Matching degree 0.83

It is a big island and located in the south. The scenery is different from Mykonos and Santorini islands.

Cote d'Azur: Ranking 3, Matching degree 0.53

Deep-blue coast.

The above results show that our proposed method effectively retrieves information relating to input keywords even when there are no matches with the input keyword itself.

Evaluation 2:

If the CFS unit memorizes knowledge about "vacation" and "sports" such as,

vacation = 1.0/vacance + 0.6/sea + 0.6/sandy beach + 0.6/the South Pacific + ..
sports = 1.0/spots + 0.6/diving + 0.6/trekking + 0.6/golf + ..

then a ranked list of Web pages appears. Table 2 shows the extended keywords generated by the activation of "vacation" and "sports."

Table 2. Extended keywords.

Ranking	Word	Activation Value
é P	Diving	0.6079
1	Surfing	0.6079
3	Sports	0.5000
3	Vacation	0.5000
5	Golf	0.3039
5	Rock Climbing	0.3039
5	Baseball	0.3039
5	Sea	0.3039
5	Paradise	0.3039
5	Sandy Beach	0.3039

Next, abstracts of the retrieved Web pages are listed. In contrast, no Web pages were matched by the TF-IDF method using "vacation and sports."

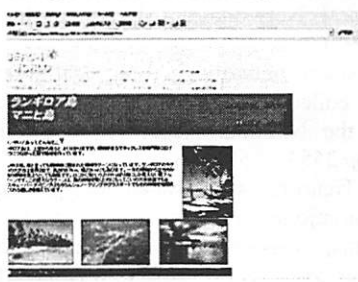
Tahiti island: Ranking 1, Matching degree 1.00

Fun for jet skiing and surfing. Diving is also enjoyable.

Boracay island: Ranking 2, Matching degree 0.91

Diving boats and cruising boats come and go. Vacationers have to climb on the boat from the water because there are no piers. It exemplifies resort life.

Rangiroa island: Ranking 3, Matching degree 0.84



Genuine diving! Even snorkeling and a glass-bottomed boat allow glimpses of its mystery.

From the results, we demonstrate the effectiveness of our proposed method. Unlike the first case, the Web pages were not retrieved by place names, but by the activities corresponding to the context of "vacation sports." Jet skiing and surfing were suggested by the CFS as a relevant sports, but baseball was not.

We shows that pertinent Web pages can be retrieved independently of the key ward, because even though the

region "sport" can encompass a huge number of different activities.

6. Conclusion

First, we showed the necessity and also the problems of applying fuzzy sets to information retrieval. Next, we introduced using conceptual fuzzy sets in overcoming those problems, and proposed the realization of conceptual fuzzy sets using Hopfield Networks. Based on above, we proposed the architecture of the search engine which can execute conceptual matching dealing with context-dependent word ambiguity. Finally, we evaluated our proposed method through two simulations of retrieving actual web pages, and compare the proposed method with the ordinary TF-IDF method. We showed that our method could correlate seemingly unrelated input keywords and produce matching Web pages, whereas the simple TF-IDF method could not.

References

[1] T. Takagi, A. Imura, H. Ushida, and T. Yamaguchi, "Conceptual Fuzzy Sets as a Meaning Representation and their Inductive Construction," International Journal of Intelligent Systems, Vol. 10, 929-945 (1995).

[2] T. Takagi, A. Imura, H. Ushida, and T. Yamaguchi, "Multilayered Reasoning by Means of Conceptual Fuzzy Sets," International Journal of Intelligent Systems, Vol. 11, 97-111 (1996).

[3] T. Takagi, S. Kasuya, M. Mukaidono, T. Yamaguchi, and T. Kokubo, "Realization of Sound-scape Agent by the Fusion of Conceptual Fuzzy Sets and Ontology," 8th International Conference on Fuzzy Systems FUZZ-IEEE'99, II, 801-806 (1999).

[4] T. Takagi, S. Kasuya, M. Mukaidono, and T. Yamaguchi, "Conceptual Matching and its Applications to Selection of TV Programs and BGMs," IEEE International Conference on Systems, Man, and Cybernetics SMC'99, III, 269-273 (1999).

[5] Wittgenstein, "Philosophical Investigations," Basil Blackwell, Oxford (1953).

[6] B. Kosko, "Adaptive Bi-directional Associative Memories," Applied Optics, Vol. 26, No. 23, 4947-4960 (1987).

[7] B. Kosko, "Neural Network and Fuzzy Systems," Prentice Hall (1992).

[8] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences U.S.A., Vol.79, pp.2554-2558 (1982)

[9] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons, Proceedings of the National Academy of Sciences U.S.A., Vol.81, pp.3088-3092 (1984)

[10] M. Kobayashi, K. Takeda, "Information retrieval on the web", ACM Computing Survey, Vol.32, pp.144-173 (2000)

[11] N. Guarino, C. Masalo, G. Vetere, "OntoSeek : content-based access to the Web", IEEE Intelligent Systems, Vol.14, pp.70-80 (1999)

Internet-based Systems for Design, Planning, Operating and Marketing in the Mining, Minerals, Metals and Materials Industry

John A. Meech
University of British Columbia, Department of Mining and Mineral Process Engineering
6350 Stores Road, Vancouver, B.C., V6T 1Z4, Canada
Tel: 604-822-3984 Fax: 604-822-5599 Email: jam@mining.ubc.ca

ABSTRACT

Internet use for corporate decision-making and mine-plant operations is expanding rapidly in the Mining and Processing sector. Application areas include property prediction and selection of materials, process design and optimization, plant control and maintenance, coordination and teleremote operation of equipment, coordination of interacting stages in a complex processing system.

On-line marketplaces are evolving to sell raw materials and to order and deliver services and supplies in the primary resources sector. Coal, iron ore, copper, nickel, zinc and other commodities are now being traded routinely over the Internet with opportunities to barter products, materials and services without the need to exchange money. Existing commodity exchanges such as the London and New York Metal Exchange are threatened by such direct activities between customer and seller.

The main goal of these systems is to provide a collaborative environment for cross-disciplinary interaction and contextual analyses of identical sets of data and/or information. Information in these systems includes databases for discrete information; knowledge bases with heuristic, fuzzy structures and rules; real-time snap-shots of the current state of a process or design problem; multiple user-interface views of the working environment; and simulation models for real-time dynamic and/or steady-state analyses of the problem-space. Groups of engineers, operators and technologists can now work on the same problem from remote locations without ever actually meeting in person. The implications of such environments on the ability of corporations to function in the Information Age is discussed together with the impact of such systems on the efficiency and effectiveness of human decision-makers.

Introduction

Mining and mineral extraction are generally viewed as separate functions within the design and operation of a mining complex. Although certain stages within a mine-mill facility are batch processes, many are continuous. Attempts to improve efficiency or effectiveness of each stage is generally carried out in isolation without regard to up- or down-stream implications of a particular change except perhaps, at the long-term corporate level. Although such an approach may "optimize" an individual stage, it does not provide a means to examine solutions that lie across several related processes.

Separate optimization of each stage does not guarantee that the overall process is optimized. Recent advances in the area of Factory Automation and Manufacturing Science suggest that the time has arrived to apply several new hardware and software techniques to the mine-mill complex. These include: intelligent database mining; just-in-time production scheduling; robotics; remote-sensing; automated mining systems; processing ore at the "face"; and final product manufacturing at the mine site.

The mining and metallurgical industries are at a crossroads. Faced with a long-term trend of ever-declining commodity prices together with increasingly complex ores and decreasing grades, mining companies today must implement one of two strategies:

1. continue the routine of cutting costs by labour-reduction and by adopting new technologies;
2. expand their organizational horizons to integrate across the mine and mill interface and to include value-added down-stream facilities.

The first option is the knee-jerk response as companies try to improve operating practices when times are tough and then take profits quickly by high-grading or pushing tonnage when resource prices improve. In the short-term (3-5 years), this policy can provide some relief, but one must pick the cycles correctly or disaster may occur and often does. Most people in our industry consider it to be far-easier to focus on what we do best -- mining, processing and, in some cases, extraction -- and avoid the confusion of the value-added end-user markets.

However for long-term sustained growth, with the advent of the InterNet and its broad support for rapid communication and distribution, a successful mining company today must implement at least part of option 2. Companies must move from a position of only producing concentrate and should examine ways to achieve value-added opportunities in their end-user market(s).

This paper examines some of the software tools being applied in the Manufacturing industries to aid in complex decision-making to reorganize or adjust a facility to meet the heuristic demands of the marketplace. These tools offer assistance in a number of interesting and creative ways that include both long- and short-term planning as well as real-time process monitoring and control. It is suggested that using these systems within the Mining industry by individual companies or by a group of sector-based enterprises can provide significant relief to the problem of long-term profitability and sustainability. Major improvements can be also gained in the image of Mining as a sustainable and profitable business center, an environmentally-friendly industry and as a modern-user of high-technology.

THE LIMITS OF "ECONOMY-OF-SCALE"

Mining companies traditionally "stick to their knitting". They do their work in areas for which they clearly have expertise and leave the downstream processing to others: companies in Japan, Korea, and existing smelters, refineries and metal producers in other parts of the world. Only rarely will a company justify the expenditures required to extend their processing facilities into such activities. Conventional wisdom states that it is more efficient and economic to centralize extraction and refining operations and receive intermediate concentrate products from distributed mining operations. However examples do exist of the opposite approach – the advent of the mini-steel mills in Canada to process scrap and raw materials into steel; the evolution of hydrometallurgical processes to produce final product at the mine; the creation of power and other energy products by coal conversion such as the South African SASOL plants.

It is unusually large or rich deposits that provide the incentive to add complexity. "Economies-of-scale" prevail to dictate a centralized approach or the use of "monster" equipment. However times are changing and some of the advantages of "economies-of-scale" are beginning to disappear. Many issues which have been impediments in the past are now opportunities:

THE ADVENT OF "COMPLEX" ANALYSIS

Companies can now examine many more options in their decision-making than ever before. There may be need for flexible design, operation and product-marketing to respond to changing commodity prices, competition from other sectors (geographically- or technologically-based such as Al vs. Cu; composite materials vs. super-alloys; fibre-optics vs. coaxial cable; etc.), complex ore changes; complex technological changes (new systems of communication, new advanced materials, robotics, nanotechnology, etc.)

Starting with plant design and examining product diversification, we must adapt our plans and processes to meet these forces. A mine must be able to adjust production on-demand and avoid stockpiling. It must react to changes in ore conditions and customer demands.

Impurities	- complex nature of our ores are increasing. - some ores contain impurities demanding a separate, unique downstream processing. - custom smelters may not accept materials with high Hg, As, Se, Bi and other undesirables.
New Processes	- many deposits are better exploited using hydro-metallurgical techniques such as Pressure Oxidation, Bio-Leaching, Electrowinning, etc. - - this provide for metal production at the mine-site.
Local Markets	- local markets may exist to sustain final product.
Recycling	- desire to recycle may help to create such markets.
Value-added	- certain products can be made relatively cheaply allowing addition of significant value with minimal investment and operating costs; e.g., gold nuggets sell at a premium ranging from $50 - $200 per ounce above the official selling price of gold. - Although nuggets represent only 1-2% of the gold jewelry market, local conditions can provide conditions to manufacturing "artificial" nuggets.
Regulations	- regulations are often contradictory. In an attempt to solve one problem, new problems crop up in another area; e.g., environmental laws on waste disposal are implemented without examining process options or the form of an element which may determine its toxicity and/or bioavailability.
Infrastructure	- sustaining mining activity in remote regions of a country can contribute to jobs and economic growth if infrastructure support is provided.
Design Impact	- availability of down-stream processing can affect decisions on the design and operation of a mine-mill enterprise to reduce costs.
Local Resources	- the presence of local resources such as power, rail, shipping ports, etc., can provide an incentive to invest in downstream processing.
Delivery costs	- savings in transportation and product delivery costs can derived from the presence of nearby smelting, refining and/or manufacturing facilities.
Complexities	- complex, interactive decision-making across an overall enterprise has not been possible because of poor data-communication, poor data-collection and poor data-analysis. - such is not the case today.

INNOVATIVE MANUFACTURING SYSTEMS

Managing a company in the 21st century requires a new way to communicate with the external environment. Companies of the Third Millennium must transform into intelligent, learning organizations able to cope with globalization of information resources. The main problem will not be access to information but the ability to mine data and transform it into useful operating and strategic resources [1].

As system models become increasingly complex, decomposition into smaller units is the usual way to structure a problem. Historically this has led to atomized

structures consisting of many **autonomous subsystems**, each of which decide on what information to receive and send out -- and when to do it. Autonomous subsystems are embedded into larger systems, since autonomy and independence are not equivalent concepts. These ideas are gaining strong interest and the atomized approach to information-flow modeling and evaluation is an idea whose time has come [2]. In the real-world, autonomous subsystems consist of groups of people and/or machines tied together by the flow of information and materials.

Advances in computer technology have led to the design of extremely complex systems in areas such as advanced manufacturing systems, transportation systems and world models (economic and ecological). The complexity of these systems requires distributed supervisory functions -- that is, an assembly of individual modules must be defined and coordinated within a comprehensive control architecture. While some controllers direct processes, others supervise. An effective architecture [3] should possess the following features:

- Users can specify high-level tasks, which are then decomposed into detailed execution tasks according to an established hierarchy or distribution network,
- Users can plan and control at different resolutions of time and level of detail,
- The system can decompose complex behaviors into manageable sub-functions,
- The system allows functions to be distributed across several intelligent controllers.

An example of such an architecture is the design implementation suggested by NASA/NIST as shown in Figure 1 [4,5,6].

Managing complexity, reacting to change and disturbances are key issues in production systems. Distributed, agent-based or holonic structures represent an alternative to hierarchical systems. Several approaches to implement such structures include: simulation modeling to develop and test agent-based architectures; the holonification of existing resources and "traditional" (centralized/hierarchical) systems. Cooperation of agent-based distributed control structures and evolutionary schedulers allow these systems to handle critical complexity, reactivity, disturbances and optimality issues simultaneously.

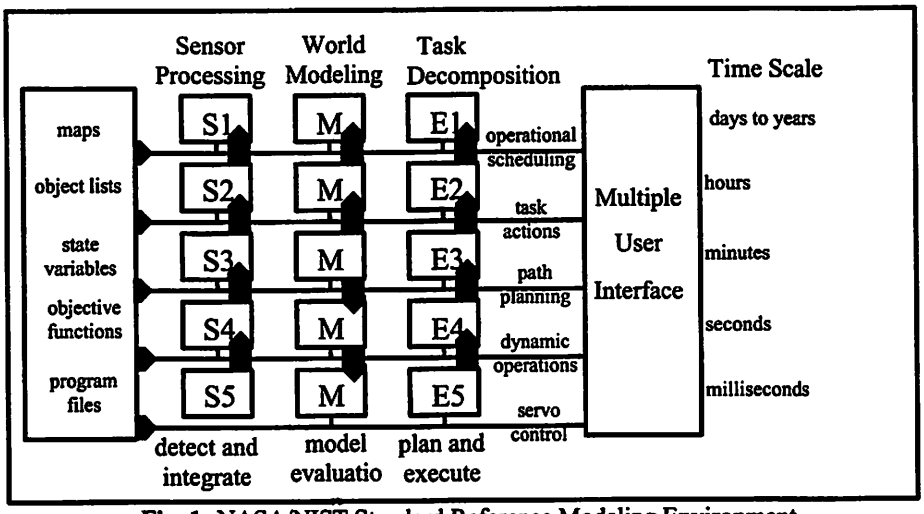


Fig. 1: NASA/NIST Standard Reference Modeling Environment (modified NASREM after Moncton, 1997 [4])

An agent is an "encapsulated" software entity with its own identity, state, behavior, thread of control, and ability to interact with other entities including people, other agents and "legacy" systems. An agent, whether real or virtual, is able to act on itself and on other agents. Its behavior is based on observations, knowledge and interactions with other agents in the system or process. An agent has several important features -- the ability to perceive at least partial representation of its environment, the ability to communicate with other agents, the ability to produce child agents, knowledge about its objectives and some rather unique autonomous behavior often characterized as selfishness [7,8].

Holonic manufacturing (Fig. 2) is a new paradigm that consists of autonomous, intelligent, flexible, distributed, cooperative agents or **holons** [9]. The word "holon" derives from the field of holography -- a holon is defined as "a part of a whole". Three basic types of holons, **resource holons, product holons** and **order holons**, have been defined [10] although other types or sub-types might be characterized for certain specific systems. These entities use object-oriented concepts such as aggregation

and specialization to perform their duties. The most promising feature of the holonic approach is that it provides a transition methodology from hierarchical to heterarchical systems, which are more representative of the real world. They will allow us to develop very-large scale simulations of complex organizations of people, systems and equipment. These can even include the emotional factors inherent in the decision-making across a large manufacturing or mining operation. Weather conditions, market factors, environmental requirements, operational difficulties -- all of these can be considered in the simulation model.

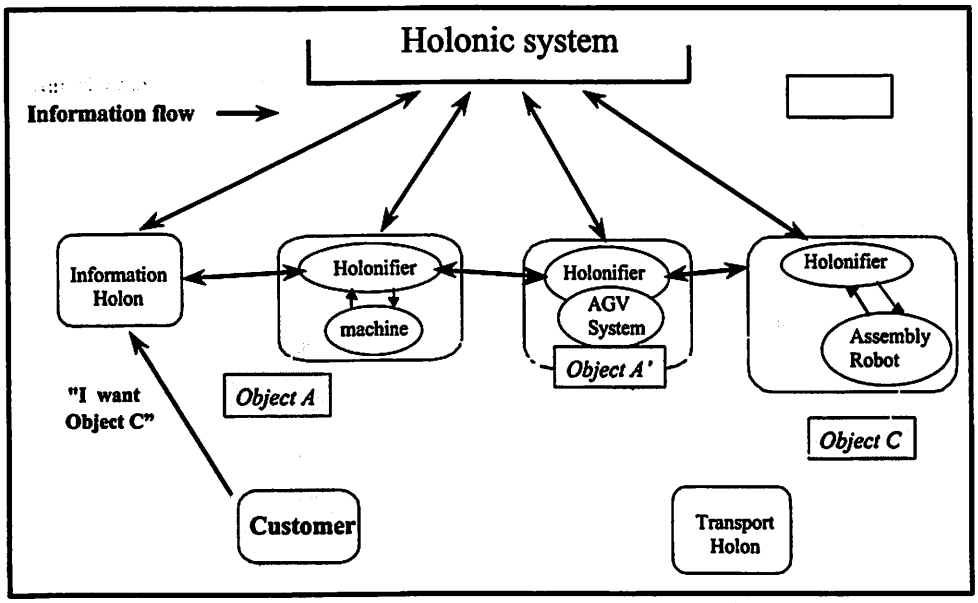


Fig. 2: Heterogeneous Holonic Manufacturing System consisting of real and soft holons. (after Monostori and Kádár, 1999 [7])

The main design issues of an agent-based system are:

- **Structure:** internal structure of agents and their level of their self-containment,
- **Communication:** protocols, common interchange language,
- **Group formation:** persuading machines/people to participate in a group, reward/penalty systems,
- **Configurability:** open systems (addition, deletion, substitution of machines/groups),
- **Scalability:** appropriateness of scale-up to the level of the extended enterprise,
- **Local optima:** reaching global optima with agents pursuing their own individual goals.

An object-oriented framework to develop a distributed system provides a model to represent a plant containing different agents. The object library holds two main types: **resource agents** and **order agents**. An order agent processes orders, announces jobs and dispatches messages among different resources or groups (Fig. 3). Many different resource agents are initialized during dynamic configuration (names, process capabilities, etc.).

Agents contain functionally-separated subagents. Each agent uses a communication subagent to handle messages by using a contract network protocol. Each resource agent involves a supervisor subagent to control real-world actions. A registration mechanism is used to startup and shutdown each agent. Local knowledge and data bases are used to store information on machine capacities, time intervals for jobs, groups in which an agent is interested, etc. Facts about the agent itself are accessed through the communication subagent by a request message [11].

Agent-based software engineering was invented to facilitate interoperability. There is much interest and development in "middle"-ware to deal with software that is already written -- **legacy software.**

The term "agent" can mean many things: mobile code, web search tool, interface tool, distributed component library, semantic broker (translator), applet, code with temporal duration or persistency, electronic commerce with message-passing entities, dynamic services, intelligent routers, robots, etc.

Many programs need task-fulfilling properties to deal with: assignment problems, burst bandwidth problems with mobile code capability, open source information, interoperability with brokering, etc. Agents can eliminate "data overload" and "information starvation" difficulties by providing "just-in-time" flow of information. Three key design aspects of an agent system are:

- number of agents required
- number of types of agents required
- number of actions that an agent can perform

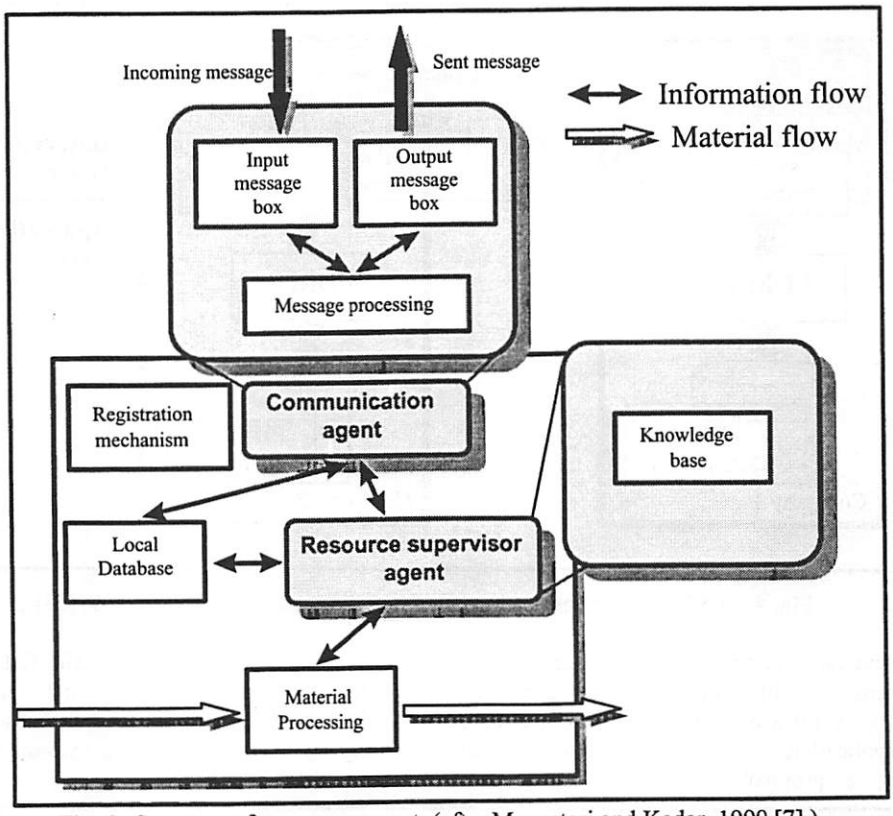


Fig. 3: Structure of a resource agent. (after Monostori and Kadar, 1999 [7])

There are four main task-functions that define an agent:

- dealing with complex **problem-solving**,
- finding, filtering and presenting **information**,
- providing **services** to other agents to solve problems cooperatively,
- **providing translation services** among agents using different standards, communication protocols, languages, etc.

Large-scale, cooperative teams consist of interacting agents from all four groups,. They offer capabilities beyond the realm of conventional software design. An infrastructure providing these features allows for the design of small pieces of code to solve specific problems that can interact with other pieces of code, rather than duplicating functions in each module. In this way, task-specific agents can be built up from smaller functional agents that can participate in multiple activities.

DATA MODELS AND COMMUNICATION

A major problem in integrating complex systems is the method of communication used across the system. A number of important communication protocols have been established -- some of these will now be described.

The STEP Standard

The STEP Standard (STandard for the Exchange of Product Model Data - ISO 10303) established in 1985 was the first to use an integrated product model approach to provide semantic data models (Application Protocols such as AP214 and AP203) together with mechanisms for data exchange [12, 13].

The file-based exchange used by STEP is based on "Processors" which transform data from each Product Data Management system into files using a standard format and a standard data-model (Part 21, ISO 10303) (see Fig. 4). The STEP standard also provides mechanisms to share data via the Standardized Data Access Interface defined by ISO 10303-22 in 1994 [14].

CORBA

The Object Management Group (OMG), a consortium of companies from all facets of the computer industry, has defined the Common Object Request Broker Architecture (CORBA). CORBA is middle-ware that allows intelligent components to discover each other and inter-operate on an object bus [15]. This object bus is referred to as the ORB (Object Request Broker). The ORB abstracts information needed for remote components to communicate with each other.

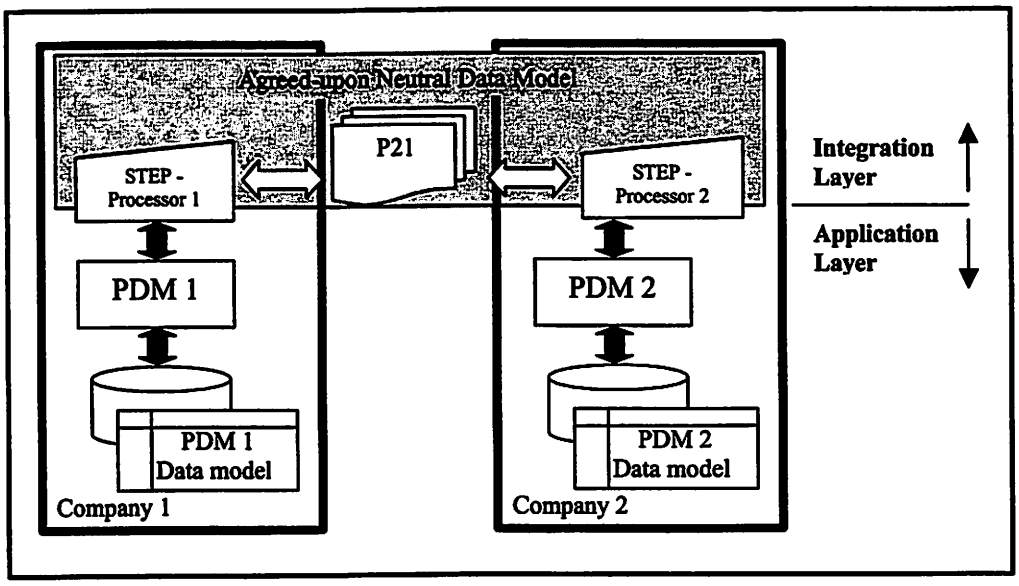


Fig. 4. A STEP based Integration Layer (after Karcher and Wirtz, 1999, [13]).

Components interact across networks, servers, and operating systems as if all resided on the same machine. This makes it easy for a developer to create networked client/server applications. The component boundaries are defined using a protocol known as the Interface Definition Language (IDL). The IDL file is compiled to generate client-side **stubs** and server-side **skeletons**, which permit components to access one another across languages, tools, operating systems and networks. This architecture can be seen in Fig. 5.

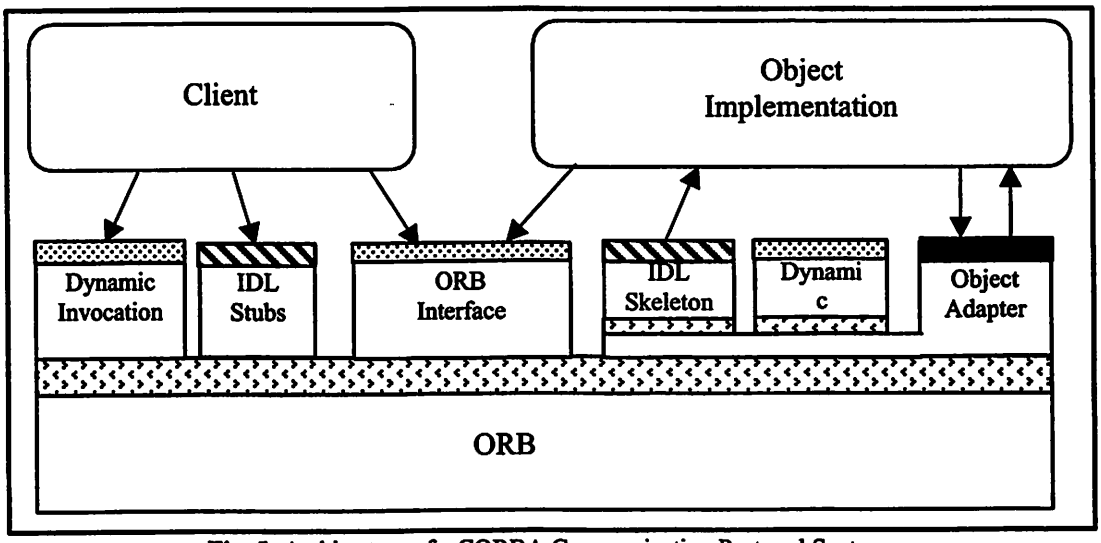


Fig. 5: Architecture of a CORBA Communication Protocol System.
(after Nicoletti, 1999 [16])

Interface identical for all ORB Implementations
There may be multiple Object Adaptors
Stubs and Skeletons are identical for each Object type
ORB dependent interface

So developers only need to worry about object interaction rather than focusing on locating remote objects and passing information along a wire. CORBA defines an extensive set of bus-related services to create

and delete objects, access them by name, store them in data-warehouses, externalize their states, and define ad-hoc relationships among them.

CORBA specifies a system that provides interoperability between objects in a heterogeneous, distributed environment that is transparent to the user. Its design is based on the OMG Object Model. OMG defines an object semantics to specify characteristics independent of the method of implementation [16].

CORBA operates as follows: *clients* request services from *objects* (servers) through a well-defined interface. This interface is specified by the IDL. A client accesses an object by issuing a *request* to the object. The request is an event containing:

1. information about the operation,
2. the object name of the service provider, and
3. any parameters.

To make a request, a client communicates with the ORB through its IDL *stub* or through a Dynamic Invocation Interface (DII). The stub presents a mapping between the language of the client implementation (C, C++, Java, and others) and the ORB core. The ORB then transfers the request to the Object Implementation which receives the request through an IDL or a dynamic skeleton as shown in Fig. 5.

APPLICATIONS IN MINING

The following list contains some applications that under study using an IMS agent-based or holonic system. Significant improvement in the day-to-day operation of a mine-mill complex can be achieved through these studies.

Intelligent Stockpiles

Stockpiling of ores can be reduced to a minimum or can be set-up to provide ore blending or individual treatment of specific ore types. IMS systems can provide the data and knowledge to begin to examine this practice [17, 18, 19].

Enhanced Comminution Systems

Establishment of optimum run-of-mine sized ore using blasting, primary crushing and autogenous grinding can be done through the employment of IMS systems [20]. Real examples of these benefits have been demonstrated by Mt. Isa [21] and Highland Valley Copper [22].

Coordinated Real-Time Maintenance

Coordination of scheduled maintenance has been done at several Canadian mines. Collection of cross-corporate data on equipment needs can enhance such coordination and lead to increased production and reduced costs. Operating times can increase significantly.

Tele-remote Operations

Baiden has demonstrated the tremendous advantages to be derived from telerobotic operation of underground equipment. These benefits include increased productivity, improved worker safety and reduced capital costs. A doubling of productivity is possible [23].

Enhanced Data Communication

CORBA-based Internet communication systems can provide baseline support to collect, store and present corporate-wide data for all enterprise levels. Wireless communication is now state-of-the-art allowing remote analysis of "live" data. Cost-savings from reduction in personnel and information coordination are considerable.

Discovery of New Ideas

By operating the models as a simulator, it is possible that new milling circuit designs can be devised to provide enhanced operation under certain heuristic situations.

Value-Added Production at the Mine

Diversification of mine production can help to withstand fluctuations in the conventional resource-sector markets. Several examples of companies who have already done this or who need to examine this option are as follows:

Polar Diamond - BHP/Ekati

BHP's new Ekati Mine is supplying a subsidiary firm in Yellowknife with a portion of their diamond production to manufacture the "Polar" Diamond with a picture of a polar bear lasered on the girdle of the cut stones.

Millennium Diamond - De Beers

De Beers have long been leaders in promoting end-use sales of their diamond production. The latest promotion is the creation of special Millennium diamonds with a strategy similar to that of BHP.

Gold Jewelry Production on-site

The Harmony Mine in South Africa have developed a novel process to produce 5-9s gold on site. A special training centre has been set up to train 40 local artisans per year to manufacture jewelry at the mine site.

Other gold mines should follow this example. In BC, there are likely opportunities to work with First Nations groups to provide new employment benefits in remote communities based on local cultural activities.

Chemical Products from Coal

The North American coal industry faces increased environmental concern for global warming from the release of green-house gases such as CO_2. As well, the competition form abroad has increased tremendously, particularly from Indonesia, leading to extremely low prices for Thermal Coal and Metallurgical Coal -- the only two products of significance in the coal industry.

With the expected increases in transportation and heating fuel costs, the industry needs to look towards value-added production of other energy-vector products such as fuels from coal and hydrogen production as was in vogue during the oil-crisis of the 1970s.

e. www.globalcoal.com

Four of the largest coal producers in the world -- Anglo American, BHP-Billiton, Glencore International and Rio Tinto, have entered into a joint venture to establish an e-

marketplace for thermal coal. A number of important coal consumers, including Endesa, Enel, EPDC and TXU Europe, are also likely to join in as founding shareholders. Together with industry participants who join in the future, the parties will fund Global Coal (www.globalcoal.com), and assist in creating initial market liquidity. The Global Coal e-marketplace is intended to benefit all participants by reducing transaction costs, increasing price transparency and enabling forward pricing. The marketplace will be neutral and open and will protect commercially sensitive information.

To meet the trading needs of the coal industry, Global Coal will support 3 transaction types. First, it will allow real-time trading for current and future standardised coal contracts. Second, it will provide an environment for custom transactions where the specification or delivery terms fall outside standard contract parameters. Third, it will facilitate a tender/request-for-proposal process for buyers.

Global Coal will also offer a broad range of coal industry news and resources. The company plans to extend its offering of value-added services to enhance its position as the premier business-to-business portal for coal industry participants. The marketplace will serve Atlantic and Asia-Pacific regions, with trading locations in the Netherlands, Colombia, South Africa, and Australia. Global Coal is the first e-marketplace in mining and the partners plan to pursue other opportunities in base metals, bulk ores and bulk alloys. At the time of writing, Global Coal was due to begin operations during the second quarter of 2001.

InterNet Commerce

Led by BHP, three of the largest Australian mining companies have announced the formation of a **B2B** enterprise (Business-to-Business). All suppliers are expected to use the facility to bid on contracts to supply goods ad services. Current suppliers have been told that if they do not adapt their business to operate over the Internet, the companies will cease doing business with them in the future.

In addition, BHP plans to market a portion of its rough diamond production over the Internet. Wholesalers and retailers alike will be able to bid on uncut diamonds directly from the mine. The process will bypass numerous intermediaries and decrease the cost of distribution and manufacturing.

CONCLUSIONS

With the increasing pressures of low commodity prices and foreign competition, mining companies must examine alternative strategies to deal with these complexities and remain sustainable. Tools currently being used within the manufacturing sector have potential to provide solutions in integrating data from across different departments.

These tools provide ways to collect, store and present real-time data for a variety of decision-making including direct or supervisory control and long-range planning.

Opportunities exist to enhance mine economics through value-added production by focusing on ways to recover and manufacture end-user products at the mine site and supply end-user customers with mine-derived products.

Failure to adopt these approaches will result in continued need for relief when commodity price drop and companies will be unable to develop the innovations required to provide future long-term sustainability.

ACKNOWLEDGEMENT

The author wishes to acknowledge the support and assistance by members of the IPMM group (Intelligent Processing and Manufacturing of Materials). IPMM consists of over 400 researchers around the world from a diverse set of backgrounds who share a common interest in intelligent processes. IPMM holds a bi-annual conference -- the 3rd IPMM Conference was held July 29 - August 3, 2001 in Vancouver, B.C. and the 4th will take place in May 2003 in Sendai, Japan.

REFERENCES

[1] Szczerbicki, E., Gomolka, Z. 1999. Management of information in complex systems: perspectives for the new Millenium. *Proc. IPMM'99* - 2nd Inter. Conf. on Intelligent Processing & Manufacturing of Materials, Honolulu, HI., Vol. 2, 749-752.

[2] Gunasekaran, A., Sarhadi, M. 1997. Planning and management in enterprise integration, *Concurrent Engineering*: Research and Applications, 235-247.

[3] Davis, W.J., 1999. Distributed intelligent control of complex systems. *Proc. IPMM'99* - 2nd Inter. Conf. on Intelligent Processing & Manufacturing of Materials, Honolulu, Hawaii, Vol. 1, 583-590.

[4] Monckton, S.P., 1997. Multiagent manipulator control. Ph.D. Thesis, *University of British Columbia*, Dep't. Mech. Eng., Vancouver, B.C.

[5] Albus, J.S., Quintero, R., 1990. Towards a reference model architecture for real-time intelligent control systems (ARTICS). *Proc. ISRAM'90*, 243-250.

[6] Lumia, R., 1994. NASREM for real-time sensory interactive robot control. *Robotica*, 12, 127-35.

[7] Monostori, L., Kádár, B., 1999. Agent-based control of manufacturing systems. *Proc. IPMM'99* - 2nd Inter. Conf. on Intelligent Processing & Manufacturing of Materials, Honolulu, HI., Vol. 1, 125-131.

[8] Koussis, K., Pierreval, H., Mebarki, N., 1997. Using multi-agent architecture in FMS for dynamic scheduling, *Journal of Intelligent Manufacturing*, 16(8), 41-47.

[9] Valckenaers, P., Bonneville, F., van Brussel, H., Bongaerts, L., Wyns, J., 1994. Results of the holonic

system benchmark at KULeuven, *Proc. CIMAT* - 4th Inter. Conf. on Computer Integrated Manufacturing & Automation Tech., Troy, New York, 128-133.

[10] van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems, *Computers in Industry*, 37(3), 255-276.

[11] Monostori, L., Kádár, B., Hornyák, J., 1998. Approaches to managing changes and uncertainties in manufacturing, *CIRP Annals*, 47(1), 365-368.

[12] ISO 10303-11, 1994. Industrial automation and integration - Product data representation and exchange - Pt. 11, *EXPRESS* language manual.

[13] Karcher, A., Wirtz, J., 1999. PDM-based virtual enterprises - bridging the semantic gap. *Proc. IPMM'99* - 2nd Inter. Conf. on Intelligent Processing & Manufacturing of Materials, Honolulu, HI., 1, 591-596.

[14] ISO 10303-22, 1994. Industrial automation systems and integration - Product data representation and exchange - Pt. 22: *Standard data access interface.*

[15] Orfali, R., Harkey, D., Edwards, J., 1997. *Instant CORBA*, Wiley Computer Publishing, New York.

[16] Nicoletti, G.M., 1999. Redefining the web: toward the creation of large-scale distributed applications. *Proc. IPMM'99* - 2nd Inter. Conf. on Intelligent Processing & Manufacturing of Materials, Honolulu, HI., Vol. 2, 1013-1018.

[17] Harris, C.A., Meech, J.A., 1987. Fuzzy Logic: A Potential Control Technique for Mineral Processing. *CIM Bulletin*, 80(905), 51-59.

[18] Chilviet, E.I., Meech, J.A., 1996. Intelligent systems to assist in SAG circuit supervision. in *Mine Simulation*, G. Panagiotou, J. Sturgul, (Eds.), *Proc. MineSim '96*, 1st Inter. Symp. on Mine Simulation via the InterNet, CyberSpace and Athens, Greece, A.A. Balkema, Rotterdam, p. 39 plus 14 HTML pages.

[19] P. Molck, R. Gonçalves, T. Caldas, J. Valentim, L. Lima, E. Newton, M. França, R. Mendes, F. Gomide' 2001. Intelligent Stockpile Building in Iron Ore Shipping Yard. Proc. IPMM-2001, Vancouver, B.C., July 29-Aug. 3, 2001. ep.11.

[20] Baiden, G., Meech, J., 1987. Simulating the Mine-Mill Interface. *Inter. J. Surface Mining*, 1(3), 191-198.

[21] Pease, J.D., Young, M.F., Johnson, M., Clark, A., Tucker G., 1998. Lessons from Manufacturing - Integrating Mining and Milling for a Complex Orebody. Mine to Mill Conference, Brisbane.

[22] Simkus, R., Dance, A., 1998. Tracking Hardness and Size: Measuring and Monitoring ROM Ore Properties at Highland Valley Copper. Mine to Mill Conf., Brisbane.

[23] G. Baiden, 1999. Telemining™ Systems Applied to Hard Rock Metal Mining at Inco Ltd., *Proc. IPMM'99* - 2nd Inter. Conf. on Intelligent Processing & Manufacturing of Materials, Honolulu, HI., 1, 53-5

Integration of Document Index with Perception Index and Its Application to Fuzzy Query on the Internet

Dae-Young Choi
Department of EECS, CS division, Univ. of California, Berkeley,
The BISC group (Visiting Scholar), 199MF Cory Bldg. Berkeley, CA 94720-1770.
dychoi@EECS.Berkeley.EDU

Absrract

Commercial Web search engines such as Yahoo !, Google, etc. have been defined which manage information only in a crisp way. Their query languages do not allow the expression of preferences or vagueness. In order to handle these problems, we propose the Perception Index (PI) that contains attributes associated with a focal keyword restricted by fuzzy term(s) used in fuzzy queries on the Internet. If we integrate the Document Index (DI) used in commercial Web search engines with the proposed PI, we can handle both crisp terms (keyword-based) and fuzzy terms (perception-based). In this respect, the proposed approach is softer than the keyword-based approach. The PI brings somewhat closer to natural language. It is a further step toward a real human-friendly, natural language-based interface for Internet. It should greatly help the user relatively easily retrieve relevant information. In other words, the PI assists the user to reflect his/her perception in the process of query. Consequently, Internet users can narrow thousands of hits to the few that users really want. In this respect, the PI provides a new tool for targeting queries that users really want, and an invaluable personalized search.

1. Introduction

The central concept of information retrieval is the notion of relevance [9]. A user with a given query for information tries to find any specific results that he/she really wants. There are several models for specifying the representations used for the documents and the queries, as well as the matching of these representations [4]. The most used model is that of the Boolean query based on set theory. Documents are represented as sets of terms and queries are Boolean expressions on terms. The retrieval mechanism does an exact match by classifying documents that satisfy the Boolean query as being relevant, all other documents as being irrelevant. This model is used by virtually all commercial textual-document retrieval systems. However, it is difficult to overcome the limitations of this model, including the inability to handle properly imprecision and subjectivity. The second model is the vector space model [9] where documents and queries are represented as vectors in the space of all possible index terms. The document vectors consist of weights based on term frequencies in the collection, while the query vectors are binary vectors on the terms. The matching is based on a similarity measure between the documents and the query (often involving the cosine of the angle between the query vector and a given document vector). To date, this model leads the others in terms of performance. The third model is the probabilistic model [9] where documents are represented as binary vectors. The queries are vectors of terms with weights based on the estimated probability of relevance of documents with those terms. Like the vector space model, the key advantage is the ability to rank documents on the likelihood of relevance. The fourth model is the generalized Boolean model, where fuzzy set theory allows the extension of the classical Boolean model to incorporate weights and partial matches, and adding the idea of document ranking.

The importance of representations of uncertainty in databases is increasing as more complex applications such as CAD/CAM and geographical information systems (GIS) are being undertaken in object-oriented and multi-media databases. Query languages are designed to express the user's retrieval requests in either a crisp manner or not. Much of the work in the database area has been in extending query languages to permit the representation and retrieval of imprecise data [2,6,7,8,10]. There are some current commercial attempts at providing fuzzy query capabilities as front ends to conventional database systems [6].

Until now, however, commercial systems including informational retrieval systems (IRS), data base management systems (DBMS), and Web search engines have been defined which manage information only in a crisp way. Moreover, (crisp) traditional query languages do not allow the expression of preferences or vagueness which could be desirable for the following reasons [4] :

- to control the size of the results;
- to express soft retrieval conditions;
- to produce a discriminated answer.

Although the commercial Web search engines such as Yahoo !, Google, Lycos, etc. help Internet users get to good information, they do not properly handle fuzzy query. For example, consider a fuzzy query that finds '***popular*** national parks in the USA'. In this case, '***popular***', 'national parks' and 'USA' are generally processed as keywords in the commercial Web search engines. As a result, search engines return a bunch of page titles (or URLs) irrelevant to user's query. For example, given a fuzzy query that finds '***popular*** national parks in the USA', Yahoo ! returns about 34,200 page titles (or URLs) and Google returns about 73,100 page titles (or URLs). Intuitively, we find that there are so many page titles (or URLs) irrelevant to user's query. It should be noted that fuzzy term '***popular***' is a constraint on the focal keyword 'national parks' rather than an independent keyword. In other words, the fuzzy term '***popular***' plays the role of a constraint on the fuzzy query. Thus, using fuzzy term(s), Internet users can narrow thousands of hits to the few that users really want. In this respect, the fuzzy terms in a query provides helpful hints for targeting queries that users really want, and an invaluable personalized search. However, commercial Web search engines tend to ignore the importance of fuzzy terms in a query. The expressive power of conventional

search engine query interfaces is relatively weak when restricted to keyword-based search (i.e., Document Index (DI)-based search) [3]. At present, the keyword-based search engines present limitations in modeling perceptual aspects of humans. In addition, they appear to have trouble with returning the targeted results. In other words, they generally return a bunch of Web pages (or URLs) irrelevant to user's query. In this respect, we need a new tool to handle both the fuzzy query and the removal of spurious results. In order to tackle these problems, we propose the Perception Index (PI) that contains attributes associated with a focal keyword restricted by fuzzy term(s) in a fuzzy query.

2. Integration of DI with PI

Search engines are the most popular tools that people use to locate information on the Web. A search engine works by traversing the Web via the hyperlinks that connect the Web pages, performing text analysis on the pages it has encountered, and indexing the pages based on the keywords they contain. A user seeking information from the Web would formulate his/her information goal in terms of a few keywords composing a query. A search engine, on receiving a query, would match the query against its Document Index (DI). All of the pages that match the user query will be selected into an *answer set* and be ranked according to how relevant the pages are with respect to the query. Relevancy here is usually based on the number of matching keywords that a page contains [3]. The DI is generally consisted of keywords that appear in the title of a page or in the text body. Based on the DI, the commercial Web search engines such as Yahoo !, Google, Lycos, etc. help users get to good information. For example, BigBook (or SuperPages) can help users to find 'Italian restaurants within a 1-mile radius from a specific address' (U.S. yellow pages services) [5]. This proximity search is processed based on crisp query with keywords (i.e., 'Italian restaurants', '1-mile', 'a specific address'). However, they do not properly process fuzzy queries. For example, find '***popular*** national parks in the USA'. In addition, they have problems as follows [3]:

- large answer set;
- low precision;
- unable to preserve the hypertext structures of matching hyperdocuments;
- ineffective for general-concept queries.

In this paper, we do not attempt to solve 'unable to preserve the hypertext structures of matching hyperdocuments' problem. However, we try to tackle in part the other problems (i.e., 'large answer set', 'low precision', 'ineffective for general-concept queries'). In order to handle these problems, we propose a Perception Index (PI). The remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations is derived from the brain's crucial ability to manipulate perceptions – perceptions of distance, size, weight, color, speed, time, direction, force, number, truth, likelihood, and other characteristics of physical and mental objects. Familiar examples of the remarkable human capability are parking a car, driving in heavy traffic, playing golf, riding a bicycle, understanding speech, and summarizing a story [12]. These perceptions are mainly manipulated based on fuzzy concepts. For processing a fuzzy query, the PI is consisted of attributes associated with a keyword restricted by fuzzy term(s) in a fuzzy query. In this respect, the restricted keyword is named as a focal keyword, whereas attribute(s) associated with the focal keyword may be regarded as focal attribute(s). The PI can be mainly derived from the contents in the text body of a Web page or from the other sources of information with respect to a Web page. For example, the PI may be consisted of distance, size, weight, color, etc. on a keyword in the text body of a Web page. Using the PI, search engines can process fuzzy concepts (terms). In the sequel, if we integrate the DI used in commercial Web search engines with the proposed PI, search engines can process fuzzy queries. For example, consider a fuzzy query that finds '***popular*** national parks in the USA'. In this case, the fuzzy term '*popular*' is processed by using the PI, whereas keywords '*national parks*' and '*USA*' are processed by using the DI. We note that 'in' and 'the' in the above fuzzy query are examples of stop words ignored by search engines (see <www.google.com>). See Table 1 for an example of integrated index (DI+PI) in the Appendix.

It should be noted that fuzzy term(s) may be regarded as a constraint on a fuzzy query. For example, consider a fuzzy query that finds '***popular*** national parks in the USA'. In this case, the fuzzy term '***popular***' play the role of a constraint on the fuzzy query. In other words, using fuzzy term(s), Internet users can narrow thousands of hits to the few that users really want. In this respect, the PI provides helpful hints for targeting queries that users really want, and an invaluable personalized search.

The expressive power of conventional search engine query interfaces is relatively weak when restricted to keyword-based search [3]. At present, commercial Web search engines based on the DI (i.e., keyword-based search engines) present limitations in modeling perceptual aspects of humans. In addition, they generally return a bunch of Web pages (or URLs) irrelevant to user's query. Although much Web search engines have been developed, they do not properly handle the fuzzy terms representing human's perception. In addition, they appear to have trouble with returning the targeted results. In order to tackle this problem, we integrate the DI used in commercial Web search engines with the proposed PI. In the proposed method, given a fuzzy query, search engine processes the fuzzy query based on the integrated index (DI+PI).

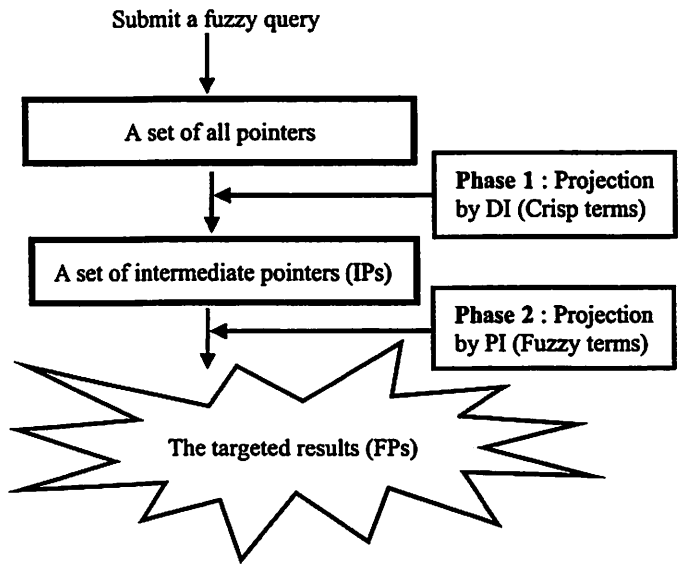


Fig. 1. A search mechanism based on the integrated index

In Fig. 1, if we submit a query with only crisp terms (keyword-based query), this search engine uses only the phase 1. By applying the DI, the phase 1 performs an elimination-based approach to eliminate the URLs which are impossible to be the answers of the query. In this case, this search engine will return the same results that the existing search engines do. On the other hand, if we submit a query with both crisp terms and fuzzy terms, this search engine uses both phase 1 and phase 2. In this case, by applying the PI, the URLs reflecting fuzzy terms are extracted. More specifically, the phase 2 evaluates the fuzzy terms in detail on the set of intermediate pointers (i.e., the candidate URLs), and then generates the final pointers (FPs) (i.e., targeted results) that user really wants. This search mechanism can be conceptually explained by SQL-like language as follows : *SELECT * FROM {a set of intermediate pointers that satisfies focal keyword(s) in the DI} [WHERE the value(s) of focal attribute(s) in the PI are satisfied by the user]*. We note that commercial Web search engines tend to ignore the importance of *[WHERE]* part. In this approach, the PI may be regarded as a constraint on the DI.

3. Fuzzy Query based on the Integrated Index

We assume that a fuzzy term in a fuzzy query is marked with an asterisk. For example, it is expressed as '***popular*** national parks in the USA'. If a query has fuzzy term(s) marked with asterisk(s), search engine displays a PI associated with a focal keyword restricted by fuzzy term(s). Then user can specify values with respect to the fuzzy terms.

3.1 Types of Fuzzy Query

Fuzzy query is largely divided into simple fuzzy query and compound fuzzy query.

(1) Simple Fuzzy Query

The simple fuzzy query does not include conjunction ('*and*') or disjunction ('*or*') connective(s) between fuzzy terms, or negation ('*not*').

Example 1. Consider a fuzzy query that finds '***popular*** national parks in the USA'. In this case, the DI, the PI, and stop words may be as follows : DI = {national parks, USA, ...}, PI = {No. of visitors, ...}, stop words = {in, the}. We note that a focal keyword 'national parks' in the DI is restricted by a fuzzy term 'popular'. In this case, the fuzzy term '*popular*' may be manipulated by the number of visitors (i.e., a focal attribute in the PI) per year, and represented by the following membership function :

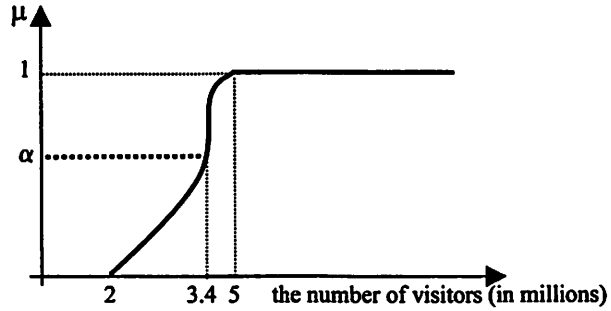


Fig. 2. A membership function of '*popular*'

Example 2. Consider a fuzzy query that finds 'national parks ***moderate*** distance from San Francisco'. In this case, the DI, the PI and stop words may be as follows : DI = {national parks, San Francisco, ...}, PI = {distance, ...}, stop words = {from}. We note that a focal keyword 'San Francisco' in the DI is restricted by a fuzzy term 'moderate'. In this case, the fuzzy term '*moderate*' may be manipulated by the degree of distance (i.e., a focal attribute in the PI) from San Francisco, and represented by the following membership function :

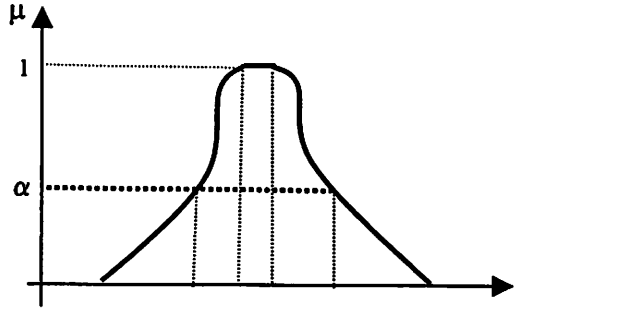


Fig. 3. A membership function of '*moderate*'

(2) Compound Fuzzy Query

The compound fuzzy query includes conjunction ('*and*') or disjunction ('*or*') connective(s) between fuzzy terms, or negation ('*not*').

3.2 Query Processing based on the Integrated Index

Now, we present how this search engine processes fuzzy queries. Let the set of national parks in the USA be $A = \{A_1, A_2, ..., A_{99}, A_{100}\}$ and each A_i, ($i = 1,2, ..., 100$) has its own PT(page title) or URL.

Example 3. Consider a crisp query that finds 'national parks in the USA' (Q_1). In this case, the PI is not used. So, this search engine uses only the phase 1 in Fig. 1. Thus, the integrated index is made as follows : See Table 2 in the Appendix.

In the crisp query case, this search engine returns the same results that the existing search engines do. We note that IPs and FPs are equal in Table 2.

Example 4. Consider a fuzzy query that finds '***popular*** national parks in the USA' (Q_2). In this case, the DI and the PI are used. So, this search engine uses both the phase 1 and the phase 2 in Fig. 1. We assume that '***popular*** national parks in the USA' are A_p, $A_p \in \{A_1, A_2, ..., A_{99}, A_{100}\}$, by using α-cut in Fig. 2. Thus, the integrated index is made as follows : See Table 3 in the Appendix.

Example 5. Consider a fuzzy query that finds 'national parks ***moderate*** distance from San Francisco' (Q_3). In this case, the DI and the PI are used. So, this search engine uses both the phase 1 and the phase 2 in Fig. 1. We assume that 'national parks ***moderate*** distance from San Francisco' are A_m, $A_m \in \{A_1, A_2, ..., A_{99}, A_{100}\}$, by using α-cut in Fig. 3. Thus, the integrated index is made as follows : See Table 4 in the Appendix.

Example 6. Consider a fuzzy query that finds 'national parks that ***popular*** *and* ***moderate*** distance from San Francisco' (Q_4). In this case, the DI and the PI are used. So, this search engine uses

both the phase 1 and the phase 2 in Fig. 1. Then the query results with respect to Q_4 become $\{A_p\} \cap \{A_m\}$. For instance, let A_p be a set $\{A_1, A_2, A_3\}$ and A_m be a set $\{A_1, A_4, A_5\}$, then $\{A_p\} \cap \{A_m\} = \{A_1\}$. Thus, the integrated index is made as follows : See Table 5 in the Appendix.

Example 7. Consider a fuzzy query that finds 'national parks that ***popular** or ***moderate** distance from San Francisco' (Q_5). In this case, the DI and the PI are used. So, this search engine uses both the phase 1 and the phase 2 in Fig. 1. Then the query results with respect to Q_5 become $\{A_p\} \cup \{A_m\}$. For instance, let A_p be a set $\{A_1, A_2, A_3\}$ and A_m be a set $\{A_1, A_4, A_5\}$, then $\{A_p\} \cup \{A_m\} = \{A_1, A_2, A_3, A_4, A_5\}$. Thus, the integrated index is made as follows : See Table 6 in the Appendix.

Example 8. Consider a fuzzy query that finds '*not* ***popular** national parks in the USA' (Q_6). In this case, the DI and the PI are used. So, this search engine uses both the phase 1 and the phase 2 in Fig. 1. Then the query results with respect to Q_6 become $\{\sim A_p\}$. For instance, let A_p be a set $\{A_1, A_2, A_3\}$, then $\{\sim A_p\} = \{A_4, A_5, \ldots, A_{99}, A_{100}\}$ if the universal set $A = \{A_1, A_2, \ldots, A_{99}, A_{100}\}$. Thus, the integrated index is made as follows : See Table 7 in the Appendix.

3.3 User Interface based on the Integrated Index

An important problem relating to personalization concerns understanding how a machine can help an individual user via suggesting recommendations [1]. In our approach, the PI can help the user to specify clearly what he/she really wants. More specifically, the user in the system is asked to specify fuzzy term(s) in a query. In this respect, the PI may be regarded as a recommendation for handling fuzzy term(s) in a query. As a result, search engine returns 'the targeted results'. Now, we describe user interface for phase 1 and phase 2 in Fig. 1.

(1) User Interface for Phase 1

Initially, user interface for phase 1 lets user specify his/her queries with only crisp terms (keywords), or both crisp terms and fuzzy terms. If user submits a query with only crisp terms, only user interface for phase 1 is used, and search results are returned based on only the DI. On the other hand, if user submits a query with both crisp terms and fuzzy terms, user interface for phase 2 is also displayed to process the fuzzy terms.

(2) User Interface for Phase 2

User interface for phase 2 displays a PI associated with a focal keyword. For example, given a fuzzy query that finds '***popular** national parks in the USA', a PI associated with a focal keyword 'national parks' is displayed as shown in Table 3. It should be noted that different people may use different conceptual comprehension (fuzzy terms, membership functions, α-cut), with respect to the same situation. It is the user's task in this user interface to examine the suggested attributes in the PI, and to specify the values of the focal attributes reflecting user's query requirements. Using the PI, search results can be restricted within narrow limit. We call it '*target search by fuzzy terms*'. In other words, search engine will return the targeted results that users really want.

Fuzzy terms are specified in user interface for phase 2. For example, they can be expressed as point value, interval value, multiple values, etc.

- ***Point value***

Example 9. In Example 1, given a α-cut, the fuzzy term '*popular*' may be specified by using a focal attribute 'no. of visitors'. More specifically, it is expressed as a point 3.4 (i.e., 'no. of visitors' $\geq$ 3.4 millions).

- ***Interval value***

Example 10. In Example 2, given a α-cut, the fuzzy term '*moderate*' may be specified by using a focal attribute 'distance'. More specifically, it is expressed as an interval (i.e., distance = [50, 150] in miles).

- ***Multiple values***

A veristic variable [11,12] which can be assigned two or more values in its universe simultaneously will be specified as multiple values.

Example 11. Let U be the universe of natural languages and let X denote the fluency of an individual in English, French and Italian. Then, X *isv* (1.0 English + 0.8 French + 0.6 Italian) means that the degrees of fluency of X in English, French and Italian are 1.0, 0.8 and 0.6, respectively [11,12].

4. Some Features of the Proposed Method

Remark 1. The higher the α in α-cut ($0 \leq \alpha \leq 1$), the smaller the number of the targeted results. This property provides continual incremental result from 'the highest constraint (i.e., $\alpha = 1$)' to 'the lowest constraint (i.e., $\alpha = 0$)'. Consequently, we can achieve 'interactive user control of the query processing' by adjusting the value of α.

Remark 2. If $\alpha = 0$, search results coincide with the results by applying only the DI (i.e., the existing keyword-based search). In this case, the results of phase 1 in Fig. 1 become search results.

Remark 3. Even though the same integrated index (DI+PI) is given, different search results are returned by adjusting the value of α or by using different focal attributes in the PI. In the case of 'using different focal attributes in the PI', for example, consider a fuzzy query that finds '*attractive* car', where '*attractive*' means 'comfortable and fast'. In this case, for the fuzzy term '*attractive*', people may use different focal attributes (i.e., size, speed, etc.) in the PI. In addition, different people may use different conceptual comprehension (fuzzy terms, membership functions, α-cut), with respect to the same situation. Thus, search engine will return the personalized search results that users really want. In the meantime, clustering (i.e., grouping similar documents together to expedite information retrieval) is adaptively determined depending on the value of α or the selected focal attributes in the PI.

Remark 4. Using the PI, Internet users can narrow thousands of hits to the few that users really want.

Remark 5. Using the PI, therefore, we can tackle in part the major problems in commercial Web search engines (i.e., 'large answer set', 'low precision', 'ineffective for general-concept queries').

Remark 6. For comparing with commercial keyword-based search engines, the ratio *[the number of FPs / the number of IPs]* can be used as a measure of performance evaluation on the proposed method. We note that the number of IPs is the result of phase 1 and the number of FPs is the result of phase 2 in Fig. 1. The smaller the ratio, the better the filtering effect of the proposed method.

5. Conclusions

The expressive power of conventional search engine query interfaces is relatively weak when restricted to keyword-based search (i.e., Document Index (DI)-based search). At present, the keyword-based search engines present limitations in modeling perceptual aspects of humans. In addition, they appear to have trouble with returning the targeted results. In other words, they generally return a bunch of Web pages (or URLs) irrelevant to user's query. In this respect, we need a new tool to handle both the fuzzy query and the removal of spurious results. In order to tackle these problems, we introduce the Perception Index (PI) that contains attributes associated with a focal keyword restricted by fuzzy term(s) in a fuzzy query. If we integrate the Document Index (DI) used in commercial Web search engines with the proposed PI, we can handle both crisp terms (keyword-based) and fuzzy terms (perception-based). In this respect, the proposed approach is softer than the keyword-based approach (i.e., commercial Web search engines). It is a further step toward a real human-friendly, natural language-based interface for Internet. It should greatly help the user relatively easily retrieve relevant information. In other words, the proposed method assists the user to reflect his/her perception in the process of query. As a consequence, Internet users can narrow thousands of hits to the few that users really want. In this respect, the PI provides a new tool for targeting queries that users really want, and an invaluable personalized search. The use of PI provides helpful hints for solving the problems of 'large answer set', 'low precision', 'ineffective for general-concept queries' suffered by most search engines.

In this paper, we also present the search mechanism based on the integrated index (DI+PI) and fuzzy query based on the integrated index (DI+PI). Moreover, we describe some features of the proposed method.

Acknowledgements

The author wishes to thank Prof. L. A. Zadeh for his inspirational address on the perceptual aspects of humans. The idea of Perception Index (PI) is inspired by his papers [11, 12] and his comments in the BISC (Berkeley Initiative in Soft Computing) seminars and group meetings. He also thanks Sun-Gyung Jung, plan & control manager/education center of Oracle Korea, Si-Young Choi and Jae-Hyun Choi, for their encouragement. This work was supported by postdoctoral fellowships program from Korea Science & Engineering Foundation (KOSEF). In addition, funding for this research is provided in part by the BISC Program of UC Berkeley and the BT Advanced Communication Technology Center (BTexaCT)-Advanced Research-Computational Intelligence (Program Manager: Ben Azvine).

References

[1] N. J. Belkin (2000) Helping people find what they don't know, Communications of the ACM 43(8) : 58-61.

[2] J. Kacprzyk and A. Ziolkowski, Retrieval from databases using queries with fuzzy linguistic quantifiers, Fuzzy logic in knowledge engineering (Edited by Prade H and Negoita C. V), Verlag TUV Rheinland, 1986.

[3] B. Kao, J. Lee, C. Y. Ng and D. Cheung (2000) Anchor point indexing in Web document retrieval, IEEE trans. on SMC(part C) 30(3) : 364-373.

[4] D. H. Kraft and F.E. Petry (1997) Fuzzy Information systems : managing uncertainty in databases and information retrieval systems, Fuzzy sets and systems 90(2) : 183-191.

[5] D. Lidsky and R. Kwon (1997) Searching the net, PC magazine Dec. 2 : 227-258.

[6] H. Nakajima, T. Sogoh and M. Arao (1993) Development of an efficient fuzzy SQL for a large scale fuzzy relational database, Proc. 5th IFSA world congress : 517-530.

[7] F. E. Petry and P. Bosc, Fuzzy databases : principles and applications, Kluwer, Norwell, MA, 1996.

[8] D. Rasmussen and R. R. Yager (1999) Finding fuzzy and gradual functional dependencies with summarySQL, Fuzzy sets and systems 106(2) : 131-142.

[9] S. Salton, Automatic text processing : the transformation, analysis and retrieval of information by computer, Addison-Wesley, Reading, MA, 1989.

[10] C. A. Testemale, Database system dealing with incomplete or uncertain information and vague queries, Fuzzy logic in knowledge engineering (Edited by Prade H and Negoita CV), Verlag TUV Rheinland, 1986.

[11] L. A. Zadeh (1997) Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy sets and systems 90(2) : 111-127.

[12] L. A. Zadeh (1999) From computing with numbers to computing with words – From manipulation of measurements to manipulation of perceptions, IEEE trans. on circuit and systems 45(1) : 105-119.

Appendix

Table 1. An example of Integrated Index (DI + PI)

Document Index (DI)	IPs	Perception Index (PI)				FPs (Results)
Keywords	URLs	Distance	Size	No. of visitors	...	Targeted URLs

(IPs : Intermediate Pointers; FPs : Final Pointers; URLs : Uniform Resource Locators)

Table 2. A snapshot of Integrated Index after processing Q_1

Document Index (DI)	IPs	Perception Index (PI)			FPs (Results)
National parks, USA	A_1	Distance	No. of visitors	...	A_1
	A_2	Distance	No. of visitors	...	A_2
	...	...	...	...	...
	A_{99}	Distance	No. of visitors	...	A_{99}
	A_{100} + Irrelevant URLs	Distance	No. of visitors	...	A_{100} + Irrelevant URLs

Table 3. A snapshot of Integrated Index after processing Q_2

Document Index (DI)	IPs	Perception Index (PI)			FPs (Results)
National parks, USA	$\{A_1,., A_{100}\}$ + Irrelevant URLs	Distance	**No. of visitors**	...	URLs w.r.t $\{A_p\}$

(Focal keyword : National parks; Focal attribute : No. of visitors)

Table 4. A snapshot of Integrated Index after processing Q_3

Document Index (DI)	IPs	Perception Index (PI)			FPs (Results)
National parks, **San Francisco**	$\{A_1,., A_{100}\}$ + Irrelevant URLs	**Distance**	No. of visitors	...	URLs w.r.t $\{A_m\}$

(Focal keyword : San Francisco; Focal attribute : Distance)

Table 5. A snapshot of Integrated Index after processing Q_4

Document Index (DI)	IPs	Perception Index (PI)			FPs (Results)
National parks, **San Francisco**	$\{A_1,.,A_{100}\}$ + Irrelevant URLs	**Distance**	**No. of visitors**	.	URLs w.r.t $\{A_p\} \cap \{A_m\}$

(Focal keyword : San Francisco; Focal attributes : Distance and no. of visitors)

Table 6. A snapshot of Integrated Index after processing Q_5

Document Index (DI)	IPs	Perception Index (PI)			FPs (Results)
National parks, **San Francisco**	$\{A_1,., A_{100}\}$ + Irrelevant URLs	**Distance**	**No. of visitors**	...	URLs w.r.t $\{A_p\} \cup \{A_m\}$

(Focal keyword : San Francisco; Focal attributes : Distance and no. of visitors)

Table 7. A snapshot of Integrated Index after processing Q_6

Document Index (DI)	IPs	Perception Index (PI)			FPs (Results)
National parks, USA	$\{A_1,., A_{100}\}$ + Irrelevant URLs	Distance	**No. of visitors**	...	URLs w.r.t $\{\sim A_p\}$

(Focal keyword : National parks; Focal attribute : No. of visitors)

Fuzzy Conceptual Graphs for the Semantic Web

T.H. Cao

Artificial Intelligence Group
Department of Engineering Mathematics
University of Bristol
United Kingdom BS8 1TR
Tru.Cao@bristol.ac.uk

Abstract
Despite its great usefulness in making information widely available to everyone, the current World Wide Web is showing its limitations with the explosion of information over the Internet. Its hypertext-based languages, like HTML, the information represented by which is mainly for human reading rather than machine processing, have hampered more advanced applications and better services on the Internet than ones currently available. For its next generation, the so-called Semantic Web, a logical formalism that can approach human expression and reasoning is a very good language candidate for Web documents. Conceptual graphs and fuzzy logic are two logical formalisms that emphasize the target of natural language, where conceptual graphs provide a structure of formulas close to that of natural language sentences while fuzzy logic provides a methodology for computing with words. This paper proposes fuzzy conceptual graphs, which combine the advantages of both the two formalisms, as a suitable Semantic Web language.

1. Introduction

We have seen how very useful the World Wide Web is to our daily life and work. It allows nearly instant access to an extremely large virtual pool of information from over the world and facilitates electronic services and businesses on the Internet. The current Web technology is however very simple. It represents information mainly in texts for human reading, using HTML (HyperText Markup Language)[1], for example, as a language to provide a structure for storing and displaying them. With the explosion of information on the Internet that we are witnessing, such a simple language without a machine-processable semantics becomes a bottle-neck for finding information as well as for maintaining and presenting it. As an example, it restricts search engines to be just keyword-based and thus, when searching for information on the Web, one often receives many useless links. That is simply because, for instance, one cannot express the difference between the two queries to find Web pages about "Publications written by Tim Berners-Lee" and "Publications written about Tim Berners-Lee", which have the same keywords "book" and "Tim Berners-Lee" but different semantics.

[1] http://www.w3.org/MarkUp

To overcome this short coming, many researchers and practitioners have started making a great effort to extend the current Web towards the so-called Semantic Web ([1]), in which information is represented by languages with expressive but well-defined semantics to be processable by computers. In our view, this challenge is similar to the one of Artificial Intelligence, namely that of how to make computers approach human expression and reasoning. In particular, this requires a formal knowledge representation language that not only has the expressive power close to that of natural language, but also can be automatically processed by computers. The context of the Web adds one more difficult task to this, that is, the represented knowledge must also be comprehensible and interoperable widely over the Internet, and this requires accompanying ontologies.

How humans process information represented in natural language is still a challenge to science in general, and to Artificial Intelligence in particular. However, it is clear that for a computer with the conventional processing paradigm to process natural language, a formalism is required. For reasoning, it is desirable that such a formalism be a logical one. A logic for handling natural language should have not only a structure of formulas close to that of natural language sentences, but also a capability to deal with the semantics of vague linguistic terms pervasive in natural language expressions.

Currently, conceptual graphs (CGs) ([13]) and fuzzy logic ([18]) are two logical formalisms that emphasize the target of natural language, each of which is focused on one of the two mentioned desired features of a logic for handling natural language. Indeed, while a smooth mapping between logic and natural language has been regarded as the main motivation of conceptual graphs ([14], [15]), a methodology for computing with words has been regarded as the main contribution of fuzzy logic ([19], [20]). Conceptual graphs, based on semantic networks and Peirce's existential graphs, combine the visual advantage of graphical languages and the expressive power of logic. Meanwhile, fuzzy logic, based on fuzzy set theory, has been developed

for approximate representation of, and reasoning with, imprecise and vague information.

This paper proposes fuzzy conceptual graphs (FCGs) ([2], [11], [16]), which combine the advantages of both conceptual graphs and fuzzy logic, as a suitable knowledge representation language for the Semantic Web. Firstly, Sections 2 and 3 briefly present the basic notions of conceptual graphs and fuzzy logic, respectively. Section 4 introduces fuzzy conceptual graphs with their recent development. Then Section 5 discusses and proposes fuzzy conceptual graphs as a suitable Semantic Web language. Finally, Section 6 concludes the paper and suggests future research.

2. Conceptual Graphs

A *simple CG* is a bipartite graph of *concept* vertices alternate with (conceptual) *relation* vertices, where edges connect relation vertices to concept vertices ([13]). Each concept vertex, drawn as a box and labelled by a pair of a *concept type* and a *concept referent*, represents an entity whose type and referent are respectively defined by the concept type and the concept referent in the pair. Each relation vertex, drawn as a circle and labelled by a *relation type*, represents a relation of the entities represented by the concept vertices connected to it. For brevity, we may call a concept or relation vertex a concept or relation, respectively. Concepts connected to a relation are called *neighbour concepts* of the relation. Each edge is labelled by a positive integer and, in practice, may be directed just for readability.

For example, the CG in Figure 1 says "John is a student. There is a subject. Computer Science is a field of study. The subject is in Computer Science. John studies the subject", or briefly, "John studies a subject in Computer Science".

Figure 1: A simple CG

In a textual format, concepts and relations can be respectively written in square and round brackets as follows:

[STUDENT: John]→(STUDY)→[SUBJECT: *]
↓
[FIELD: Computer Science]←(IN)

Here, for simplicity, the labels of the edges are not shown.

In this example, [STUDENT: John], [SUBJECT: *], [FIELD: Computer Science] are concepts with STUDENT, SUBJECT and FIELD being concept types, whereas (STUDY) and (IN) are relations with STUDY and IN being relation types. The referents John and Computer Science of the concepts [STUDENT: John] and [FIELD: Computer Science] are *individual markers*. The referent * of the concept [SUBJECT: *] is the *generic marker* referring to an unspecified entity. Two concepts with two different individual markers are assumed to refer to two different entities, while concepts with the same individual marker are assumed to refer to the same entity.

The first-order predicate logic semantics of conceptual graphs is defined through operator Φ ([13]) that maps a CG to a first-order predicate logic formula. Basically, Φ maps each vertex of a CG to an atomic formula of first-order predicate logic, and maps the whole CG to the *conjunction* of those atomic formulas with all variables being existentially quantified. Each individual marker is mapped to a constant, each generic marker is mapped to a variable, and each concept or relation type is mapped to a predicate symbol. For example, let G be the CG in Figure 1, then $\Phi(G)$ is:

$\exists x$ (student(John) $\wedge$ subject(x) $\wedge$
field(Computer Science) $\wedge$
study(John, x) $\wedge$ in(x, Computer Science)).

Conceptual graphs can also be nested. A *nested CG* is recursively defined as a simple CG extended by adding a *descriptor* field to each of its concepts, where a descriptor is either empty or a nested CG, which describes the referent of a concept. With this structure, *negation* of a proposition can be represented in conceptual graphs as a unary relation of type NEG whose connected concept is of type PROPOSITION and has the CG representing that proposition as its descriptor. This gives conceptual graphs the full expressive power of order-sorted predicate logic. For example, the following CG represents the negation of "John studies Literature":

(NEG)→[PROPOSITION: *
[STUDENT: John]
↓
(STUDY)→[SUBJECT: Literature]].

A fundamental operation on CGs is *CG projection*, which maps a CG to another more or equally specific one, by mapping each vertex of the former to a vertex of the latter that has a more or equally specific concept type and referent, or relation type. The mapping must also preserve the adjacency of the neighbour concepts of a relation. As such, if a CG has a projection to another one, then the latter logically implies the former. Figure 2 illustrates a projection from G to H, provided that STUDENT is a subtype of PERSON and STUDY is a subtype of ACT.

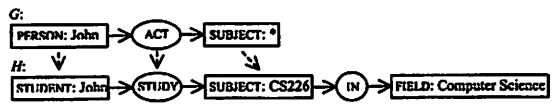


Figure 2: A CG projection

3. Fuzzy Logic

Often concepts encountered in the real world are vague in nature, like *young* or *old*, *short* or *tall*, *cheap* or *expensive*, as reflected in natural language. These concepts, or their expressions in natural language, are vague in the sense that, in most contexts, there is no clear-cut boundary between them and *not young* or *not old*, *not short* or *not tall*, *not cheap* or *not expensive*, respectively. In other words, the membership of an object in the extension of such a concept is not a matter of "to be or not to be", but rather a matter of degree. Classical set theory, in which the membership grade of an element in a set can only be either 0 or 1, is thus inadequate to deal with vague concepts. This was the main motivation of Zadeh founding fuzzy set theory ([17]), which generalizes classical set theory by defining membership grades to be real numbers in the interval [0, 1].

Fuzzy set theory then gave birth to fuzzy logic ([18]). In the literature, the term *fuzzy logic* has been used for different logic systems that have originated from the theory of fuzzy sets. However, they may have so different characteristics that they need to be distinguished to avoid confusion. We classify them into three main groups, namely, *partial truth-valued logic*, *possibilistic logic*, and *fuzzy set logic*. We call a fuzzy logic that deals with partial truth partial truth-valued logic, e.g. [12], which is a special multiple-valued logic, whose formulas are associated with real numbers in the interval [0, 1] interpreted as truth degrees of formulas. In contrast, in possibilistic logic ([7]), although formulas are also associated with real numbers in the interval [0, 1], they have the meaning of uncertainty degrees. Finally, in the broadest sense, we call a fuzzy logic whose formulas involve fuzzy set values fuzzy set logic, e.g. [3].

Fuzzy logic is supposed to be complementary to other approaches, in particular the long time and very well founded probability theory, for dealing with uncertainty and imprecision, which apparently exist in the real world and thus human expression and reasoning. The most controversial aspect of fuzzy set theory and fuzzy logic is also their cornerstone, the definition of fuzzy sets, that is, what a membership grade means. As answers, there have been so far two main schools of thoughts, one of which take membership grades as a primitive notion and the other links them with probability.

In the latter school, the voting model interpretation of fuzzy sets is as follows (cf. [9]). Given a fuzzy set A on a domain U, each voter has a subset of U as his/her own crisp definition of the concept that A represents. For example, a voter may have the interval [0, 35] representing human ages from 0 to 35 years as his/her definition of the concept *young*, while another voter may have [0, 25] instead. The membership function value $\mu_A(u)$ is then the proportion of voters whose crisp definitions include u. As such, A defines a probability distribution on the power set of U across the voters, and thus a fuzzy proposition "x is A" defines a family of probability distributions of the variable x on U.

However, we argue that the current lack of a unique and clearly defined semantics of membership grades of fuzzy sets should not discourage us from developing the theory. We, human beings, use many such vague concepts and degrees in [0, 1] in daily life, but it is still a challenge to science in general, and to Artificial Intelligence in particular, how we actually process these cognitively. It is fuzzy logic that takes this challenge, moving forward from the traditional probability theory, which is apparently not adequate to give the answer. We think theoretical research and practical experiments in both of the schools of fuzzy logic mentioned above should be encouraged. In practice, fuzzy logic has been successfully applied to several areas, such as expert systems, knowledge acquisition and fusion, decision making, and information retrieval, among others.

4. Fuzzy Conceptual Graphs

Firstly, *simple FCGs* extend simple CGs with linguistic labels defined by fuzzy sets as individual markers ([2], [11], [16]). For example, in Figure 3, the simple FCG G expresses "Peter, a Swede, is *tall*", where *tall* is the linguistic label of a fuzzy set. The contracted form of G is G^*, where the concepts [SWEDE: Peter] and [HEIGHT: *@*tall*] are called *fuzzy entity concept* and *fuzzy attribute concept*, respectively.

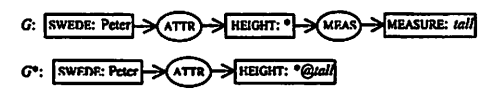


Figure 3: Simple FCGs

A concept type or a relation type can be a *fuzzy type* to represent uncertainty and/or partial truth about the type of the entity referred by a concept or the type of a relation, where a fuzzy type is defined as a pair of a *basic type* and an uncertainty and/or truth degree ([6]). An uncertainty and/or truth degree can be a *fuzzy truth value*, i.e., a fuzzy set on [0, 1] whose values are interpreted as truth degrees, or a *fuzzy probability value*, which is also defined by a fuzzy set on [0, 1] whose values are however interpreted as probability degrees. Examples of linguistic labels of a fuzzy truth value are *more or less true* and *not very false*, while ones of a fuzzy probability value are *quite likely* and *very unlikely*. A simple FCG with fuzzy types is shown in Figure 4, which says "It is *very true* that John is an American man, who is *young*, and it is *more or less true* that he likes a car, whose colour is *blue*", where

very true, *more or less true*, *young* and *blue* are linguistic labels of fuzzy sets.

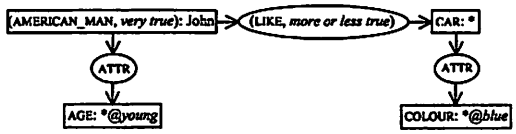


Figure 4: A simple FCG with fuzzy types

FCG projection, which matches an FCG to another one, extend CG projection with matching degrees between fuzzy sets and fuzzy types in the matched pairs of concepts and relations in the two FCGs. For representing complex information, we have introduced *nested FCGs* ([4]). As for nested CGs, a nested FCG is recursively defined as a simple FCG extended by adding a descriptor field to each of its concepts, where a descriptor is either empty or a nested FCG, which describes the referent of a concept. For example, the nested FCG in Figure 5 expresses "Tom, a Scot, shows a picture of a monster which is *about 20 feet* in length, and tells about a *very unlikely* situation that it exists in Lake Loch Ness". Here, the dotted line is a *coreference link* denoting that the two concepts [MONSTER: *] refer to the same entity; in the textual format, coreference is represented by variables of the same name as concept referents. Meanwhile, representation of such nested pieces of information in linear notations, e.g. predicate logic, would be difficult to follow.

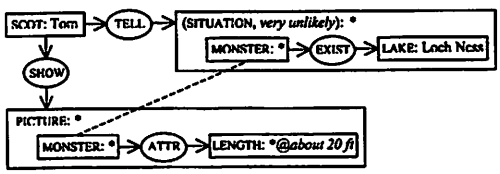


Figure 5: A nested FCG

One aspect of natural language the logic for which has been the quest and focus of significant research effort is one of generalized quantifiers. In reasoning with quantifiers, *absolute quantifiers* and *relative quantifiers* on a set have to be distinguished, where the quantities expressed by the latter are relative to the cardinality of the set. Examples of absolute quantifiers are *only one*, *few*, or *several*, while ones of relative quantifiers are *about 9%*, *half*, or *most*. In practice, there are quantifying words, e.g. *few* and *many*, that may be used with either meaning depending on the context. For instance, *few* in "*Few* people in this conference are from Asia" may mean a small number of people, while *few* in "*Few* people in the United Kingdom are from Asia" may mean a small percentage of population. Classical predicate logic with only the existential quantifier, equivalent to the absolute quantifier *at least 1*, and the universal quantifier, equivalent to the relative quantifier *all* or *every*, cannot deal with general quantification in natural language.

In the crisp case, absolute quantifiers can be defined by natural numbers, and relative quantifiers by non-negative rational numbers that are not greater than 1 measuring a proportion of a set, where 0 means 0% and 1 means 100%. Correspondingly, in the fuzzy case, absolute quantifiers can be defined by fuzzy sets on the set N of natural numbers, i.e., fuzzy numbers whose domain is restricted to N, and relative quantifiers by fuzzy numbers whose domain is restricted to the set of rational numbers in [0, 1]. For simplicity without rounding of real numbers, however, we assume absolute quantifiers to be defined by fuzzy numbers on [0, +∞] and relative quantifiers by fuzzy numbers on [0, 1]. The existential quantifier in classical logic corresponds to *at least 1* in natural language, which is an absolute quantifier whose membership function is defined by $\mu_{at\ least\ 1}(x) = 1$ if $x \geq 1$, or $\mu_{at\ least\ 1}(x) = 0$ otherwise. Meanwhile, the universal quantifier, which corresponds to *all* or *every* in natural language, is a relative quantifier and its membership function is defined by $\mu_{all}(1) = 1$ and $\mu_{all}(x) = 0$ for every $0 \leq x < 1$.

An absolute quantifier Q on a type T whose denotation set in a universe of discourse has the cardinality $|T|$ corresponds to the relative quantifier $Q_T = Q/|T|$. A relative quantifier Q in a statement "Q A's are B's" can be interpreted as the proportion of objects of type A that belong to type B, i.e., $Q = |A \cap B|/|A|$, which is a fuzzy number. Equivalently, it can also be interpreted as the fuzzy conditional probability, which is a fuzzy number on [0, 1], of $B(x)$ being **true** given $A(x)$ being **true** for an object x picked at random uniformly. We have applied this interpretation and the fuzzy set-theoretic semantics of generalized quantifiers to formally define the semantics of *generally quantified FCGs* as probabilistic logic rules comprising only simple FCGs ([5]). For example, Figure 6 shows a generally quantified FCG G and its defining expansion E, expressing "*Most* Swedes are *tall*", where {*} denotes a *set referent*, *most* and *tall* are linguistic labels of fuzzy sets, and *most* = *Pr(tall(x)* | Swede(*x*)).

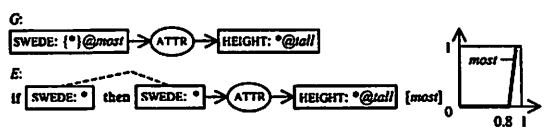


Figure 6: A generally quantified FCG and its defining expansion

Furthermore, the type in a generally quantified concept can be represented by a simple FCG as a lambda expression defining that type. We call such a simple FCG a *lambda FCG*, which is like a simple FCG

except that it has one concept, which we call a *lambda concept*, whose referent is denoted by λ to be distinguished from the generic and individual referents. For example, Figure 7 illustrates a generally quantified FCG *G* and its defining expansion *E*, expressing "*Most* people who are *tall* are *not fat*". As such, a lambda FCG corresponds to a relative clause in natural language.

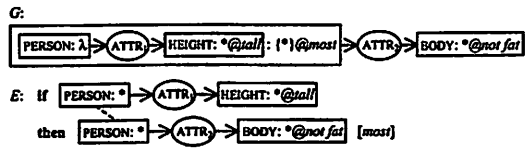


Figure 7: Quantification on a type defined by a lambda FCG

5. Towards the Semantic Web

As mentioned in the introduction section, despite its great usefulness in making information widely available, with the explosion of information on the Internet the simple hypertext-based Web is showing its limitations, so new knowledge representation languages are required for its next generation, the Semantic Web. As a first step, XML (eXtensible Markup Language)[2] has been introduced to allow everyone to add arbitrary structure to their documents and make use of that structure in sophisticated ways for domain- and task-specific purposes. Then higher-level languages based on XML, e.g., RDF (Resource Description Framework), OIL (Ontology Inference Layer), or DAML (DARPA Agent Markup Language), have been developed with more logical and ontological modelling primitives ([8]).

Although, as one of its goals, XML was designed for Web documents written in it to be human-legible and reasonably clear, it turned out to be too verbose. That is why other more readable languages like the ones mentioned above have been introduced. Comparing Web languages to programming languages, we view XML as a low-level language, while others are high-level languages, which are closer to human expression and reasoning. As such, Web documents may be written in any high-level language and then translated to XML for machine processing or information exchange. As an alternative, Web documents written in a high-level language can be processed directly by a specialized language processor.

A formal language that has a smooth mapping to and from natural language, like conceptual graphs, and can deal with imprecision and vagueness, like fuzzy logic, is a very good candidate for the Semantic Web. Being a logical formalism itself is an advantage of a language for reasoning on knowledge represented by it. Therefore we propose fuzzy conceptual graphs as a suitable Semantic Web language, which combines the mentioned advantages of both conceptual graphs and fuzzy logic as summarized below:

- The language is more human-readable than XML-based languages and other logical formalisms, like predicate logic. A query in natural language can be smoothly translated to an FCG to be matched to FCGs in Web documents, and an answer FCG can also be smoothly translated to natural language for human reading.
- The inherent graphical notation of the language is useful for knowledge visualization, in particular for presenting nested information whose representation in linear notations is difficult to follow. The structure for abstracting knowledge into nested concepts is helpful for knowledge organization and storage.
- The language itself is a logical formalism, which is desirable for knowledge reasoning. With its order-sorted feature, knowledge can be represented more concisely and reasoning can be performed more efficiently via inheritance. Its fuzzy logic core is significant for representation and reasoning with uncertain and imprecise information.

For example, Web pages about "Books written by Tim Berners-Lee" and "Books written about Tim Berners-Lee" as mentioned in the introduction section can be indexed by the following CGs

[PERSON: Tim Berners-Lee]
↑
(AGNT)←[WRITE: *]→(OBJ)→[BOOK: *]

and

[PERSON: *]
↑
(AGNT)←[WRITE: *]→(OBJ)→[BOOK: *]→(CONT)
↓
[PERSON: Tim Berners-Lee]

where (AGNT) and (OBJ) are relations between the action concept [WRITE: *] and its agent and object concepts, and (CONT) is a relation between an entity concept and its content as another entity concept. Then the query for "Publications written by Tim Berners-Lee" can be translated into the CG

[PERSON: Tim Berners-Lee]
↑
(AGNT)←[WRITE: *]→(OBJ))→[PUBLICATION: *]

which can then be matched to the first indexing CG above by CG projection and the corresponding Web pages are retrieved, provided that BOOK is a subtype of PUBLICATION. We note that this query CG cannot be projected to the second indexing CG above because "Tim Berners-Lee", referring to a specific entity, is more specific than "*", referring to an unspecified entity.

[2] http://www.w3.org/XML

For the fuzzy case, let us take an example adapted from [10], which discusses the advantages of conceptual graphs over XML-based languages and uses them for embedding knowledge to Web documents, in particular for indexing document elements. For instance, the following CG can be used to index an image, among others, about a red vehicle:

[VEHICLE: *]→(ATTR)→[COLOR: red]

where "red" is a non-interpreted individual marker. The implemented knowledge processor, called WebKB, is equipped with ontologies of concept types and relation types, in order to access knowledge represented in conceptual graphs. A query to retrieve an image about something that is red can be translated in to the following CG

[SOMETHING: *]→(ATTR)→[COLOR: red]

which can then be projected to the indexing CG above and the corresponding image is retrieved.

That work however did not consider vague linguistic terms, of which *red* is an example. Therefore, a query to find an image about something that is *very red* would fail, as its corresponding CG could not match to the indexing CG. Meanwhile, using fuzzy conceptual graphs, such an image can be indexed by the FCG

[VEHICLE: *]→(ATTR)→[COLOR: *@*red*]

where *red* is a linguistic term defined by a fuzzy set. Then the query about something *very red* represented by the following FCG

[VEHICLE: *]→(ATTR)→[COLOR: *@*very red*]

can be answered to a certain degree defined by the matching degree of the fuzzy set defining *very red* to that defining *red*, using FCG projection from the query FCG to the indexing FCG. Here, for dealing with fuzziness, besides the ontologies of concept and relation types, one needs also an ontology of vague concepts with their defining fuzzy sets.

6. Conclusion

We have presented the basic notions of conceptual graphs and fuzzy logic, showing that these two logical formalisms actually have the common target of natural language. We then have introduced fuzzy conceptual graphs, which combine the advantages of both conceptual graphs and fuzzy logic, as a knowledge representation language for Artificial Intelligence approaching human expression and reasoning. In particular, we have proposed fuzzy conceptual graphs as a suitable knowledge representation language for the Semantic Web, in which information is not only for human reading but also for machine processing.

There are still many aspects to be investigated for this proposal. A specific subset of fuzzy conceptual graphs with a balance between the expressive power and the computational tractability in the Web context is to be identified. The management of an ontology of vague concepts is a problem to be studied. Followed up is the implementation of tools for processing fuzzy conceptual graphs in Web documents. These are some of the topics that we suggest for future research.

References

[1] T. Berners-Lee, J. Hendler and O. Lassila (2001). The semantic web. *Scientific American.*

[2] T.H. Cao (1999). Foundations of order-sorted fuzzy set logic programming in predicate logic and conceptual graphs. PhD Thesis, University of Queensland.

[3] T.H. Cao (2000). Annotated fuzzy logic programs. *International Journal for Fuzzy Sets and Systems*, 113, 277-298.

[4] T.H. Cao (2000). Fuzzy conceptual graphs: A language for computational intelligence approaching human expression and reasoning. In Sincak, P. et al. (eds.), *The State of the Art in Computational Intelligence*, Physica-Verlag, pp. 114-120.

[5] T.H. Cao (2001). Generalized quantifiers and conceptual graphs. In Proceedings of the 9th International Conference on Conceptual Structures, Springer-Verlag, to appear.

[6] T.H. Cao and P.N. Creasy (2000). Fuzzy types: A framework for handling uncertainty about types of objects. *International Journal of Approximate Reasoning*, 25, 217-253.

[7] D. Dubois, J. Lang and H. Prade (1994). Possibilistic logic. In Dov M. Gabbay et al. (eds), Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3, Oxford University Press, pp. 439-514.

[8] D. Fensel (ed.) (2000). The semantic web and its languages. *IEEE Intelligent Systems.*

[9] B.R. Gaines (1978). Fuzzy and probability uncertainty logics. *Journal of Information and Control*, 38, 154-169.

[10] P. Martin and P. Eklund (1999). Embedding knowledge in web documents. In Proceedings of the 8th International World Wide Web Conference.

[11] S.K. Morton (1978). Conceptual graphs and fuzziness in artificial intelligence. PhD Thesis, University of Bristol.

[12] J. Pavelka (1979). On fuzzy logic. *Zeitschrift fur Mathematik Logik und Grundlagen der Mathematic*, 25, Part I: 45-72, Part II: 119-134, Part III: 447-464.

[13] J.F. Sowa (1984). *Conceptual Structures - Information Processing in Mind and Machine.* Addison-Wesley Publishing Company.

[14] J.F. Sowa (1991). Towards the expressive power of natural language. In J.F. Sowa (ed.), *Principles of Semantic Networks - Explorations in the Representation of Knowledge.* Morgan Kaufmann Publishers, pp. 157-189.

[15] J.F. Sowa (1997). Matching logical structure to linguistic structure. In N. Houser, D.D. Roberts and J. Van Evra (eds), *Studies in the Logic of Charles Sanders Peirce.* Indiana University Press, pp. 418-444.

[16] V. Wuwongse and M. Manzano (1993). Fuzzy conceptual graphs. In G.W. Mineau, B. Moulin and J.F. Sowa (eds), *Conceptual Graph for Knowledge Representation*, LNAI 699, Springer-Verlag, pp. 430-449.

[17] L.A. Zadeh (1965). Fuzzy sets. *Journal of Information and Control*, 8, 338-353.

[18] L.A. Zadeh (1975). Fuzzy logic and approximate reasoning. *Synthese*, 30, 407-428.

[19] L.A. Zadeh (1978). PRUF - A meaning representation language for natural languages. *International Journal of Man-Machine Studies*, 10, 395-460.

[20] L.A. Zadeh (1996). Fuzzy logic = Computing with words. *IEEE Transactions on Fuzzy Systems*, 4, 103-111.

A Reference Model for Intelligent Information Search

Ivan Ricarte and Fernando Gomide
State University of Campinas-FEEC-DCA
13083-970 Campinas, São Paulo, Brazil
ricarte,gomide@dca.fee.unicamp.br

Abstract

The paper aims a tutorial review of the current state of the art in the area of Web search to address information retrieval models and a reference model for intelligent information search. We first review current information Web search models and methods, followed by contributions brought by machine learning, artificial and computational intelligence. As a result, a reference model is sketched. Its purpose is to summarize the main relationships between computational intelligence and information search systems as a means to promote innovative, intelligent information search systems development.

1. Introduction

A decade ago, computer technology was evolving towards cheaper and faster hardware with software components breaking the limits of the feasible with sophisticated interfaces, data and knowledge bases, and information processing systems and engines. In the meantime, the phenomenon of the World Wide Web has surprised most of the technical and social world, revolutionizing the way people access information. However, the Web brings difficulties to classical information processing and retrieval methodologies. Typically, classical information systems are designed for structured environments and have been used to index static collections of centralized, directly accessible documents. On the contrary, the Web is a distributed, dynamic, open, and rapidly growing information resource.

There are two major ways to search for documents in the Web. One way is to use a Web agent, software programs that receive a user query and systematically explore the Web to locate documents, evaluate their relevance, and return a rank-ordered list of documents. Currently, this approach is impractical due to the considerable dimension of the Web space. An alternative, the one currently adopted by most if not all search tools available, is to search a precompiled index built and periodically updated by Web traversing agents. The index is a searchable database that gives reference pointers to Web documents. Clearly, we note two different, albeit related flows of search tasks: the first to mediate the user with the database during information retrieval, and the second to interface Web traversing agents with the database to store and update documents.

The need to make sense of the swelling mass of data and misinformation that fills the Web brings a crucial information search problem. The search engine, along with browsers, domain name servers, and hypertext markup language, becomes an essential ingredient to turn the Web useful. The ability to search and retrieve information from the Web efficiently and effectively is a key technology for realizing its full potential. Current search methods and tools retrieve too many documents, of which only a small fraction is relevant to the user query. Furthermore, the most relevant documents do not necessarily appear at the top of the query output order. The design and development of current-generation search methods and tools have focused on query-processing speed and database size. Modern search engines have applied techniques from the past forty years to the Web-based text. The focus should instead shift to provide a short, ranked list of meaningful documents. That is, search should shift from lexicographical spaces to conceptual spaces of documents because this is where sense may be found in a compact and meaningful form from the point of view of the user query.

Considerable research effort is being developed to fill the need for improved search engines. Improved search requires more effective information retrieval. Ideally, effective information retrieval means high recall and high precision, but in practice it means acceptable compromises. Recall and precision strongly depends on the indexing, query, matching, ranking and feedback techniques used by the information system. Many approaches have been suggested to improve search. Among them we note filtering and routing techniques [1], context and page analysis [2], relevance feedback [3], semantic similarity [4], and

case-based reasoning. However, these techniques address only part of the problem because they are imbedded within classical information models. Efforts should instead focus on the information retrieval model itself, especially its constituents and computational paradigms that support them.

The aim of this paper is twofold. The first is to provide a tutorial review of the current state of the art in the area of Web search. The second is to address information retrieval models that induce a reference model to develop intelligent search techniques. For this purpose, we first review current information Web search models and methods from the point of view of information retrieval systems. Next we discuss the potential contributions that machine learning, artificial and computational intelligence brought to improve information retrieval models to enhance information search effectiveness and to develop intelligent information search. As a result of the review and discussions above a reference model is sketched. Its purpose is to summarize the main relationships between computational intelligence and information search systems as a means to promote innovative, intelligent information search systems development. Remarks concerning relevant, related issues not emphasized here and suggestions for further developments conclude the paper.

2. Information Search in the Internet

The Web, the most extensive and popular hypertext system in use today, is a large distributed dynamical digital information space built on top of the Internet. Hypertext is fundamentally an information storage and retrieval system that provides nonsequential method to access information. Its essential features are nodes and hyperlinks. In general, nodes may contain documents, graphics, audio, video, and other media. For simplicity, here we assume nodes with documents only.

Four main modules characterize information retrieval systems models: query representation, matching strategies, ranking methods and query output, complemented by the user interface, document indexing and database support.

Information retrieval can be broadly grouped into four main classes: set-theoretic, algebraic, probabilistic, and hybrid models [1], depending on the method selected for each of the four modules. In particular, set theoretic models include fuzzy-set based models to generalize classical set operations in queries, document representation, and ranking [5].

Currently, most Web search schemes function similarly. A document-collecting program, an agent, explores the hyperlinked documents of the Web looking for pages to index. These pages are stored in a database or repository. Finally, a retrieval program takes a user query and creates a list of links to Web documents that matches the query. These schemes completely hide the organization and content of the index from the user. There are other schemes that feature hierarchically organized subject catalog or directory of the Web, which is visible to the users as they browse and query [1,6]. They may or may not use agents to update their databases. Therefore, Web search systems are built upon extensions of the information system model. If we assume the Web itself as an information source and add a search agent module to traverse the information source and catch documents, extract the information required to represent and to store them in a database, then a Web search system model emerges, figure 1.

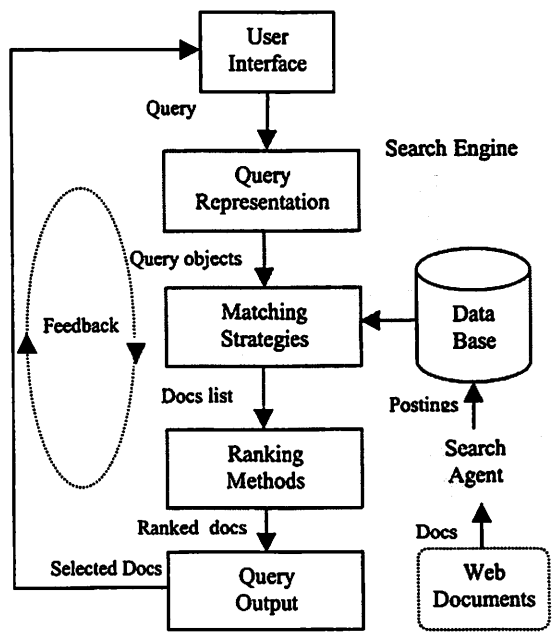


Figure 1: Information retrieval system model.

Traditional Web search engines are based on users typing in keywords or expressions to query for the information they want to receive. These are translated

into query objects whose form depends on the document representation method used. Query objects are matched against document representation to check their similarity or adjacency of their postings. Postings are tuples consisting of a term, a document identifier, and the weight of that term in that document. Postings are sometimes called inverted files and are the result of indexing. Indexing is the process of developing a document representation by assigning content descriptors or term to the document. The matching documents are ranked according to a relevance measure. Relevance measure considers the number of query terms contained in the document, the frequency of the query terms used in the documents, proximity of the query terms to each other in the document. The position where the terms occur in the document and the degree to which query terms match individual words is also considered in most systems.

All search agents use essentially the same scheme to traverse the Web to fetch pages, to extract representation information and to store this information in a database. In practice the agents use a graph search strategy that embeds page fetch, document descriptor extraction and encoding according to an information retrieval model, and an appropriate database storage procedure. They create a queue of pages to be explored, with at least one Web page in the queue, and choose a page from the queue to explore. Next, they fetch the page chosen and extract all the links to other pages and add any unexplored page links to the queue. The page fetched is processed to extract representation parameters such as title, headers, key words or other representation information and store in a database. Next, a new page is selected from the top of the queue. Search algorithms differs only in the way new elements are added to the queue, with depth-first, breadth-first and best-first being among the most used.

Efficient search in information retrieval is not a bottleneck, but retrieval effectiveness is a different matter. Two main parameters, recall and precision, respectively, can explain retrieval effectiveness. Ideally, we would like to achieve both high recall and high precision. In practice, we must accept a compromise. Indexing and query terms that are too specific yields higher precision at the expense of recall. Indexing and query terms that are too broad yields higher recall at the expense of precision [1]. Current search tools retrieve too many documents, of which only a small fraction are relevant to the user query. This is a result of two main issues. The first refers to the limitations of the information retrieval and search models in use today. The second concerns the lexicographical view of the models, which uses the word structure of documents content only. Enhanced models should strive to include conceptual views of documents to enhance search effectiveness.

3. Intelligent Information Search

Information retrieval using probabilistic methods has produced a substantial number of results over the past decades. In the eighties, artificial intelligence and fuzzy set theory also made an impressive contribution to intelligent information retrieval and indexing. More recently, attention has shifted to inductive learning techniques including genetic algorithms, symbolic learning and neural networks. Most information retrieval systems still rely on conventional inverted index and query techniques, but a number of experimental systems and computational intelligence paradigms are being developed.

Knowledge-based information retrieval systems attempt to capture information specialists' domain knowledge, search strategies and query refinement heuristics. Most of these systems have been developed based on the manual knowledge acquisition process, but data mining and knowledge discovering techniques indicate a major source for automatic knowledge elicitation. Computer-assisted systems have shown to be more useful and several systems of this type have been developed over the past decade [7]. Many of them embody forms of semantic networks to represent domain knowledge, accept natural languages queries, and include a knowledge base of search strategies and term classification similar to a thesaurus.

Systems and methods of information retrieval that are based upon the theory of fuzzy sets have been recognized as more realistic than the classical methods [8]. Extensions of Boolean models to fuzzy set models redefine logical operators appropriately to include partial membership and process user query in a manner similar to the Boolean model. Fuzzy models represent document characteristics and user queries can be in natural language propositions. Similarly to classical knowledge-based systems, various forms of thesaurus have been added to improve retrieval performance. In general, fuzzy information retrieval systems consider each piece of knowledge as a pattern represented as a set of attributes, objects and values. The values of the attributes can be either quantitative or qualitative and represented by possibility distributions in the attribute

domain. Fuzzy connectives allow the user to elaborate complex queries. Each query is fuzzy-matched with the characteristic patterns of each document in the database. Matching degrees provides information to rank document relevance and query output.

Unlike the manual knowledge acquisition process and the linguistic-based natural language processing technique used in knowledge-based systems design, learning systems rely on algorithms to extract knowledge or to identify patterns in examples of data. Various statistics-based algorithms have been used extensively for quantitative data analysis. The computational intelligence techniques, the neural network approach, evolution-based genetic algorithms and fuzzy set theory provide quite different schemes for data analysis and knowledge discovery.

Neural networks seem to fit well with conventional models such as vector space model and the probabilistic model. Assuming a broader view of connectionist models, vector space model, cosine measures of similarity, and automatic clustering and thesaurus can be combined into a network representation. Neural nets have been used for document clustering and to develop interconnected, weight/labeled networks of keyword for concept-based information retrieval [7]. Generally speaking, neural networks provide a convenient knowledge representation for information retrieval applications in which nodes typically represent objects such as keywords, authors, and citations and their weighted associations of relevance.

The self-adaptiveness property of genetic algorithms is also very appealing for information retrieval systems. For instance, genetic algorithms find the keywords that best describe a set of user provided documents [7]. In this case, a keyword represents a gene, a user-selected document represents a chromosome, and a set of user-selected documents represents the initial population. The fitness of each document is based on its relevance to the documents in the user-selected set measured by the Jaccard's score. Genetic algorithms have also been used to extract keywords and to tune keywords weights. Genetic algorithms and genetic fuzzy systems have been ones of the most fertile computational intelligence tools to Web search engines. This is to be contrasted with the knowledge and neural network-based approaches, which has shown modest presence in the Web world. Genetic algorithm can dynamically take a user's selected starting homepages and search for the most closely related homepages in the Web based on the links and keyword indexing. A more advanced system uses a genetic information retrieval agent filter approach for documents from the Internet using a genetic algorithm with fuzzy sets genes to learn the user's information needs [9]. A similar, collaborative approach to develop for personalized intelligent assistant uses a metagenetic algorithm to evolve populations of keywords and logic operators. A primary genetic algorithm creates a population of sets of unique keywords selected at random from a dictionary. A secondary genetic algorithm then creates a population of sets of logic operators for each of the primary genetic algorithm members.

4. The Reference Model

In this section we propose a model that incorporates the generic aspects of information retrieval systems, along with hooks in the model to incorporate specific techniques that are used in the information retrieval process.

The model is a set of interrelated abstract classes —or a framework— along with method specifications that constitute brainware insertion points. In actual implementations, concrete classes are derived from these abstract classes, defining methods that implement a specific technique. The model focus on what has to be done rather than on how it is done. It establishes in which methods each aspect is contemplated and how the implemented classes should collaborate to perform the application tasks.

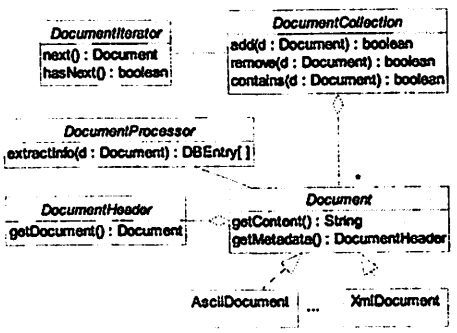


Figure 2: Document components.

The basic element in the framework is the Document class, figure 2. From any Document object we obtain the document content, via getContent() method, as well

as information about it, via getMetadata(). Representation of what constitutes document metadata is abstracted introducing the DocumentHeader class. Instances of classes derived from Document include AsciiDocument, HtmlDocument, WordDocument, and XmlDocument, each of which with its specific way to express and obtain the document header and content.

In an information retrieval application, documents occur in collections: restricted (a closed world or limited collection of documents, as in a traditional information retrieval) or unrestricted (an open world domain of documents, as in the Web). These collections are abstracted by the DocumentCollection class. The elements in a object of this class can be accessed using a DocumentIterator, which provides methods such as next(), returning a Document, and hasNext(), returning a Boolean value. Implementation examples of next() include search strategies such as depth first, breadth-first or any appropriate heuristic search scheme, including the fuzzy search methods.

The document returned by next() is processed to get the postings, the information to be stored in the application database. In this model, the structure of the information to be stored in the database is represented by the DBEntry class. Classes associated with concretizations of DBEntry shall be aware of specific structures used in underlying databases, e.g. a tuple for a relational database or an object for an object-oriented database system.

DBEntry objects for a given document are created by the method extractInfo() from the DocumentProcessor class. This method encapsulates keywords extraction, associated concepts taken from a thesaurus, semantic networks or frames, and clustering strategies used to classify documents. Clustering strategies could be as simple as "extract most frequent words" or as elaborate as the fuzzy set-based or neural networks approaches described earlier. This method output could adopt an appropriate automatic index method such as single or multi-term phrase indexing, using probabilistic, statistical or linguistic methods.

Operations to access the underlying application databases are generalized through the methods specified in the class DatabaseAccessor, figure 3. Along with DBEntry, this adapter class abstracts which type of database management system (relational, object-oriented, hierarchical) is being used in the application. The application could actually use multiple and heterogeneous databases.

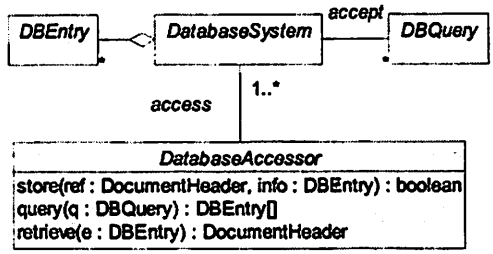


Figure 3: Database components.

The model also contemplates the many possibilities that users have to express their queries, as well as the alternatives for query processing. The user entry is a string, but this does not necessarily imply that the user interface has to be textual. This string might represent selections from forms, voice interfaces, or virtual reality environment. The user entry should be parsed, processed, and transformed into an internal representation format. This format is an object of the UserQuery class, figure 4, created by the factory method createUserQuery(). Implementations of this method in derived classes might incorporate strategies such as Boolean expressions, natural language processing, and precisiated natural languages propositions [10]. New UserQuery objects can be created from an existing object of this class by aggregating feedback, which is contemplated by the method introduceFeedback().

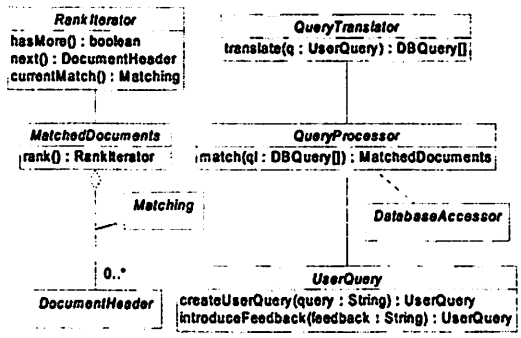


Figure 4: Retrieval components.

The UserQuery is translated into one or more DBQuery objects by the QueryTranslator class. The application databases can then be inquired using the methods query() and retrieve() of DatabaseAccessor

class. The mediation between the UserQuery and postings formats is performed by the QueryProcessor class, whose method match() returns a MatchedDocuments object, a collection of document headers matching the user request along with its matching degree. This is expressed in the model as an object of the Matching class; there is no need to restrict this measure to scalar values. Implementations of match() could consider techniques such as cosine, possibility measures or matching operators of fuzzy set theory, for example.

Note that an object of the MatchedDocuments class is a collection and, as such, can benefit from the already mentioned Iterator pattern. In this case, the underlying sequence of access to the list of documents is ordered by a ranking criterion. The method rank() returns a RankIterator object, and currentMatch() gets the matching degree for the document returned by next().

Learning methods, especially participatory learning paradigm [11], emerges from two main sides. The first concerns the need to adapt document representation index and classification from a set of concepts and keywords, and vice-versa. The second concerns elaboration of the query objects by algorithms such as genetic or alternative adaptive approaches. These could be encapsulated in the extractInfo() and createUserQuery(), respectively.

5. Conclusions

The state of the art of information retrieval and Web search systems has suggested the reference model sketched in this paper. The purpose of the model was to introduce an object-oriented framework to establish the relationships between computational intelligence and information search systems as a means to promote innovative, intelligent systems development. Future developments shall address context in Web search, the issue of hyperlink structures in system complexity and the role of the Semantic Web. Hopefully the model proposed here could grow and contribute for developing and implementing future generation systems.

6. References

[1] V. Gudivada, V. Ragahavan, W. Gorski, and R. Kasanagottu, "Information Retrieval on the World Wide Web". *IEEE Internet Computing*, Vol. 1, No. 5, September/October 1997, pp. 58-68.

[2] S. Lawrence, and C. Giles, "Context and Page Analysis for Improved Web Search". *IEEE Internet Computing*, Vol. 2, No. 4, July/August 1998, pp. 38-46.

[3] C. Chang, and C. Hsu, "Enabling Concept-Based Relevance Feedback for Information Retrieval on the WWW". *IEEE Trans. on Data and Knowledge Engineering*, Vol. 11, No. 4, July/August 1999, pp. 595-609.

[4] S. Green, "Building Hypertext Links by Computing Semantic Similarity". *IEEE Trans. on Data and Knowledge Engineering*, Vol. 11, No. 5, September/October 1999, pp. 713-730.

[5] T. Radecki, "Fuzzy Set Theoretical Approach to Document Retrieval". *Information Processing and Management*, Vol. 15, 1979, pp. 247-259.

[6] M. Mauldin, "Lycos: Design Choices in an Internet Search Service". *IEEE Expert: Intelligent Systems and Applications*, Vol. 12, No.1, January/February 1997, pp. 8-11.

[7] H. Chen, "Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms". *Journal of the American Society for Information Science*, Vol. 43, No. 3, 1995, pp. 194-216.

[8] S. Miyamoto. *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Kluwer, Boston, 1990.

[9] M. Matin-Bautista, and M. Vila, "A Fuzzy Genetic Algorithm Approach to an Adaptive Information Retrieval Agent", *Journal of the American Society for Information Science*, Vol. 50, No. 9, 1999, pp. 760-771.

[10] L. Zadeh. "A New Direction in AI". *AI Magazine*, Spring 2001, pp. 73-84.

[11] R. Yager, "A Model of Participatory Learning", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 5, 1990, pp. 1229-1234.

Content Based Vector Coder for Information Retrieval

Shuyu Yang and Sunanda Mitra
Department of Electrical and Computer Engineering , Texas Tech University
Lubbock,TX79409-3102
Shu.yang@ttu.edu, Sunanda.Mitra@coe.ttu.edu

Abstract
Retrieval of relevant information and its efficient transmission over the Internet to worldwide users are of utmost interest in many applications. Content-based source encoding is, therefore, essential in enhancing information retrieval. We present a novel multi-scale source vector encoder whose codebook is generated by neuro-fuzzy clustering of salient information features in the wavelet domain. Our results demonstrate that the performance of such encoder is equivalent to an optimized statistical approach while, providing a drastic reduction in execution time. The performance of the new vector encoder surpasses that of the well-known scalar coder, the Set Partitioning in Hierarchical Trees (SPIHT) in the fidelity of reconstructed data, at low bit rates and comparable at high bit rates as exemplified by a set of magnetic resonance (MR) brain images.

1. Introduction

Vector quantization (VQ) has been theoretically proved to be a more efficient coding method than scalar quantization [1]. The design of an efficient VQ encoder involves global codebook generation by selecting a good clustering algorithm and using appropriate features extracted from the training data set. In image vector coding, codebooks obtained from a set of training images tend to retain common features of the samples, thus, when used to decode a particular image, special features of the image will be lost, which usually leads to a blurred reconstructed image. In this paper, we are presenting a content-based vector coder whose codebooks are generated using a neuro-fuzzy clustering algorithm, adaptive fuzzy leader clustering (AFLC) [2] together with a new feature extraction method. Usually, the fidelity of a reconstructed compressed image depends on the nature of the codebook used for decoding. Thus, it is highly desirable to choose an efficient clustering algorithm for codebook generation. At the same time, selecting appropriate features from a set of training images is equally critical.

Through multi-level wavelet transform, an image can be decorrelated in both space and frequency, resulting in a hierarchical structure in which most of the image information being packed into a few large-magnitude coefficients at higher levels, while a large population of the coefficients concentrates around zero [3], leaving much room for compression. Successful wavelet based scalar quantization methods such the Embedded Zerotree Wavelet (EZW) [4] and SPIHT [5] exploit this dependency between inter-scale coefficients by retaining and quantizing coefficients according to their significance.

Although SPIHT has been regarded as one of the best scalar quantization schemes, the intra-scale dependency among wavelet coefficients are not explicitly exploited. On the other hand, traditional wavelet-based VQ schemes tend to take advantage of the intra-scale dependency by grouping adjacent coefficients in the same level into vectors [3, 6], without taking the relationship among inter-scale coefficients into consideration. Here we will present a new vector forming method that exploits both dependencies.

Besides generating wavelet feature vectors in a new way, we compare two efficient clustering algorithms that can be used for codebook training. The AFLC algorithm, based on neuro-fuzzy concepts, has been proven to be quite successful in many clustering tasks such as image vector quantization, image segmentation and noise removal [6, 7]. Deterministic annealing (DA) [8] is entirely based on a statistical approach, whose performance yielding global optimum. We compare the clustering accuracies of both methods in terms of the distortion of reconstructed image from codebooks generated from these two methods respectively.

After the image is vector quantized, the error residual, which contains image-specific information rather than the common image features found in the codebook, is scalar quantized using SPIHT. Such a combination allows the detail contents of the image to be preserved, rendering a higher fidelity reconstructed image than coded otherwise.

In section 2, the AFLC and DA algorithms will be briefly described. Our new feature extraction method will be explained in section 3. The coding method and

results are presented in section 4. Section 5 presents the conclusions.

2. AFLC and DA

2.1 AFLC

AFLC is an integrated neural-fuzzy clustering algorithm that can be used to learn cluster structure embedded in complex data sets in a self-organizing manner. The algorithm can be described as a two-layered structure (Figure 1). The first layer uses a self-organizing neural network similar to ART1 [9-11] to find hard clusters. Let C be the current number of centroids and $\mathbf{v}_i$ (i=1, ..., C) be the centroids. When a new sample $\mathbf{x}_k$ comes in, it is normalized and then initially classified into the cluster whose centroid has the largest dot product with the sample vector:

$$y_{i^*} = \max_i\{y_i\} = \max_i\left\{\sum_{j=1}^{n} b_{ij}\bar{x}_{kj}\right\}, \quad i = 1,...,C,$$

$$b_{ij} = \frac{v_{ij}}{\|\mathbf{v}_i\|}, \quad \bar{x}_{kj} = \frac{x_{kj}}{\|\mathbf{x}_k\|},$$

and i^* is the index of the winning cluster.

This initial classification is verified through a vigilance test in the second layer, the recognition layer. When the test fails, a new cluster is created, otherwise the system is optimized and clusters are updated. The vigilance test consists of calculating a ratio between the distance of the sample to the winning cluster and the average distance of all the samples in this cluster to the cluster centroid,

$$R = \frac{\|\mathbf{x}_j - \mathbf{v}_i\|}{\frac{1}{N_i}\sum_{k=1}^{N_i}\|\mathbf{x}_k - \mathbf{v}_i\|}.$$

When this ratio is higher than a user-defined threshold, the test fails and a new cluster is created, taking the sample as the initial centroid. Otherwise, the sample is officially classified into the winning cluster, and then its centroid and the fuzzy membership values are updated using fuzzy C-means [12] (FCM) equations:

$$\mathbf{v}_i = \frac{1}{\sum_{j=1}^{p}(u_{ij})^m}\sum_{j=1}^{p}(u_{ij})^m\mathbf{x}_j, \qquad i = 1,2,...,c$$

$$u_{ij} = \frac{(1/\|\mathbf{x}_j - \mathbf{v}_i\|^2)^{1/m-1}}{\sum_{k=1}^{c}(1/\|\mathbf{x}_j - \mathbf{v}_k\|^2)^{1/m-1}}, \qquad i = 1,2,...,c, j = 1,2,...,p$$

The number of clusters finally generated depends on the user-specified threshold.

2.2 The deterministic annealing (DA) algorithm

Deterministic annealing is an optimization algorithm based on principles of information theory and statistical mechanics. Specifically, it minimizes the

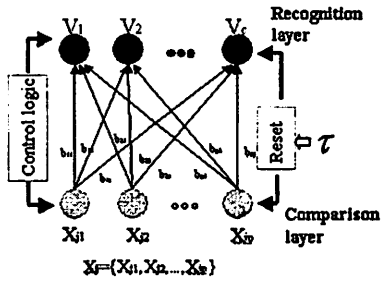


Figure 1: The AFLC algorithm structure

expected distortion of a given system while maximizing its randomness (Shannon's entropy). Let D be the average distortion,

$$D = \sum_x\sum_y p(x,y)d(x,y),$$

where p(x,y) is the joint probability distribution of input x and codeword y, p(y|x) is the association probability that relates x to y, d(x,y) is the distortion measure. Shannon's entropy of the system is given by

$$H(X,Y) = -\sum_x\sum_y p(x,y)\log p(x,y).$$

The optimization is achieved by minimizing the Lagrangian F=D-TH, where T is the Lagrange multiplier.

When used for image vector quantization, mass-constrained DA is preferred, where the constraint $\sum_i p(y_i) = 1$ is added into the Lagrangian, resulting in the optimization of F' with respect to the centroids y_i and the encoding rule $p(y_i|x)$:

$$F' = D - T[H + \sum_i p(y_i) - 1].$$

The encoding rule and the centroids are thus given by:

$$y_i = \frac{\sum_x xp(x)p(y_i|x)}{p(y_i)},$$

where

$$p(y_i|x) = \frac{p(y_i)e^{-(x-y_i)^2/T}}{\sum_{j=1}^{K}p(y_j)e^{-(x-y_j)^2/T}},$$

$$p(y_i) = \sum_x p(x)p(y_i|x).$$

When the variance of a cluster reaches certain limiting value ruled by the covariance matrix of the cluster, splitting of the cluster happens. The Lagrangian parameter at which a splitting occurs is called critical temperature T_c, from the terminology of statistical physics. It is calculated by $T_c=2\lambda_{max}$, where λ_{max} is the maximum eigenvalue of the covariance matrix $C_{x|y}$ of the posterior distribution p(x|y) of the cluster corresponding to codeword y:

$$C_{x|y} = \sum_x p(x \mid y)(x-y)(x-y)' .$$

As the Lagrange multiplier decreases, the number of clusters increases. Implementation of DA can be found in its original paper in [8].

2.3 The Performances of AFLC and DA

When considering the use of AFLC or DA for codebook training, a comparative evaluation of the two is necessary. However, unlike DA, which is derived strictly from a statistical framework, a mathematical analysis of AFLC is difficult because of its complicated structure. Therefore, to get around the problem, it would be appropriate to evaluate their performances based on the reconstructed image quality (relating to the clustering accuracy) and the time it takes for both to generate the same amount of clusters under the same condition.

Our experiment consists of using 11154 samples extracted from a set of 26 MR images (which will be described in detail in section 4). Figure 2 and Figure 3 shows the performances of both algorithms in terms of reconstructed image PSNR (VQ only) and codebook generation time. It is noteworthy that the time is plotted in logarithmic scale so that the large difference of the two can be illustrated. We can clearly see that the reconstructed image quality is comparable while the speed is drastically different. For example, for the MR images used, DA takes 156 hours as oppose to 2.97 by AFLC to generate around 800 codewords.

Because of its statistical computational complexity involving updating the association probabilities and centroids, the clustering speed of DA slows down considerably when the population and dimension of samples as well as the number of existing clusters increase.

AFLC provides a good compromise between clustering accuracy and speed. However, what plagues AFLC is just the same problem that troubles algorithms with ART-1 structure—the sample-order-dependency. Just as the first sample is considered the first initial centroid, the centroids of the following new clusters created in the recognition layer of AFLC are initialized with the sample that does not fit in with any existing clusters. In other words, the algorithm "sees"

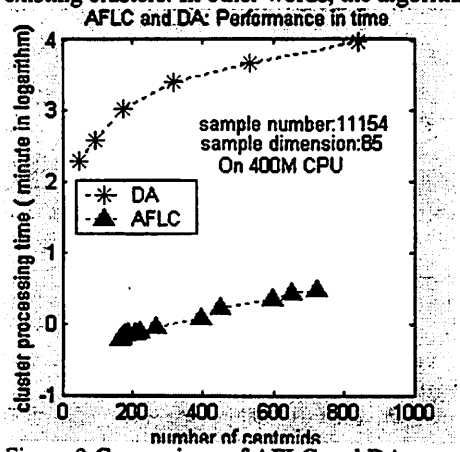


Figure 2 Comparison of AFLC and DA in clustering speed

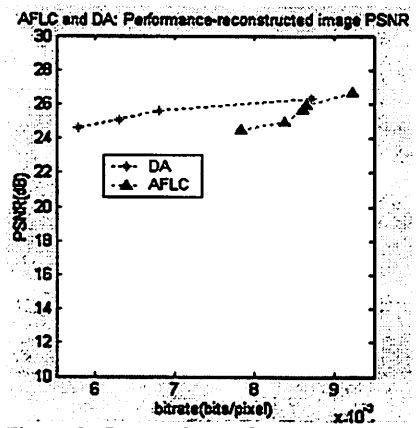


Figure 3. Comparison of AFLC and DA in reconstructed image quality

and only considers the samples in the order they are presented.

To eliminate the sample-order-dependency, proper initialization of the centroids is essential. Instead of assigning randomly incoming samples as initial centroids, the algorithm can be forced to "see" the entire sample population when a decision is made to increase the number of clusters and assign a new centroid. An algorithm aiming to improve AFLC using such a method is under development in our laboratory.

3. Feature Extraction

Vectors formed in the spatial domain by partitioning the spatial image into non-overlapping blocks have a random distribution that cannot be embedded into the optimization of the clustering algorithm. For the DA clustering used above, we approximate the distribution function of the training sample vectors with uniform

distribution, which is not necessarily true for any type of images. Since vector samples with a general distribution is always preferred over undefined distributions, it is of interest to look for a way to extract feature vectors from the image so that a generalized distribution can be applied.

The study of wavelet transformed coefficients of images show that wavelet coefficients do have such desirable property. In [3], it is shown that the wavelet coefficients can be fitted to a generalized Gaussian distribution. Further more, for multi-level wavelet transformed coefficients, it can be observed that if a wavelet coefficient at certain scale is insignificant with respect to a given threshold, then all the coefficients of the same orientation in the same spatial location at finer scales are insignificant with a high probability. This hypothesis has been successfully applied to code wavelet coefficients at successive decomposition levels using coefficient's significance maps [4, 5]. Although such zerotree based scalar coding takes advantage of the redundancy among different scales, redundancy inside the scales are not exploited.

In contrast to scalar coding of wavelet coefficients, intra-scale redundancy can be easily exploited by VQ, since VQ removes redundancy inside the vector. One of the most popular but a simple way of achieving this is by blocking individual wavelet coefficients inside one scale as feature vectors, as is done in [3].

Our new way of forming sample vectors takes both dependencies into consideration. Vectors are formed by stacking blocks of wavelet coefficients at different scales at the same orientation location. Since the scale size decreases as the decomposition level goes up, block size at lower level is twice the size of that of its adjacent higher level. The same procedure is used to extract feature vectors for all three orientations. This is illustrated in Figure 4 for a three level wavelet transform. The dimension of the vector is fixed once the decomposition level is fixed.

Since wavelet coefficients obey generalized Gaussian distribution, the vectors formed in this way can be approximated by a multi-dimensional generalized Gaussian distribution.

4. Results

The training data we used is a group of slices (slice 1 to slice 31) from a 3D simulated MR image of a human brain [13]. Thus, the training images are reasonably different because of the span from top of the brain to the lower part of the brain despite belonging to the same class. Figure 5 shows some of the images from the training set. A few slices inside the group, for example, slice 6 and slice 12, etc are randomly chosen and excluded from the training set and later used as test images.

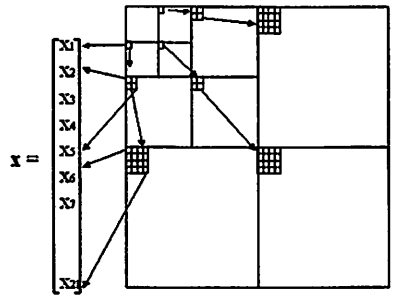


Figure 4: Vector extraction from multi-level wavelet coefficients

After the codebooks are generated, VQ is performed on the test image (slice 6, shown in Figure 7 (a), is used here as an example). The residual obtained by subtracting the vector-quantized coefficients from the original wavelet is then scalar-quantized with SPIHT. Figure 6 compares the reconstructed image PSNR of our content-based coder and that of using SPIHT alone. Our coder surpasses SPIHT in all bit rates but by far at low bit rates. Figure 7 shows the reconstructed images at various bit rates. It is to be noted that at low bit rates the SPIHT reconstructed images appeared to be much blurrier than the content-based encoder even though PSNRs are comparable.

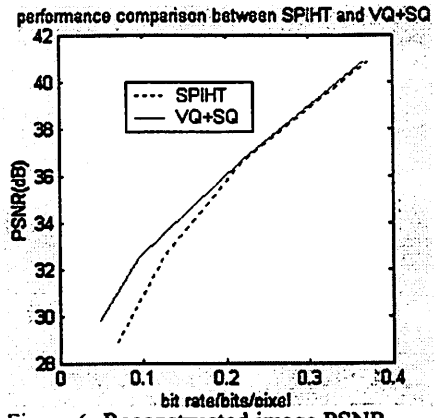


Figure 6. Reconstructed image PSNR for VQ+SQ and SPIHT

5. Conclusion

We have presented a content-based vector coder combining vector and scalar quantization to preserve important image features in the wavelet domain. We also demonstrate that this coder yields much better performance than the state-of-the-art scalar quantization scheme, namely, SPIHT, at low bit rates.

The training set we used for this paper is an entire set of simulated brain MR image slices that spans all the features of the human brain. Our results demonstrate that even at very low bit rates, the content-based coder can preserve more details of the image than SPIHT (refer to Figure 7 (c)-(f) and (d)-(g)). This shows a promising application of this encoder to medical image compression in which a codebook can be generated for specific types of medical images (for example, X-ray images).

We also would like to point out that the codebook we use now is a non-structured codebook. Based on the distribution of the feature vectors, we can impose certain structure on the codebook to make the coding more efficient. In addition, a better clustering algorithm that eliminates the sample-order dependency of AFLC can be used for codebook generation. These issues are being investigated currently.

6. References

[1] T. Berger, Rate Distortion Theory, Englewood Cliffs, NJ: Prentice-Hall, 1971.

[2] S. C. Newton, S. Pemmaraju, and S. Mitra, "Adaptive fuzzy leader clustering of complex data sets in pattern recognition," *IEEE Trans. Neural Networks* vol 3, pp.794-800, 1992.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. on Image Processing, vol. 1, No. 2, April 1992.

[4] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Transactions on Signal Processing, Vol.41, No.12, Dec, 1993.

[5] A. Said and W.A Pearlman, "A new,fast and effcient image codec based on set partitioning in hierarchical trees," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 3, June,1996.

[6] S. Mitra, and S.Y. Yang, "High Fidelity Adaptive Vector Quantization at Very Low Bit Rates for Progressive Transmission of Radiographic Images", Journal of Electronic Imaging, Vol. 11, No. 4, Suppl. 2, pp. 24-30, November 1998.

[7] R. Castellanos, H. Castillo, and S. Mitra, "Performance of nonlinear methods in medical image restoration," SPIE Proceedings on Nonlinear Image Processing, Vol. 3646, 1999.

[8] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," Proc. of IEEE, 86, 1998.

[9]. G. A. Carpenter, and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," Computer Vision, Graphics and Image Processing, Vol. 37, pp. 54-115, 1987.

[10] G. Carpenter and S. Grossberg, "Art-2: Self organization of stable category recognition codes for analog input patterns," Appl. Opt., Vol. 26, pp. 4919-4930, 1987.

[11] G. Carpenter and S. Grossberg, "Art-3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," Neural Networks, Vol. 3, pp. 129-152, 1990.

[12] Bezdek J., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, NY, 1981.

[13] Montréal Neurological Institute, McGill University, "*BrainWeb:* Simulated Brain Database," http://www.bic.mni.mcgill.ca/brainweb, May, 2001

Acknowledgments

This research has been partially supported by funds from the Advanced Research Program (ARP) (Grant No. 003644-176-ARP), the Advanced Technology Program (ATP) (Grant # 003644-0280-ATP) of the state of Texas, and the NSF grant EIA-9980296.

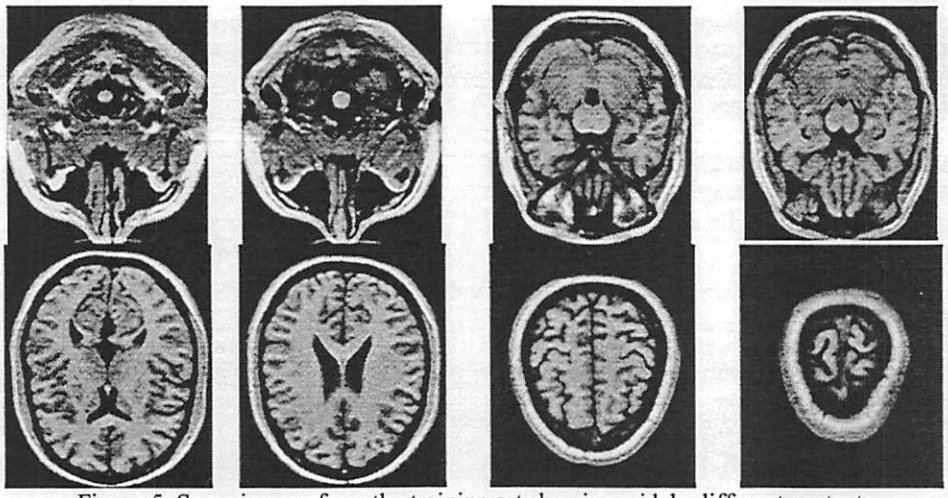
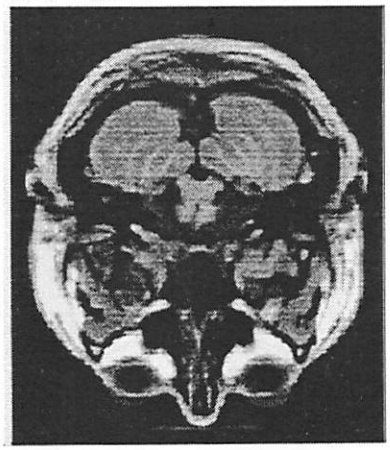
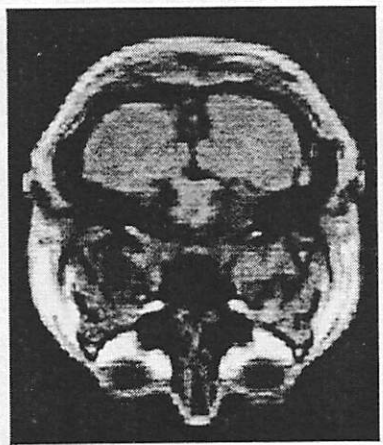


Figure 5. Some images from the training set showing widely different contents

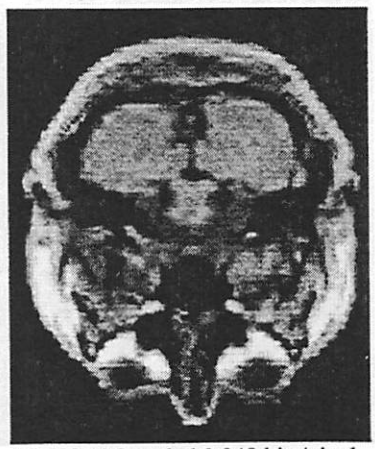


(a) Original test image (MRI slice 6)

(b) VQ+SQ coded 0.36bits/pixel
PSNR:40.87dB



(c) VQ+SQ coded 0.095 bits/pixel
PSNR:32.51 dB



(d) VQ+SQ coded 0.048 bits/pixel
PSNR: 29.81 dB

(e) SPIHT coded 0.37 bits/pixel
PSNR: 40.86 dB

(f) SPIHT coded 0.1266 bits/pixel
PSNR: 32.53 dB

(g) SPIHT coded 0.07 bits/pixel
PSNR: 28.87 dB

Figure 7 Reconstructed image co mparison: content-based coder and SPIHT

Presentations of Thursday, August 16

Recognition Technology, Data Mining, Summarization, Information Aggregation and Fusion

BERKELEY

The BISC Decision Support System

Masoud Nikravesh

BISC Program (Berkeley Initiative in Soft Computing)
Department of Electrical Engineering and Computer Sciences
University of California

Berkeley, CA 94720
Tel: (510) 643-4522 Fax: (510) 642-5775
Email: nikravesh@cs.berkeley.edu
http://www-bisc.cs.berkeley.edu

Decision Support System and Ranking

- use intelligently the vast amounts of important data in organizations in an optimum way as a decision support system
- share intelligently and securely company's data internally and with business partners and customers that can be process quickly by end users
- Decision support system is an approach or a philosophy which can be used for:
  - strategic planning such as resource allocation
  - management control such as efficient resources utilization
  - operational control for efficient and effective execution of specific tasks

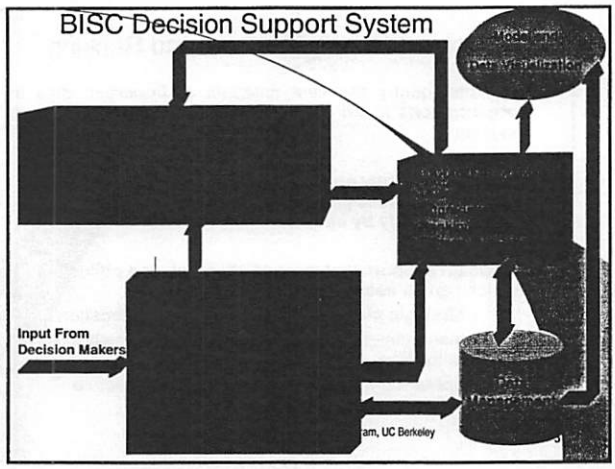
BISC Decision Support System

Objectives: Develop soft-computing-based techniques for decision analysis

- Tools to assist decision-makers in assessing the consequences of decision made in an environment of imprecision, uncertainty, and partial truth and providing a systematic risk analysis
- Tools to be used to assist decision-makers answer "What if Questions", examine numerous alternatives very quickly and find the value of the inputs to achieve a desired level of output
- Tools to be used with human interaction and feedback to achieve a capability to learn and adapt through time

BISC Decision Support System Components and Structure

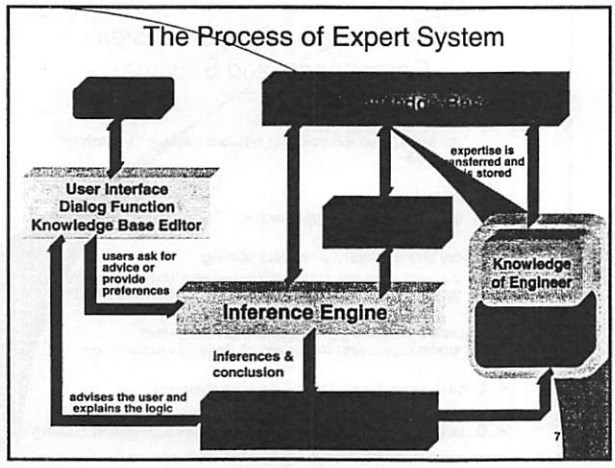
- Data Management
  - database(s) which contains relevant data for the decision process
- User Interface
  - users and DSS communication
- Model Management and Data Mining
  - includes software with quantitative and fuzzy models including aggregation process, query, ranking, and fitness evaluation
- Knowledge Management and Expert System
  - model representation including linguistic formulation
- Evolutionary Kernel and Learning Process
- Data Visualization and Visual Interactive Decision Making



Data Visualization and Visual Interactive Decision Making

Allows end-user or decision makers to recognize trends, patterns, and anomalies that can not be predicted or recognized by standard analysis methods through

- **Visual interactive modeling (VIM): user can intervene in the decision-making process and see the results of the intervention**
- **Visual interactive simulation (VIS): users may interact with the simulation and try different decision strategies**



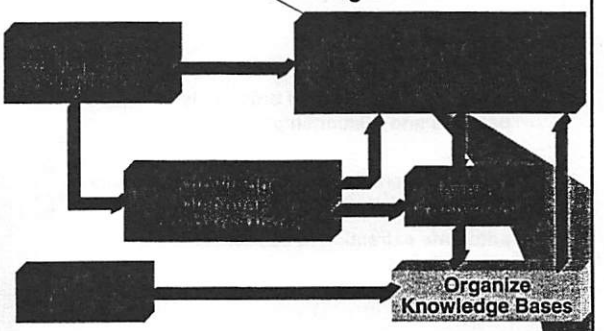
The Components of ES

- **the knowledge base contains engineering knowledge for model representation which provide problem solving environment**
- **the inference engine provide reasoning, conclusions, and recommendation**
- **the user interface and knowledge based editor provide dialog environment for questions and answers**
- **the advisor and translator can translate the machine inference to a human understandable advice, recommendation, and logical explanation**

Fuzzy Logic and Case-Based Reasoning (CBR)

- **Case-Based Reasoning (CBR)**
  - solve new problems based on history of g[illegible]en solved old problems
  - Provide a framework for knowledge acquisitio[illegible] information system development
  - enhance learning capability
  - generate explanations and recommendation to user[illegible]
- **Fuzzy Logic**
  - simulating the process of human reasoning
  - framework to computing with word and perception, and linguistics variables.
  - deals with uncertainties
  - creative decision-making process

The Data Management



Knowledge Management

- knowledge discovery and data mining- using search engines, databases, data mining, and online analytical processing, the proper knowledge must be found, analyzed, and put into proper context
- organize knowledge bases - it stores organizational knowledge and best practices
- knowledge acquisition - determines what knowledge (information) is critical to decision making
- knowledge representation - target audiences are defined and technologies are put into place to enable knowledge delivery when needed

Online Analytical Processing (OLAP)

- real time analysis of data by end-users
- data aggregation
- capture association
- capture relationships between many types of business elements
- data representation
- complex data manipulation

Data Mining for Decision Support

- automated discovery of previously unknown patterns and relationships
- automated prediction of trends and behaviors
- automate extraction of association rules among items

Case : Physical Stores or E-Store

A computer system that could instantly track sales and inventory at all of its stores and recognize the customer buying trends and provide suggestion

- regarding any item that may interest the customer
- to arrange the products
- on pricing, promotions, coupons, etc
- for advertising strategy

Case: Profitable Customers

A computer system that uses customer data that allows the company to recognize good and bad customer by the cost of doing business with them and the profits they return

- keep the good customers
- improve the bad customers or decide to drop them
- identify customers who spend money
- identify customers who are profitable
- compare the complex mix of marketing and servicing costs to access to new customers

Case : Internet-Based Advising

A computer system that uses the expert knowledge and the customer data (Internet brokers and full-service investment firms) to recognize the good and bad traders and provide intelligent recommendation to which stock buy or sell

- reduce the expert needs at service centers
- increase customer confidence
- ease-of-use
- Intelligent coaching on investing through the Internet
- allow customers access to information more intelligently

Case: Managing Global Business

A computer system responding to new customers and markets through integrated decision support activities globally using global enterprise data warehouse

- **information delivery in minutes**
- **lower inventories**
- **intelligent and faster inventory decisions in remote locations**

Case: Resource Allocator

A computer system that intelligently allocate resources given the degree of match between objectives and resources available

- **resource allocation in factories floor**
- **for human resource management**
  - **find resumes of applicants posted on the Web and sort them to match needed skill and can facilitate training and to manage fringe benefits programs**
  - **evaluate candidates**
  - **predict employee performance**

Intelligent Systems to Support Sales

- **A computer system that matching products and services to customers needs and interest based on case-based reasoning and decision support system to improve:**
- **sale**
- **advertising**

Case: Enterprise Decision Support

An interactive computer-based system that facilitates the solution of complex problems by a group of decision makers either by speeding up the process of the decision-making process and improving the quality of the resulting decisions through expert and user (company customer) collaboration and sharing the information, goals, and objectives.

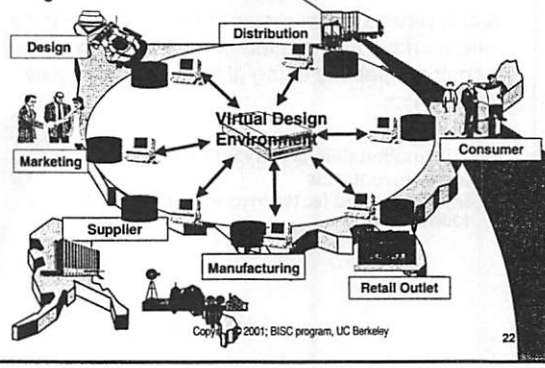
Case: Fraud Detection

An Intelligent Computer that can learn the user's behavior through in mining customer databases and predicting customer behaviours (normal and irregularities) can be used to uncover, reduce or prevent fraud

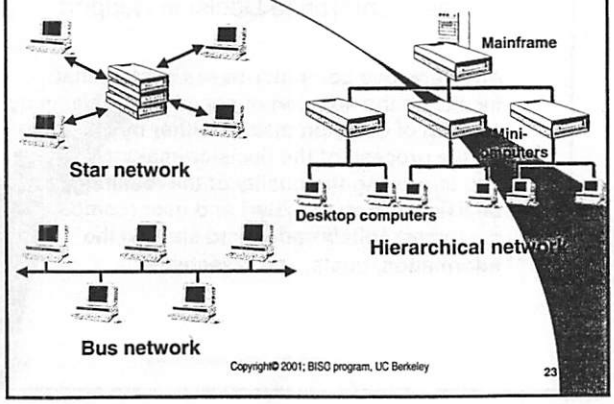
- in credit cards
- stocks
- financial markets
- telecommunication
- insurance

Supply-Chain Management (SCM)

Global optimization of design, manufacturing, supplier , distribution, planning decisions in a distributed environment



Network Typologies



Other Applications

Application	Description
Finance	• stock prices and characteristics, credit scoring, credit card ranking
Military	• battlefield simulation and decision making
Medicine	• diagnosis
Marketing	• store and product display • electronic shopping
Internet	• provide knowledge and advice to large numbers of user
Education	• university admission

Granular Fuzzy Web Search Agents

Yanqing Zhang and Shi Hang
Department of Computer Science
Georgia State University
Atlanta, GA 30303
USA
yzhang@cs.gsu.edu

Tsau Young Lin
Department of Mathematics and Computer Science
San Jose State University
San Jose, CA 95192
USA
tylin@cs.sjsu.edu

Yiyu Yao
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
yyao@cs.uregina.ca

Abstract

With explosive growth of the Internet, Web sites and Web databases, the Internet waiting times of a huge number of Web users are going up because current precise-key-word-based search engines generate a large number of search pages and the search results are not necessarily clustered and ordered intelligently and efficiently. To increase Internet search speed and improve the Internet QoS (Quality of Service), this paper presents new granular fuzzy search agents based on granular computing, fuzzy computing, linguistic computing, data mining, personalization and intelligent agent technology. The personalized fuzzy Web search agent, the personalized granular Web search agent, and the Chinese Web search agent are proposed.

1. Introduction

With explosive growth of the Internet, Web sites and Web databases, there are more and more data and information on the Web. The resulting problem is that the Internet waiting times of a huge number of Web users are going up because current precise-key-word-based search engines generate a large number of search pages and the search results are not necessarily clustered and ordered intelligently and efficiently. Therefore, generating a small number of very relevant search pages and ordering search results based on similarity and relevancy are two major approaches to increasing Internet search speed and improving the Internet QoS (Quality of Service).

Many search engines support Boolean operators, field searching, and other advanced techniques. A user simply enters key works and Boolean operators, and then clicks on the Search button. Thousands of hits may be generated by the precise-key-word-based search engines. In many cases, the first several pages may not include highly relevant results for different users. The basic premise of relevancy searching is that results are sorted, or ranked, according to certain criteria. Criteria can include the number of terms matched, proximity of terms, location of terms within the document, frequency of terms, document length, personal preferences and other factors. The exact numerical formula of these criteria is the ranking algorithm. This paper proposes a personalized granular-fuzzy-relevancy-matrix-based Web search algorithm which can sort search results based on a fuzzy relevancy matrix, granular relevancy matrix and users' preferences. In addition, fuzzy operators and granular operators like fuzzy AND, fuzzy OR, granular AND and granular OR are also used to connect key words so as to make fuzzy or granular searches.

Different users have different interesting areas. It is desirable that the search engine can be personalized. For example, when a user inputs a key word "bridge", engineers get search results about bridges, gamers get some things about the card game, and tourists get information about some bridge locations if the search engine is personalized. If the search engines can keep a record of searched areas of the users, it will be helpful for the users in the future searching. This paper proposes two new smart search agents: (1) the general granular-fuzzy-key-word-based search agent which can serve all different users like eBay and Yahoo, and (2) the personalized granular-fuzzy-key-word-based search agent which can serve an individual user like My eBay and My Yahoo. Data mining techniques are used to update fuzzy relevancy matrices of the two different search agents.

There is another problem during the searching. That is the searching based on exact quantities is a not suitable for all situations. Sometimes, the users want to seek a hotel at a proper price. But proper price may not be an exactly number (say $49.99). Under this situation, we should use fuzzy terms like around $50 and about 120 miles for fuzzy search. In general, a

linguistic search agent is an ideal system which can use flexible linguistic terms and different languages.

In addition, granular computing can be used to design pure granular search agents. For example, rough sets can be used to design a rough search agent. The rough search agent may provide more relevant results by using rough sets and data mining techniques. The interval search agent can select possible search results based on interval computing.

In summary, this paper presents several granular fuzzy Web search agents such as the fuzzy Web search agent, the granular search agent like rough search agent and interval search agent, and the Chinese fuzzy search agent based on soft computing, granular computing, linguistic computing, data mining, personal profile, and intelligent agent technology [15][16][18-24].

2. History

Only a few experts predicted in 1989 that the web would emerge from the Mesh proposal- originally a funding pitch to CERN management for a project that would use hypertext to manage general information about particle accelerators and physics experiments at the lab. Even less predictable was that a humble link directory created by a couple of graduate students to help people find their way around the nascent Web, or that a few experimental indexing services quaintly dubbed "search engines", would ultimately become media superstars.

Search Engine: software that provides Web site addresses that contain one or more terms or keywords specified in a user's query. The term search engine is sometimes used, incorrectly, to mean a manual index of the Web compiled by editor.

The Web search engines have broad popular appeal. They are the tools that have finally made end-user searching a reality. While they cater to the general public, the Web search engines are increasingly important tools for the information professional as well. Today a typical Internet user faces mundane, repetitive tasks such as browsing, filtering, and searching for relevant information. In 1998, the global information search space on the Internet consists of an estimated 320 million HTML pages in the Web. The Web may very well be the most elegant real-world manifestation of the central metaphor of chaos theory. It follows that the Web has also become the largest, most complex search space ever. World Wide Web Search Engines have become the most heavily-used online services, with millions of searches performed each day.

3. Current Techniques

Many search engines support Boolean operators, field searching, and other advanced techniques, but with relevancy searching users simply enter their terms and click the Search button. While searches may retrieve thousands of hits, search engine produces claim their systems place items that best match the search query at the top of the results list.

The basic premise of relevancy searching is that results are sorted, or ranked, according to certain criteria. Criteria can include the number of terms matched, proximity of terms, location of terms within the document, frequency of terms (both within the document and within the entire database, document length, and other factors. The exact "formula" for how these criteria are applied is the "ranking algorithm" and varies among search engines.

For example, *Hotbot* describes term frequency and location as primary factors. *AltaVista* considers these factors, as well as the number of terms matched and the proximity of search terms. *Infoseek* gives extra weight to terms in the title and metatags. *Lycos* considers terms in metatags. *Excite* analyzes the content of the documents for related phrases in a process it calls Intelligent Concept Extraction (ICE).

Users are much more likely to scan their result lists and retrieve only selected documents. The user may consider a number of factors in deciding whether or not to retrieve a document, but a key factor is the number of terms matched. While a user will typically browse only the first few pages of results, the ranking of those results provided by the search engine is crucial to the success of the search session and the user's perception of his satisfaction with the results.

As the Web continues it relentless growth, search tools must evolve and adapt to remain useful. "Currently, search is simply bad." Joel Truher said, "It is like interacting with a snotty French waiter. The service is bad, you get served things you did not ask for, you often have order again and again, and you can not get things that are listed on the menu."

In the early months of 1999, the phase "Web search" generally means a keyword search on some kind of index or directory of textual documents. When we search, we still most commonly think of finding information on Web "pages," each having discrete URLs (Uniform Resource Locators, or "addresses").

As we moved to the new century, improvements in computer processing and storage capabilities, together with the stunning increase in available

bandwidth for data communications, means we will be seeing far more types of media on the web. And increasingly, information will be kept in databases that serve content dynamically, rather than stored as static Web pages.

4. Fuzzy Web Search Agent

Different fuzzy techniques have been used in Web search [8][9]. However, some "fuzzy" Web search engines don't use fuzzy logic. Actually they can just search approximate results based on criteria like key words. For clarity, here we use fuzzy logic as a basic tool to design a fuzzy-logic-based Web search agent (a fuzzy Web search agent in short) for better QoS of Web search. The novel technique proposed here can use not only traditional fuzzy-key-word-based search method but also fuzzy-user-preference-based search algorithm so as to get more satisfactory personalized search results for a particular user. In this sense, if user A and user B type in the same search key words with fuzzy operators such as fuzzy AND or fuzzy OR, user A and user B will get two different search results because user A has a different profile from user B. Clearly, personalized fuzzy Web search agent is more powerful than traditional fuzzy Web search engine because a user's profile is taken into account. Therefore, the traditional fuzzy Web search engine is a bottom building block of the personalized fuzzy Web search agent. The logical architecture of the Personalized Fuzzy Web Search Agent (PFWSA) is given in Fig. 1. In general, the personalized fuzzy Web search agent consists of the basic Fuzzy Web Search Engine (FWSE), the Personalized Fuzzy DataBase (PFDB), and the final Fuzzy Fusion System (FFS). In Fig. 1., X is a user's inputs such as key words and logical operators, Y is updated user's personal information from either a user or an automatic data mining system, and Z is the final Web search results generated by the FFS based on results generated by the FWSE and personal information from the PFDB.

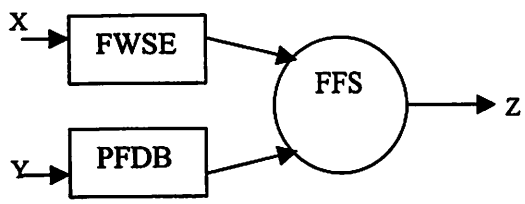


Fig. 1. Logical Architecture of the PFWSA

A fuzzy relevancy matrix is used to show similarity between two fuzzy search terms. A problem is the fuzzy relevancy matrix could be very large. To reduce complexity of the fuzzy relevancy matrix, a personalized fuzzy relevancy matrix is used to actually design a personalized fuzzy Web search algorithm. Since a user has a small number of frequently used search words, the personalized fuzzy relevancy matrix will be small. The personalized fuzzy relevancy matrix is updated dynamically based on the PFDB which is also updated periodically by mining the user's Web usage and preferences.

The personalized fuzzy Web search algorithm is described logically as below:

Begin

Step 1: Use input key words and operators to match the personalized fuzzy relevancy matrix, and find out relevant key words ranked by similarity;

Step 2: Use these key words to do regular Web search to find out candidate results;

Step 3: Use personal profile in the PFDB to select a small number of personalized results from the candidate results based on ranked personal preferences.

Step 4: Display the final results in the ranked order

End

5. Granular Web Search Agent

Granular computing (GrC) is a label of theories, methodologies, techniques, and tools that make use of granules in the process of problem solving [15][16][18][19]. Basic ingredients of granular computing are subsets, classes, and clusters of a universe [15][16]. There are many fundamental issues in granular computing, such as granulation of the universe, description of granules, relationships between granules, and computing with granules.

In designing a GrC based Web search agent, we will focus on the three important components of a search system, a set of Web documents, a set of users (or a set of queries, as user information needs are typically represented by queries), and a set of retrieval algorithms. For text based Web documents, they are normally represented through a set of index terms or keywords. A GrC based agent will explore potential structures on these four sets of entities, in order to improve efficiency and effectiveness of Web search. Granulation of the set of documents has been considered extensively in the design of cluster-based retrieval systems, in order to reducing computational costs [11][13]. In this approach, a collection of documents is divided into clusters such that each

cluster consists of similar documents. A center is constructed for each cluster to represent all the documents in that cluster. A hierarchical clustering of documents is produced decomposing large clusters into smaller ones. The large clusters offer a rough representation of the document. The representation becomes more precise as one moves towards the smaller clusters. A document is then described by different representations at various levels. Hence, a cluster-based retrieval system implicitly employs multi-representation of documents.

Retrieval in the system is carried out by comparing a query with the centers of the larger clusters. If the center of the current cluster is sufficiently close to the query, then the query will be compared against the centroids of the smaller clusters at a lower level. In other words, if it is concluded that a document is not likely to be useful using a rough description, then the document will not be further examined using more precise descriptions. Different retrieval methods strategies may also be employed at different levels. It is important to realize, however, that the use of document clustering only reduces the dimensionality of the document collection while the dimensionality of index terms remains the same. That is, the same number of terms is used for the representation of cluster centers regardless of the level in the document hierarchy.

The notion of constructing a term hierarchy to reduce the dimensionality of terms has been studied [10][12]. A main consideration is the existing trade-off relationship between the high dimensionality of index terms and the accuracy of document representation. One may expect a more accurate document representation by using more index terms. However, the increase of the dimensionality of index terms also leads to a higher computational cost. It may also be argued the addition of index terms may not necessarily increase the accuracy of document representation as additional noise may be added. Recently, Wong et al. [14] suggested granular information retrieval. It is explicitly demonstrated that document clustering is an intrinsic component of term clustering. In other words, term clustering implies document clustering. In a term hierarchy, a term at a higher level is more general than a term at a lower level. A document is then described by fewer more general terms at a higher level, while is described by many specific terms at a lower level. Retrieval in a term hierarchy can be done in a manner similar to retrieval in a document hierarchy. There are many advantages to our proposed approach of granular information retrieval. As already mentioned, the proposed method reduces the dimensionality of both the document and term spaces. This provides the opportunity to focus on a proper level of granulation of the term space. In general, the method provides a model for developing knowledge based intelligent retrieval systems.

In a similar way, we can granulate the set of users. If queries or user profiles are represented in a similar form as documents, the process is much simpler. Granulation of users can be done either by pure hierarchical clustering or through concept hierarchy. A hierarchical structure may also be imposed on the document retrieval functions. Many retrieval functions have been developed for information retrieval, including exact Boolean matching, co-ordination level matching, fuzzy logic matching, inner product, and cosine similarity measure. Obviously, these functions do not share the same computational complexity and accuracy characteristics. For example, the co-ordination level matching is less expensive to compute than the cosine similarity measure, while at the same time being less accurate. At the higher levels of the term hierarchy involving more general descriptions, a simpler less expensive retrieval function may be used. On the contrary, a more expensive retrieval function can be used at the lower levels of the term hierarchy.

In summary, GrC based retrieval agents will explore the structures of various of entities in a retrieval system through granulation. In particular, different granulated views can be developed, and an agent chooses suitable views to achieve the best results. The framework of granular Web search agents allows multi-representation of Web documents and users, as well as multi-strategy retrieval. The challenging issues will be the granulation of documents, terms, users and retrieval algorithms, the representation of various objects under different granulated views, and the selection of suitable granulated views.

It is expected that granular Web search agents will be a potential solution to many difficulties involved in Web search.

6. Chinese Web Search Agent

There are a lot of challenges to the Internet users, including how to cope with the problems of the cross-social, cross-cultural and multi-lingual cyberspace. Right now, there are more and more Internet users in China. They also face the mundane information when they are looking through the web. They need good

search engines to look for the useful information. There are many Chinese websites support search services, but those kinds of services are not real search engines in definition. Those kinds of searching through the Web are based on Web Index searching or a searching in the limited area and websites. The limitation of the Chinese database searching restricts the development of the Internet in China.

There were many attempts to develop the Chinese web search engines during the past few years, such as the *SkyNetwork* of Beijing University, *WebCompass* of Tsinghua University and so on. Because of the shortage of researchers and the limits of the hardware, there is not any Chinese Web Search Engine successfully in business.

At present the search engine services of Chinese websites are mainly shifted from English Web Search Engines. For example, *SOHU.com* bought the *Verity* software as the its Chinese Web search engine, *Netease.com* uses the search engine from *AltaVista,* and *SINA.com* uses the software of *Openfind* company from Taiwan. But all these search engine services have some kinds of limitation. There are many problems to be solved for the development of Chinese Web Search Engines.

The main difference between the Chinese and English in the linguistics is the difference between the character and word. In English the character usually has no meaning, but in Chinese every character has at least one meaning. Although there are only about 3000 basic characters in Chinese, the simple combination of these characters can have many meanings. This difference causes the different mechanism in the Chinese and English search engines. Sometimes the mixture of the Chinese and English words is given to the search engines, but the searching is failed. It is because the search engines cannot support the techniques of searching for the mixed language in the web. There are still some problems of the Chinese search engines: the Chinese character code. Due to the historical reasons, there are three major Chinese character codes: GB2312, GBK and BIG5, used in the Chinese websites. The different standards have the different character libraries. So sometimes one character has the different address in the different character libraries. And this causes failure of the searching under different code.

Hence we want to introduce a new method to solve the problems of the Chinese web search engines. In this method, we use fuzzy logic to identify the Chinese words during the searching.

First, after the users typed the Chinese word into the search engines. The search engines identify the word to distribute the English characters, the numerical characters and Chinese characters by fuzzy logic analysis. In the analysis, the search engines compare the combination of these characters with their own word matrix to find the related words. Then with these words, search engines will compare the meaning of these words, then select the most meaningful words as the keywords for the searching. Through this analysis, the search engines will not search the information by the different characters, but by the related meaningful words.

Second, from the results of the first step, the search engines look through their related Web databases and websites to find the matched information. The amount of databases of the search engines connected is an important factor for the results of searching. The more databases it related, the more accurate the useful information will be searched.

Third, after the results of the searching come out, the search engines need to do some sorting for the results and give the most matched information to the user. With the different sorting criteria, the sorting results will be different. And the criteria are based on the requirement of the most users.

In this paper, fuzzy logic is used in the Chinese search agent. The Chinese fuzzy search agent can accept fuzzy Chinese words, and then use them to narrow down search results based on a fuzzy Chinese character relevancy matrix and personal preferences.

7. Conclusion and Future Work

This paper presents a brief introduction of the search engines, the history, current status and future trends of search engines. By upgrading search engines to search agents, fuzzy Web search agents, granular Web search agents and the Chinese Web search agents are proposed to reduce Web search redundancy, increase Web search relevancy, and decrease users' Web search time.

In the future, advanced intelligent techniques such as soft computing, granular computing, rough sets, linguistic computing, different languages, intelligent agent technology, and distributed computational intelligence will be used to design a hybrid personalized multi-language Web search agent to continue to improve QoS of the Web search for people in different countries. In addition, wireless hybrid personalized multi-language Web search agent will be very useful on mobile hand held devices.

References

[1] Agrawal, A.Arning, T.Bollinger, M.Mehta, J.S hafer, R.Srikant, The quest data mining system, *Proc. of the 2nd Int'l Conference on Knowledge Discovery in Databases and Data Mining, Portland*, Oregon, August, 1996.

[2] Mathew Schwartz, "Search Engines", *Computerworld*, vol. 34, no. 19, pp. 78, 2000.

[3] Chris Sherman, "The Future of Web Search", *Online*, pp. 54-61, May 1999

[4] Berry, Michael J.A & Linoff, Gordon, *Data mining techniques for marketing, sales and customer support*, 1997.

[5] Manish Mehta, Rakesh Agrawal and Jorma Rissanen: SLIQ: A Fast Scalable Classifier for Data Mining.

[4]. http://sac.uky.edu/~stang0/DataMining/WhatIs.html What is Data Mining.

[5]. http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologi/datamining.html Data Mining: What is Data Mining?

[6]. http://www3.shore.net/~kht/text/dmwhite.html An introduction to data mining

[7]. http://192.35.251.71/datamine/trees.htm Data mining techniques: Decision Trees

[8]. http://zaptron.com.

[9]. http://aptronix.com.

[10] H. Chen, T. Ng,, J. Martinez, B. Schatz, "A Concept Space Approach to Addressing the Vocabulary Problem in Scientific Information Retrieval: An Experiment on the Worm Community System," *Journal of the American Society for Information Science*, vol. 48, no. 1, 17-31, 1997.

[11] E. Rasmussen, "Clustering Algorithms.," In: Frakes, W., Baeza-Yates, R. (Eds.): *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, Englewood Cliffs, USA, pp. 419-442, 1992.

[12] K. Spark Jones, "Automatic Keyword Classification for Information Retrieval," Butterworths, London, UK, 1971.

[13] Willett, "Recent Trends in Hierarchic Document Clustering: A Critical Review," *Information Processing and Management*, vol. 24, no. 5, pp.577-597, 1988.

[14] S.K.M. Wong, Y.Y. Yao, and C.J. Butz, " Granular information retrieval, " *Soft Computing in Information Retrieval: Techniques and Applications*, Crestani, F. and Pasi, G. (Eds.), Physica-Verlag, Heidelberg, pp. 317-331, 2000.

[15] Y.Y. Yao, "Granular computing: basic issues and possible solutions," *Proceedings of the 5th Joint Conference on Information Sciences*, pp. 186-189, 2000.

[16] Y.Y. Yao. and N. Zhong,, "Potential applications of granular computing in knowledge discovery and data mining," *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, pp. 573-580, 1999.

[17] L.A. Zadeh, "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems*, vol. 19, pp. 111-127, 1997.

[18] Y.T. Lin, "Granular Computing: Fuzzy Logic and Rough Sets," *Computing with words in information/intelligent systems*, L.A. Zadeh and J. Kacprzyk (eds), Springer-Verlag, 1999.

[19] Y.T. Lin, "Data Mining: Granular Computing Approach," *Proceedings of he Third Pacific-Asia Conference on Knowledge Discovery and Data Mining*, April 26-28, Springer-Verlag, Lecture Notes in Artificial Intelligence series, 1999.

[20] Y.-Q. Zhang, M. D. Fraser, R. A. Gagliano and A. Kandel, "*Granular Neural Networks for Numerical-Linguistic Data Fusion and Knowledge Discovery*," Special Issue on Neural Networks for Data Mining and Knowledge Discovery, *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, pp.658-667, May, 2000.

[21] Y.-Q. Zhang and A. Kandel, "Compensatory Genetic Fuzzy Neural Networks and Their Applications," Series in Machine Perception Artificial Intelligence, Volume 30, World Scientific, 1998.

[22] Walter Brenner, Rudiger Zarnekow and hartmut Witting, "Intelligent Software Agents: foundations and applications," Springer, 1998.

[23] H. S. Nwana and N. Azarmi, "Software Agents and Soft Computing: Towards Enhancing Machine Intelligence," Springer-Verlag, 1997.

[24] Dimitris N Chorafas, "Agent Technology Handbook," McGraw Hill, 1997.

Fuzzy Neural Web Agents for Stock Prediction

Yanqing Zhang and Somasheker Akkaladevi
Department of Computer Science
Georgia State University
Atlanta, GA 30303 USA
yzhang@cs.gsu.edu

George Vachtsevanos
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332 USA

Tsau Young Lin
Department of Mathematics and Computer Science
San Jose State University
San Jose, CA 95192

Abstract

A fuzzy neural Web-based stock prediction agent is developed using the Granular Neural Network (GNN) which can discover fuzzy rules. After learning from the past stock data, the GNN is able to use discover fuzzy rules to make future predictions. After doing simulations with six different stocks (msft, orcl, dow, csco, ibm, km), it is conclusive that the fuzzy neural stock prediction agent is giving less average errors with large amount of past training data and high average errors in case of less amount of past training data. To implement this stock prediction Web agent, Java Servlets, Java Script and jdbc are used. SQL is used as the back-end database. The performance of the GNN algorithm is compared with the performance of the BP algorithm by training the same set of data and predicting the future stock values. The average error of the GNN is less than that of BP algorithm. It is also possible to make this system as a commercial application by updating such that it gives more user-friendly functionality and by giving more valuable information to the users.

1. INTRIDUCTION

In Individuals, companies and governments spend vast quantities of resources attempting to predict financial markets. Careful thought will tell you that there can be no "common knowledge" system that can predict financial markets. Assuming there is such a model, then everybody would know the best time to sell. Ideally everybody would be selling at this time. If everybody is selling, then who is buying? Therefore, there cannot be any publicly available system for predicting markets. On the other hand, a prediction software would be very useful to assist individual in reaching a final decision. Then (assuming that it is possible to predict markets), this paper is developed which uses Granular Neural Networks technique to predict the future stock values. Overall, time series forecasting provides reasonable accuracy over short periods of time, but the accuracy of time series forecasting diminishes sharply as the length of prediction increases.

Recurrent neural networks have an architecture where the outputs from intermediate neurons are fed back to the input layer. In this way the network will contain memory of previous values of the input data. A single value y(t-1) is used as input and a single value O(t) is produced as output from the neural network. The temporal dependencies are modeled in the weights in the feedback loops in the network. Applications of recurrent networks can be found in Ellman [2]. Since a moving average is a past estimate, a technical trader often misses a lot of the potential in the stock movement before the appropriate trading signal is generated. Thus, although technical analysis may yield insights into the market, its highly subjective nature and inherent time delay does not make it ideal for the fast, dynamic trading markets of today.

When compared to these techniques, the Granular Neural Network (GNN) is able to discover fuzzy rules from training data and then the trained GNN can predict future values for new inputs. The basic strategy is to produce a system that predicts the price or the value of a property or index in the future.

The rest of this paper is organized as follows. Section 2 discusses the GNN architecture and the data mining algorithm for stock prediction. Section 3 describes details on system implementation. Section 4 analyzes stock prediction performance and gives comparisons. Section 5 summarizes conclusions and future work.

2. Granular Neural Networks

Generally speaking, a GNN (Granular Neural Networks) is capable of processing various granular data (granules) [18][19]. Granules could be a class of numbers, a cluster of images, a set of concepts, a group of objects, a category of data, etc. These granules are inputs and outputs of GNNs as multimedia data are inputs and outputs of biological neural networks in the human brain.

Therefore, granular-data-based GNNs are more useful and more effective to process multimedia granules than conventional numerical-data-based neural networks. Here, the FNN is a powerful Fuzzy Neural Network with Knowledge Discovery (FNNKD) [19].

Since FNNKDs are basic building blocks of a FGNN, the FNNKD is discussed first, and then the architecture of the FGNN will be described later. The detailed 5-layer architecture of the FNNKD is given in [19]. A 2-input-1-output FGNN consists of 3 layers. The functions of different layers are described layer by layer as follows:

Layer 1: Linguistic Feature Extraction Layer

In this layer, numerical-linguistic X and Y are transformed to corresponding fuzzy feature vectors *(a1, b1, c1, d1)* and *(a2, b2, c2, d2)*, respectively.

Layer 2: Multi-FNNKD Layer

This layer consists of 4 dedicated FNNKDs (i.e., FNNKD1, FNNKD2, FNNKD3, and FNNKD4). FNNKD1 is a $2 \times k_1 \times 1$ fuzzy neural network which generates a crisp center a of an output fuzzy set by using k_1 fuzzy rules. FNNKD2 is a $2 \times k_2 \times 1$ *fuzzy* neural network which generates a crisp width b of an output fuzzy set by using k_2 fuzzy rules. FNNKD3 is a $2 \times k_3 \times 1$ fuzzy neural network which generates c of an output fuzzy set by using k_3 fuzzy rules. FNNKD4 is a $2 \times k_4 \times 1$ fuzzy neural network which generates d of an output fuzzy set by using k_4 fuzzy rules.

An n-input-1-output normal fuzzy system has m fuzzy IF-THEN rules which are described by

IF x_1 is A_1^k and ... and x_n is A_n^k THEN y is B^k, where x_i and y are input and output fuzzy linguistic variables, respectively.

Layer 3: Output Layer

Case 1: A fuzzy linguistic output is Z represented by a fuzzy feature vector *(a, b, c, d)*.

Case 2: A crisp numerical output is a.

For simplicity, the detailed learning algorithm is given in [18]. Once the learning procedure has been completed, all parameters for a FNNKD have been adjusted and optimized. As a result, all m fuzzy rules have been discovered from training data. Finally, the trained FNNKD can generate new values for new given input data.

The GNN consists of 3 inputs and 1 output. For stock prediction the 3 inputs are open, high, low values of past-historical data of a particular stock company and 1 output is the close amount for each day in the historical data. Neural network takes this historical-data for training. After training it will create a weight file for the given historical data. According to the algorithm to predict the future stock values, we have to give the number of days to predict the future values.

For example: If we consider 3 input and one output, the training function y = f (f1, f2, f3) where f1, f2 and f3 are 3 inputs and y is the output, which is a function of 3 inputs.

3. SYSTEM IMPLEMENTATION

To implement this stock prediction system, Java Servlets, Java Script and Jdbc are used. SQL is used as the back-end database.

3.1 Concepts of Input Data

The system can predict future for any stock or market index. For example to predict Intel stock values for some days ahead, historical data (past) is required. You can change the prediction to one-day, one-week, or one-year ahead by substituting the historical data.

Data for neural networks is probably the most important aspect for training. How well the network performs is very dependent on the quality of the input data.

A data set is a collection of variables used to make a prediction. Examples of the variables are various stocks and indexes, and each stocks date, open, low, high, close and volume data. The output of the Predictor is a distribution chart or values from which we can easily understand the future prospects of that particular stock. Database tables have been generated using stock-historical data from www.yahoo.com site.

Without user intervention a particular stock data in the form of an Excel spread sheet is obtained. Preprocessing operation is needed as preparatory step for next stage. As an example of preprocessing, the

downloadable data is in DD-MM-YY format. The format should be changed to DD-MM-YYYY so that it can be properly inserted into the database tables. The data is saved in ASCII format. A Java program does the process of inserting this data into the database. This program coverts the whole file line by line into values and inserts them into the table. The same name is used for the text file as the stock symbol name. So the program uses the text file name to create a table with the same name.

Each line of the data set must contain 5 values date, open, high, low, close values of the stock. Each value is separated by space (see Table 1).

dd/mm/yyyy	date	date of the day
double	open	Initial Price
double	low	Lowest traded price
double	high	Highest traded price
double	close	Price of the last trade
double	volume	number of traded stocks

Table 1 Meaning of data parameters

3.2 Software Parameters

There are some parameters, which we have to use during the training period. They are error, threshold, tolerance, etc. The accurate results depend on these parameters.

The prediction algorithm takes all these parameters as input. This algorithm is called when we click predict future from the system after entering the stock symbol. This algorithm makes the neural network learn. The algorithm returns the future values as output witch are eventually stored in results table for each stock. We will keep track of these results until we click on the average error for all simulations for this particular stock. Java program displays all the results of predictions for any particular stock at a time on the web page and clears the contents of the table. This means that now the table is ready to store any new predictions on that stock symbol.

3.3 Overview of Implementation

A full run of the program implementation will be described, going through all the main features of the program:

- Download historical data from the Internet.
- Copy the whole data into a text and run the program, which inserts the data into the database.
- A program is written through which users can buy and sell the stock shares, and the corresponding data is stored in the database.
- A program is written through which each user can see his/her own transactions.
- Algorithm, which trains the granular neural networks using the mean square error as stop criterion for learning, while never exceeding the maximum number of cycles which can take testing data from the initial date to the user entered date, and predicts the future stock closing values.
- A program is written, which compares the predicted values with real values.

One of the most important factors here is to construct a neural network deciding on what the network will learn. A neural network must be trained on some input data. The two major problems in implementing the training are

- Defining the set of input to be used (the learning environment)
- Deciding on an algorithm

When the program is used for prediction on new values, the network should be trained up to the date to be predicted, before making forecast. A new learning set has to be made, which contains the values up to the desired date. It is also possible to make an entirely new prediction (i.e. a prediction where the target is unknown.) In this case we do not compare the predicted values with the real values because we do not have real values for the future.

4. PERFORMANCE ANALYSIS

Using the developed system to predict the future stock values using Granular Neural Networks we can do some simulations to know the performance of the algorithm.

4.1 Predicting A Stock Using Complete Data

By using the complete past historical data, if we predict stock values for future 30days from the algorithm we are now able to compare the predicted values with the real values. The average error for this simulation is 1.582.

4.2 Predicting A Stock Using Less Data

In another sample simulation, we take the data from 1981 to 1994 and predict the stock values for some future days. The average error for this simulation is 2.11. The dotted curve shows the

predicted values and the solid curve shows the real values.

The average error for this simulation is greater than the previous simulation. In this graph the results are not close to the real values compared to the previous chart. So based on average errors and the graphs we can conclude that, more the data we have the better training the neural network gets and gives more close results. This means that, more the available data for predicting financial markets, the greater the chances of an accurate forecast.

4.3 Predicting A Stock by varying training error

For the same test case if we decrease the maximum training error parameter from 0.0001 to 0.000015, we are getting more close results. For dow stock the curve is shown in Fig. 9. The dotted curve shows the predicted values with error 0.0001 and the solid dotted lines shows the predicted values with maximum training error 0.000015 and the solid curve is the real data.

The average error is only 1.05 when compared to 1,582 in case of high training error. By comparing average error, we can conclude that, the resulting future stock values are closer than the future stock values with high training error.

Simulations are done on six stocks (msft, orcl, dow, csco, ibm, km). From the simulation results it is conclusive that, the average error for simulations using lot of data is small compared to the average error using less data and the more data for training the neural network, the better prediction it gives.

4.4 Comparison Between GNN and BP

The performance of the GNN algorithm is compared with the performance of the BP algorithm by training the same set of data and predicting the future stock values. If the training error was set at 0.03 and the neural network was trained for dow stock data using both the algorithms. The GNN took 2 minutes 58 seconds to train the neural network where as BP took 2 hours and 55 minutes. The GNN's average error was 1.39 where as BP gave 3.38.

The average error for GNN is less compared to the average error for BP algorithm. From the average error and the graph it is conclusive that, GNN produced closer future stock values with the real stock values compared to the BP algorithm using less training error. If the training error was set at 0.07 and the neural network was trained for csco stock data using both the algorithms. The GNN took 2 minutes to train the neural network where as BP took 1 hour and 48 minutes. The GNN's average error was 6.09 where as BP gave 7.16.

The average error for GNN is less compared to the average error for BP algorithm. From the average error and the graph it is conclusive that, GNN produced closer future stock values with the real stock values compared to the BP algorithm using less training error. Based on the above two simulations, the overall performance with GNN technique is better than BP technique.

5. CONCLUSION AND FUTURE WORK

After completing several simulations for predicting several stocks based on the past historical data, it is conclusive that, the average error for simulations using lot of data is small compared to the average error using less amount of data. This means that, the more data for training the neural network, the better prediction it gives. If the training error is low, predicted stock values are close to the real stock values. The results are good when we use the GNN algorithm when compared to the BP algorithm.

One possibility for future work is to update the system, so that it can read the past stock data automatically from the web and store them in the database. In this way the system will become Internet ready for predicting any stock market and is ready at any time. Another possibility for future work is to update the system, which can allow to trade the stock, which means users can manage to buy and sell the stock after seeing the prediction values from this system. In this way the system can eventually become Internet ready to be used anywhere in the world at any time.

The system can be updated, so that it will consider the other stock information as inputs to train the neural network. Then the system would become more reliable in the real world. The system can also be updated for mutual fund applications, which is similar to the stock prediction application. The system can be updated so that the, stock prediction results from all simulations can be compared with the existing neural network techniques such as mat lab or which are existing on online on the internet.

It is also possible to make the fuzzy neural Web-based stock prediction agent system as a commercial application by updating such that it gives more user-friendly functionality and by giving more valuable information to the users.

Acknowledgments

The authors would like to thank Dr. Saeid Belkasim and Dr. Raj Sunderraman very much for their comments.

References

[1] G. Cybenko, "*Approximation by superpositions of a sigmoidal function*," Technical Report, University of Illinois, 1998.

[2] J. Ellman, "*Finding structure in time.*,"*Cognitive Science*, pages 179-211, 1990.

[3] Thomas HellStrom and Kenneth Hollmstrom, "*Predicting the Stock Market*," Malardalen University, Technical Report Screen IMa-TOM-1997-07, 1998.

[4] J. Hertz, K. Anders and G. Palmer, "*Introduction to the Theory of Neural Computation*," Addison-Wesley Publishing Company, 1991.

[5] http://www.yahoo.com/.

[6] I. Huntley, ShareHolder, "*Top 500 Companies*," Ian Huntley Pty Limited, 1996.

[7] N. Karayiannis and A. Venetsanopoulos, "*Artificial Neural Networks. Learning Algorithms, Performance Evaluation and Applications*," Kluwer Academic Publishers, 1993.

[8] C. Klimasauskas, "*Applying neural networks.* In *Neural Networks in Finance and Investing*," Section 3, pages 47–72., Probus Publishing Company, 1993.

[9] Ramon Lawrence, "*Using Neural Networks to Forecast Stock Market Prices*," University of Manitoba, 1997,

[10] MetaStock Software Package.

[11] J. Robert and Van Eyden, "*The Application of Neural Networks in the Forecasting of Share Prices*," Finance and Technology Publishing, 1996.

[12] A. Sing, "*Application of neural networks for predicting financial markets*," University of Queensland, 1997.

[13] Manfred Steiner and Hans-Georg Wittkemper, "*Neural networks as an alternative stock market model*," In *Neural Networks in the Capital Markets*, Section 9, pages 137–148, John Wiley and Sons, 1995.

[14] F. Takens, "*Detecing strange attractors in fluid turbulence*," In D. Rand and L-S. Young, editors, *Dynamical Systems and Turbulence*, Springer, 1981.

[15] E.A. Wan, "*Time Series prediction by using a connectionist network with internal delay lines*," In A. Weigend and N. Gershenfeld, editors, *Time Series Prediction. Forecasting the Future and Understanding the Past*, SFI Series in the Sciences of Complexity, ProcAddison-Wesley, 1994.

[16] H. White, "*Economic prediction using neural networks: The case of IBM daily stock returns,*" Proceedings of the IEEE International Conference on Neural Networks (ICNN'88), San Diego, California, 1988.

[17] Y. Yoon and G. Swales, "*Predicting stock price performance: A neural network approach. In Neural Networks in Finance and Investing*, " Section 19, pages 329–342, Probus Publishing Company, 1993.

[18] Y.-Q. Zhang, M. D. Fraser, R. A. Gagliano and A. Kandel, "*Granular Neural Networks for Numerical-Linguistic Data Fusion and Knowledge Discovery*," Special Issue on Neural Networks for Data Mining and Knowledge Discovery, *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, pp.658-667, May, 2000.

[19] Y.-Q. Zhang and A. Kandel, "Compensatory Genetic Fuzzy Neural Networks and Their Applications," Series in Machine Perception Artificial Intelligence, Volume 30, World Scientific, 1998.

Scientific and Philosophical Contribution of L.A. ZADEH*

I.BURHAN TÜRKŞEN
President IFSA
Director, Information / Intelligent Systems Laboratory
Mechanical and Industrial Engineering
University of Toronto
Toronto, Ontario, M5S 3G8
CANADA
turksen@mie.utoronto.ca

Abstract: - Zadeh's contributions are both scientific and philosophical. On the one hand, with his seminal papers on fuzzy sets and fuzzy logics, he has caused a "grand paradigms shift" in scientific taught process. This has lead to many scientific and engineering discoveries and generation of novel solutions to complex electro-mechanical systems. On the other hand, with his celebrated papers on "Concept of a Linguistic Variables", "Computing with Words" and "Computing with Perceptions" he has expounded on new concepts that were germane to analysis and discovery of newer patterns of thinking" in psychology, linguistic, social sciences, economics, bio-medicine, genomics, etc.

Key-Words: - Fuzzy Sets, Logics, Approximate Reasoning, Linguistic Variables, Computing with Perceptions, Decision-Making, Humanistic Systems.

Introduction

L. A. Zadeh's initial contributions expressed in his seminal papers, e.g., "Fuzzy Sets" (1965), "Probability Measures of Fuzzy Events" (1968), "Outline of a New Approach to the Analysis of Complex Systems..." (1973), "Fuzzy Sets as a Basis for a Theory of Possibility" (1978), "Theory of Approximate Reasoning" (1979), etc., had a direct impact on mathematics, science and engineering and caused the development of fuzzy-neural system modeling and creation of novel solutions for electro-mechanical systems.

However, I believe, his main thesis has been that, in humanistic systems, human reasoning and decision making is not just "measurement" based, although it is an important component, but it is rather linguistic and perception based. The concepts and notions embedded in his celebrated papers, e.g., "Concept of a Linguistic Variable..." (1975-1976), "The Role of Fuzzy Logic..." (1983), "Syllogistic Reasoning..." (1985), "Computing with Words" (1996-2001), suggested novel approaches to complex humanistic systems for their potential analysis and solutions in humanistic terms in manners akin to "Human-like Reasoning".

In order to understand the deeper meanings of Fuzzy Theory, let us delve very briefly into the etymological origins of the word fuzzy.

R. Hodge (2001) states that " 'Fuzzy Logic' was born out of Zadeh's acute sense of different logic(s) inherent in human languages. ... his concern for the strengths as well as weaknesses of natural languages in scientific thought...". Zadeh uses `fuzzy` "to apply to categories of language or thought, not to the nature of (physical, mechanistic) phenomena". His use of "fuzzy" is an "example of his genius with language" with a background in " Indo-European languages", i.e., Russian and Iranian, and Turki languages, i.e., Azeri-Turkish.

Hodge (2001) further states that 'fuzzy' comes from a word "fusus" that refers to fire and water and their effects, to energies particles as well as liquids: to a world with unstable outlines, a world in flux. The family of English words that descend from it reflects the range of meanings of "fusus". They include 'infuse', 'con-fuse' and 'transfuse' from 'melted or joined' with 'de-fuse' part of the same

* Supported in Part by Natural Sciences and Engineering Research Council of Canada, Nortel Networks and Information Intelligence Co.

branch ...(from) remove; diffuse' from 'spread out, extended' and 'profuse' and 'effuse' from 'pour out in abundance'. All these words are formed by the addition of a prefix to 'fusus', to limit or constrain the fuzzy range of meanings of fusus to a more specific (but still somewhat fuzzy) meaning.

In general, initially, "Zadeh used 'fuzzy' to apply to categories of language and thought, [but] not to the nature of phenomena. However, recently, he has developed a typology of edges (boundaries or borders in phenomena, not in categories) in terms of categories that describe them" (Hodge, 2001).

This typology of edges, in terms of categories, is introduced in "Toward a Theory of Information Granulation..." (1997). For example, Zadeh provides description of how humans perceive and identify in categorical terms such natural phenomena as nose, cheek, etc., of a human face with fuzzy boundaries.

Edges, which are distinct, can be described with fuzzy membership functions. This is the base of Type 1 fuzzy theory. But the edges that can be described with fuzzy membership categories can be represented with Type 2 or Type 3, etc., fuzzy theories. That is "crisp" edges can be represented with Type 1 membership functions, where as "fuzzy" edges can be represented with Type 2 or higher levels of fuzziness.

These Type 2 or higher levels of fuzzy membership functions represent membership of membership values, i.e., imprecision of imprecise membership functions and contain uncertainty associated with membership values for varying shades of meaning in words. In real life human communication words have imprecise meanings, sometime known as vagueness, ambiguity, etc., even in a given context such as human decision processes and descriptions of natural phenomena.

In particular, Type 2, and higher levels of fuzzy theory expose risks associated with managerial decision making in OR and MS studies. (Türkşen, 1986, 2001).

Contributions

Let me next attempt to articulate Zadeh's scientific and philosophical contributions in a bit more detail.

In his seminal paper "Fuzzy Sets" (1965), Zadeh introduced the notion of "...a continuum of grades of membership" along with the "complement", "containment", "union" and "intersection" operations with "Max-Min" and with "Algebraic sum and Product", as well as "convex combination of " fuzzy sets, "Fuzzy sets induced by mappings", and "separation of convex fuzzy sets".

Implicit within these introductory concepts are the relaxation of the "Law of Excluded Middle", LEM, and its dual the "Law of Contradiction", LC. This naturally is unacceptable in classical set theory. But, it is known throughout human struggle, one needs to break away from traditional ways of thinking for the discovery and development of novel theories. L.A. Zadeh, in fact, broke away from the essential axiom of the classical theory in 1965.

In "Probability Measures of Fuzzy Events", he introduces the notion of a fuzzy event with examples such as "It is a *warm* day", "X is *approximately* equal to 5", "in twenty tosses of a coin there are *several* more heads than tails". These expressions "are fuzzy because of the imprecision of the meaning of the underlined words" (1968). Moreover, he generalizes the mathematical expressions of mean, variance and entropy in probability theory to "the mean variance, ...(and) entropy of a fuzzy event...".

In "Decision-Making in a Fuzzy Environment", Bellman and Zadeh (1970), they introduce "...a decision process in which the goals and/or constraints, but not necessarily the system under control, are fuzzy in nature".

Furthermore, they illustrate a new framework "...by examples involving multistage decision processes...".

In "Similarity Relations and Fuzzy Orderings", Zadeh (1971) discuss the *similarity relation* to be "a fuzzy relation which is reflexive, symmetric and transitive" together with "*fuzzy linear ordering* ... fuzzy preordering, (and) fuzzy weak ordering...".

Until the early'70's, Zadeh and his follower's essentially developed the foundations of fuzzy mathematics of fuzzy sets. This may be considered the first stage of fuzzy theory which had no practical applications.

In "Outline off a New Approach to Analysis of Complex Systems and Decision Process" (1973), he introduces the concepts of "A linguistic variable..." and "the compositional rule of Inference...". He then stresses that this new "...approach provides an

approximate and yet effective means of describing the behaviour of systems which are too complex or too ill-defined to admit of precise mathematical analysis. Its main applications lie in economics, management science, artificial intelligence, psychology, linguistics, information retrieval, medicine, biology and other fields in which the dominant role is played by the animate rather than inanimate behaviour of system constituents". In this paper, we are also introduced to the notion of "Computation of the Meaning of Values of a Linguistic Variables".

"The Outline of a New Approach..." is a landmark paper. It is on the bases of this paper, Mamdani and Assilian (1975) developed first practical laboratory version of an applied fuzzy system model and its use in industrial fuzzy control. This gave rise to the wide spread "fuzzy control" application in electro-mechanical systems.

In "A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts" (1976), we read "The high standards of precision which prevail in mathematics, physics, chemistry, engineering and other 'hard' sciences stand in sharp contrast to the imprecision which pervades much of sociology, psychology, political science, history, philosophy, linguistics, anthropology, literature, art and related fields". In this paper, we also find the definitions of "fuzzy truth", as well as S and Π membership functions.

As well, in this paper, we find an exposition on the relation between classificational and attributional questions, their analytical representations together with a graphical interpretation that demonstrates "cylindrical extension" and "projection" which shows what we recently come to recognize as "projection anomaly" in fuzzy clustering techniques in fuzzy system modeling. (Uncu, Türkşen, 2001).

In "Fuzzy Sets as a Basis for a Theory of Possibility" (1978), he states that "...when our main concern is with the meaning of information-rather than with its measure (in Weiner and Shannon sense of the statistical theory of Communication) the proper framework for information analysis is possibilistic rather than probabilistic in nature...". In this paper, we are introduced to "the concept of a possibility distribution", "possibility measure", truth qualification", "probability qualification", etc.

In his celebrated papers, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning – I, II, III, (1975-1976), we read, "By a *linguistic variable* we mean a variable whose values are words or sentences in a natural or artificial language."

It is stated that "Given our veneration for what is precise, rigorous and quantitative, and our disdain for what is fuzzy, unrigorous and qualitative, it is not surprizing that the advent of digital computers... have proved highly effective in dealing with *mechanistic* systems, that is, with inanimate systems whose behaviour is governed by the laws of mechanics, physics, chemistry, and electromagnetism. Unfortunately, the same cannot be said about *humanistic* systems,...".

Professor Zadeh indicates that "...the ineffectiveness of computers in dealing with humanistic systems is a manifestation of what might be called the *principle of incompatibility* – a principle which asserts that high precision is incompatible with high complexity.

In this paper on "the Concept of a Linguistic Variable...", term sets are specified for *Age*, Appearance, Truth, and Probability, etc.

As well we find the notions of *interaction* and *non-interaction* of fuzzy sets, linguistic variables and their linguistic terms, the *extension principle*, *type n*, n = 2, 3,... fuzzy sets, *linguistic truth variables* and *fuzzy logic*, *Truth Tables* and *Linguistic Approximation*, *Linguistic Probabilities* and *their computations*, *composition rule of inference* which was later identified as GMP, etc.

In "A Theory of Approximate reasoning" (1979) rules of inference and approximate reasoning are further discussed in detail as projection principle, entailment principle, semantic equivalence, etc.

In "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems", (1983) which is dedicated to Prof. Eli Sanchez, it is stressed that "Management of uncertainty is an intrinsically important issue in the design of expert systems because much of the information in the knowledge base of a typical expert system is imprecise, incomplete or not totally reliable".

In this article, it is shown that there are effects of fuzziness in facts and rules. Types of proposition are discussed with inference in fuzzy logic. Translation rules are revisited. We are also introduced to

inference from quantified propositions", e.g., Q_1A's are B's, and Q_2 (A and B)'s are C's $\geq Q_1 \otimes Q_2$) A's are C's, etc.

Next, we find "Syllogistic Reasoning in Fuzzy Logic and Its Application to Usuality and Reasoning with dispositions" (1985). In this paper, Zadeh views fuzzy logic "...as a generalization of multivalued logic in that it provides a wider range of tools for dealing with uncertainty and imprecision in knowledge representation, inference, and decision analysis. Such topics as "intersection/product syllogism", dispositional modus ponens". In this paper, they are treated together with "fuzzy quantifiers", "compositionality", "robustness", and "usuality". In this context, we observe the use of Σ count (.) in many examples. Fuzzy syllogisms and reasoning with dispositions are exemplified with "MPR(Major Premise Reversibility) chain Syllogism", "Antecedent Conjunction Syllogism", "Consequent Conjunction Syllogism", etc.

These and other related issues are treated further in "A Computational Approach to Fuzzy Quantifiers in Natural Languages"(1983). Unfortunately, works in this area are very few, e.g., Narazaki and Türkşen. (1994)

Next, Zadeh proposes "A Theory of Commonsense Knowledge"(1984). He states, "The conventional knowledge representation techniques based on the use predicate calculus and related methods are not well-suited for the representation of commonsense knowledge because the predicates in propositions which represent commonsense knowledge do not, in general, have crisp denotations. For example, the proposition *Most Frenchmen are not tall* cannot be represented as a well-formed formula in predicate calculus because the sets which constitute the denotations of the predicate *tall* and the quantifier *most* in their respective universes of discourse are fuzzy rather than crisp. "Meaning representation" is further discussed with "Test-Score Semantic", "Composition of Elastic Constraints", together with "Rules pertaining to modification... composition... quantification" as well as "representation of dispositions" with "inference from dispositions" and the applications of "sigma count" and relative sigma count."

In a series of papers, "Fuzzy Logic=Computing with Words" (1996) "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic" (1997), and "From Computing with Numbers to Computing with Words –from manipulation of measurements to manipulation of perceptions" (2001), Zadeh writes "...the main contribution of fuzzy logic is a methodology for computing with words. No other methodology serves this purpose". He goes on to state "in its traditional sense, computing involves ... manipulation of numbers and symbols. By contrast, humans employ mostly words in computing and reasoning, arriving at conclusions expressed as words from premises expressed in a natural language or having the form of mental perception".

He then traces the origins of this development stating "The concept of CW is rooted in several papers starting with ["Outline of a New Approach..."(1973)] in which the concept of a linguistic variable and granulation were introduced. The concepts of fuzzy constraint and fuzzy constraint propagation were introduced in ["Calculus of Fuzzy Restrictions"(1975)], and developed more fully in ["A Theory of Approximate Reasoning"(1979)], etc.

In these works, there are schemas that show how one gets started with the notion of granulation and first arrive at information and action granules and then apply divide and conquer principle. Next, one identifies crisp and fuzzy information granules, CIG, FIG. The examples of CIG are given as "time → years → months → weeks → days → ..." and FIG as "age → very young + young + middle-aged + old + very old".

As well examples of mental and physical granulation are demonstrated as – mental granulation: "body → head + neck + left arm + chest + right arm + ..." and physical granulation: "speach, walking, eating."

The generalized constraint that was introduced in previous papers are re stated as "X isr R, where isr (pronounced ezar) is a variable copula which defines the way in which R constrains X."

The role of R in relation to X in defined by the value of the discrete variable r where r could take on values:

" e: equal (abbreviated to =)
d: disjunctive (possibilistic)
(abbreviated to blank)
c: conjunctive
p: probabilistic
λ: probabilistic value
u: usuality

rs: random set
rsf: random fuzzy set
fg: fuzzy graph
ps: rough set (Pawlak set)"

Next, we are exposed to fuzzy constraint propogation and the rules of inference in fuzzy logic under the headings of: Conjunctive Rule 1, Conjunctive Rule 2, Disjunctive Rule 1, Disjunctive Rule 2, Conjunctive Rule, Projective Rule, Subjective Rule, as well as Derived Rules under the headings of: Compositional Rules, Extension Principle (Mapping Rule), Inverse Mapping Rule, Generalized Modus Ponens, Generalized Extension Principle, Syllogistic Rule, Constraint Modification Rule, etc. Once again we are treated will the example of "Balls in a Box".

"A box constrains ten balls of various sizes of which several are large and a few are small. What is the probability that a ball drawn at random is neither large nor small?"

"To be able to answer this question, it is necessary to be able to define the meanings of *large*, *small*, *several large balls*, *few small balls*, and *neither large nor small*. This is a problem in semantics, which falls outside the probabilistic theory, neurocomputing and other methodologies.

Conclusion

Zadeh, thus concludes "In our quest for ... Machine intelligence (high MIQ), we are developing a better understanding of the fundamental importance of the remarkable human capacity to perform a wide variety of physical and mental tasks without any measurements and any computations. Underlying this remarkable capability is the brain's crucial ability to manipulate perceptions – perceptions of distance, size, weight, force, color, numbers, likelihood, truth and other characteristic of physical and mental objects. A basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy."..."Humans employ words to describe perceptions ... (in this regard) ... manipulation of perceptions is reduced to computing with words... In coming years, computing with words and perceptions is likely to emerge as an important direction in Science and Technology."

References

[1] R.E. Bellman and L.A. Zadeh, "Decision-Making in a Fuzzy Environment", *Management Science*, 17:4 (December 1970) 141-164.

[2] R. Hodge, "Key Terms in Fuzzy Logic: Deep Roots and New Understandings", *Fuzzy Sets and Systems*, (2001) (Submitted).

[3] E.H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", Int. J. of Machine Studies, 7,1 (1975) 1-13.

[4] H. Narazaki and I.B. Türkşen, "An Integrated Approach for Sylogistic Reasoning and Knowledge Consistency Level Maintenance", IEEE-Trans on SMC, 24, 4 (1994) 548-563.

[5] I.B. Türkşen, "Interval-Valued Fuzzy sets Based on Normal Forms", *Fuzzy Sets and Systems*, 20 (1986) 191-210.

[6] I.B. Türkşen, "Sources, Measurements and Models of Type 2 Fuzziness in the New Millennium", in: Fuzziness in the New Millennium, V. Dimitrov (ed), Springer Verlag (2001) (in press).

[7] O. Uncu and I.B. Türkşen, "A Novel Fuzzy System Modeling Approach: Multidimensional Structure Identification and Inference", 10th IEEE – Int. Conference on Fuzzy Systems, Dec. 2-5'01, Melbourne, Australia (2001) (Submitted).

[8] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, Vol.8, New York: Academic Press (1965), 338-353.

[9] L.A. Zadeh, "Probability Measures of Fuzzy Events", *J.Math. Analysis and Appl.*, 10 (1968) 421-427

[10] L.A. Zadeh, "Similarity Relations and Fuzzy Ordering", *Information Sciences*, 3 (1971) 177-200.

[11] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. Systems*, Man, and Cybernetics, SMC-3 (1973) 28-44.

[12] L.A. Zadeh, "A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts", *Int. J. Man-Machine Studies*, 8 (1976) 249-291.

[13] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, (1978) 3-28.

[14] L.A. Zadeh, "The Concept of a Linguistic Variable and its Application to Approximate Reasoning", Parts 1 and 2, *Information Sciences*, 8 (1975) 199-249, 301-357.

[15] L.A. Zadeh, "Calculus of Fuzzy Restrictions", Fuzzy Sets and Their Applications to Cognitive and Decision Progresses, L.A. Zadeh, K.S Fu, K. Tanaka and M. Shimura (eds), Academic Press, New York, (1975) 1-39.

[16] L.A. Zadeh, "The Concept of a Linguistic Variable and its Application to Approximate Reasoning", Parts 3, *Information Sciences*, 9 (1976) 43-80.

[17] L.A. Zadeh, "A Theory of Approximate Reasoning", *in J. Hayes, D. Michie, and L.I. Mikulich (eds) Machine Intelligence*, Halstead Press, New York, Vol. 9 (1979) 149-194.

[18] L.A. Zadeh, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems", *Fuzzy Sets and Systems*, 11 (1983) 199-227.

[19] L.A. Zadeh, "Syllogistic Reasoning in Fuzzy Logic and its Application to Usuality and Reasoning with Dispositions", *IEEE-Trans.SMC*, 15 (1985) 754-763.

[20] L.A. Zadeh, "A Computational Approach to Fuzzy Quantifiers in Natural Language", *Comp. and Maths. with Appls.*, 9 (1983) 149-184.

[21] L.A. Zadeh, "A Theory of Commonsense Knowledge", *in H.J. Skala, S. Termini, and E. Trillas, Eds., Aspects of Vagueness*, Dodrecht: D. Reidel (1984) 257-296.

[22] L.A. Zadeh, "Fuzzy Logic = Computing With Words", *IEEE-Trans on Fuzzy Systems*, 4, 2(1996), 103-111.

[23] L.A. Zadeh, "Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic", *Fuzzy Sets and Systems*, 90 (1997) 111-127.

[24] L.A. Zadeh, "From Computing with Numbers to Computing with Words - From Manipulation of Measurements to Manipulation of Perceptions", in: P.P. Wang(ed.) *Computing With Words*, Wiley Series on Intelligent Systems, Wiley and Sons, New York, (2001) 35-68.

On Supporting Complex Relationships and Knowledge Discovery in the Semantic Web

Amit Sheth
Large Scale Distributed Information Systems Lab
University of Georgia

Abstract
BISC Plenary Talk, Berkeley CA, August 16, 2001

Current research in Semantic Web focuses [Lee et al 2001] on semantic tagging of Web-based content, use of ontologies to facilitate shared understanding, and use of multi-agent architectures. Significant work is being devoted to developing DAML+OIL markup supporting more semantic annotation, and description logic based reasoning. Complementing these activities and building upon knowledge-based systems and information integration research, we are investigating following capabilities:

- involvement of all types, formats and media of information and corresponding logical integration of information (not only Web pages, but also dynamic and perishable content, content feeds, corporate repositories, and various digital media)
- more comprehensive semantic description of information involving domain modeling and use of multiple ontologies, and
- support for representation of, and reasoning involving more complex relationships (beyond is-a and subsumption)
- support for human-assisted knowledge discovery from autonomous information sources and all types of content

The InfoQuilt system [IQ] uses a multi-agent information brokering architecture to research and prototype the above capabilities. In this talk, I will present the support for complex relationship and examples of knowledge discovery. Complex relationships involve attempt to model cause-and-effect relationships involving multiple parameters that are often found in natural phenomena.

[Lee et al 2001] TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA, The Semantic Web: http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[IQ] http://lsdis.cs.uga.edu/proj/iq/iq_pub.html

Fuzzy Logic and Integrated Network Management

Seyed A. Shahrestani

School of Computing and Information Technology
University of Western Sydney
Penrith Campus, Locked Bag 1797
PENRITH SOUTH DC NSW 1797
AUSTRALIA

seyed@ieee.org

Abstract

Modern networks contain a large number of physical and logical elements that must be managed. The increasing speed and complexity of the networks require radical changes in network management approaches. One of the prerequisites for successful management of these complex systems is the ability to handle large amounts of information. The information may contain incoherent, missing, or unreliable data that need to be filtered and processed. In this respect, AI techniques offer a number of appealing solutions that merit to be considered. In particular, the power of fuzzy logic in handling uncertainties and its capability to coordinate and manage several models and rules make it an appropriate choice for taking up a significant part in network management. This paper focuses on this topic, highlighting in a conceptual manner the role of fuzzy logic in identifying or improving solutions to network management problems. Some specific application areas of functional importance that demonstrate the effectiveness of fuzzy logic in improved management of the networks are also discussed.

1. Introduction

The ever-increasing complexity of the networks has some profound technical implications for management systems. Modern networked systems result in an overwhelming amount of data because of comprehensive monitoring abilities. Conventional computer applications provide a degree of automation to process and filter the data to identify relevant information, but human interactions remain essential, as the data is often incomplete and conflicting. In principle, artificial intelligence (AI) techniques could limit the need for human intervention [1]. In particular, several characteristics of fuzzy logic make it an effective approach for use in an integrated network management environment. For instance, its flexibility in handling uncertainties and its capability to coordinate and manage several models and rules can be mentioned. Hence, the application of fuzzy logic to network management is rational and merits to be studied in some detail.

The main objective of this work is to discuss the ways that fuzzy logic can be used in an integrated network management environment. This is achieved in the remainder of this paper by using the following structure. Section 2 presents the integrated network management environment, highlighting why AI and in particular fuzzy logic should be considered. Section 3 gives a brief review of fuzzy logic. Noting that fuzzy logic can also be used to improve most other AI approaches applied to network management problems, a very brief overview of some other AI techniques is also given in this section. Section 4 focuses on specific applications of fuzzy logic in network management. The concluding remarks are given in Section 5.

From a broad point of view, the ability to handle huge amounts of information is a prerequisite for management of complex systems. The experience gained on a problem represents the knowledge that can be of value in the future. Information retrieval (IR) tools are the very bases for any process that deals with large databases. This is the case even if they only support only data collection and the actual task of extracting the information is left to the user (human being or software agent). Expressiveness and adaptivity are fundamental features for a data model. The abstraction associated with an object should capture all its peculiarities in an easily manageable representation. In a standard IR context, uncertainty pervades the behavior of both the system and the users. To undertake uncertainty and adaptivity problems simultaneously, fuzzy logic offers excellent solutions.

Among the other functional areas that fuzzy logic can be of great value, fault management can be mentioned. For instance, although the case-based reasoning (CBR) paradigm is reported to give good solutions to alarm correlation problem [3], its high sensitivity to the accuracy of knowledge description should not be ignored. Uncertainty permeates the entire diagnostic process and its management is a fundamental issue in actual diagnostic systems. The information regarding the context of encountered problems and the type of models that can be built to represent them are among crucial aspects of a diagnostic system. While traditionally the main components used in the definition of a context are observations (facts), the data on relevance and confidence may add precious information [4]. The latter piece of information can be easily amended and handled by fuzzy logic based approaches.

2. Integrated Network Management

The need for distribution of network management is already well established. This is evident by the approaches such as definitions of remote monitoring MIB (RMON) or mid-level manager MIB, for example. In general, the integrated network management is concerned with a combination of issues relating to equipment, services, applications, and enterprise management. In this context, various new requirements need to be met by network management solutions. Some of these requirements are mentioned in this section, while some possible enabling approaches for complying with them are discussed in later parts.

Help desk systems are designed to provide customer support through a range of different technology and Information Retrieval (IR) tools play a fundamental role in this activity. Efficiency and effectiveness in data retrieval being crucial for the overall problem solution process heavily depend on the abstraction models. The abstraction associated with an object should capture all its peculiarities in an easily manageable representation. Identification of relevant features of achieving an object abstraction is a complex task and presence of uncertainties makes this task even harder [5].

For diagnosis purposes, focusing on case-based reasoning (CBR) paradigm, models that capture the relevance and uncertainty of information in a dynamic manner are essential. This is a requirement for models used in both diagnostic knowledge and processes. Based on such models a conversational CBR shell implementing nearest-neighbor (NN) retrieval mechanisms for example can then be utilized to achieve high precision case-retrieval [4].

Distributed applications are evolving towards compositions of modular software components with user interfaces based on web browsers. Each of these components provides well-defined services that interact with other components via network. The increase in the complexity of distribution makes it more difficult to manage the end-to-end Quality-of-Service (QoS). The challenge derives in part from the need for interaction of different management scopes of network and computing domains. A management system deployed to diagnose QoS de-gradation should address two major issues. First, to measure the performance of applications, it needs a low-overhead, scalable system for measuring software components. Second, the performance management system must monitor selected measurements, diagnose QoS degradation, adapt to the environment and integrate with network management systems [6].

3. Fuzzy Logic and Artificial Intelligence

The interest in building machines and systems with human-like capabilities has lead to considerable research activity and results. Important features of human capabilities that researchers are interested in implementing in artificial systems include learning, adaptability, self-organization, cognition (and recognition), reasoning, planning, decision-making, action, and the like. All of which are related to `intelligence. These research activities form the core of Artificial Intelligence (AI) [7]. To achieve higher levels of automation, a number of AI techniques have already been applied to network management problems [1].

Although the focus of this work is on fuzzy logic applications in network management, it can be noted that fuzzy logic can be used to improve most other AI approaches, e.g. knowledge presentation in expert systems. Therefore, while this section gives a more elaborate treatment to fuzzy logic, a very brief overview of some other AI approaches that are found to be of value in network management is also given. This allows for discussing the possible improvements of utilizing fuzzy logic in other AI techniques applied to network management problems in later parts.

3.1. Fuzzy Logic

The subject of fuzzy logic is the representation of imprecise descriptions and uncertainties in a logical manner. Many artificial intelligence based systems are mainly dependent on knowledge bases or input/output descriptions of the operation, rather than on deterministic models. Inadequacies in the knowledge base, insufficiency or unreliability of data on the particular object under consideration, or stochastic relations between propositions may lead to uncertainty. In expert systems, lack of consensus among experts can also be considered as uncertainty. In addition, humans (operators, experts...) prefer to think and reason qualitatively, which leads to imprecise descriptions, models, and required actions. Zadeh introduced the calculus of fuzzy logic as a means for representing imprecise propositions (in a natural language) as non-crisp, fuzzy constraints on a variable [8].

3.2. Knowledge Based and Expert Systems

Knowledge Based Systems (KBS) are modular structures in which the knowledge is separate from the inference procedure. Knowledge may be utilized in many forms, e.g. collection of facts, heuristics, common sense, etc. When the knowledge is acquired from (and represents) some particular domain expert, the system is considered an expert system. In many cases, knowledge is represented by production rules or specification of the conditions that must be satisfied for the rule to become applicable. Also included are the provisions of what should be done in case a rule is activated. Production rules are IF--THEN statements; a 'conclusion' is arrived at, upon the establishment of validity of a 'premise' or a number of premises [9].

Rule-based systems are popular in AI because rules are easy to understand and readily testable. Each rule can be considered independent of the others, allowing for continual updating and incremental construction of the AI programs. Broadly speaking, systems relying on heuristic rules are considered brittle. When a new situation falls outside the rules, they are unable to function and new rules have to be generated. Thus, a very large knowledge base must be created and stored for retrieval purposes. In general, heuristic rules are hard to come up with and are always incomplete. The rules are usually inconsistent; i.e. no two experts come up with the same set [9].

3.3. Artificial Neural Networks

Artificial Neural Networks (ANNs) are dense parallel layers of simple computational nodes. The strengths of the links between the nodes are defined as connection weights. In most cases, one input layer, one output layer, and two internal (hidden) layers will be considered adequate to solve most problems [10]. This is considered as a Multi-Layer Perceptron (MLP) and is widely popular. The connection weights are usually adapted during the training period by back-propagation of errors, which results in a feed-forward network.

3.4. Pattern Recognition

Pattern recognition is the ability to perceive structure in some data; it is one of the aspects common to all AI methods. The raw input data is pre-processed to form a pattern. A pattern is an extract of information regarding various characteristics or features of an object, state of a system, etc. Patterns either implicitly or explicitly contain names and values of features, and if they exist, relationships among features. The entire act of recognition can be carried out in two steps. In the first step, a particular manifestation of an object is described in terms of suitably selected features. The second step, which is much easier than the first one, is to define and implement an unambiguous mapping of these features into class-membership space [10].

Patterns whose feature values are real numbers can be viewed as vectors in n-dimensional space, where n is the number of features in each pattern. With this representation, each pattern corresponds to a point in the n-dimensional metric feature space. In such a space, distance between two points indicates similarities (or differences) of the corresponding two patterns. Partitioning the feature space by any of the many available methods, e.g. maximum likelihood, K-nearest neighbors, decision surfaces and discriminate functions then carry out the actual classification.

3.5. Case-based Reasoning

Case-based reasoning (CBR) paradigm [11] starts from the assumption that cognitive process is structured as a cycle. The first step is to gather some knowledge, then the knowledge is used to solve a problem and, depending on the result, one may decide to keep track of the new experience. Experience is accumulated either by adding new information or by adapting the existing knowledge. The idea is to solve a problem with the existing skills and, at the same time, improving these skills for future use. From the actual implementation point of view, the focus is on how to

aggregate and store the information (cases) and how to retrieve them. The solution of a problem depends on the ability of the system to retrieve similar cases for which a solution is already known. The more common retrieval techniques are inductive retrieval and nearest neighbor [4].

4. Fuzzy Logic and Network Management

Several characteristics of fuzzy logic make it an effective approach for use in an integrated network management environment. In particular, its flexibility in handling uncertainties and its capability to manage several models and rules are of great value. We start this section by giving a broad view of integrated network management tasks that fuzzy logic is suitable for. We then proceed to discussing some specific application areas.

4.1. Classification of Tasks in Management Layers

Consider the hierarchical model for network management shown in Table 1. As this model in its essential form and the functions of various layers has been discussed in [1], we do not elaborate on them.

Layer	*Tasks/ Requirements*	*Information Flow*	*Control Flow*
Business	Decision Support	↑	↓
Service	Information Retrieval (IR)	↑	↓
Network	Resource Management	↑	↓
Element	Fast Control (Connection Admission)	↑	↓

Table 1, Management layers (adapted from [1])

At the highest layer, the problems can be associated with an overwhelming amount of data. The AI techniques should process the data and present only the relevant information by acting as a decision-support tool. At this layer, the response time is important but not critical. This type of task is well suited for techniques that implement search techniques, e.g. genetic algorithm. Also, model-based expert systems can be used to hide the network complexity behind several abstraction levels [1]. In this context, fuzzy logic can be used to handle model/data uncertainties and ambiguities while interpolating between (possibly) several emerging models. The resultant aggregate model will also have some degree of confidence attached to it that will assist the operators in dealing with the presented information.

While in the next section we take a closer look at some of the tasks in the service layer, it can be noted that the above discussion holds for both service and network management layers. For example, AI based network management systems that deal with the problems at network layer, are mostly based upon expert system techniques [12]. At the elements management layer though, the time response becomes the critical factor. It must be noted that fuzzy logic (and ANN) implementations can be hardware-based to achieve fast response (while most other AI approaches are software-based). At this layer, the environment changes rapidly and a slow solution will become irrelevant. The available information is often incomplete and incoherent [1]. The fuzzy logic character in dealing with uncertainties along with its capabilities in handling several sources of information (via interpolations and taking a supervisory role), make it an excellent choice for management support at this layer.

4.2. Advanced Help Desk

In a competitive business environment, customer satisfaction is a vital objective for many companies: high-quality products and high-quality customer service are two strategic aspects. In this context, help desk systems play an important role providing customer support and functions like change, configuration and asset management. The two main components of a help desk system are the front-end and the back-end ones: the former manages the interaction with customers while the latter deals with information retrieval (IR) issues.

The core functionality is the retrieval of data from a database whose abstraction matches the description of an ideal object, inferred from a query. Implementation issues are critical both for the overall performance of the system and the accuracy of the retrieved information. Customers usually provide data with different degrees of confidence depending on how that information has been collected. Current IR tools do not explicitly model the uncertainty associated with information but they mix the measure of relevance associated to information with the relative measure of

confidence. They don't even manage the feedback provided by users about the accuracy and usefulness of the retrieved solutions. An effective use of that information is the key to enable a process of system adaptation. The explicit management of relevance and confidence on information, integrated with an adaptivity process is the key factor for improving the retrieval precision of a help desk system [5].

Fuzzy logic can be used to form an integrated approach to both uncertainty and adaptivity problems. Keywords are still at the base of the abstraction model, but together with relevance information, they will be enriched with information on confidence degree implemented using membership functions.

4.3. Network Diagnostic Systems

The precise identification of the context in which a problem occurs is fundamental in order to diagnose its causes and, eventually to fix it. The more accurate the information on the context, the more precise the diagnosis can be. The goal of a diagnostic system is to maintain and extract from an information base facts, rules and any other type of indications that can help in identifying the problems. The starting point is a set of facts (observations) but the same fact may have different relevance in different contexts. Collecting information on the relevance of facts allows being more precise in the retrieval (matching) process and precision is fundamental when the dimension of the system knowledge base grows.

The problem is that, while observations are hardly disputable, the relevance associated with them may depend on the experience of the observer. Traditionally the confidence and relevance are empirically merged in a single value and this may corrupt the information. A different (or complementary) solution is to explicitly model and manage the uncertainty associated with the observation. The idea is to capture in this way the fact that there is something missing even if we don't know what it is. Certainty may be reinforced or reduced and adaptivity plays a fundamental role in this kind of process [4]. This type of modeling and reinforcement can be best achieved by incorporating fuzzy sets and fuzzy logic.

Furthermore, the association of confidence with the information through fuzzy sets to establish explicit uncertainty models may prove to be beneficial in other respects as well [13]. For example, the fact that confidence values for a symptom are low suggests that a clear understanding of its meaning does not exist and it may need to be investigated more carefully. Looking at the confidence distribution of different symptoms of the same case, we can obtain indications on the reliability of the associated diagnosis proposals: if there is uncertainty on the causes of a problem (case) we may be more careful considering the proposed diagnosis. Qualitative analysis of case descriptors may give indications on the system users, their needs and their problems. This extra layer of information provides a starting point for a more user focused diagnostic system where effectiveness derives not only from technological issues but also from a clearer understanding of the user (human or software agent) [4].

4.4. Quality-of-Service

Distributed applications are increasingly composed of modular off-the-shelf software components and custom code. Current management systems that monitor thresholds and trigger alarms rely on correct interpretation by the operator to determine causal interactions. This approach does not scale as the number of thresholds and alarms increase. For a scaleable solution, a management system should be able to monitor, diagnose and reconfigure application components to ensure that user-level Quality-of-Service (QoS) goals are maintained. The management system must be pro-active and coordinate with existing network management systems. Emerging problems are corrected before QoS failures occur. The use of knowledge-based systems is ideal for management of these distributed applications [6]. These systems can conceptually be significantly enhanced by incorporation of fuzzy logic. Such incorporation will improve diagnostic rules that are more capable of handling ambiguity and incomplete information.

5. Concluding Remarks

To cope with the increasing complexity of the networks, their management systems have become highly complicated as well. The management system must deal with an overwhelming amount of data that may be incoherent and inconsistent or unreliable. Compared to more conventional techniques, AI approaches are more suitable for this type of tasks. In particular, the capabilities of fuzzy logic in handling vague concepts or systems with uncertainties are of prime significance. We described several ways that fuzzy logic can be used in identifying or improving the

solutions to problems encountered in an integrated network management environment. In this work, in addition to a conceptual discussion of this topic, several areas with functional importance are also considered. For instance, it is noted that a key aspect of help desk services is related to information retrieval where uncertainty in data is a major peculiarity. A comprehensive solution for uncertainty management can be based on the notion of fuzzy sets in which relevance and confidence is used to enrich the descriptive power of keyword paradigm. In the case of diagnostic systems, fuzzy logic can be used for the explicit modeling of the uncertainty that in turn leads to an actual improvement in terms of case-selection precision.

References

[1] C. Muller, P. Veitch, E. H. Magill and D. G. Smith, "Emerging AI techniques for network management," in *Proc. IEEE GLOBECOM '95*, 1995, pp. 116-120.

[2] D. Benech, "Intelligent agents for system management," in *Proc. Distributed Systems: Operation and management*, 1996.

[3] L. Lewis and P. Kaikini, "An approach to the alarm correlation problem using inductive modeling technology," Technical Note ctron-lml-93-03, Cabletron Systems R&D Center, Merrimack, 1993.

[4] G. Piccinelli, "Uncertainty modeling in diagnostic systems: An adaptive solution," Technical Report HPL-98-37, HP Laboratories, Bristol, 1998. Available at http://www.hpl.hp.com/techreports/98/HPL-98-37.html

[5] G. Piccinelli and M. C. Mont, "Fuzzy-set based information retrieval for advanced help desk," Technical Report HPL-98-65, HP Laboratories, Bristol, 1998. Available at http://www.hpl.hp.com/techreports/98/HPL-98-65.html

[6] J. Martinka, J. Pruyne and M. Jain "Quality-of-service measurements with model-based management for networked applications," Technical Report HPL-97-167, HP Laboratories, Palo Alto, 1998. Available at http://www.hpl.hp.com/techreports/97/HPL-97-167R1.html

[7] P. H. Winston, *Artificial Intelligence*. Addison-Wesley, USA, 1984.

[8] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[9] S. A. Shahrestani, H. Yee, and J. Ypsilantis, "Adaptive recognition by specialized grouping of classes," in *Proc. 4th IEEE Conference on Control Applications*, Albany, New York, 1995, pp. 637-642.

[10] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Addison Wesley, USA, 1989.

[11] K.D. Althoff and S. Wess, "Case-based reasoning and expert system development," in *Contemporary Knowledge Engineering and Cognition*, Springer-Verlag, USA, 1991.

[12] N. Nuansri, T. S. Dillon and S. Singh, "An application of neural network and rule-based system for network management," in *Proc. 30th Hawaii International Conference on System Sciences*, 1997, pp. 474-483.

[13] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems," *Fuzzy Sets and Systems*, vol. 11, pp. 199-228, 1983.

Personalized Library Search Agents Using Data Mining Techniques

Yu Tang and Yanqing Zhang
Department of Computer Science
Georgia State University Atlanta,
GA 30303 USA
yzhang@cs.gsu.edu

Abstract

In a traditional library system, a search result will be exactly the same if different users with different preferences use the same search criteria. It is obvious that the traditional library system cannot provide high QoS (Quality of Service) for different users. To solve this problem, a personalized library search agent technique is proposed based on data mining technology. By mining the training data sets, the attributes that are related to borrowing tendency of users are analyzed, and then users are divided into different groups. In addition, the SLIQ (Supervised Learning In Quest) algorithm is used to mine data. The Web-based personalized library search agent system is implemented successfully using JSP (JavaServer Pages) that call the java class and ServletExac JSP server, and Oracle 8.1.6. Simulations have shown the personalized library search agent can generate personalized search results based on users' preferences and usage. Therefore, a user can use the personalized library search agent to get good and quick result. In the future, advanced intelligent techniques such as soft computing, granular computing, and distributed computational intelligence will be used in the personalized library search agent system to continue to improve QoS of a library system and other information systems.

1. INTRIDUCTION

Currently, a Web-based library provides a lot of on-line service. People can search what they want through the Internet. Using the traditional search agent, every user is treated as the same way even if they have the different backgrounds – they have the different job, different favorite, different age and so on. But for the system, they have no difference, the search result are exactly same. Sometimes, such system can't provide high quality of service. Suppose a user is interested in graphics, he (or she) wants to find some book about java used in graphics. If he only types 'java' as keyword to search for books, the results displayed on the screen may begin with the books about java used in other fields such as java servlet, java server page and java language and so on, which he doesn't like. In this case, he has to take more time to find the books that he really needs. In the worst case – the books he needs appear on the end of the list – he maybe looses patience before he finds the books. It is obvious that the traditional searching system is time-consuming. To solve this problem, in the traditional library system, user has to give more detail information to narrow the search result. The goal is to develop a personalized search agent to enable users get the different sequence of search result according to their personality. Users can get the different sequence of search results even if the same search criteria are used. For example, a user belongs to group A. In this group, users tend to borrow books about network. Suppose he types 'java' as keyword to search for books, the books listed in the firstly page will be the books about network with 'java' in their titles and then the other books will display. By this way, users can find what they want quickly than the traditional way. Our approach is based on data mining techniques [2][3][5][6][8][9].

In this paper, a library search agent using data mining techniques is proposed. Since every user has his own personality and borrowing history, querying the same database by the same way will get the different results that may be meaningful and useful for the given user. Employing the data mining techniques, users can be divided into different groups, by this way, the system not only can provide the default list of books for every user according to the group that the user assorted, buts also it can provide different sequence of search result for different kind of users. The advantage of using such system is time saving and providing more satisfying service.

2. DATA MINING OVERVIEW

In the past three decades, statistics is used to analyze the collect data. But there is a drawback: the statistics used to analyze data starts with a hypothesis about the relationship among the data attributes, and then prove or disprove that hypothesis. If the data with a lot of attributes, this hypothesis-and –test methodology is time spending.

Another element makes the thing worse: with the development of techniques, the ability of the computer to store data is increasing. We can now store and query terabytes and even petabytes of data in one management system. The explosion of stored data requires an effective way to analysis data and to get the useful and meaningful information. It is clearly that using the statistics to analyze such massive amount of data is impossible.

For these reasons, we need to develop a new means to analysis data. Fortunately, power of computation gets great improvements while the increasing of the power of store. Meanwhile, artificial intelligence (AI) also develops. The algorithm of AI is opposed to statistical techniques; it can automatically analyze data and build data models that make us to understand the relationships among the attributes and class of data. This algorithm employs "test-and-hypothesize" paradigm instead of "hypothesize-and-test" that used in statistics.

Data mining process is interactive and iterative, it often starts with a large, arbitrary data set and with as few assumptions as possible. The initial data are treated as if there is no information available, the system must extract potential rules or patterns from that data, and then use algorithm to choose among them. This technique that used to get the important information is data modeling.

Modeling is simply the act of building a model in one situation where you know the answer and then applying it to another situation that you don't know [6]. The way that computer built the model is like the way that people build the model. First, computer is loaded a lot of data include variety of situations and their results, and then the system runs through all of data and extracts the characteristics of the data; finally, it builds the model by using this information. Once the model is built, it can be used to give the answer for the similar situations.

Data mining analyzes the relationships and patterns; it implements the any of the following types of function:

Classification: Stored data items are mapped or classified into several predetermined exclusive groups by a function. The members in the same group are as "close" as possible to each other and the members in the different group are as "far" as possible from one another where distance is measured with respect to specific variable(s) that the system try to predict [6].

Regression: Stored data items are mapped into a real-valued prediction variable by a particular function.

Clustering: Stored data items are divided into different groups according to the logical relationship or consumer preferences. The members in the same group are as "close" as possible each other and the members in the different group are as "far" as possible from one another.

Summarization: A report/documentation or a compact description for a subset of data is consolidated.

Dependency modeling: A model that describes significant dependencies between variables is found by the particular methods. It exists at two levels: the structure level and the quantitative level.

Change and deviation detection: The significant changes in the data from the historic pattern or normative values are discovered.

Data mining process includes many steps. Brachman and Anand gave a practical view of such process [1]. The main steps are following:

1. *Analyzing problem* This step involves analyzing the business problem, understanding the application domain, the relevant prior knowledge, and what the result of data mining the end-user wants to get.
2. *Preparing data* This step involves creating a target data set, data cleaning and preprocessing, data reduction and projecting. In this step, system will select a data set or a subset of variables or data samples, then it will have some basic operations on the selected data such as noise removing, necessary information collecting, and will transform the selected data to the format required by the data mining algorithms.
3. *Choosing the data mining task* In this step, the aim of data mining process will be decided, the possible aims of the process could be classification, regression, clustering, or others.
4. *Choosing the data mining algorithms* In this step, the appropriate algorithm for searching for the pattern will be selected. It includes selecting the

appropriate model and appropriate parameters that may match the particular data mining method.

5. *Generating pattern* In this step, system will generate the pattern by using rule induction (automatic or interactive) and the selected algorithm. The pattern could be in a particular representational form or a set of such representations: classification rules or trees, regression, clustering or others.

6. *Interpreting pattern* In this step, pattern will be validated and interpreted, it is possible to return to the previous step for further iteration.

7. *Consolidating knowledge* In this step, pattern will be deployed, and the guideline or reports will be produced. The related knowledge will be incorporated into the real-world performance system, or simply reported to interested parties.

8. *Monitoring pattern* This step assures that data mining strategy is correct. The historic patterns are regularly monitored against new data to detect the change in this pattern as early as possible.

3. CLASSIFICATION ALGORITHMS

3.1 Algorithms

ID3 algorithm: It is a decision tree building algorithm. It determines the classification of data by testing the value of the properties and builds the tree in a top down fashion. It is a recursively process.

C4.5 algorithm: It is an algorithm that recursively partitions the given data set to generate a classification decision tree. It considers the entire possible test that can split the data set and then select the best test. The decision tree uses Depth-first strategy. This algorithm was proposed first by Quinlan in 1993.

SLIQ(Supervise Learning In Quest) algorithm: It is a decision tree classifier designed to classify large training data [1]. It uses a pre-sorting technique in the tree-growth phase. The decision tree uses Breadth-first strategy. This algorithm was proposed and developed by IBM's Quest project team. The details of this algorithm will be introduced in the next chapter.

Naïve-Bayes algorithm: It is a simple induction algorithm. It assumes a conditional independence model of attributes given the label. It was firstly proposed by Good in 1965 and developed by Domingos and Pazzani.

Nearest-neighbor algorithm: It is a classical algorithm. It has options for setting, normalizations and editing. It was firstly proposed by Dasarathy in 1990 and developed by Aha in 1992 and Wettschereck in 1994.

Lazy decision tree algorithm: It is a tree building algorithm. It builds the "best" decision tree for every test instance. This algorithm was proposed by Friedman, Kohavi and Yun in 1996.

Decision table algorithm: It is a simple but useful algorithm. It uses a simple lookup table to select the feature subset.

The classification algorithms have much in common with traditional work in statistics and machine learning. It describes a model that based on the features present in a set of training data for each class in the database. The advantage for the algorithms is clearly: it is easy to understand for users and easy to implement on all kind of system. But the drawbacks are also obviously: if there are millions of data in the database or each data has a large number of attributes, the time for implementing will be huge, and the algorithm is not realistic.

3.2 SLIQ algorithm

The **SLIQ** (Supervised Learning In Quest) algorithm is used to classify the training dataset. It is introduced in [7].

• Basic principle of SLIQ

As many other classic classification algorithms, the SLIQ also can be implement in two phases: tree building phase and tree pruning phase. Because it is fit for both numerical and categorical attributes, there are a few differences in handling the two kinds of attributes. In the tree building phase, it uses a pre-sorting technique in the tree-growth phase for numerical attributes for evaluating splits while it uses a fast subsetting algorithm for categorical attributes for determining splits. This sorting procedure is integrated with a breadth-first tree growing strategy to enable classification of disk-resident datasets. In the pruning phase, it uses a new algorithm that based on the MDL (Minimum Description Length) principle and gets the results in compact and accurate trees.

• Details of the algorithm

SLIQ algorithm is fit for both numerical and categorical attributes. In this system, the attribute history we will consider are numerical and the others are categorical.

Phase of building tree: In this phase, there are two operations happen. First operation is to evaluate of splits for each attribute and to select the best split; second operation is partition the training dataset using the best split. The algorithm is described as following:

```
MakeTree(Training Data T)
   Partition (T);
Partition (Data S)
   if(all records S are in the same class))then return;
   Evaluate splits for each attribute A
   Use the best split to partition S into S1 and S2;
   Partition (S1);
   Partition (S2);
```

Before we analyze the numerical attributes, we partition the dataset by attributes -favor and job. Both of them are categorical. Let S(A) is the set of possible values of the attribute A, the split for A is of the form $A \in S'$, where S' is subset of S. The number of possible subset for an attribute with n possible value is 2 ** n. If the cardinality of S is large, the evaluation will be expensive. Usually, if the cardinality of the S is less than a threshold, MAXSETSIZE (the default value is 10), all of the subsets of S are evaluated. Otherwise, we use the greedy algorithm to get the subset. The algorithm starts with an empty S' and adds one element of S to S' that be the best split, these process will be repeated until there is no improvement in the splits.

For the attributes history, we pre-sorted first. Because in this system, we suppose there are four kinds of books in library, we divided the history into 4 parts, each part is for the number of one kinds of book that the users had borrowed. That means there are 4 numerical attributes need to be considered. To achieve this pre-sorting, we used the following data structure, we created a separate list (historyList[][]) for each attribute of the training dataset. history[][0] store the attribute value, history[][1] store the according index in the dataset. Then we sorted these attributes list independently.

After sorted the attributes list, we processed the splitting. As the algorithm is given below,

```
EvaluateSplits()
 for each attribute A do
   traverse attribute list of A
   for each value v in the attribute list do
      find the corresponding entry in the class list,
      and hence the corresponding entry class and the
          leaf node (say l)
      update the class histogram in the leaf l
          if A is numeric attribute then
            compute splitting index for test(A <= v)
          if A is a categorical attribute then
            for each leaf of the tree do
               find subset of A with best split.
```

For the numerical attribute, we need to compute the splitting index. In this algorithm, we use gini index (L.Breiman et.al.), which proposed by Wadsworth and Belmont. gini(T) defined as

$$gini(T) = 1 - \Sigma pj*pj$$

In this formula, T is a dataset that contains set of examples from n classes; pj is the relative frequency of class j in dataset T.

To calculate all of the ginin indexes for each attribute value, we compute the frequency for each class in the group first, and then found the best one for split the group. Because the value between vi and vi+1 will divide the list into two same parts, we choose the midpoint as the split point. For each group, one part is the examples that the value of attribute less than or equal to the split point, the other part is the examples that the value of attribute larger than the split point. We did the split as the same way one by one attribute until the node is the pure node (that is to say, all the examples in the node are the same class).

Phase of pruning tree: In this phase, the initial tree that built by using the training data will be examined and the sub-tree with the least estimated error rate will be chosen. The strategy is based on the principle of Minimum Description Length (MDL). It includes two parts: Data encoding and model encoding, comparison of the various sub-tree of T.

SLIQ is an attractive algorithm for data mining for its advantages:

1. The pre-sorting technique used in tree building phase and the MDL principles used in tree pruning phase make the result exhibits the same accuracy characteristics while the executes time is much shorter and the tree is smaller.

2. It can get the higher accuracies by classifying larger (disk-resident) datasets that can't be handled by other classifiers.

3. It can scale for large data sets and classify datasets irrespective the number of records, attributes and classes.

4. SYSTEM DESIGN

Although data mining techniques have been used in scientific and business field successfully for tracking behavior of individuals and groups, processing medical information, selecting market, forecasting financial trends and many other applications for several years, their uses in library system are limited. Many people argue that the current data mining technique is not appropriate for library system because of its lack of standards, its

unproven in library and the big technical hurdles remain. With the development of the size of database, we have to admit that the traditional catalogs can't satisfy the user's need, but the efficient new way is not discovered now. So, in this system, we try to find an alternative way to access it: to save time and make user more satisfied.

In the traditional library's searching system, all the users will get the same sequence of the search result if they use the same query to the same database, but they have own favorite field and their own need, so they are may be not interested in the books that will listed on the firstly pages.

In this system, data mining technique is used to analysis the information about the users. There are many attributes in the user's information. Here, our major interesting is the user's personality (includes his/her favorite field and profession) and the user's borrowing history. Users are classified into different classes by this information. In the same class, all the users mostly tend to borrow the same kind of books.

For example, a user's favorite fields are graphics and network, his profession is programmer, and in his borrowing history, the number of books about graphics he borrowed is 40% of the total number of books he borrowed, the number of the books about network he borrowed is 25% of the total number of books, and the number of books about program language is 30% of the total number. For the information about this specified person, using the mining result, we may classify him into the class GPGN (the priority of the kind of books for the member of this class is: graphics, program language, network and the others). So, when he uses the keyword search and types the keyword "java", all the books displayed on the screen will begin with the books about graphics with "java" appearing in the title, then all the books about program language with "java" appearing in the title, and then all the books about network with "java" appearing in the title and the other books. By this way, the user can find his/her wanted books faster than the traditional way.

In this phase, data are prepared for mining. There are a lot of records in the database and many attributes for each record, not only we need to select a relative small data set, but also we need to select the attributes that have effect on borrowing tends about the record. For the data about the user, we needn't consider all of the attributes because not all of attributes have impacts on the tendency of borrowing such as address, email, social security number and password and so on. After analyzing a lot of data, we found the impactions come from these features: the favorite fields, the borrowing history, the profession and the age of the person. To make the problem simple, we just consider the three of them: the favorite fields, the borrowing history and the profession of the user in the dataset.

SLIQ algorithm is used to generate pattern in this system, it is described in details in the last chapter. We get the decision tree by SLIQ algorithm, and we need to incorporate this knowledge. We will obtain the features for each pure node from the decision tree and put the results in the database, a table contains class type and the class features. When we need to decide the type of one member, we can get this information from database and make a decision.

As we analysis before, the goal of using data mining techniques is providing the faster and more satisfactory service for users. In this system, we can approach this goal by classifying the members into different class type and giving the different search result sequence for the different class members. The mining result is used in the two phases:

When a new member registers, he will be asked to provide his personal information. Using this information and mining result, the type of such user can be decided. Once the type is determined, the personalized search agent will sort the books for him according to his type. And this type is not immutable – the mining result will be used again and again. Because user's borrowing history is an important attribute that will impact on the member' s class type, when the user logout with some books, these books will change his borrowing history, his type maybe also changes. So, his class type will be calculated again. Or if the user modifies his profile, such as he change jobs or transfer his interesting to other fields, in such case, his class type will be determined again.

The personalize search agent will help user find what he wants quickly in two ways: it provides the default list for every user. According to the type of the user assorted to, system will automatically give the top 10 popular books of that kind that the user may most interested in. The other way is it will provide different search result for the different type of users.

In the application, with the increasing of the number of members and the changes of the member's class type, the initial decision tree may be lost its accuracy. It is necessary to mining data again. We will run the mining process after a period time to try to keep its accuracy; it is time consuming but can

saving time in searching process. It is obvious that in this system user can get the more useful search result quickly.

Tables 1 and 2 show the different search results:

	job	Favor	Search result
User A	Student	Programming language, e-business	e-business, programming language, graphics, system management
User B	Engineer	Graphics	Graphics, system management, Programming language, e-business

Table 1. search result by the personalized agent

	job	Favor	Search result
User A	Student	Programming language, e-business	programming language, graphics, system management, e-business
User B	Engineer	Graphics	programming language, graphics, system management, e-business

Table 2. research result by the traditional agent

5. CONCLUSION AND FUTURE WORK

In this system, we have successfully used data mining techniques to save search time for users. This system can provide more satisfied service for users. It is an improvement based on the existed library system and is a successful example that data mining used in practice. More important, it proves the possibility of using the data mining technique in library system. We are sure data mining technique will be used in more fields and become more popular in business.

There are more works need to do to improve the library system. We can consider all the attributes that will impact on the borrowing trend of the users. If we can do that, the accuracy of decision tree will be increased. We can use data mining technique in catalog. This will save more time for user in searching. In the future, advanced intelligent techniques such as soft computing, granular computing, and distributed computational intelligence will be used in the personalized library search agent system to continue to improve QoS of a library system and other information systems [2][10].

References

[1]. .Agrawal, A.Arning, T.Bollinger, M.Mehta, and J.S hafer, R.Srikant, "The quest data mining system," Proc. of the 2nd Int'l Conference on Knowledge Discovery in Databases and Data Mining, Portland, Oregon, August, 1996.

[2] Krzysztof Cios, Witold Pedrycz, and Roman Swiniarski, *Data Miming methods for Knowledge Discovery*, Kluwer Academic Publishers, 1998.

[3]. Yonagjian Fu, Kanwalpreet Sandhu and Ming-Yi Shih. Clustering of Web Users Based on Access Patterns, International Workshop on Web Usage Analysis and User Profiling (WEBKDD'99), San Diego, CA, 1999.

[4]. http://192.35.251.71/datamine/trees.htm Data mining techniques: Decision Trees.

[5]. http://www3.shore.net/~kht/text/dmwhite.html An introduction to data mining

[6] Huan Liu, Hiroshi Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Kluwer Academic Publishers, 1998.

[7]. Manish Mehta, Rakesh Agrawal and Jorma Rissanen, "SLIQ: A Fast Scalable Classifier for Data Mining,".

[8]. Berry, Michael J.A & Linoff, Gordon, "Data mining techniques for marketing, sales and customer support", 1997.

[9] Christopher Westphal, and Teresa Blaxton, "*Data Mining Solutions Methods and Tools for Solving Real-World Problems*," John Wiley & Sons, Inc., 1998.

[10] Y.-Q. Zhang, M. D. Fraser, R. A. Gagliano and A. Kandel, "*Granular Neural Networks for Numerical-Linguistic Data Fusion and Knowledge Discovery*," Special Issue on Neural Networks for Data Mining and Knowledge Discovery, *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, pp.658-667, May, 2000.

Dialogue-based approach to intelligent assistance on the web

Ana García-Serrano
Department of Computer Science, Technical University of Madrid
Spain
agarcia@dia.fi.upm.es

Paloma Martínez
Department of Computer Science, Carlos III University
Spain
pmf@inf.uc3m.es

David Teruel
Department of Computer Science, Technical University of Madrid
Spain
dteruel@isys.dia.fi.upm.es

Abstract
In this contribution we present the work performed in the ADVICE project, an on-going European Commission research project (IST 1999-11305). The overall objective is to design and implement an advice-giving system for E-commerce, supporting a move from the current catalogue-based customer services to a customer adapted intelligent assistance, emulating in some way the performance of a human seller. With this aim, the main elements of the ADVICE approach include an agent-based architecture with an Interface Agent to manage the multimedia presentation, the Interaction Agent to support an advanced user-system interaction and the Intelligent agent incorporating a knowledge-based model of the e-business, that supports the reasoning for advise-giving according with the user needs and the dialogue evolution.

1. Introduction

Searching for and selecting complex products on the web is a difficult task for consumers mainly due to the lack of intelligent support or assistance. An advanced solution in an e-commerce setting has to optimize the user search by a customer adapted intelligent assistance that emulates in some way the performance of a human seller including customer-adapted suggestion and explanation of product types, features, alternatives and special offers on digital markets.

The intelligent assistance needs to be supported by techniques capable to model the problem solving steps carried out by a person used to provide customer service. The knowledge-based technologies are being applied to develop an e-commerce prototype with several kinds of domain knowledge identified:

(a) To search for appropriate products according to the user needs, considering factors such as the client preferences, general characteristics of the products at different levels of abstraction, relations between products and preferences, constraints about certain product configurations, market strategies, etc.
(b) To the management of dialogue-based interaction, considering well-known dialogues in such scenario or previous experiences (conversation models or dialogue scripts), client profile (frame of features), kind of explanations needed etc.

The dialogue-based interaction has to be supported by techniques capable to model joint commitments during the dialogue, to perform pro-active system participation as well as to manage the whole process to a high performance of the system offering the user the right information every time.

The general architecture of the ADVICE system supporting these services contains three main agents:

- The Interface Agent responsible for the multimedia input-output activities of the system. This agent will collect users utterances (English sentences, clicks on the settled items, such as icons, menus, etc.) and transform them into semantic structures (streams of speech acts).
- The Interaction Agent is responsible for the adequate management of the interaction between the Interface Agent and the Intelligent Agent, as well as with the user. It means that has to (1) manage the evolution of the conversation in a coherent way, (2) deliver to the Intelligent Agent the query of the user together with relevant information about this user that may influence the production of the appropriate offer and (3) send to

the Interface Agent the question or information to be presented to the user at every moment.

- The Intelligent Agent responsible of the generation of the information required by the customer is supported by a knowledge model that contains the reasoning model as well as the domain structure.

These agent produces a configurable offer tree that contains the different products that suits the user identified needs. The prune of the tree comes from the new identification of user requirements during the dialogue.

We are currently working in the validation of the approach with the developed prototype to demonstrate the possibilities of the ADVICE approach to improve the results of the conventional E-commerce applications. This paper is devoted to the dialogue management and personalization of web-assistance in an e-commerce application.

2. Interaction Agent

A dialogue is a full-convene process where both interlocutors need to be tuned up for the best performance. Classical interfaces used to help the user through the interaction, instead of sharing a commitment. For a flexible dialogue is required at least one joint commitment so the speakers can understand one another. These commitments motivate the clarifications and confirmations always present in conversations. Research to model the commitment has been carried out in the late years, such as theoretical models of joint action [1]. Recent IST project Trindi [2] claims for the need of a 'common ground' between user and system. ADVICE approach joins this line of research prototyping the 'threads model'.

The interaction agent manages three different sorts of information from the user participation in the dialogue. First, has to extract the data that shapes the circumstance, the so called static information. Secondly should get the underlying intentions of the dialogue, that is the dynamic information. Finally, it has to attend to the structure of the interaction, for attaining a valid dialogue.

The main components of the interaction agent are lined up with this assortment. Hence, the *Session Model* will deal with the context and the details of the interaction, while the *Dialogue Manager* pay attention to the state of the interaction and checking the coherence preservation with the user.

On a second division, in order to handle de steps taken by both interlocutors through the interaction, there is a first component which role is to identify a valid dialogue helping to understand upcoming user movements. At the same time provides a set of adequate steps for the system to take. The other constituent of the dialogue manager stands for the intentional processing containing the user thread, the system thread and the thread joint, that supports a common ground for keeping the coherence of the interaction.

The third main component of the Interaction Agent is the *discourse generator*. It has to find a pattern that fits the needed discourse and then to fill it up with the context that will provide the session model. When needed some domain dependent information, will construct a request to the Intelligent Agent that will analyze the context and then provide the information requested updating the context. In this process, some events may occur, and could even originate new threads. This would force to restart the discourse generation.

A satisfactory management of dialogue requires in general both semantic representation (content of what has been expressed) and pragmatic information. The semantic structures used to link interface and interaction agents in the Advice project are based on Searle's speech acts [3],[4]. The current identified set of speech acts is shown in the figure 1:

Courtesy acts	
Salute: < c, a >	[c]: conventional (formal/informal)
Farewell: < c, a >	[a]: allowable (open/close)
Thank: < c, a >	Examples:
Disannoy: < c, a >	*Nice to see you Ana: salute(i,c)*
Empathetic: <c,a>	*Excuse me...* : disannoy(f,c)
Satisfy: < c, a >	*Don't worry...* : empathetic(i,c)
Wish: < c, a >	*Excellent* : satisfy(f,c)

Representative acts	
Inform: <t,m,s,c >	[t]: type (confirmation / data /...) [m]:matter (approve/deny/identity/..) [s]: subject (product/user/system/...) [c]: content (...) Example: *I'm Ana*: inform(data,identity,user,Ana)

Authoritative acts	
Authorize: <m,a >	[m]: matter (start / offer / task / ...) [a]: allowable (open / closed) Example: *Can I help you?* authorize(task,open)

Directive acts	
Request:<t,m,s,c> Command:<t,s,c>	[t]: type (choice/data/comparison/...) [m]: matter (approve/deny/identity/..) [s]: subject (user/system/...) [c]: content (values...) Examples: *Who are...?*: request(data,identity,...,) *Show me some saws*: command(search, system, product)

Null Speech	
Null: < >	*Well,...* : null()

Figure 1: Set of Speech Acts for the Advice

Further step is to model some kind of management for no crisp nor certain sentences of the user.

3. Natural Language Processing Components

One of the main aspects of the buying-selling interaction in the web is the capability of the web site of generating some kind of trust feeling in the buyer, just like a human shop assistant would do in a person-to-person interaction. Things like understanding the buyer needs, being able to give him technical advice, assisting him in the final decision are not easy things to achieve in a web selling site. Natural language (NL) techniques can play a crucial role in providing this kind of enhancements. Another good motivation to integrate natural language technology in this kind of sites is to make the interaction easy to those people less confident with the Internet or even computer technologies. Inexperienced users feel much more comfortable expressing themselves and receiving information in natural language rather than through the human standard ways.

The Interface Agent in ADVICE project includes the NL Interpreter and Generator components. The input user utterances are interpreted in order to obtain a feature-typed semantic structure (one or more due to ambiguity) that contains speech act information and some features about the relevant items of the user utterance.

At this moment, two interpretation strategies are defined. Firstly, message extraction techniques useful in specific domains are used in ADVICE, implemented by means of semantic grammars reflecting e-commerce generic sentences and idioms, sublanguage specific patterns and keywords. Secondly, if the pattern matching analysis does not work successfully, a robust processor that makes use of several linguistic resources (Brill tagger [5], WordNet [6], EuroWordNet [7] and a Phrase Segmenter[8]) integrates syntactic and semantic analysis in different ways. The complexity of the Natural Language applications makes almost compulsory to get profit of the existing resources that are available even though these resources were not full compatible with our requirements. The complexity of the Natural Language applications makes almost compulsory to get profit of the existing resources so are using a knowledge-based methodology in order to adapt and reuse existing English resources [8].

The lexicon is structured in three classes of words: general vocabulary, e-commerce vocabulary and domain specific vocabulary. The grammar rules contain no-terminal symbols that represent the domain concepts (tool, model, task, accessory and so on) and terminal symbols that represent the lexicon entries (vocabulary) in our application (saw, sander, to buy, to need and so on).

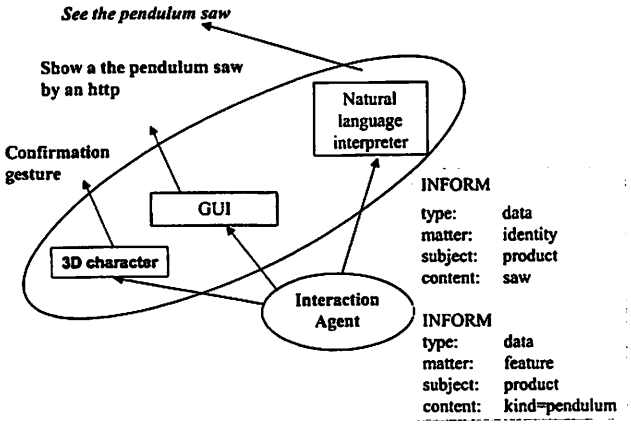


Figure 2: Generation process

Concerning the generation of answers, a domain-specific template-based approach is currently used. The templates used to generate natural language

answers to the user can be propositional (if they require arguments to fill the slots in) or not (for instance, agreements, rejections and topic movements).

In the current working prototype, templates do not contain issues concerning the User Model although they are ready to cover them in next step of the project. The user features that are considered in this first version are: Some templates include a special argument (ExpertiseLevel) that causes different levels of explanation in the system answers displayed to the user (pop-up links). These explanations are in a glossary containing each domain term (tool class, accessory and so on) together with its different explanations; Moreover, each template includes several possibilities of answer. Thus, the system does not generate always the same answer under the same conditions in order to achieve natural dialogues.

4. Intelligent Agent

In the current state of knowledge engineering, a knowledge model can be conceived as a hierarchically structured problem-solving model which implies the characterization of several classes of problems to be solved, i.e. tasks to be performed. Our proposal in this direction is to consider the organizational principle that we call the *knowledge-area oriented principle.* This principle establishes that a knowledge model can be organised by a hierarchy of knowledge-areas where each one defines a body of expertise that explains a specific problem solving behavior. The top-level area represents the whole model and is decomposed into simpler areas that encapsulate the expertise that support the reasoning methods. The bottom knowledge areas in the structure are called primary knowledge areas.

Every knowledge area is described with two parts: (i) its own knowledge represented by other simpler knowledge areas and (ii) its functionality represented by a set of tasks. In their turn, each task is another description entity that represents a basic function provided by a knowledge area (e.g., predict evolution or classify situation) and includes problem-solving methods to describe the strategy of reasoning to achieve the goals represented by the task. Each primary knowledge area have associated an inference method attached to the representation paradigm. It is also required to define a shared vocabulary to unify the concepts that have to manage the different knowledge areas.

The generic structure of the knowledge model for the ADVICE Intelligent Agent is subdivided into three Knowledge areas, *Profession Specialist, Works Specialist and Product Specialist.* Each area, or specialist, is an expertise of a sub-domain of the conceptual model.

Profession Specialist handles high level knowledge about the tasks or works that use to be done in a profession. The Works Specialist contains knowledge relating to these works, and what kinds of tools are needed to accomplish them. Finally, the Product Specialist manages knowledge about the features of the tools and accessories, and currently uses that knowledge to match the needs of the user with the products of the catalogue.

The craft domain of the current prototype includes the complete model for one kind of products, the circular saws. The generic knowledge bases identified are filled with the specific information of the saws using frames that contains the concepts and attributes specified in the generic vocabulary as well as the specific values to the product.

```
PATTERN atf55EBplus
DESCRIPTION (product)
type = saw [a], name = 'atf55EBplus' [m],
(saw) subtype = plunge [b],
       power = '1200' [c],
       weight = '4800' [d],
       saw blade speed <= 4800 [e],
       saw blade diameter = '160' [f],
       bevel cuts <= 45 [g],
       cutting depth <= 55 [h],
       dust extractor connection = 36 [i],
       systainer = 'yes'              [j],
       rail guide = 'no'              [k],
       electronics = 'yes'    [l],
RELEVANCE OF CHARACTERISTICS
a -> 50%, b -> 50%, c -> 50%, d -> 100%,
f -> 50%, h -> 50%, j -> 50%, k -> 50%,
l -> 50%, m -> 100%,
c,b -> 50%, b,f -> 50%, b,h -> 50%,
c,b,j,k,l -> 100%, b,f,j,k,l -> 100%,
b,h,j,k,l -> 100%, c,d,e,f,g,h,i,j,k,l-> 100%.
```

Figure 3. Description of a concrete product

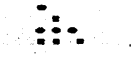
In the frame description (figure 3) can be distinguished three parts: the name of the frame, the description of the frame, a list of labeled (object – attribute – value) that represents the features of the product and the relevance of characteristics. Rules are used to represent which combination of the matched attributes allows the deduction of the whole frame in a

given situation. The IF-part is a simple expression (colon is an and operator) and the THEN-part contains a percentage indicating the matching degree of the frame. For example c,b → 50% means that if the c and the b slots are matched, the whole frame is matched with a certainty of the 50 per cent. Given that several rules can be used in a current situation, a simple uncertainty model computes the matching degree of the frame. The matched frames are organized into a tree structure according whit the domain knowledge. The nodes with alternatives have attached the attribute-value pairs which is expected will allow the discrimination of the alternatives according with next steps of the dialogue.

When the Interaction Agent send a request message to the Intelligent Agent, its answer after the reasoning process is an offer tree with the products that fulfil the identified user requirements. The Interaction Agent explores the tree and can find a solution node or selects from the alternative nodes the pairs to generate the next step in the dialogue. This information is send to the Interface agent that for example, can generate a new question to the user (pro-activity of the intelligent assistant) or produce an explanation showing some information.. When the required information from the user is received and returned to the Interaction agent, the prune of the tree according to the user answer is performed.

5. Working Prototype

•Welcome, I'm the virtual sales assistant. Who are you?
The first one is a neutral message
The second one is a starting message, (for user ident.).
•Hi there. I'm John Smith.
The system receives information (new user)
Then the system play neutral or starting messages
•Hello Mr. Smith. Nice to meet you. What can I do for you?
•Well, I want a circular saw.
Intelligent agent produces an offer configuration decision tree



•What kind of saw do you want: a pendulum or a plunge cut saw?

Figure 4: Some advice-customer interaction steps

This section is devoted to present some details of the performance of current working prototype for the dialogue example that first interaction steps are shown in figure 4. In next step in the dialogue the user answers the system question : "What kind of saw do you want: a pendulum or a plunge cut saw?". The figure 5 contains all the information extracted from the answer: "I think I need a pendulum cut saw".

From the utterance (or user clks on the saw alternative):
"I think I need a pendulum-cut saw"

• Identification of the comunicative acts:
null, inform(data, identity, product, saw)
inform(data,feature,product,type=pendulum-cut)

• Identification of a new state into the (current) question-answer dialogue pattern:
"solve" (the previous state was "request(product)")

• Identification of the topic:
product type pendulum-cut (add to the session model).

Figure 5:Informational contents of an utterance

The processing steps after user answer are presented in the sequel. The user answer contains an 'inform act' that is stored in the session model. The dialogue manager changes the dialogue state from 'require clarification' to 'solve task'. The thread will close system 'asking about type' element, and is again the user's 'request product' element next to be reach. The discourse maker fails in constructing a response with the final solution, because the tree of solutions has yet several of them. Hence, adds a new element in the system thread: 'request more data'. The discourse maker now is able to act sending to the Interface Agent the question referring next bifurcation in the offer tree. Finally, the action taken by the system will be used for updating the dialogue state (again to 'require data' to further step processing).

6. Conclusions

The work presented in this paper corresponds to a real-world experience with a complex problem, the development of an intelligent virtual assistant, where the solution based on the use of structures of problem-solving methods has shown to be successful to the design because of the capability of offering an understandable view of the reasoning and the incorporated knowledge.

One main advantage of the ADVICE approach comes from the integration of an advanced human-computer understanding and expression capabilities with the intelligent configuration of answers that leads to the generation of the right information in the right way and moment.

In the current prototype, the inter-agent message communication is supported by sockets and XML

contents coded. The Intelligent Agent was developed in C++ and Java and the Interaction Agent as well as the NLP components in Ciao Prolog [9].

7. Acknowledgements

We thank the Intelligent Systems Research Group (ISYS) and to the ADVICE Consortium for their support. (http:www.advice.iao.fhg.de)

8. References

[1] Cohen, P.R., Levesque, H.J. (1991) *Confirmation and Joint Action*, Proceedings of International Joint Conf. on Artificial Intelligence, 1991

[2] Cooper & Larsson, 99

[3] Searle, J.R., (1969). *Speech Acts: an essay in the philosophy of language.* Cambridge Univ. Press.

[4] García-Serrano, A. and Peñas, A. Interpretación de mensajes en un entorno de comunicación libre: Una aplicación a las conversaciones de correo electrónico. Technical Report FIM/110.1/IA/99. Technical University of Madrid, 1999

[5] Brill, E., Some advances in rule-based part of speech tagging. Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, Wa. , pp. 722-727, 1994

[6] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K., Introduction to WordNet: An On-line Lexical Database. (Revised August 1993). Princeton University, New Jersey, 1990.

[7] Vossen, P., Bloksma, L., Rodríguez, H., Climent, H., Calzolari, N., Roventini, A., Bertagna, F., Alonge, A., Peters, W., The EuroWordNet Base Concepts and Top Ontology. Version2. EuroWordNet (LE 4003) Deliverable, 1998.

[8] Martínez, P. and García-Serrano, A., The role of knowledge-based technology in language applications development. Expert Systems with Applications 19 , 31-44, 2000.

[9] Bueno, F., Cabeza, D., Carro, M., Hermenegildo, M., López, P., Puebla, G. (1999). *The Ciao Prolog System: Reference Manual.* The Ciao System Documentation Series Technical Report CLIP 3/97.1, The CLIP Group, School of Computer Science, Technical University of Madrid.

Accelerating Imprecise Temporal Queries for Video Navigation

Marcin Detyniecki
Berkeley initiative in Soft Computing (BISC)
Computer Science Division- Department of EECS
University of California, Berkeley, CA 94720
United Sates of America
marcin@eecs.berkeley.edu

Abstract
In this paper we present first a new method for computing temporal queries based on fuzzy time vocabulary. And secondly, since temporal analysis is usually heavy, we propose a parameterized pseudo-associative t-norm that reduces the computational time, without loosing generality.

1. Introduction

In this paper, we focus on how to navigate in an annotated video by making temporal queries. The annotations may be in a database with other information. We assume that the annotations are precisely time-indexed, but their attached information may be uncertain. In other words, we know precisely at what time (of the video) something happens, but we are not completely sure about everything associated with the event.
We introduced in [4], following the spirit of Zadeh's idea [14] of "Computing with words", a dictionary with the basic time related concepts. With this vocabulary and the logic tools introduced, the user is able to realize human type queries. Here, we focus on the time related queries [11],[1],[12]. We show a general way of how to compute a solution. However, the operators involved are usually heavy from the computational point of view. We propose a solution that reduces computational effort by relaxing some basic properties, but without loosing generality.
Let us start by explaining how the classical video query systems work and what we exactly propose.

2. Fuzzy Continuous Annotations

The actual works on query systems for video are based on the use of annotations (see [7], [8], [6], [10], [12]). These annotations can be considered as information contained in a database associated to the video and indexed by the time.
We call fuzzy annotation a classical annotation accompanied by a degree of certainty of the information (and not of the time indexing this annotation). This degree is usually a value between 0 and 1 (zero for completely uncertain and one for completely certain). So, for example an annotation can be: *"At minute 6 the actor on the scene is Robert with a degree of certainty 0.75"*. Which means that we think that the actor is Robert but we are not totally sure. We notice that the indexing time (6 minutes) is considered as certain.
We speak about *continuous* annotations because we have the information for every time. Now, we can represent this information on a graph, where the x-axis is the indexing time of the film and the y-axis is the degree of certainty. Note that the actor appears for a period of time so that we have a curve and not a point.

Figure 1. Fuzzy Annotation "Robert appears".

3. Placing the video player

Placing the video player at the starting time is not a trivial issue. Since we have fuzzy annotations, we do not know exactly when the event starts. Let us assume that we want to see when Robert appears. If we just use the certain information, the video player will always be placed after the real start (i.e. not when Robert appears but some seconds after). This will force the user to rewind in order to see the beginning, and it is not something we want. Taking this into account, we may think that a good solution is to start at the point where the certainty is not null (i.e. where the membership function starts). This time the video player will be placed too far in advance and the user will have to wait until the event happens, which is also not a good solution.
The action of indicating the exact start time can be seen as a defuzzification process. We use an approach, based on the alpha-cuts. The idea is to work by alpha-cuts. Here, we propose to simply take as starting

time(s) the minimum(s) of the (intervals) of the 1/2 cut. This gives us a point(s) where we are more or less sure that it starts. We pre-select this alpha-cut, but we leave the possibility to the advanced user to change its attitude for the defuzzification, by of increasing and decreasing the alpha value (see figure 2).

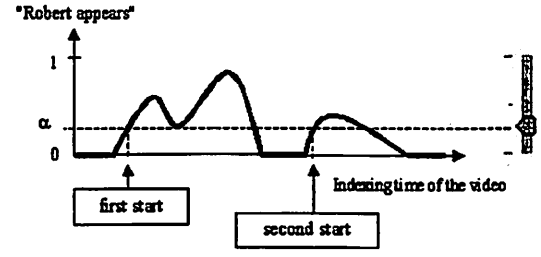


Figure 2. Placing the player.

For more details on this kind of defuzzification method in more general framework refer to our paper [2].

4. Fuzzy time vocabulary

In the spirit of Zadeh's idea of "computing with words" [14], we propose in [4] to construct a fuzzy time related dictionary. This thesaurus allows us to "precisiate" natural time querying. Using this we are able to use time positioning definitions such as *beginning*, *end* and *middle*, to use imprecise time duration such as *about five minutes*, *long* and *short time* and to use time relationship like *after*, *before* and *close*. In [4] we also present how to modify and combine this notions.

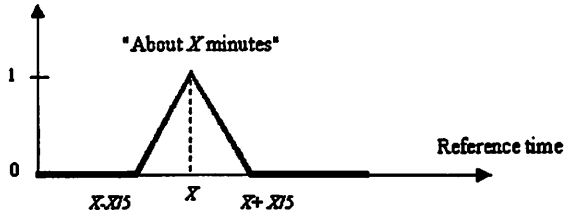


Figure 3. Example of fuzzy time vocabulary.

5. Fuzzy time relationship

We also defined relationships between time events, like for instance *after* and *before*. We based our approach on Yager's general framework for relative temporal relationship (see [12]). In this framework we have for example that the definition of "*after*" will be: If X-Y <0 then the degree of satisfaction of the concept "*X after Y*" is 0 and if X-$Y \geq 0$ the degree will be 1. In a symmetric way we can define the notion "*before*".

We proposed in [4] to use the time descriptors in order to generate new notions as for instance: *"About 10 minutes after"* or *"About 10 minutes before"*.

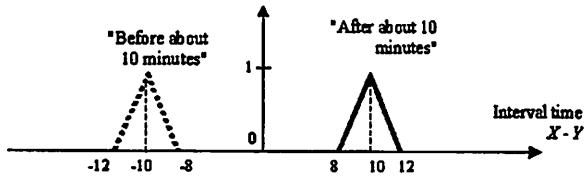


Figure 4. About 10 minutes after (or before).

6. Resolving time relationships

Now, using these relationships we may want to point to a particular moment in the video. For instance to answer the query: "*About 10 minutes after the crash*". Let *Crash*(y) be the membership degree of the annotation "*the crash*" at the time y. Then the membership function of "*About 10 minutes after the crash*" indexed by the time x will be obtained by following formula, where T is a t-norm :

$$About\_10m\_after\_Crash(x) = \max_y[T(after\_about\_10m(x-y), Crash(y))] \quad (1)$$

Formula (1) computes the "best answer" (max) for the logical conjunction (t-norm) of *"after about 10 minutes"* AND the *"crash"* event.

Let us generalize this result and let R be the membership function of a time relationship and E the membership function of an event, then we can point to a new moment of the video by using the general formula:

$$R \circ E(x) = \max_y[T(R(x-y), E(y))] \quad (2)$$

We remark that the event E can also be a time positioning, like *beginning*, *middle* or *end* of the video.

7. Choosing the t-norm

Formula (1) allows calculating simply the degree of membership at the time x of specific temporal query. However, the choice of the t-norm T is not clear. So we initially proposed to use a parameterized t-norm, so that the user can adjust the attitude of his aggregation.

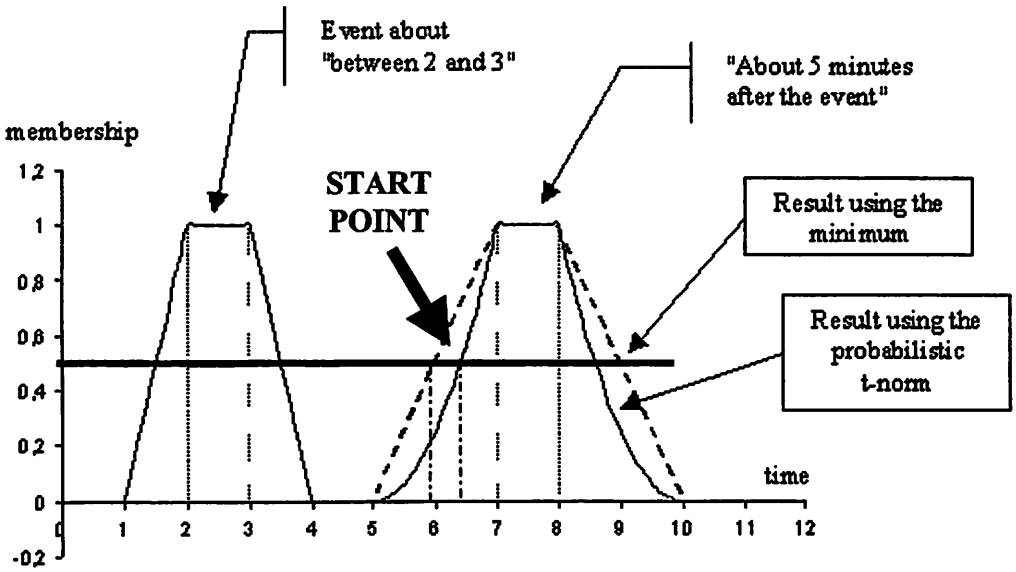


Figure 5. Pointing to "about 5 minutes after an event that happened more or less between 3 and 5 minutes"

In figure 5 we are looking for "about 5 minutes after an event that happened more or less between 3 and 5 minutes". Here "about 5 minutes" is a triangular fuzzy number being tolerant for more or less one minute. We notice that for a fixed defuzzification alpha-cut, we can have different starting points when using different t-norms. For instance when using the minimum we obtain a considerable earlier start than when using the product.

It is to notice that we are differentiating the t-norms by their attitude when aggregating the uncertainty. Since this is a personal choice we left the possibility of changing the attitude by using a parameterized t-norm. In [3] we present a methodology that is intended to help the user in the choice of the attitude by changing the parameter.

It is also important to note that all the t-norms have the same behavior when aggregating a certain value (see [3]). This induces that no matter what t-norm we use, we will always obtain the same certain interval (i.e. the same core). We can conclude that if we do not care about the uncertainty, then we can use any t-norm. This may happen if we just want to point the certain areas. However, the practice shows that this approach is deficient, since it is too strict. With just certain information the video-player is usually placed after the beginning of the event, forcing the user to rewind. Also a "only certain" approach will ignore all the imperfect annotations, in particular most of the automatic ones.

8. Using fast operators

It is clear that an interesting solution for the previous choice of the t-norm is to pick out a parameterized t-norm with a large attitude range. Like this we have always the possibility of choosing (from a large spectrum) the attitude with respect to the full uncertainty. For instance, we may select the Yager t-norm [13] (see [5] more complete justifications):

$$Y(u,v) = 1 - \left[(1-u)^p + (1-v)^p\right]^{1/p} \qquad (3)$$

However, looking at (2) we notice that we have to compute for every time x of the video, the aggregation (by the t-norm) of every time y in order to take the maximum. Taking into account that at least we have 25 frames per second, we have very quickly a great number of calculi. For instance just for *one* time relationship in a *one-hour-video*, we will have to compute 8.100.000.000 times the t-norm.

Examining the proposed algorithm and equation (2) we observe that we never use the associativity property of the t-norms. In fact each time we just aggregate two values. Here a fast operator that having similar properties as the Yager t-norm may be interesting. In [8] we study a family of operator with a relax type of associativity (pseudo-t-norms). From this work and obtain the following operator:

$$R(u,v) = \max\left((1-2t)\cdot(\max(u,v)-1)+\min(u,v),\ 0\right) \quad (4)$$

Where t is a parameter in the range [0,1/2]. Note that this parameter translates the attitude with respect to the total uncertainty. In fact it is the result of the aggregation at the 1/2 alpha-cut:

$$R^{\min-\max}\left(\frac{1}{2},\frac{1}{2}\right) = t \quad (5)$$

It is also to notice that this pseudo t-norm generalizes some basic t-norms. In fact, if we choose that the attitude with respect to the "total fuzziness" should be relaxed (i.e. t =1/2) then the operator (4) becomes the minimum (the largest t-norm). And for the strictest attitude (i.e. t =0), (4) becomes the Lukasiewicz t-norm. It is possible to obtain stricter operators than the Lukasiewicz t-norm by using negative parameter t. These particular cases will be bigger than the limit drastic t-norm, but we consider that their attitude is too strong. In fact all these last particular cases cut off everything with a degree under 1/2.

Now, coming back to the computational issues. Here for illustration proposes we are going to compare operator (4) to the Yager t-norm. But note that the same comparison can be done to any parameterized t-norm and we would obtain the same result.

Yager t-norm: Y(u,v)	*Operator: R (u,v)*
1 division - 1/*p*	1 product
1 additions	1 addition
3 subtractions	1 subtraction
3 power operations	1 comparison (*min*, *max*)

Table 1. Effort comparison between Yager t-norm and the reduced min.

Looking at Table 1, we notice that the new operator is computationally lighter. In fact, we observe that Yager t-norm has a division that has the same complexity as a product.
We immediately see that the calculus of the Yager t-norm is more arduous. In fact, the two operators have then same amount of additions and product, but Yager t-norm has one more subtraction. This does not make a large difference. The main difference appears when we observe that Yager t-norm needs 3 power operations, while the new operator only needs one comparison: the bigger number will be used for the max and the other one for the min.
It is to notice that in the n-ary case (see [5]) aggregation using Yager t-norm is heavier. In other words the new operator (4) is interesting when aggregation several times just couple of values, but also for the calculus of large number of arguments.

The fact that the reduced minimum is computationally faster than the Yager t-norm is interesting, but we may think that the price for this is that we lost the associativity property. In fact, the reduced minimum is not associative, but it is pseudo-associative as explain in [5]. The idea is that we can still aggregate by packages only by keeping in memory the maximum and minimum.

9. Application

A Java based Video Search Engine was developed by the multimedia indexing group at LIP6 (University Paris 6) for the Esprit / Avir project (see [9]).

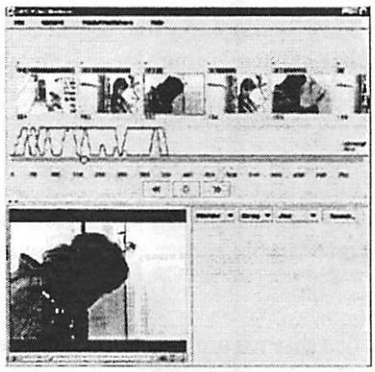


Figure 6. Java based Video Search Engine.

10. Conclusions

In this paper we presented a model for answering to imprecise temporal query. The system uses fuzzy annotations and a time related dictionary. This thesaurus "precisiates" what the user understands for a particular concept. This allows us to achieve a human friendly interface.
In this paper we focused on the fact that the calculus of time relationship is extremely heavy, so we propose an operator that lightens this computation. It is to notice that the presented operator is not only interesting for this particular example, but it is also an interesting solution for practical efficient applications, where use of a t-norm is required. Its computational lightness combined to its pseudo-associativity, without

forgetting its generalization property make of it a power-full tool.

11. Acknowledgment

This work was founded by the French Government in the form of grants. I would like to thank Claude Seyrat for the development of the Java Video Search Engine and Ronald Yager with whom I started this work.

12. References

[1] Cobb, M. and Petry, F., Fuzzy querying binary relationships in spatial databases, *Proceedings of the 1995 IEEE,* International Conference on Cybernetics and Society, Vancouver, 1452-1458, 1995.

[2] Detyniecki M. and Yager R., Ranking fuzzy numbers using α-weights, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, to appear in 2000.

[3] Detyniecki M., Yager R. and Bouchon-Meunier B., Specifying t-norms based on the value of T(1/2,1/2), *Mathware & Soft Computing vol.VII*, number 1, 2000.

[4] Detyniecki M., Seyrat C. and Yager R., Interacting with Web Video Objects, *Proceedings of 18th International Conference of the North American Fuzzy Information Society - Nafips'99*, pp.914-917, New York, USA, June 1999.

[5] Detyniecki M., Mathematical Aggregation Operators and their Application to Video Querying, *Doctoral Thesis - LIP6 research reports 2001,* number 002, November 2000.

[6] Hibino, S. Rundenstein, E. A., A visual multimedia query language for temporal analysis of video data, Multimedia Database Systems, Kluwer Academic Publisher: Norwell, MA. 123-159, 1996.

[7] Hjelsvold, R. and Midstraum, R., Modeling and querying video data, *Proceedings of the 20th VLDB Conference,* Santiago de Chile, 686-694, 1994.

[8] Mackay, W.E., EVA : An experimental video annotator for symbolic analysis of video data, *SIGCHI Bulletin 21*, 68-71, 1989.

[9] Seyrat C. and Detyniecki M. On Multimedia Indexing, *Second International Workshop on Knowledge Representation for Interactive Multimedia Systems - KRIMSII*, Trento - ITALY. June 1998.

[10] Snodgrass, R., The temporal query language TQUEL, *ACM Transactions on Database Systems* 12, 247-298, 1987.

[11] Vitek, M., Fuzzy information and fuzzy time, *Proceeding of IFCA Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis,* Marseille, 159-162, 1983.

[12] Yager, R. R., Fuzzy temporal methods for video multimedia information systems, *Journal of Advanced Computational Intelligence 1*, 37-45, 1997.

[13] Yager, R. R., On a general class of fuzzy connectives, *Fuzzy Sets and Systems 4*, 235-242, 1980.

[14] Zadeh, L. A., Fuzzy Logic = Computing with Words, *IEEE Transactions on Fuzzy Systems* 59, 125-148, 1993.

Clustering of Document Collections using a Growing Self-Organizing Map

Andreas Nürnberger
University of California at Berkeley
EECS, Computer Science Division
Berkeley, CA 94720, USA
E-mail: anuernb@eecs.berkeley.edu

Abstract

Clustering methods are frequently used in data analysis to find groups in the data such that objects in the same group are similar to each other. Applied to document collections, clustering methods can be used to structure the collection based on the similarities of the contained documents and thus support a user in searching for similar documents. Furthermore, the discovered clusters can be automatically indexed by keywords. Therefore the user does not depend on manually defined index terms or a fixed hierarchy, which often did not reflect recent changes in the underlying document collections. In this article we present an approach that clusters a document collection using a growing self-organizing map. The presented method was implemented in a software tool, which combines keyword search methods with a visualization of the document collection.

1. Introduction

Index terms and (hierarchical) classification methods are frequently used to structure object collections, e.g. document archives or libraries, and thus simplify the access for a user who is searching for specific documents. One of the main drawbacks of this approach is that the maintenance of these indexes is very expensive. Furthermore, they are usually not applied consistently and – since they are not updated frequently – they usually did not reflect recent changes of the structure of the underlying document collection.

Clustering is a well-known approach to structure prior unknown and unclassified datasets. Applied to document collections, clustering methods can be used to structure the collection based on the similarities of the contained documents and thus support a user in searching for similar documents. Furthermore, the discovered clusters can be automatically indexed by keywords without the need to manually define index terms.

In the following we present an approach that clusters a document collections using a growing self-organizing map. The method was implemented in a software tool that combines conventional keyword search methods with a visualization of the document collection. With the approach presented in this paper we resolved some of the problems of a first prototype that was implemented using conventional self-organizing maps [9, 15]. The main disadvantage of this architecture was that the size and shape of the map had to be defined in advance. Therefore a map had to be trained several times to obtain an appropriate solution. Especially for huge collections of documents this process is usually very time-consuming.

Clustering a document collection using self-organizing maps requires a preprocessing of the document collection to obtain numerical data describing each document. Similar to most of the existing models for document retrieval our approach is based on the vector space model [18]. The vector space model represents terms and documents as vectors in k-dimensional space. The currently most popular models using this approach are Latent Semantic Indexing (LSI) [2], Random Projection [8], and Independent Component Analysis (ICA) [7].

The vector space model enables very efficient analysis of huge document collections due to its simple data structure without using any explicit semantic information. A document is described based on a 'statistical fingerprint' of word occurrences and the semantically information is considered based on statistical correlations in further processing steps, e.g. the so-called 'latent semantic' in the LSI approach [13, 14]. However, some approaches try to consider semantic information by a preprocessing step, see e.g., the analysis of three-word contexts discussed in [5]. In spite of the insufficiencies the vector space model enables the processing of large document collections efficiently. Furthermore, using self-organizing maps document collections and search results can be visualized in an intuitive way.

In the following section we will briefly review the concepts of self-organizing systems and the implemented growing self-organizing map approach. In Sect. 3 we describe the document pre-processing steps and the

methods used for grouping the text documents based on different similarity measures. In Sect. 4 we present the implementation of this approach.

2. Self-organizing maps

Self-organizing maps [10] are a special architecture of neural networks that cluster high-dimensional data vectors according to a similarity measure. The clusters are arranged in a low-dimensional topology that preserves the neighborhood relations in the high dimensional data. Thus, not only objects that are assigned to one cluster are similar to each other (as in every cluster analysis), but also objects of nearby clusters are expected to be more similar than objects in more distant clusters. Usually, two-dimensional grids of squares or hexagons are used. Although other topologies are possible, two-dimensional maps have the advantage of an intuitive visualization and thus good exploration possibilities.

Self-organizing maps are trained in an unsupervised manner (i.e. no class information is provided) from a set of high-dimensional sample vectors. The network structure has two layers (see Figure 1). The neurons in the input layer correspond to the input dimensions. The output layer (map) contains as many neurons as clusters needed. All neurons in the input layer are connected with all neurons in the output layer. The weights of the connection between input and output layer of the neural network encode positions in the high-dimensional data space. Thus, every unit in the output layer represents a prototype. Before the learning phase of the network, the two-dimensional structure of the output units is fixed and the weights are initialized randomly. During learning, the sample vectors are repeatedly propagated through the network. The weights of the most similar prototype w_s (*winner neuron*) are modified such that the prototype moves toward the input vector w_i. As similarity measure usually the scalar product is used. The weights w_s of the winner neuron are modified according to the following equation: $\forall i: w'_s = w_s + \delta \cdot (w_s - w_i)$, where δ is a learning rate.

To preserve the neighborhood relations, prototypes that are close to the winner neuron in the two-dimensional structure are also moved in the same direction. The weight change decreases with the distance from the winner neuron. Therefore, the adaptation method is extended by a neighborhood function v:

$$\forall i : w_{s'} = w_s + v(c,i) \cdot \delta \cdot (w_s - w_i)$$

where δ is a learning rate. By this learning procedure, the structure in the high-dimensional sample data is non-linearly projected to the lower-dimensional topology. After learning, arbitrary vectors (i.e. vectors from the sample set or prior 'unknown' vectors) can be propagated through the network and are mapped to the output units. For further details on self-organizing maps see [11].

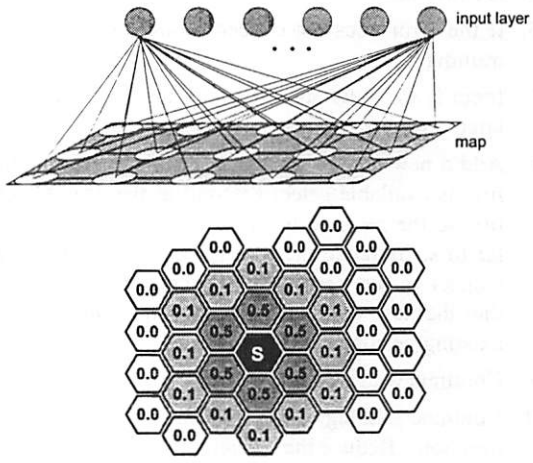


Figure 1. Structure of a rectangular self-organizing map (top) and possible neighborhood function for a structure based on hexagons (bottom).

Unfortunately, the standard model of self-organizing maps requires a predefined map structure. Therefore, the complete learning process has to be repeated if the size of the map was to small (the classification error which can be defined by $|w_s - w_i|$ for every pattern is usually very high and thus very dissimilar vectors are assigned to the same unit) or to large (similar vectors spread out on the map).

Growing self-organizing map approaches try to solve this problem by a learning method which modifies the size (and structure) of the map by adding new units to the map, e.g. if the accumulated error on a map unit increases a specified threshold. In the following the approach which is used in the presented application is briefly described.

2.1. A growing self-organizing map approach

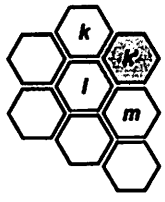
The proposed method is mainly motivated by the growing self-organizing map models presented in [1, 4]. In contrast to these approaches we use hexagonal map structure and restrict the algorithm to add new units to the external units if the accumulated error of a unit exceeds a specified threshold value.

The algorithm can be briefly described as follows:

1. Predefine the initial grid size (usually 2×2 units)
2. Initialize the assigned vectors with randomly selected values. Reset error values e_i for every unit i.

3. Train the map using all inputs patterns for a fixed number of iterations. During training increase the error values of a winner unit *s* by the current error value for pattern *i*.
4. Identify the unit with the largest accumulated error.
5. If the error does not exceed a threshold value stop training.
6. Identify the external unit *k* with the largest accumulated error.
7. Add a new unit to the unit *k*. If more than one free link is available select the unit at the nearest position to the neighboring unit which is most dissimilar to *k*. Initialize the weights of the new unit with respect to the vectors of the neighboring units so that the new vector is smoothly integrated into the existing vectors (see Figure 2).
8. Continue with step 3.
9. Continue training of the map for a fixed number of iterations. Reduce the learning rate during training.

This process creates an incremental growing map and it also allows training the map incrementally by adding new documents, since the training algorithms affects mainly the winning units to which new documents are assigned. If these units accumulate high errors, which means that the assigned documents cannot be classified appropriately, this part of the map starts to grow. Even if the consider neuron is an inner neuron, than the additional documents pushes the prior assigned documents to outer areas to which new neurons had been created. This can be interpreted, e.g. as an increase in publications concerning a specific topic. Therefore also dynamic changes can be visualized by comparing maps, which were incrementally trained by newly published documents.



k: winner unit
l, m: neighboring units
k': new unit
Computation of new weight vector for *k'*:
$w_{k'} = w_k + w_m - w_l$

Figure 2. Example of a computation of a vector for a new inserted unit

3. Building a map of a document collection

For the training of self-organizing maps, the documents must be encoded in form of numerical vectors. To be suited for the learning process of the map, to similar documents similar vectors have to be assigned, i.e. the vectors have to represent the document content. After training of the map, documents with similar contents should be close to each other, and possibly assigned to the same neuron. So, when a user has discovered a document of interest on the map, he or she can search the surrounding area.

The presented approach is based on statistical evaluations of word occurrences. We do not use any information on the *meaning* of the words since in domains like scientific research we are confronted with a wide and (often rapidly) changing vocabulary, which is hard to catch in fixed structures like manually defined thesauri or keyword lists. However, it is important to be able to calculate significant statistics. Therefore, the number of considered words must be kept reasonably small, and the occurrences of words sufficiently high. This can be done by either removing words or by grouping words with equal or similar meaning. A possible way to do so is to filter so-called *stop words* and to build the stems of the words (see e.g. [3]).

Although these document preprocessing steps are well known, they are still rarely used in commercially available document retrieval approaches or search engines. An overview of document pre-processing and encoding is given in Figure 3.

3.1. Stemming and filtering

The idea of stop word filtering is to remove words that bear no content information, like articles, conjunctions, prepositions, etc. Furthermore, words that occur extremely often can be said to be of little information content to distinguish between documents. Also, words that occur very seldom are likely to be of no particular statistical relevance.

Stemming tries to build the basic forms of words, i.e. strip the plural 's' from nouns, the 'ing' from verbs, or other affixes. A stem is a natural group of words with equal (or very similar) meaning. We currently used the stemming algorithm of [16], which uses a set of production rules to iteratively transform (English) words into their stems.

For the further reduction of relevant words we use two alternative approaches. The first reduces the vocabulary to a set of index words. These words are not selected manually, but automatically chosen by an information theoretic measure. The second approach is based on the work discussed in [17] and [6]. It uses a self-organizing map to build clusters of similar words, where *similarity* is defined based on a statistical measure over the word's context.

3.2. Selection of index words based on their entropy

For each word *a* in the vocabulary we calculate the entropy as defined by [14]:

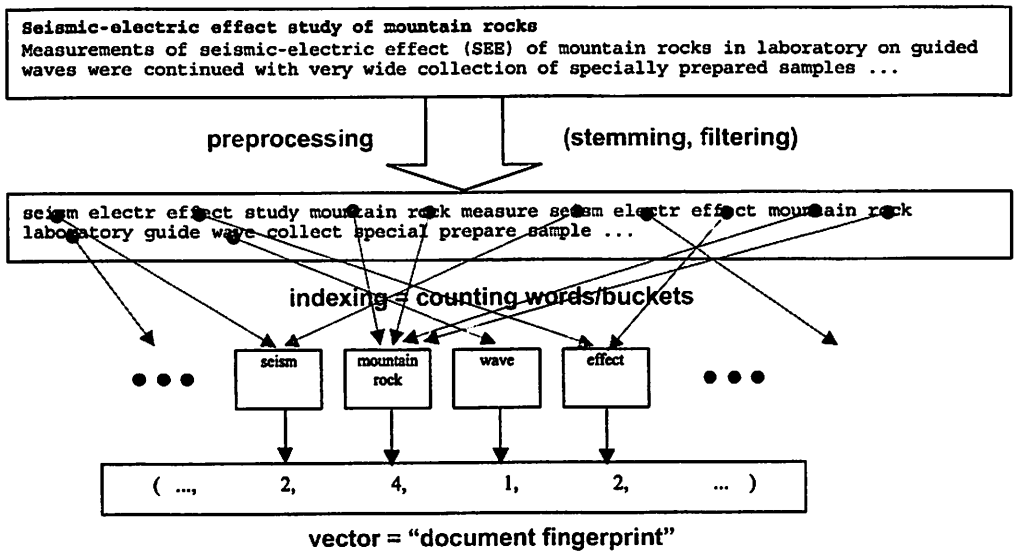


Figure 3. Document pre-processing and encoding

$$W(a) = 1 + \frac{1}{\ln(m)} \sum_{i=1}^{m} p_i(a) \cdot \ln(p_i(a)) \text{ with}$$

$$p_i(a) = \frac{n_i(a)}{\sum_{j=1}^{m} n_j(a)},$$

$n_i(a)$ is the frequency of word a in document i and m is the number of documents in the document collection. Here, the entropy gives a measure how well a word is suited to separate documents by keyword search. E.g. words that occur in many documents will have low entropy. The entropy can be seen as a measure of importance of words in the given domain context. We choose a number of words that have a high entropy relative to their overall frequency (i.e. from words occurring equally often we prefer those with the higher entropy). This procedure has empirically been found to yield a set of relevant words that are suited to serve as index terms.

3.3. The word category map

This approach does not reduce the number of words by removing irrelevant words from the vocabulary, but by building groups of words which are frequently used in similar (three-word-)contexts. A self-organizing map is used to find appropriate clusters of words. To be able to use words for training of a self-organizing map, the words have to be encoded. Therefore, to every word a random vector $\vec{w}$ with 90 dimensions is assigned (discussions concerning the number of dimensions can be found in [5]). This encoding does not imply any word ordering, as random vectors of dimensionalities that high can be shown to be 'quasi-orthogonal': the scalar product for nearly every pair of words is approximately zero. Then, the three-word-context of a word a is encoded by calculating the element-wise mean vectors of the words before $\vec{w}_{before}$ and after $\vec{w}_{after}$ the considered word over all documents and all occurrences of a. These mean (or expectation value) vectors $< \vec{w}_{before} >$ and $< \vec{w}_{after} >$ over the random vectors of enclosing words are used to define the context vector $\vec{w}_c$ of the considered word: $\vec{w}_c = (< \vec{w}_{before} >, \vec{w}, < \vec{w}_{after} >)$.
The obtained context vectors have 270 (=3·90) dimensions. Words a, b that often occur in similar contexts have similar expectation values and therefore similar context vectors $\vec{w}_c^{(a)}, \vec{w}_c^{(b)}$. The vectors $\vec{w}_c$ are finally clustered on a two-dimensional hexagonal grid using a self-organizing map. Words that are used in similar contexts are expected to be mapped to the same or to nearby neurons on this so-called word category map. Thus, the words in the vocabulary are reduced to the number of clusters given by the size of the word category map. Instead of index terms, the word categories *buckets* are used for the document indexing.
The most apparent advantage of this approach over the index term approach is that no words are removed from the vocabulary. Thus, all words are considered in the document clustering step. Furthermore, the word category map can be used as an expedient for the visual exploration of the document collection, because one often finds related words clustered together in the same or adjacent neurons of the word category map. From these clusters the user may choose related keywords which are appropriate for a (new) keyword search to reduce (or increase) the number of considered documents. However, due to the statistical peculi-

arities of the approach and the rather weak semantic clues the context vectors give, there are often additional words in the clusters that stand in no understandable relation to the others. The main drawback of this approach is that the words in one cluster become indistinguishable for the document indexing.

3.4. Generating characteristic document vectors

Figure 3 shows the principle of the proposed document encoding. At first, the original documents are preprocessed, i.e. they are split into words, then stop words are filtered and the word stems are generated (Sect. 3.1). Afterwards the considered vocabulary is reduced to a number of groups or *buckets*. These buckets are the index words from Sect. 3.2 or the word category maps from Sect. 3.3. The words of every document are then sorted into the buckets, i.e. the occurrences of the word stems associated with the buckets are counted. Each of the n buckets builds a component in a n-dimensional vector that characterizes the document. These vectors can be seen as the *fingerprints* of each document.

For every document in the collection such a fingerprint is generated. Using a self-organizing map, these document vectors are then clustered and arranged into a hexagonal grid, the so-called *document* map. Furthermore, each grid cell is labeled by a specific keyword that describes the content of the assigned documents.

The labeling method we used is based on methods proposed in [12]. It focuses on the distribution of words used in the documents assigned to the considered grid cell compared to the whole document database. The labeled map can then be used in visual exploration of the document collection, as shown in the following section.

4. Using the maps to explore document collections

To assess the usability of this approach a software prototype has been developed. The interactive user interface has been implemented in Java. The tool processes the documents as described above and stores the indexes and maps in a simple database. So we finally have a document map, where similar documents are grouped, and a word category map (if this approach is chosen) where the grouping of words is shown. In Figure 4 a screenshot of the software tool is shown.

The document map opens up several appealing navigation possibilities. Most important, the surrounding grid cells of documents known to be interesting can be scanned for further (similar) documents. Furthermore, a keyword search method has been implemented that – besides providing an ordered result set – visualizes the distribution of keyword search results by coloring the grid cells of the document map with respect to the number of hits for specific keywords or combinations of keywords. This allows a user to judge e.g. whether the search results are assigned to a small number of (neighboring) grid cells of the map, or whether the search hits are spread widely over the map and thus the search was – most likely – too unspecific and should be further refined.

If the highlighted nodes build clusters on the map we can suppose that the corresponding search term was relevant for the neighborhood relations in the learning of the self-organizing map. In this case the probability to find documents with similar topics in adjacent nodes can be expected to be higher.

Furthermore, the labels (index terms) assigned to the grid cells can be used to search for specific topics and thus supports the user in navigating through the document collection.

4.1. Using the word category map

The word category map can be used e.g. to look for related keywords for searching. If, for example, the number of search hits seems to be very small, so we would like to broaden our search. On the other hand, we would like the query to be still specific. In the word category map we can visualize the fingerprints of the matching documents. The highlighted nodes give us visual hints on which important keywords the document contains in addition to those keywords we have been searching for. Furthermore, we may find groups of documents with visually similar fingerprints (i.e. similar highlighted regions) and thus similar content. Therefore, we are supported in finding some keywords which describe the document content and which can then be used to refine the search by adding (or prohibiting) these keywords.

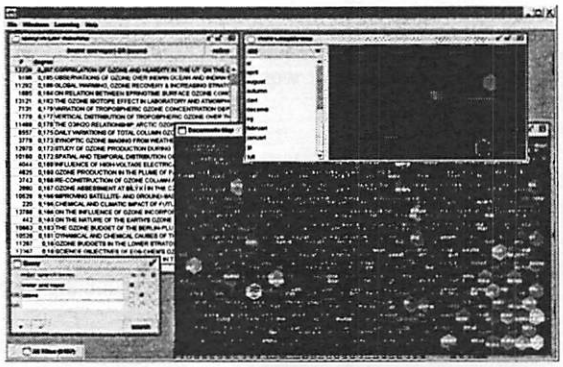


Figure 4. Screenshot of the software tool: Ranked search results (top left), word category map (top right), query window (bottom left), and document map with colored grid cells according to search results (bottom right)

5. Conclusions

The presented approach enables a user to search for specific documents, but also to enlarge obtained result sets (without the need to redefine search terms) by navigating through groups of documents with similar contents surrounding the search hits. Furthermore, the user is supported in finding appropriate search keywords to reduce or increase the documents under consideration by using a word category map, which groups together words used in similar contexts.

The methods proposed in this article combine (iterative) keyword search with grouping of documents based on a similarity measure in an interactive environment without the need to manually define lists of index terms or a classification hierarchy, which usually require expensive maintenance. Especially in rapidly changing document collections – like collections of publications of scientific research – classification systems that are not frequently updated are usually not accepted by the users, for whom especially new topics are of high importance.

Acknowledgements

The work presented in this article was partially supported by BTexact Technologies, Adastral Park, Martlesham, UK.

References

[1] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery, *IEEE Transactions on Neural Networks,* 11(3), pp. 601-614, 2000.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, and T. K. Landauer, Indexing by latent semantic analysis, *Journal of the American Society for Information Sciences,* 41, pp. 391-407, 1990.

[3] W. B. Frakes, and R. Baeza-Yates, *Information Retrieval: Data Structures & Algorithms,* Prentice Hall, New Jersey, 1992.

[4] B. Fritzke, Growing cell structures - a self-organizing network for unsupervised and supervised learning, *Neural Networks,* 7(9), pp. 1441-1460, 1994.

[5] T. Honkela, Self-Organizing Maps in Natural Language Processing, Helsinki University of Technology, Neural Networks Research Center, Espoo, Finland, 1997.

[6] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, Newsgroup Exploration with the WEBSOM Method and Browsing Interface, Technical Report, In: Helsinki University of Technology, Neural Networks Research Center, Espoo, Finland, 1996.

[7] C. L. Isbell, and P. Viola, Restructuring sparse high dimensional data for effective retrieval, In: *Proc. of the Conference on Neural Information Processing (NIPS'98),* pp. 480-486, 1998.

[8] S. Kaski, Dimensionality reduction by random mapping: Fast similarity computation for clustering, In: *Proc. Of the International Joint Conference on Artificial Neural Networks (IJCNN'98),* pp. 413-418, IEEE, 1998.

[9] A. Klose, A. Nürnberger, R. Kruse, G. K. Hartmann, and M. Richards, Interactive Text Retrieval Based on Document Similarities, *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy,* 25(8), pp. 649-654, Elsevier Science, Amsterdam, 2000.

[10] T. Kohonen, Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics,* 43, pp. 59-69, 1982.

[11] T. Kohonen, *Self-Organization and Associative Memory,* Springer-Verlag, Berlin, 1984.

[12] K. Lagus, and S. Kaski, Keyword selection method for characterizing text document maps, In: *Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks,* pp. 371-376, IEEE, 1999.

[13] T. K. Landauer, P. W. Foltz, and D. Laham, An Introduction to Latent Semantic Analysis, *Discourse Processes,* 25, pp. 259-284, 1998.

[14] K. E. Lochbaum, and L. A. Streeter, Combining and comparing the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval, *Information Processing and Management,* 25(6), pp. 665-676, 1989.

[15] A. Nürnberger, A. Klose, R. Kruse, G. Hartmann, and M. Richards, Interactive Text Retrieval Based on Document Similarities, In: G. Hartmann, A. Nölle, M. Richards, and R. Leitinger (eds.), *Data Utilization Software Tools 2 (DUST-2 CD-ROM),* Max-Planck-Institut für Aeronomie, Katlenburg-Lindau, Germany, 2000.

[16] M. Porter, An algorithm for suffix stripping, *Program,* pp. 130-137, 1980.

[17] H. Ritter, and T. Kohonen, Self-organizing semantic maps, *Biological Cybernetics,* 61(4), 1989.

[18] G. Salton, A. Wong, and C. S. Yang, A vector space model for automatic indexing, *Communications of the ACM,* 18(11), pp. 613-620, (see also TR74-218, Cornell University, NY, USA), 1975.

Presentations of Friday, August 17

E-Commerce, Intelligent Agents, Customization and Personalization

Soft Knowledge as Key Enabler of Future Services

Ebrahim Mamdani
Imperial College of Science, Technology and Medicine, University of London

Evolution Towards Digitisation

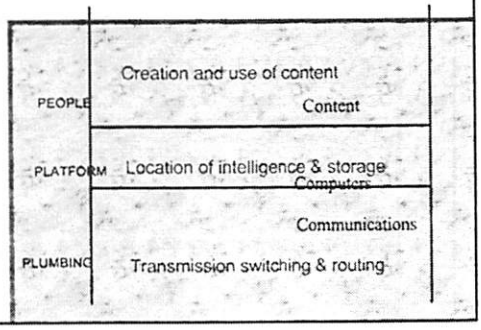
50 years since the computer has been invented has been marked the following key developments:

- Integrated circuits
  - The operation of Moore's law providing six orders of magnitude performance gain in only 50 years
- The development of the Ethernet
  - providing novel means of communication between computers
    - shared medium giving statistical multiplexing gain
- Digitisation of all kinds of content
  - Analogue forms superseded by digital
  - Laying the foundation of CONVERGENCE

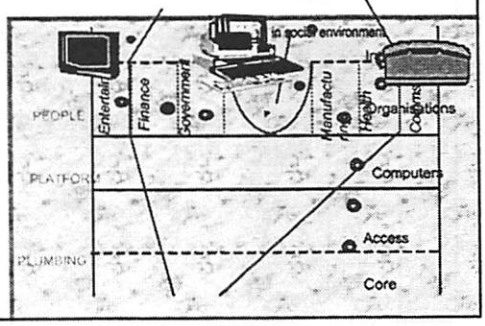
Convergence?

- Digitisation of all kinds of content
- Communication of digital content
- Blurring of established commercial boundaries
- New opportunities as well as threats

Overview - Reference Model



Overview - Reference Model



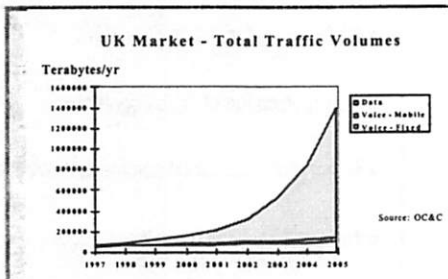
Observations

- Convergence implies increased competition between hitherto separate traditional businesses
  - Offers possibility of substituting one service with another
    - Fax vs e-mail
    - E-mail vs phone
- Current areas of interest include:
  - Fixed – mobile
  - Phone – internet

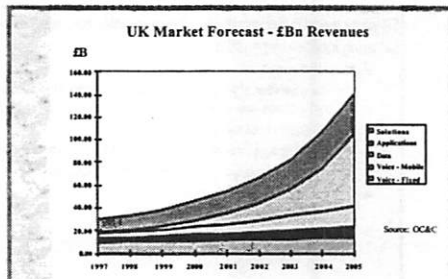
Audio-visual – Internet Convergence

- Currently push vs Pull
- Very little live content
- Tendency towards multi-cast
- Increasingly sharing common standards
  - Distribution of each other's traditional content
- But vastly differing cultures and mind-sets
- Sit-up vs sit-back

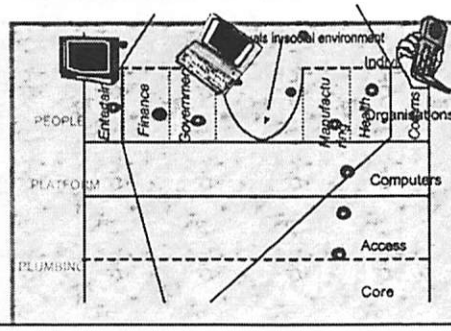
Datawave Changes Everything



UK Growth by Segment



Overview - Reference Model



Digital Devices

- Digital devices are essentially digital computers
- They carry out a number of key functions:
  - some key task scheduler
  - variety of I/O device drivers
  - a range of network drivers
  - screen management
  - Keyboard / keypad management
  - a set of applications
  - such that the set makes some coherent sense
  - Other key functions are placed at accessible locations
    - e.g. home or office machine, ISP servers

Digital Devices

- Digital convergence offers a huge variety of functions and combinations
- Devices are packages of functional subsets
  - Compromise between "function creep" and ease of use
  - Need to inter-work with other functions (i.e. other devices)
  - Foreign functions located at "home" or at other service providers
  - Critically dependant upon battery technology
- Middleware is essential for support of such inter-working

Content As Commodity

- Timing
  - Content is created
  - content is advertised
  - It is sold
  - Funds are collected
  - Content is transported
  - content is received
  - content is used
- Storage at many points in the network allows trading of content

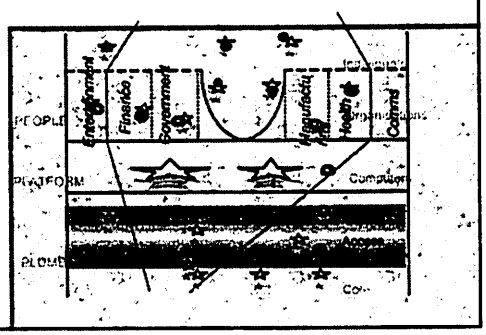
Digital Content

- Intellectual Property
  - rights need protection
- Personal
  - Security
    - anonymity; confidentiality; authentication; non-repudiation
- Cultural
- Meta-content critical commodity in its own right
  - libraries; directories (white and yellow pages); dictionaries (digital ontologies)

New Services: Communication

- Human to Human
  - minor need for live contact between two or more individuals
- Human to archive
  - Growing market of direct access
- Human to Machine
  - Games and simulations
- Machine to machine
  - Unprecedented degree of automation
  - Essential societal support functions
  - Monitoring proper functioning of people & properties

Overview - Reference Model



Types of Agents

- Agents serving individuals
  - Personal profiles, terminal preferences
- Agents serving organisations
  - selling, advertising products and services
- Agents in the middleware
  - Providing a set of services to other agents
    - Location services; Management services (registration etc.)
    - Federation services (combine a number of service providers)
- Agents in the network
  - Optimizing network resources

Unplanned Consequences

- Noteworthy failures
  - Iridium Satellite phone system
  - Video on demand
  - WAP
- Unplanned successes
  - E-mail
  - World wide Web
  - Mobile telephony
  - SMS
- Doubtful future:
  - IM
  - Digital TV

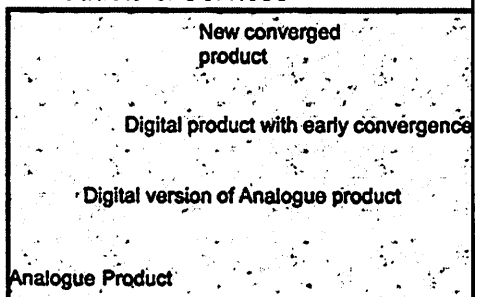
Unplanned Consequences

- Digital devices & services are versatile
- Designers underestimate the ingenuity and creativity of the users
  - Users are free to take advantage of the versatility presented to them
- User's first priority is to gain life advantages
- Thus, they may use the devices & services in a manner that gives them such advantages
  - mostly by fair means
  - but also by foul means

Darwinian Evolution Applies

- Technology capable of endless variation
  - Programmability
- Payoff function as social advantage
  - Network externalities
  - Near universal service has a very large payoff
- Selection works as long as no dependencies are created
  - Qwerty phenomenon
- Evolution is all tactics and no strategy

Evolving Convergence Products & Services



Example Services

- Bundled personalised services.
  - Audio-visual entertainment packages.
  - Travel service bundles.
- Intelligent appliances.
- Intelligent spaces.
- Management of business processes.
  - Automatic diffusion of relevant information across alliances, departments etc.
  - Manufacturing around the world.

Relevant Questions for the Future

- How can we transfer energy from our bodies to power our devices
- How to make devices aware of us?
  - Aware of us as individuals but also when in a group?
  - How to use this information usefully for configuring devices for optimal interaction?
  - How to make devices socially aware?
  - What new services are possible with such socially aware devices?
- How long before we a have paperless world?

Relevant Questions for the Future

- What are the social consequences of Convergence?
  - What is the truth about digital divide?
  - Difference between knowledge workers and service workers?
    - Those who tell computers what to do and those who are told by computers what to do?
- How important is the Internet?
  - What is its true impact on Information Ecosystem?

Pointers

- Intelligence at the edge
  - Loss of control feared by the providers?
  - Empowerment of the user?
- Storage at the edge of the network
  - Loss of IPR feared by the content owners? Pointers
- From 1000 persons per computer to 1000 computers per person
- Ease of use of device is an important factor
  - Managing user expectations of the intelligence
  - Costs against social gains have to be measured
  - Users need to learn the value of IPR
  - Analogue has its place also

Intelligent Information Processing and Analysis

Masoud Nikravesh

BISC Program, Computer Sciences Division, EECS Department
University of California, Berkeley, CA 94720, USA
Email: nikravesh@cs.berkeley.edu

Abstract

The process of ranking (scoring) has been used to make billions of financing decisions each year serving an industry worth hundreds of billion of dollars. To a lesser extent, ranking has also been used to process hundreds of millions of applications by U.S. Universities resulting in over 15 million college admissions in the year 2000 for a total revenue of over $250 billion. College admissions are expected to reach over 17 million by the year 2010 for total revenue of over $280 billion. In this paper, we will introduce fuzzy query and fuzzy aggregation as an alternative for ranking and predicting the risk for credit scoring and university admissions which currently utilize an imprecise and subjective process.

Introduction

Consider walking into a car dealer and leaving with an old used car paying a high interest rate of around 15% to 23% and your colleague leaves the dealer with a luxury car paying only a 1.9% interest rate. Consider walking into a real estate agency and finding yourself ineligible for a loan to buy your dream house. Also consider getting denied admission to your college of choice but your classmate gets accepted to the top school in his dream major. Welcome to the world of ranking, which is used both for deciding college admissions and determining credit risk. In the credit rating world, FICO (Fair Isaac Company) either makes you or breaks you, or can at least prevent you from getting the best rate possible [1]. Admissions ranking can either grant you a better educational opportunity or stop you from fulfilling your dream.

When you apply for credit, whether it's a new credit card, a car loan, a student loan, or a mortgage, about 40 pieces of information from your credit card report are fed into a model. That model provides a numerical score designed to predict your risk as a borrower. When you apply for university or college admission, more than 20 pieces of information from your application are fed into the model. That model provides a numerical score designed to predict your success rate and risk as a student. In this paper, we will introduce fuzzy query and fuzzy aggregation as an alternative for ranking and predicting risk in areas which currently utilize an imprecise and subjective process.

The areas we will consider include: credit scoring (Table 1), credit card ranking (Table 2), and university admissions (Table 3). Fuzzy query and ranking is robust, provides better insight and a bigger picture, contains more intelligence about an underlying pattern in data and is capable of flexible querying and intelligent searching [2]. This greater insight makes it easy for users to evaluate the results related to the stated criterion and makes a decision faster with improved confidence. It is also very useful for multiple criteria or when users want to vary each criterion independently with different degrees of confidence or weighting factor [3].

Fuzzy Query and Ranking

In the case of crisp queries, we can make multi-criterion decision and ranking where we use the functions AND and OR to aggregate the predicates. In the extended Boolean model or fuzzy logic, one can interpret the AND as a fuzzy-MIN function and the OR as a fuzzy-MAX function. Fuzzy querying and ranking is a very flexible tool in which linguistic concepts can be used in the queries and ranking in a very natural form. In addition, the selected objects do not need to match the decision criteria exactly, which gives the system a more human-like behavior.

Table 4. Measures of Association

Simple Matching Coefficient :	$\lvert A \cap B \rvert$
Dice's Coefficient :	$2\dfrac{\lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert}$
Jaccard's Coefficient :	$\dfrac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
Cosine Coefficient :	$\dfrac{\lvert A \cap B \rvert}{\lvert A \rvert^{1/2} \times \lvert B \rvert^{1/2}}$
Overlap Coefficient :	$\dfrac{\lvert A \cap B \rvert}{\min(\lvert A \rvert, \lvert B \rvert)}$
Dissimilarity Coefficient :	$\dfrac{\lvert A \Delta B \rvert}{\lvert A \rvert + \lvert B \rvert} =$
1 – Dice's Coefficient :	$\lvert A \Delta B \rvert = \lvert A \cup B \rvert - \lvert A \cap B \rvert$

Measure of Association and Fuzzy Similarity: There are five commonly used measures of association and these are given in Table 4. In this study, the fuzzy-based Jaccard's model has been used to calculate the similarity. Suppose the fuzzy sets to be measured are fuzzy sets A and B with membership functions $\mu_A(x)$ and $\mu_B(x)$ respectively;

$$E(A,B) \equiv degree(A = B) \equiv \frac{|A \cap B|}{|A \cup B|}$$

Table 5. Properties of aggregation operators for triangular norms and triangular conorms.

• *Conservation* $t(0,0) = 0; t(x,1) = t(1,x) = x$	• *Conservation* $s(1,) = 1; s(x,0) = s(0,x) = x$
• *Monotonicity* $t(x_1,x_2) \le t(x_1',x_2')$ *if* $x_1 \le x_1'$ *and* $x_2 \le x_2'$	• *Monotonicity* $s(x_1,x_2) \le s(x_1',x_2')$ *if* $x_1 \le x_1'$ *and* $x_2 \le x_2'$
• *Commutativity* $t(x_1,x_2) = t(x_2,x_1)$	• *Commutativity* $s(x_1,x_2) = s(x_2,x_1)$
• *Associativity* $t(t(x_1,x_2),x_3) = t(x_1,t(x_2,x_3))$	• *Associativity* $s(s(x_1,x_2),x_3) = s(x_1,s(x_2,x_3))$

Given the following properties for aggregation operators (see Table 5): conservation, monotonicity, commutativity, associativity, the triangular norm and triangular co-norm will be used to calculate fuzzy aggregation [3-7]. There are several triangular norms/triangular co-norms pairs (see Table 6) which can be used such as [3-6]:

Minimum/Maximum;
Drastic Product/Drastic Sum;
Bounded Difference/Bounded Sum;
Einstein Product/Einstein Sum;
Algebraic Product/Algebraic Sum;
Hamacher Product/Hamacher Sum

In this study, Fuzzy-Min (and/conjunction) and Fuzzy-Max operators (or/disjunction) have been used (see Table 7) [3, 6,7].

Table 6. Triangular norm/triangular co-norm pairs.

$$\text{Minimum}: t(x_1,x_2)=\min\{x_1,x_2\}$$
$$\text{Maximum}: s(x_1,x_2)=\max\{x_1,x_2\}$$

$$\text{Drastic Product}: t(x_1,x_2)=\begin{cases}\min\{x_1,x_2\} & \text{if } \max\{x_1,x_2\}=1\\ 0 & \text{othewise}\end{cases}$$

$$\text{Drastic sum}: s(x_1,x_2)=\begin{cases}\max\{x_1,x_2\} & \text{if } \min\{x_1,x_2\}=0\\ 1 & \text{othewise}\end{cases}$$

$$\text{Bounded difference}: t(x_1,x_2)=\max\{0,x_1+x_2-1\}$$
$$\text{Boubded sum}: s(x_1,x_2)=\min\{1,x_1+x_2\}$$

$$\text{Einstein product}: t(x_1,x_2)=(x_1\cdot x_2)/(2-(x_1+x_2-x_1\cdot x_2))$$
$$\text{Einstein sum}: s(x_1,x_2)=(x_1+x_2)/(1+x_1\cdot x_2)$$

$$\text{Algebraic product}: t(x_1,x_2)=x_1\cdot x_2$$
$$\text{algebraic sum}: s(x_1,x_2)=x_1+x_2-x_1\cdot x_2$$

$$\text{Hamacher product}: t(x_1,x_2)=(x_1\cdot x_2)/(x_1+x_2-x_1\cdot x_2)$$
$$\text{Hamacher sum}: s(x_1,x_2)=(x_1+x_2-2x_1\cdot x_2)/(1-x_1\cdot x_2)$$

Table 7. Fuzzy-Min and Fuzzy-Max Operators.

Conjunction rule : $\mu_{A\wedge B}(x)=\min\{\mu_A(x),\mu_B(x)\}$
Disjunction rule : $\mu_{A\vee B}(x)=\max\{\mu_A(x),\mu_B(x)\}$
Negation rule : $\mu_{\neg A}(x)=1-\mu_A(x)$
$\mu_{A\wedge A}(x)=\mu_A(x)$
$\mu_{A\wedge(B\vee C)}(x)=\mu_{(A\wedge B)}(x)\vee\mu_{(A\wedge C)}(x)$

If: $\mu_A(x)\le\mu_A(x')$ **AND** $\mu_B(x)\le\mu_B(x')$

Then: $\mu_{A\wedge B}(x)\le\mu_{A\wedge B}(x')$

If Query (A) and Query (B) are equivalent:
$\mu_A(x)=\mu_B(x)$

Precisions and Recall Measure: Table 8 and Fig. 1 show the definition of precision, recall and their relationship. Given a user's criteria, the data provided for modeling, and the strategy defined in Fig. 2, the recall/precision relationship has been optimized. Therefore, a user will get better precision and recall in fuzzy or imprecise situations.

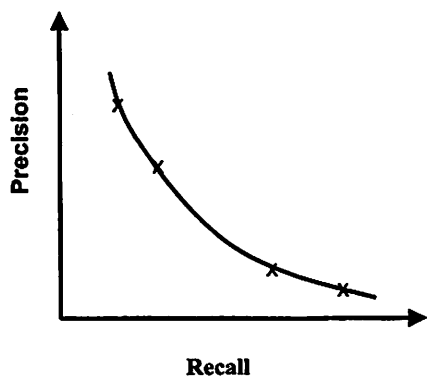


Fig. 1. Inverse relationship between Precision and Recall.

Table 8. Measures of Precision, Recall and several other relevant attributes.

$$\text{Precision}: P=\frac{|A\cap B|}{|B|}$$
$$\text{Recall}: R=\frac{|A\cap B|}{|A|}$$
$$\text{Fallout}: F=\frac{|\bar{A}\cap B|}{|\bar{A}|}$$
$$\text{Generality}: G=\frac{|A|}{N}$$
$$\text{Retrieved / Relevent}: A\cap B$$
$$\text{Retrieved / Non-Relevent}: \bar{A}\cap B$$
$$\text{Not-Retrieved / Relevent}: A\cap\bar{B}$$
$$\text{Not-Retrieved / Not-Relevent}: \bar{A}\cap\bar{B}$$

Search Strategy: There are several ways to search and query in databases such as latent semantic indexing (LSI), full text scanning, inversion, and the use of signature files. While LSI has limitations, it is highly rewarding, since it is easy to implement and update; it is fast; it works in a reduced domain; it is scaleable; and it can be used for parallel processing. One solution to its Boolean model is to use an extended Boolean model or fuzzy logic. In this case, one can add a fuzzy quantifier to each term or concept. In addition, one can interpret the AND as a fuzzy-MIN function and the OR as a fuzzy-MAX function respectively.

The most straightforward way to search is **full text scanning**. The technique is simple to implement; has no space overhead; minimal effort on insertion or update is needed; a finite state automaton can be built to find a given query; and Boolean expressions can be used as query resolution. However, the algorithm is too slow.

The **inversion** method is the most suitable techniques followed by almost all commercial systems (if no semantics are needed).It is easy to implement and fast. However, storage overhead is up to 300% and updating the index for dynamic systems and merging of lists are costly actions. In this study, in addition to inversion techniques, Fuzzy-Latent Semantic Indexing (**FLSI**) originally developed for text retrieval has been used [2, 3]. Fig. 2 shows a schematic diagram of the performance of **FLSI**. Fig. 3 and Fig. 4 show the performance of **FLSI** for text retrieval purposes. The following briefly describes the **FLSI** technique [2, 3]:

- Fuzzy-based decompositions are used to approximate the matrix of document vectors.
- Terms in the document matrix may be presented using linguistic terms (or fuzzy terms such as most likely, likely, etc) rather than frequency terms or crisp values.
- Decompositions are obtained by placing a fuzzy approximation onto the eigensubspace spanned by all the fuzzy vectors.
- Empirically, we establish our technique such that the approximation errors of the fuzzy decompositions are close to the best possible; namely, to truncated singular value decompositions.

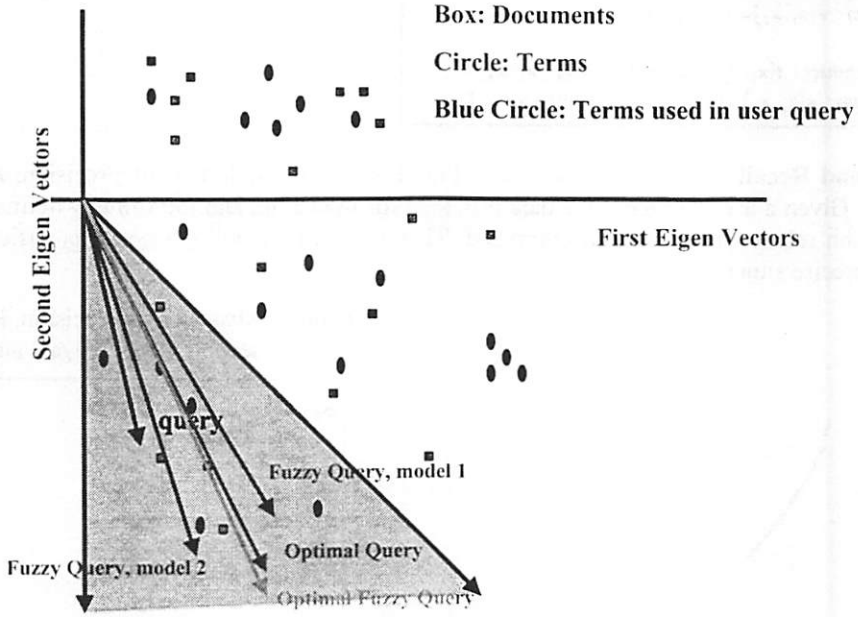


Fig. 2. Schematic diagram of the performance of the Fuzzy-Latent Semantic Indexing method.

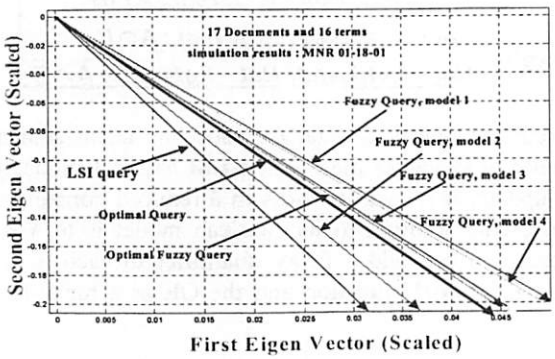


Fig. 3. Example 1 of FLSI for text retrieval.

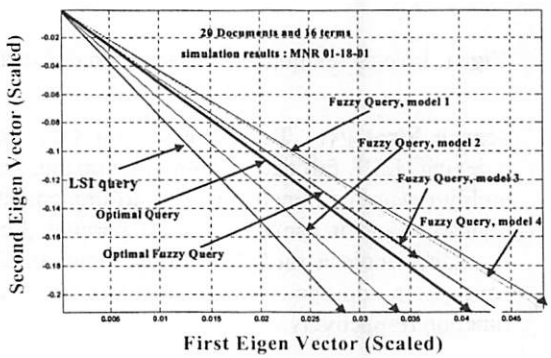


Fig. 4. Example 2 of FLSI for text retrieval.

Implementation

In this section, we introduce fuzzy query and fuzzy aggregation for credit scoring, credit card ranking, and university admissions.

Application to Credit Scoring: Credit scoring was first developed in the 1950's and has been used extensively in the last two decades. In the early 1980's, the three major credit bureaus, Equitax, Experian, and TransUnion worked with the Fair Isaac Company to develop generic scoring models that allow each bureau to offer an individual score based on the contents of the credit bureau's data. FICO is used to make billions of financing decisions each year serving a 100 billion dollar industry. Credit scoring is a statistical method to assess an individual's credit worthiness and the likelihood that the individual will repay his/her loans based on their credit history and current credit accounts. The credit report is a snapshot of the credit history and the credit score is a snapshot of the risk at a particular point in time. Since 1995, this scoring system has made its biggest contribution in the world of mortgage lending. Mortgage investors such as Freddie Mac and Fannie Mae, the two main government-chartered companies that purchase billion of dollars of newly originated home loans annually, endorsed the Fair Isaac credit bureau risk, ignored subjective considerations, but agreed that lenders should also focus on other outside factors when making a decision.

When you apply for financing, whether it's a new credit card, car or student loan, or a mortgage, about 40 pieces of information from your credit card report are fed into a model (Table 1). This information is categorized into the following five categories with different level of importance (% of the score):

- Past payment history (35%)
- Amount of credit owed (30%)
- Length of time credit established (15%)
- Search for and acquisition of new credit (10%)
- Types of credit established (10%)

When a lender receives your Fair Isaac credit bureau risk score, up to four "score reason codes" are also delivered. These explain the reasons why your score was not higher. Followings are the most common given score reasons [1];

- Serious delinquency
- Serious delinquency, and public record or collection filed
- Derogatory public record or collection filed
- Time since delinquency is too recent or unknown
- Level of delinquency on accounts
- Number of accounts with delinquency
- Amount owed on accounts
- Proportion of balances to credit limits on revolving accounts is too high
- Length of time accounts have been established
- Too many accounts with balances

By analyzing a large sample of credit file information on people who recently obtained new credit, and given the above information and that contained in Table 1, a statistical model has been built. The model provides a numerical score designed to predict your risk as a borrower. Credit scores used for mortgage lending range from 0 to 900 (usually above 300). The higher your score, the less risk you represent to lenders. Most lenders will be happy if your score is 700 or higher. You may still qualify for a loan with a lower score given all other factors, but it will cost you more. For example, given a score of around 620 and a $25,000 car loan for 60 months, you will pay approximately $4,500 more than with a score of 700. You will pay approximately $6,500 more than if your score is 720. Thus, a $25,000 car loan for 60 months with bad credit will cost you over $10,000 more for the life of the loan than if you have an excellent credit score.

Given the factors presented earlier and the information provided in Table 1, a simulated model has been developed. A series of excellent, very good, good, not good, not bad, bad, and very bad credit scores have been recognized (without including history). Then, fuzzy similarity and ranking have been used to rank the new user and define his/her credit score. Fig. 5 shows the simplified flow diagram and flow of information for PNL-Based Fuzzy Query. In the inference engine, the rules based on factual knowledge (data) and knowledge drawn from human experts (inference) are combined, ranked, and clustered based on the confidence level of human and factual support. This information is then used to build the fuzzy query model with associated weights. In the query level, an intelligent knowledge-based search engine provides a means for specific queries. Initially we blend traditional computation with fuzzy reasoning. This effectively provides validation of an interpretation, model, hypothesis, or alternatively, indicates the need to reject or reevaluate. Information must be clustered, ranked, and translated to a format amenable to user interpretation.

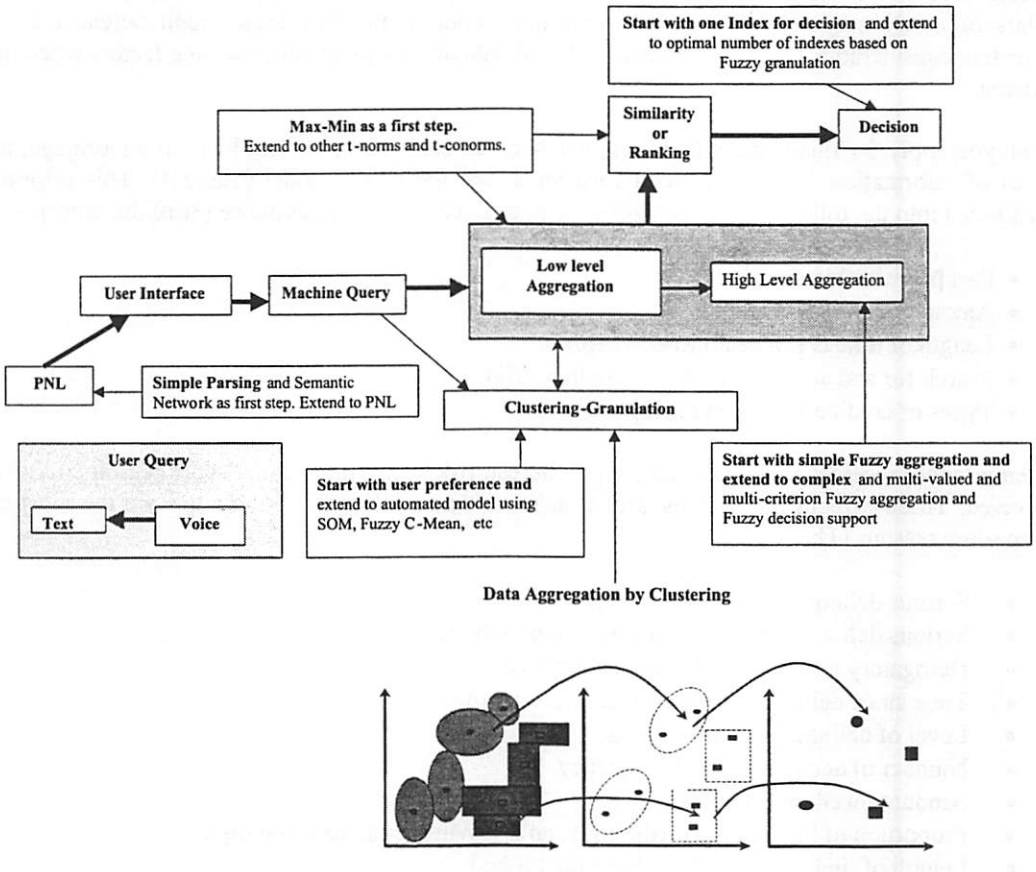


Fig 5. Simplified flow diagram and flow of information for PNL-Based Fuzzy Query.

Fig. 6 shows a snapshot of the software developed for credit scoring. Table 1 shows the granulation of the variables that has been used for credit scoring/ranking. To test the performance of the model, a demo version of the software is available at: http://zadeh.cs.berkeley.edu/ [2]. Using this model, it is possible to have dynamic interaction between model and user. This provides the ability to answer "What if?" questions in order to decrease uncertainty, to reduce risk, and to increase the chance to increase a score.

Application to Credit Card Ranking: Credit ratings that are compiled by the consumer credit organization such as the U.S. Citizens for Fair Credit Card Terms (CFCCT) [8] could simply save you hundreds of dollars in credit card interest or help you receive valuable credit card rebates and rewards

including frequent flyer miles (free airline tickets), free gas, and even hundreds of dollars in cash back bonuses.

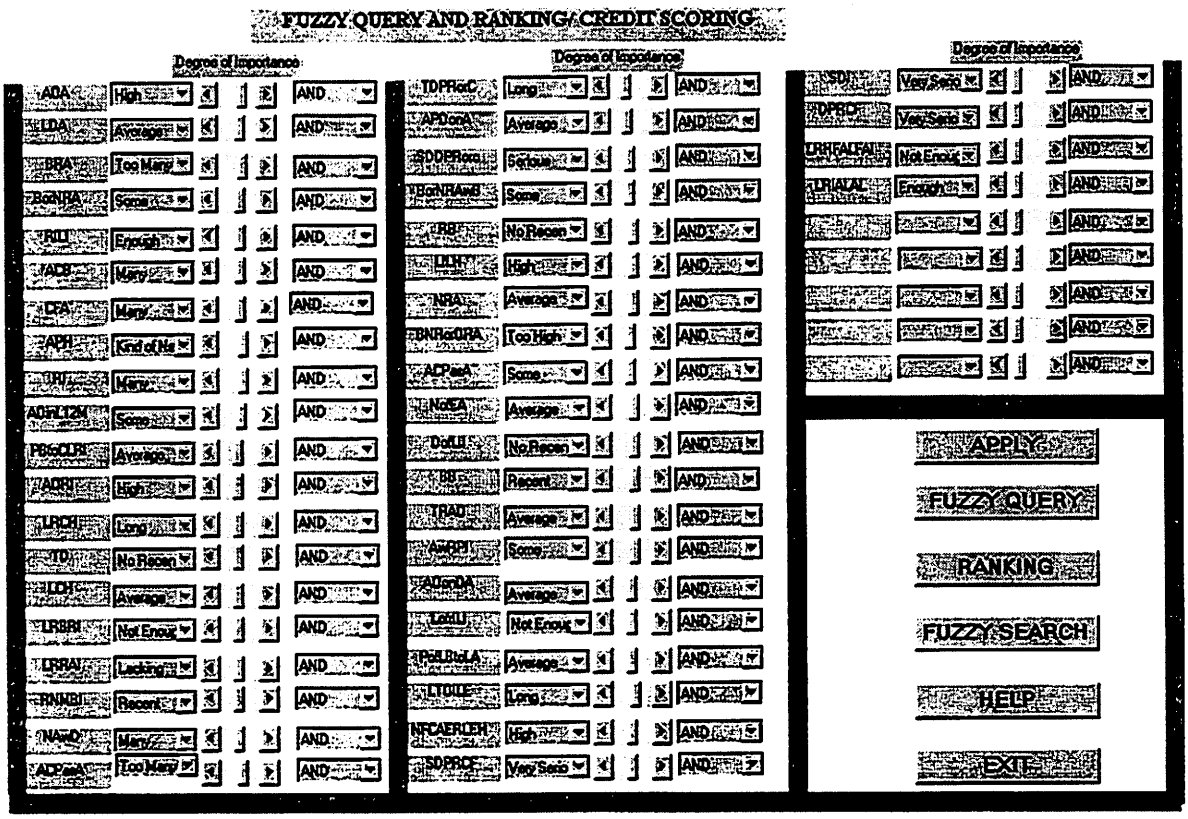


Fig. 6. A snapshot of the software developed for credit scoring.

CFCCT has developed an objective-based method for ranking credit cards in US. In this model, interest rate has the highest weighting in the ranking formula. FCC rates credit cards based on the following criteria [8]:

- Purchase APR
- Cash Advance APR
- Annual Fees
- Penalty for cards that begin their grace periods at the time of purchase/posting instead of at the time of billing
- Bonuses for cards that don't have cash advance fees
- Bonuses for cards that limit their total cash advance fees to $10.00
- Bonuses for introductory interest rate offers for purchases and/or balance transfers
- Bonuses for cards that have rebate/perk programs
- Bonuses for cards that have fixed interest rates.

Table 9 shows the top 10 classic cards, the top 10 gold cards, and the top 10 platinum cards which have been ranked by the CFCCT method [8] as of March 2001. Given the above factors and the information provided in Table 8, a simulated model has been developed. A series of excellent, very good, good, not good, not bad, bad, and very bad credit cards have been recognized for the credit cards listed in Table 9. Then, fuzzy similarity and ranking has been used to rank the cards and define a credit score. Fig. 7 shows a snapshot of the software developed to rank credit cards. Table 2 shows the granulation of the variables that

has been used for the rankings. To test the performance of the model, a demo version of the software is available at: http://zadeh.cs.berkeley.edu/ [2].

Table 9. Credit cards ranked by the CFCCT.

Classic Cards	Type	Gold Cards	Type	Platinum Cards	Type
Pulaski B& T	V	Pulaski	MC	Capital One	VP
Ark. Natl	MC/V	Capital One	VP	NextCard	VP
Capital One	V	SFNB	V	BofA	VP
NextCard	V	NextCard	V	Simmons	VP
Wachovia	V	BofA	V	G&L Bank	MCP/VP
MCP/VPBlue	AMEX	Wachovia	V	Aria	VP
Helena Natl	MC/V	Blue	AMEX	Ever	VP
Simmons	V	Helena	MC/V	Blue	AMEX
Metro. Natl.	V	Simmons	V	AF	VP
Umbrella	V	Metro.	V	Banco	VP

V=Visa; MC=MasterCard; AMEX=American Express

University Admissions: Hundreds of millions of applications were processed by U.S. universities resulting in more than 15 million enrollments in the year 2000 for a total revenue of over $250 billion. College admissions are expected to reach over 17 million by the year 2010, for total revenue of over $280 billion. In Fall 2000, UC Berkeley was able to admit about 26% of the 33,244 applicants for freshman admission [8]. In Fall 2000, Stanford University was only able to offer admission to 1168 men from 9571 applications (768 admitted) and 1257 women from 8792 applications (830 admitted), a general admit rate of 13% [10].

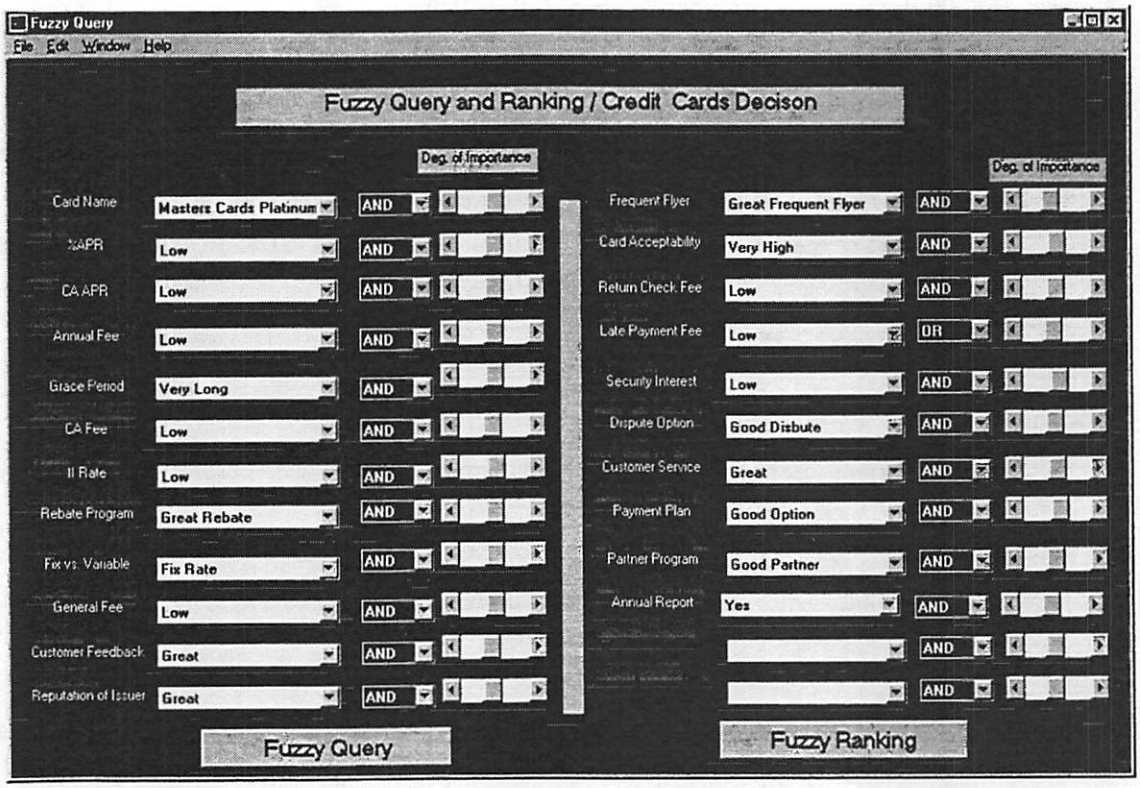


Fig. 7. A snapshot of the software developed to rank credit cards.

The UC Berkeley campus admits its freshman class on the basis of an assessment of the applicants' high school academic performance (approximately 50%) and through a comprehensive review of the application including personal achievements of the applicant (approximately 50%) [9]. For Fall 1999, the average weighted GPA of an admitted freshman was 4.16, with a SAT I verbal score range of 580-710 and a SAT I math score range of 620-730 for the middle 50% of admitted students [9]. While there is no specific GPA for UC Berkeley applicants that will guarantee admission, a GPA of 2.8 or above is required for California residents and a test score total indicated in the University's Freshman Eligibility Index must be achieved. A minimum 3.4 GPA in A-F courses is required for non-residents. At Stanford University, most of the candidates have an un-weighted GPA between 3.6 and 4.0 and verbal SAT I and math SAT I scores of at least 650 [10]. At UC Berkeley, the academic assessment includes student's academic performance and several measured factors such as:

- College preparatory courses
- Advanced Placement (AP)
- International Baccalaureate Higher Level (IBHL)
- Honors and college courses beyond the UC minimum and degree of achievement in those courses
- Uncapped UC GPA
- Pattern of grades over time
- Scores on the three required SAT II tests and the SAT I (or ACT)
- Scores on AP or IBHL exams
- Honors and awards which reflect extraordinary, sustained intellectual or creative achievement
- Participation in rigorous academic enrichment
- Outreach programs
- Planned twelfth grade courses
- Qualification for UC Eligibility in the Local Context

All freshman applicants must complete courses in the University of California's A-F subject pattern and present scores from SAT I (or ACT) and SAT II tests with the following required subjects:

a. History/Social Science - 2 years required
b. English - 4 years required
c. Mathematics - 3 years required, 4 recommended
d. Laboratory Science - 2 years required, 3 recommended
e. Language Other than English - 2 years required, 3 recommended
f. College Preparatory Electives - 2 years required

At Stanford University, in addition to the academic transcript, close attention is paid to other factors such as student's written application, teacher references, the short responses and one-page essay (carefully read for quality, content, and creativity), and personal qualities.

The information provided in this study is a hypothetical situation and does not reflect the current UC system or Stanford University admissions criteria. However, we use this information to build a model to represent a real admissions problem. For more detailed information regarding University admissions, please refer to the University of California-Berkeley and Stanford University, Office of Undergraduate Admission [9,10].

Given the factors above and the information contained in Table 3, a simulated-hypothetical model (a Virtual Model) was developed. A series of excellent, very good, good, not good, not bad, bad, and very bad student given the criteria for admission has been recognized. These criteria over time can be modified based on the success rate of students admitted to the university and their performances during the first, second, third and fourth years of their education with different weights and degrees of importance given for each year. Then, fuzzy similarity and ranking can evaluate a new student rating and find it's similarity to a given set of criteria.

Fig. 8 shows a snapshot of the software developed for university admissions and the evaluation of student applications. Table 3 shows the granulation of the variables that was used in the model. To test the performance of the model, a demo version of the software is available at: http://zadeh.cs.berkeley.edu/ [2]. Incorporating an electronic intelligent knowledge-based search engine, the results will eventually be in a format to permit a user to interact dynamically with the contained database and to customize and add information to the database. For instance, it will be possible to test an intuitive concept by dynamic interaction between software and the human mind.

This will provide the ability to answer "What if?" questions in order to decrease uncertainty and provide a better risk analysis to improve the chance for "increased success" on student selection or it can be used to select students on the basis of "diversity" criteria. The model can be used as for decision support and for a more uniform, consistent and less subjective and biased way. Finally, the model could learn and provide the mean to include the feedback into the system through time and will be adapted to the new situation for defining better criteria for student selection.

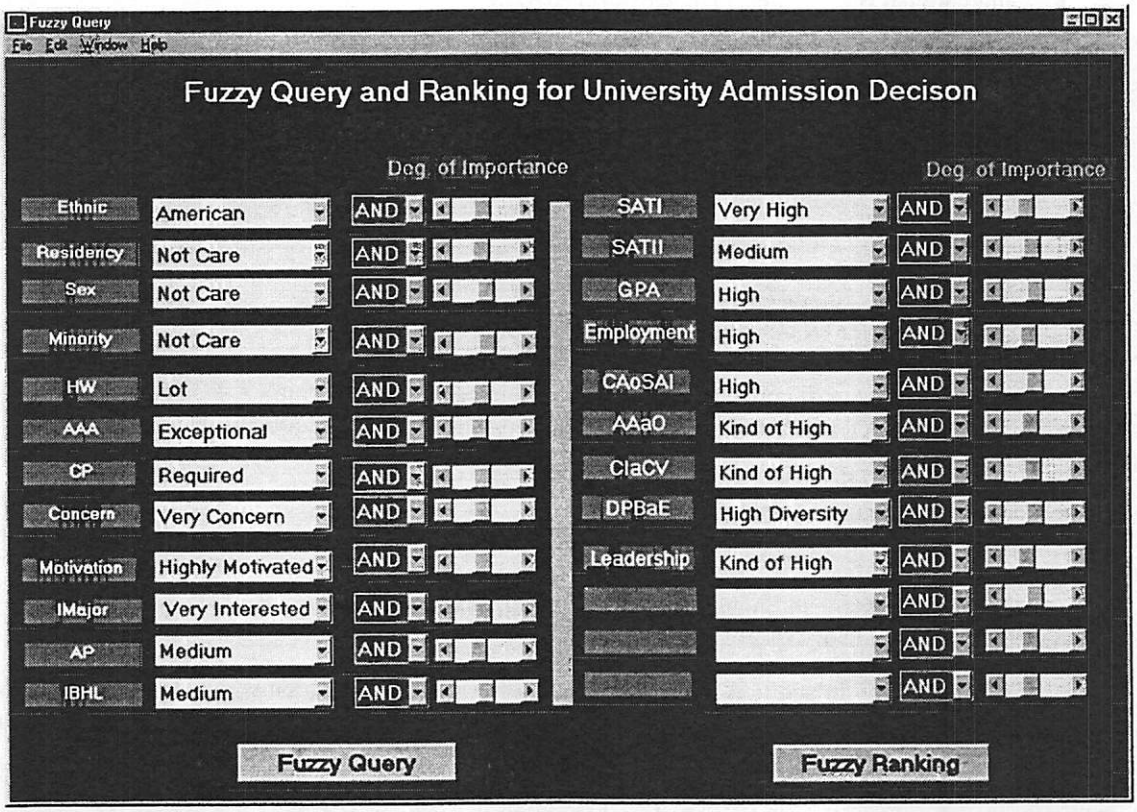


Fig. 8. A snapshot of the software developed to rank credit cards.

Conclusions

In this study, it has been found that ranking and scoring is a very subjective problem and depends on user perception and preferences in addition to the techniques used for the aggregation process. Therefore, user feedback and an interactive model are recommended tools to fine-tune the preferences based on user constraints. This will allow the representation of a multi-objective optimization with a large number of constraints for complex problems such as credit scoring or admissions. To solve such subjective and multi-criteria optimization problems, GA-fuzzy logic and DNA-fuzzy logic models [2] are good candidates.

In the case of the GA-Fuzzy logic model, the fitness function will be defined based on user constraints. For example, in the admissions problem, assume that we would like to select students not only on the basis of their achievements and criteria defined in Table 3, but also on the basis of diversity which includes gender distribution, ethnic background distribution, geophysical location distribution, etc. The question will be

"what are the values for the preferences and which criteria should be used to achieve such a goal?" In this case, we will define the genes as the values for the preferences and the fitness function will be defined as the degree by which the distribution of each candidate in each generation match the desired distribution. Fuzzy similarity can be used to define the degree of match.

The following important points have been found in this study:

- No single ranking function works well for all contexts
- Most similarity measures work about the same regardless of the model
- There is little overlap between successful ranking functions
- The same model can be used for other applications such as the design of a more intelligent search engine which includes the user's preferences and profile [2,3].

Acknowledgement

Funding for this research was provided by the British Telecommunication (BT) and the BISC Program of UC Berkeley.

References

[1] Fair, Isaac and Co.: http://www.fairisaac.com/.
[2] M. Nikravesh, 2001. Perception-based information processing and retrieval: application to user profiling, 2001 research summary, EECS, ERL, University of California, Berkeley, BT-BISC Project. (http://zadeh.cs.berkeley.edu/ & http://www.cs.berkeley.edu/~nikraves/ & http://www-bisc.cs.berkeley.edu/).
[3] M. Nikravesh, 2001. Credit Scoring for Billions of Financing Decisions, Joint 9th IFSA World Congress and 20th NAFIPS International Conference. IFSA/NAFIPS 2001 "Fuzziness and Soft Computing in the New Millenium", Vancouver, Canada, July 25-28, 2001.
[4] P.P. Bonissone, K.S. Decker, 1986. Selecting Uncertainty Calculi and Granularity: An Experiment in Trading; Precision and Complexity, in Uncertainty in Artificial Intelligence (L. N. Kanal and J. F. Lemmer, Eds.), Amsterdam.
[5] M. Mizumoto, 1989. Pictorial Representations of Fuzzy Connectives, Part I: Cases of T-norms, T-conorms and Averaging Operators, Fuzzy Sets and Systems 31, 217-242.
[6] R. Fagin, 1998. Fuzzy Queries in Multimedia Database Systems, Proc. ACM Symposium on Principles of Database Systems, 1-10.
[7] R. Fagin, 1999. Combining fuzzy information from multiple systems. J. Computer and System Sciences 58, 83-99.
[8] U.S. Citizens for Fair Credit Card Terms; http://www.cardratings.org/cardrepfr.html.
[9] University of California-Berkeley, Office of Undergraduate Admission, http://advising.berkeley.edu/ouars/.
[10] Stanford University Admission, http://www.stanford.edu/home/stanford/facts/undergraduate.html

Table 1. Variables, Granulation and Information used to create the Credit Rating System Model.

AOA: Amount owed on accounts is too high. 01	AOA={'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LDA: Level of Delinquency on accounts. 02	LDA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
BRA: Too few bank revolving accounts. 03	BRA ={'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
BorNRA: Too many bank or national revolving accounts. 04	BorNRA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
RILI: lack of recent installment loan information: 04	RILI = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
ACB: Too many accounts with balances. 05	ACB = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
CFA: Too many Consumer finance accounts. 06	CFA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
APH: Account payment history too new to rate. 07	APH = {'Too New'; 'New'; 'Kind of New'; 'Established'; 'Well Established'; 'Not Care'};
RI: Too many recent inquiries in the last 12 months. 08	RI = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
AOinL12M: Too many accounts opened in the last 12 months. 09	AOinL12M = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
PBtoCLRI: Proportion of balances to credit limits is too high on revolving accounts. 10	PBtoCLRI= {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
AORI: Amount owed on revolving accounts is too high.11	AORI = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LRCH: Length of revolving credit history is too short. 12	LRCH ={'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long'; 'Not Care'};
TD: Time since delinquency is too recent or unknown. 13	TD = {'Too Recent'; 'Recent'; 'No Recent';'Unkown'; 'Not Care'};
LCH: Length of credit history is too short. 14	LCH= {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long'; 'Not Care'};
LRBRI: Lack of recent bank revolving information. 15	LRBRI = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
LRRAI: Lack of recent revolving account information. 16	LRRAI= {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
RNMBI: No recent non-mortgage balance information. 17	RNMBI= {'Too Recent'; 'Recent'; 'No Recent';'Unkown'; 'Not Care'};
NAwD: Number of accounts with delinquency. 18	NAwD = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
ACPasA: Too few accounts currently paid as agreed. 19	ACPasA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many';'Not Care'};
TDPRorC: Time since derogatory public record or collection. 20	TDPRorC ={'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
APDonA: Amount past due on accounts. 21	APDonA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
SDDPRorC: Serious delinquency, derogatory public record, or collection. 22	SDDPRorC= {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
BorNRAwB: Too many bank or national revolving accounts with balances. 23	BorNRAwB = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
RB: No recent revolving balances. 24	RB = {'Too Recent'; 'Recent'; 'No Recent'; 'Not Care'};
LILH: Length of installment loan history 25	LILH= {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
NRA: Number of revolving accounts. 26	NRA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
BNRorORA: Number of bank revolving or other revolving accounts. 26	BNRorORA ={'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
ACPasA: Too few accounts currently paid as agreed. 27	ACPasA = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
NofEA: Number of established accounts. 28	NofEA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
DofLI: Date of last inquiry too recent. 29	DofLI = {'Too Recent'; 'Recent'; 'No Recent'; 'Not Care'};
BB: No recent bankcard balances. 29	BB = {'Too Recent'; 'Recent'; 'No Recent'; 'Not Care'};
TRAO: Time since most recent account opening too short. 30	TRAO = {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
AwRPI: Too few accounts with recent payment information. 31	AwRPI = {'Too Few'; 'Few'; 'Some'; 'Many'; 'Too Many'; 'Not Care'};
AOonDA: Amount owed on delinquent accounts. 31	AOonDA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LofrILI: Lack of recent installment loan information. 32	LofrILI = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
PofLBtoLA: Proportion of loan balances to loan amounts is too high. 33	PofLBtoLA = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
LTOILE: Length of time open installment loans have been established * 36	LTOILE = {'Too Short'; 'Short'; 'Average'; 'Long'; 'Too Long';'Not Care'};
NFCAERLFH: Number of finance company accounts established relative to length of finance history 37	NFCAERLFH = {'Too Low'; 'Low'; 'Average'; 'High'; 'Too High';'Extremely High'; 'Not Care'};
SDPRCF: Serious delinquency and public record or collection filed X 38	SDPRCF = {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
SD: Serious delinquency X 39	SD = {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
DPRCF: Derogatory public record or collection filed X 40	DPRCF = {'Not Serous'; 'Serious'; 'Very Serious'; 'Extremely Serious'; 'Not Care'};
LRHFALFA: Lack of recent history on finance accounts, or lack of finance accounts * 99	LRHFALFA = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};
LRIALAL: Lack of recent information on auto loan, or lack of auto loans * 98	LRIALAL = {'Lacking'; 'Not Enough'; 'Enough';'Not Care'};

Table 2. Variables, Granulation and Information used to create the Credit Card Ranking System Model.

```
% Vis: Vissaa
% VisG: Vissaa Gold
% VisP: Vissaa Platinum
% MSCS: Masters Cards
% MSCSG: Masters Cards Gold
% MSCSP: Masters Cards Platinum
% Amaexs: Americana Experesses
% APR: Annual Percentage Rate
% APRC: Cash Advance APR
% AF: Annual Fee
% GP: Grace Periods
% CAF: Cash Advance Fee
% IIR: Introductory Interest Rate
% RBP: Rebate Programs
% FVR: Fix vs. Variable Rate
% GF: General Fee
% CF: Consumer Feedback
% RI: Reputation of Issuer
% FF: Frequet Flyer
% CA: Card Acceptability
% RCF: Return Check Fee
% LPF: Late Payment Fee
% SI: Security Interest
% DO: Dispute Option
% CS: Customer Service
% SPP: Special Payment Plan
% PP: Partner Programs
% IYR: Itemize Annual Report
```

```
CARDName= {'Vissaa'; 'Vissaa Gold'; 'Vissaa Platinum'; 'Masters Cards'; 'Masters Cards Gold'; ...
      'Masters Cards Platinum'; 'Americana Experesses'; 'Not Care'};
  APR={'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  APRC={'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  AF= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  GP={'Extremely Short'; 'Very Short'; 'Short'; 'Medium'; 'Long'; 'Very Long'; 'Not Care'};
  CAF=  {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  IIR= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  RBP= {'No Rebate'; 'Some Rebate'; 'Good Rebate'; 'Great Rebate'; 'Not Care'};
  FVR= {'Fix Rate'; 'Not Quite Fix'; 'Not Quite Vaiable'; 'Variable'; 'Not Care'};
  GF=  {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  CF= {'Vey Bad'; 'Bad'; 'Not Bad'; 'Average'; 'Good'; 'Great'; 'Not Care'};
  RI=  {'Vey Bad'; 'Bad'; 'Not Bad'; 'Average';  'Good'; 'Great'; 'Not Care'};
  FF={'No Frequent Flyer'; 'Some Frequent Flyer'; 'Good Frequent Flyer'; 'Great Frequent Flyer'; 'Not Care'};
  CA= {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  RCF = {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  LPF = {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  SI =  {'Very Low'; 'Low'; 'Average'; 'High'; 'Very High';'Extremely High'; 'Not Care'};
  DO={'No Disbute'; 'Some Disbute'; 'Good Disbute'; 'Great Disbute'; 'Not Care'};
  CS = {'Vey Bad'; 'Bad'; 'Not Bad'; 'Average'; 'Good'; 'Great'; 'Not Care'};
  SPP= {'No Option'; 'Some Option'; 'Good Option'; 'Great Option'; 'Not Care'};
  PP={'No Partner'; 'Some Partner'; 'Good Partner'; 'Great Partner'; 'Not Care'};
  IYR ={'Yes', 'No'};
```

Table 3. Variables, Granulation and Information used to create the University Admission System Model.

% AP : Advanced Placement	EthnicName = {'American'; 'Chinese'; 'French'; 'Greek'; 'Indian'; 'Irish'; 'Italian'; 'Japanese'; 'Mediterranean ';'Persian'; 'Spanish'; 'Taiwanese', 'Not Care'};
% IBHL : International Bacculaureat Higher Level (IBHL)	Residency={'California Resident'; 'US Resident'; 'International', 'NotCare' };
% HW: Honors and Awards	Sex={'Male'; 'Female, 'Not Care' };
% GPA: 12th Grade Courses GPA	Minority={'No'; 'Yes'; 'Not Care'};
% CP: Course pattern	HW= {'Few'; 'Some'; 'Lot'; 'Not Care'};
% GPAP: Pattern of Grades through time	AAA= {'Kind of Active'; 'Active'; 'Exceptional'; 'Not Care'};
% SAT II	CP={'Less Than Required'; 'Required'; 'Recommended'; 'Above Recommendation' };
% SAT I	Concern={'Kind of Concern'; 'Concern'; 'Very Concern'; 'Enthusiast'};
% CAoSI: Creative Achievement or Sustained Intellectual	Motivation={'Kind of Motivated'; 'Motivated'; 'Highly Motivated'; 'Enthusiast'};
% AAaO: Academic Achievement and Outreach	IMajor={'Kind of Interested'; 'Interested'; ' Very Interested'; 'Enthusiast'};
% CIaCV: Contribution to the intellectual and cultural vitality	AP= {'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% DPBaE: Diversity in the Personal Background and Experience	IBHL= {'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% Leadership	SATI={'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% Motivation	SATII={'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% Concern: Concern for Community and others	GPA= {'Very Low'; 'Low'; 'Medium'; 'High'; 'Very High' };
% AAA: Achievements; Art or Athletics	Employment={'Few'; 'Average'; 'Kind High'; 'High'; 'Lot'};
% Employment	CAoSAI= {' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional' };
% IMajor: Interest in the Major	AAaO={' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional'};
	CIaCV={' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional'};
	DPBaE={' Low Diversity'; 'Kind Low Diversity'; 'Diverse'; 'Kind of High Diversity'; 'High Diversity'; 'Exceptional'};
	Leadership={' Low'; 'Kind Low'; 'Average'; 'Kind of High'; 'High'; 'Exceptional' };

The E-Business Technologies: Past, Present, and Future

Ming-Chien Shan
Hewlett Packard Laboratories, Palo Alto, California
Email: mcshan@exch.hpl.hp.com

Electronic business will become the trademark of the 2000s. Today, enterprises view the Internet presence as a logical extension of their existing business models in terms of business operations, distribution channels and marketing media. To compete successfully, enterprises are demanding effective ways to implement best-practices processes for electronic business on the Internet and beyond. Many companies have focused their development on the front-end supporting personalized web-based interfaces. However, the backend operations are often not adequate to support the offering, especially causing many dotcoms failed in fulfillment of their orders.

In this talk, I will review these technologies and their applications in various E-business domains. The main topics to be covered include:

- What are E-commerce, E-business, E-service and M-service?
- What operational system supports are required?
- Major supporting technologies and their e-business applications.

An Interface that maps Intent to Functionality: The Agent-Oriented Approach

Babak Hodjat
Dejima Inc.

San Jose, CA, USA
Babak@dejima.com

Makoto Amamiya
Kyushu University,
Department of Intelligent
Systems
Fukuoka, Japan
amamiya@is.kyushu-u.ac.jp

Abstract

Interfaces are getting more complex due to the number of functions they provide and the limitations of our access mechanisms (i.e., the five senses). We need to reverse the paradigm in which we contort the expression of our intent to the dictates of machine functionality and instead turn functionality into an expression of our intent, thus placing the burden of mapping our intent to the functionality on the shoulders of the application interface. In this paper we propose an Agent Oriented representation of the functionality of an application in which the agents try to interpret the intent of the users and map it onto the functionality. The Adaptive Agent Oriented Software Architecture (AAOSA) uses a community of collaborative claim-based and message-driven components known as agents as a basis for constructing software. This semantic approach makes it possible for applications to better incorporate context into the user interactions and make the resulting applications more flexible and easy to use. We will describe an AAOSA-based application and a comparison study of it to an existing interface.

1. Introduction

The best interface to a machine with a single function is probably a single button. Making the button smarter would probably not be much of an improvement. There is a simple correlation between the operation of the button and our means of operating it. This is the reason why first remote controls and the first Graphical User Interfaces (GUIs) were simple to use. The complexity of the interface used to be easily managed. This is no longer true; today's applications have more complex functions, and even when the functions are simple, the sheer number of them makes them hard to access through conventional interfaces. This is where the new UI paradigm, with buttons that listen, would make a difference. Instead of adding buttons to a UI with limited real estate, the approach allows for scalable UIs by simply having the existing UI listen for the new functionality.

The buttons in the new paradigm still represent a function, and in addition they are also responsible for interpreting the intent of the user beyond recognizing the user physically pushing the button. In our approach we represent the functionality of a system, i.e., the buttons, with independent software modules we call *agents*. Our agents are smart enough to coordinate their decisions with one another. Thus, if multiple agents believe that they are the best candidates based on their interpretation of your intent, coordination mechanisms make sure the agents with the strongest case will serve you. There are times when you intend more than one function to be activated, and there are times when the agents simply cannot make a decision based on the available criteria. In these cases, the agents make themselves known by interacting with you to resolve any uncertainty. This is the only time you need to become aware of the existence of the agents. As a side effect, a user will learn the functionality of the system through interactions, and these interactions only occur if the system fails to map intent to functionality. Using this approach, the buttons disappear; the system is now interpreted in terms of its functionality rather than in terms of its UI. Using the TV set as an example, without the conventional UI you would not have to memorize the TV channel numbers. In the new paradigm, you would not even have to know the network names like ABC and ESPN. The important information would be the type of program *you* would want to see. Leaving it to the agents to map the program types to appropriate channels and network names. This behavior of mapping behavior, making claims and collaborating is the essence of the *Adaptive Agent Oriented Software Architecture* (AAOSA).

2. AAOSA and Natural Language Interfaces (NLIs)

AAOSA was designed for Agent Oriented Software Engineering [1] [2]. AAOSA provides a programming environment that supports designers and developers by managing the complexities of how to communicate between agents, and how to coordinate the community of agents [3]. A direct result of this is that agent definitions are declarative rather than procedural. These properties enable designers that use AAOSA to focus on the task of modeling the real application instead of spending time on basic support functions.

AAOSA imposes no preferred size on the collaborative agents. Rather, the granularity of the network of agents corresponds to the complexity and inherent parallelism of the application. After designing the architecture of the agent network, which in the case of the UI mimics the semantic structure of the application, the software designer provides each agent with a set of interpretation policies. The interpretation policies are a set of rules used to decide when that particular agent should return a claim to handle part of the input. The interpretation criterion is not limited to the message content of the input. Other factors such as, process history, probabilities, and outside information (e.g., interaction with other agents) may be used to derive a claim. These interpretation policies are defined using OPAL, a declarative programming language designed as part of AAOSA.

The interpretation policies are rules defined in terms of claims, delegations, and disambiguation actions. Agents use policies to decide if they are responsible for processing any part or all of the input. An agent that claims an input can either process that input, delegate the input to its down-chain agents, or both. Agents can also make use of other contextual clues in making claims. When there are conflicting claims or the agent cannot choose the best claim to propagate up-chain, an ambiguity has occurred and the disambiguation process is triggered. This may in turn trigger an interaction with the user to determine their intent.

No centralized control is enforced in a network of AAOSA agents. In this architecture, agents introduce themselves and their abilities to one another at the beginning or during execution. Agents can therefore be added to or removed from the application at runtime.

The AAOSA Software Engineering platform applies powerful Distributed Artificial Intelligence (DAI) techniques [4][5][6] which bring about the following capabilities:

Complete encapsulation of responsibilities: Each agent is responsible for all its data structures and operations and the decision to employ them.

Emergence: When input to a system is unpredictable or restricted, we rely on emergent behavior to achieve the desired output. The emergent behavior is based on agent interactions and is the result of coordination mechanisms in the core of AAOSA. Applications can be designed based on input strings paired with the expected output, relying on the emergent behavior to provide the best result for previously unencountered input.

- Learning: Each agent has learning capabilities to improve its behavior over time.
- Extendibility: The functionality of an agent network is extended by adding agents or by merging with other agent networks. This will cause minimal disruption in the original network behavior.
- Reusability: Agents and agent sub-networks can be reused in different applications and across multiple applications.
- Wrapping legacy code: Existing code can be wrapped and used as an agent in the network.
- Fault tolerance: Removing an agent or a link from the network only reduces the functional scope and does not affect the overall application. Conflicts that occur due to agents misfiring claims are handled by the coordination mechanism in the core system.
- Distributability: This comes as a result of the system being message-based.
- Multiple entry and exit points: Any agent, regardless of level, accepts multiple simultaneous inputs to the system. In the same manner, output from the system can originate from multiple agents within the agent network.

When AAOSA is applied to a Natural Language Interface (NLI), the interpretation policies should be much less restricted than the production rules of a grammar parser. For instance, rather than requiring the claims on which a new claim is based to be in sequence, we can require them only to be exclusive. The interpretation policies will determine what the best reduction condition is and each agent will compute a confidence factor for its claims based on how much its claim differs from the desired. Using a threshold, claims of higher confidence are used as results. Another main difference between the parser and our proposed natural interaction system is that the context considered in the reductions of a context-sensitive grammar is limited to the input. In the real world, the decision to make a claim may be made

based on context information that is not necessarily present in the input. For instance, an agent may decide to make a claim based on the history of successful claims made, or the status of the semantic domain it is representing, or even based on interactions with the user.

3. Case Study: The Financial Application

Online systems for getting stock information such as Yahoo Finance are in wide use these days but suffer from a usability problem noted in the introduction. This problem arises from the limitation of the interfaces to such applications in mapping users' intents to the provided functionality. We have applied the proposed AAOSA methodology to the creation of an NLI for the finance application (www.dejima.com/demos).

3.1 The functionality and scope

The first step in creating an NLI is to determine the scope and functionality of the application. The functionality should be created using two main sources:

1) Functionality available to the users in existing applications,
2) The expectations of the users from the system: If the users expect something that is technologically possible it should be handled by the NLI.

Users' expectations can be collected through Wizard of Oz tests [7]. These tests are also useful for creating a corpus, which the development of the agent network will be based upon.

The functionality of the system included displaying 19 different information on S&P 500 companies such as price, annual high and low, dividend, volume, yield, ... (e.g., "What is IBM trading at?"). This information would be presented in tables where they could be compared and the format and information in the tables would be completely dependant on user requests (e.g., "Volume and dividend for Sun and Coke"). Price and volume of different companies could also be charted against each other in different price ranges (e.g., "Plot DIS vs. INTC"). Context could be used to add companies or parameters to a given query. The user set this context by marking a check box on the screen. The users could also filter out companies based on criteria they set on the above-mentioned parameters (e.g., "What stocks have ask greater than close?").

Based on initial tests other functionality was also added to the system, including functionality to support such queries as: "List all stocks", "Show company names", "What is Dejima's address", "What is p/e ratio?".

3.2 The Agent Network

The first step is to create a prototype system to be used in user test iterations. In order to do so a corpus is fabricated which at least contains one sample command per application functionality.

Figure 1 shows a portion of the agent network created for the Finance NLI. The networks such as the ones under NUMBERS, DURATION, and DATE, are examples of agent sub-networks that have been reused for this project. There are a total of 90 agents in this network. Each agent in the network includes interpretation/actuation/ambiguity policies written in a declarative language called OPAL. For instance the OPERATOR agent includes a simple policy as follows:

```
(P30: 'above' | 'over' | '>' | '> than' | 'or greater'
        {title:'greater than operator'}
        {action: {execute:'&gt;'}}),
```

Policy P30 includes an interpretation portion which is triggered upon the existence of tokens such as 'above', 'over', '>', ... If this claim is delegated to by the up-chain agent, RHS, the action '&gt' is reported. The RHS agent in turn includes the following policy:

```
(P10: OPERATOR (['the'|'their']) PRICE
        {title:'current price'}
        {action:
        {execute:'<operator>',OPERATOR,'</operato
r><rhs>',PRICE,'</rhs>'}}),
```

This policy is selected if the claim from OPERATOR is possibly followed by 'the' or 'their' and a claim from the PRICE agent. If this claim is delegated to, the actuation string returned by OPERATOR, '&gt', is inserted in the appropriate place in the action being suggested by this policy. As you can see in this case the agents are collaborating in completing an XML actuation output.

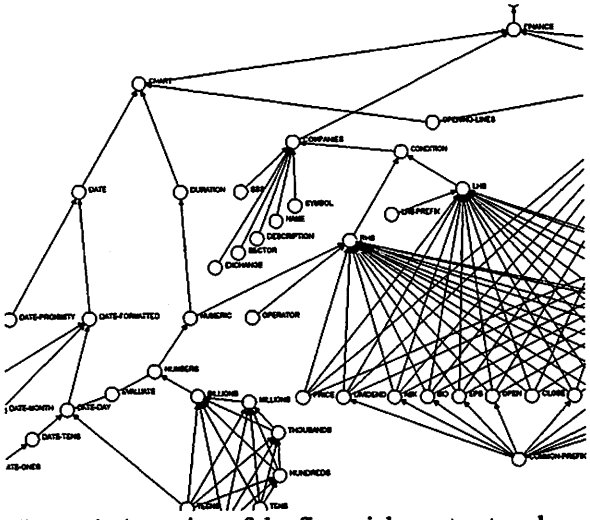


Figure 1. A portion of the financial agent network.

3.3 Testing and Tuning

17 people participated in three rounds of user testing, each followed by tuning of the agent network and its interface. Each user spent approximately 2 hours working with the system. A total of 29 scenarios were given to users. 10 of the scenarios left it up to the users to freely enter whatever query they liked. The other 19 prompted for specific actions such as:

- There is a particular stock that you have been researching. You want to know it's current trading price and volume.
- Check how high Tektronix, Sunoco and Starbucks went during the last year and what they are currently trading.
- You need a list of companies that have a P/E less than 10 and a current trading price less than 15 dollars.
- In addition to the previous question, add volume and closing prices to the list.

The total development time including the user tests was about two months. We used 9 people for the final testing round. In general, users found the application to be quite useful and would like to have this technology applied to their respective financial applications. They were able to find more information faster. The final results show an 82.3% success rate with 9.2% of the entries being classified as out of defined functional scope. The corpus had 1486 entries.

3.4 Comparison Studies

After the deployment of the final system a comparison study between the Agent-based Financial application and Yahoo Finance (http://finance.yahoo.com/) was conducted. Users found the interface highly intuitive, easy to use, productive, and an enjoyable experience. They also found the ability of the system to maintain context very useful. 5 people took part in these tests each performing 15 tasks with both systems.

The disparity among the agent-based application and Yahoo increases dramatically when the user does not know the ticker symbol, wants a comparison of several companies or needs to search companies using specific parameters such as daily high, volume, P/E... For example:

- Check how high Tektronix, Sunoco, and Starbucks went during the past year and what their current price is.
- Get a list of companies that have a P/E ratio less than 10 and a current trading price less than 15 dollars.

Average mouse click with the agent-oriented interface was 2, as opposed to 8 in the case of Yahoo. Average time per query with the AAOSA interface was 36 seconds as opposed to 61 seconds spent with Yahoo Finance. For instance for the query "Find a list of companies with a price greater then $25 and a P/E greater than 15" on average users spent 28 seconds with the agent-oriented interface and 108 seconds Using Yahoo Finance.

3. Conclusions

The problem of creating an NLI is a complex one for which AAOSA provides an elegant solution. Ungrammatical and unpredicted input will be accepted by the system, which downgrades gracefully to a dialogue system to get the user to the desired functionality. Different inputs can be used to accomplish the same functional result. Contextual information can be used to resolve contradicting claims from agents. Interacting with the user can also be used as a means to resolve ambiguities. This brings about the means to carry out smart dialogues where users may deviate from answering the system directly by adding extra information or changing the context established in prior interactions of the dialogue.

4. References

[1] N. R. Jennings and M Wooldridge (2000) "Agent-Oriented Software Engineering" http://www.mmrg.ecs.soton.ac.uk/nrj/publications/agt-handbook.pdf, in Handbook of Agent Technology (ed. J. Bradshaw) AAAI/MIT Press. (to appear)

[2] Iglesias, C.A., Garijo, M., & Gonzalez, J.C. A Survey of Agent-Oriented Methodologies. In Intelligent Agents V—Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL-98), Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 1998.

[3] B. Hodjat, M. Amamiya, Introducing the Adaptive Agent Oriented Software Architecture and its Application in Natural Language User Interfaces, in *Agent-Oriented Software Engineering*, Springer-Verlag, 2000.

[4] G. Agha, P. Wegner, A. Yonesawa, Research directions in concurrent object oriented programming. The MIT Press: Cambridge, MA, 1993.

[5] Edmund H. Durfee and Jeffrey S. Rosenschein. Distributed problem solving and multi-agent systems: Comparisons and examples. In Proceedings of the 13th International Workshop on Distributed Artificial Intelligence (IWDAI-94), pages 94-- 104, 1994.

[6] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Trends in cooperative distributed problem solving. IEEE Transactions on Knowledge and Data Engineering, 1(1):63--83, March 1989.

[7] N. Dahlback, A Jonsson, L. Ahrenberg, Wizard of Oz Studies – Why and How, *Intelligent User Interfaces '93*, ACM, 1993.

Incorporating Fuzzy Ontology of Term Relations in a Search Engine

Dwi H. Widyantoro and John Yen
Department of Computer Sciences
Texas A&M University
College Station, TX 77843-3112, USA
dhw7942, yen@cs.tamu.edu

Abstract

This paper presents our approach to automatically build a fuzzy ontology of term relations from a collection of abstracts of paper. The ontology describes whether a term is narrower or broader than other terms. The construction of ontology involves two steps: creating a full ontology using the notion of fuzzy *narrower term relation*, and pruning the full ontology to eliminate excessive and unnecessary term associations. Preliminary results from directly evaluating the quality of ontology suggest that the basic approach adopted as well as the technique developed for practical use are intuitively promising. The ontology built using this approach has been incorporated in a domain-specific search engine namely Personalized Abstract Search Services (PASS) to help refine a user's query.

1. Introduction

Web-based search engines have become common tools to locate information in cyberspace. Typical search engines retrieve information based on keywords given by users and return the information found as a list of search results. In spite of their popularity, keyword-based search engines have a weakness in that they often return a large list of search results with many of the top list of search results are irrelevant. This problem can be trivially avoided if users know exactly the right query terms to use. These terms are the ones that are unique or at least very specific, causing the search engines to bring only the relevant information to the top list of search results. However, such query terms are often very hard to find, and in many cases, they do not even exist. In order to get the information needed, finding the right query terms can become additional task during information seeking activity. Unfortunately, this problem has not been widely addressed by most search engines currently available on the Web.

This paper presents our strategy on refining a query term and describes how the strategy is applied in a Personalized Abstract Search Services (PASS). Our approach is based on a fuzzy ontology of term associations. Given a query term, PASS uses its knowledge about term associations to suggest a list of broader and narrower terms in addition to providing the search results based on the original query term. A term x is said to be *broader* than a term y if the semantic meaning of x subsumes or covers that of y. For example, *fuzzy system* is broader than *fuzzy controller*, while the latter term is broader than *lyapunov stability*. The definition of *narrower* term is the inverse of *broader* term definition. The fuzzy ontology of term associations is created automatically using information directly obtained from a corpus. The construction of this ontology involves two steps: (1) creating a full ontology using the notion of fuzzy *narrower term relation*, and (2) pruning the full ontology to eliminate excessive and unnecessary term associations.

In the following section, some backgrounds from which our approach departs will be described. Section 3 presents our ontology-based approach for query refinement. We then briefly discuss how to use the fuzzy ontology for query refinement and how the technique might improve the effectiveness of information seeking activities. Section 5 describes the implementation of fuzzy ontology construction in PASS search engine and shows some of its results based on current PASS abstract collection. A discussion of related works in query refinement method will be presented in Section 6, followed by conclusions in Section 7.

2. Background

The basic construct needed to build a fuzzy ontology of term associations is knowledge about the relation between two terms. The knowledge can be acquired from either a manually (e.g., *WordNet* [1]) or automatically [2, 3] constructed thesaurus. In this work, we focus on the construction of fuzzy ontology of term associations that are derived from an automatically built, co-occurrence-based thesaurus. Furthermore, we are interested in exploiting *narrower* and *broader* term relations

as building blocks in the fuzzy ontology construction.

2.1. Fuzzy Narrower Term Relation

Let $C = (a_1, a_2, \ldots, a_n)$ be a collection of articles a_is, where each article $a = (t_1, t_2, \ldots, t_m)$ is represented by a set of terms t_js.

Given a term t_j, the occurrence of t_j in article a is represented by $occur(t_j, a)$ and its membership value is defined by $\mu_{occur}(t_j, a) = f(\mid t_j \mid)$. The function f is a general membership function that takes the frequency occurrence of t_j in a as its argument. In the information retrieval community, the function f can be viewed as a normalized within document term weighting method.

Let $NT(t_i, t_j)$ denote that t_i is `narrower-than` t_j. The membership degree of $NT(t_i, t_j)$, represented by $\mu_{NT}(t_i, t_j)$, is defined by

$$\mu_{NT}(t_i, t_j) = \frac{\sum_{a \in C} \mu_{occur}(t_i, a) \otimes \mu_{occur}(t_j, a)}{\sum_{a \in C} \mu_{occur}(t_i, a)} \quad (1)$$

where $\otimes$ denotes a fuzzy conjunction operator. The equation above is equivalent with the fuzzy narrower term described in [4]. If we define f as a binary function such that

$$f(\mid t_i \mid) = \begin{cases} 1 & \text{if} \mid t_i \mid \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

then it can be easily shown that Equation 1 will collapse into the same expression regardless the selection of fuzzy conjunction operators.

The definition simply says that the membership degree that term t_i is `narrower-than` term t_j is the ratio between the number of co-occurrences of both terms and the number of occurrences of term t_i. Therefore, the more frequent terms t_i and t_j co-occur and the less frequent term t_i occurs in documents, t_i is `narrower-than` t_j with higher degree of confidence. A membership value of 1.0 is obtained when a term always co-occurs with another term. In contrast, the membership value of *narrower* term relation between two terms that never co-occur will be 0.

2.2. Fuzzy Broader Term Relation

The definition of `broader-than` relation between two terms is the inverse of the definition of terms `narrower-than` relation. Let $BT(t_i, t_j)$ denote that t_i is `broader-than` t_j. Since the meaning of statement "t_i is `broader-than` t_j" is equivalent to "t_j is `narrower-than` t_i", then

$$BT(t_i, t_j) \Longleftrightarrow NT(t_j, t_i) \quad (2)$$

Similarly, the membership degree of $BT(t_i, t_j)$, or $\mu_{BT}(t_i, t_j)$, can be defined by

$$\mu_{BT}(t_i, t_j) = \mu_{NT}(t_j, t_j) \quad (3)$$

suggesting that the computation of BT's membership value can be derived directly from the corresponding NT's membership value.

3. Fuzzy Ontology Construction

This section describes our approach in constructing a fuzzy ontology based on the narrower and broader term relations. The technique employed can also be considered

In general, the fuzzy ontology construction can be grouped into two stages. The first stage is to create a full ontology from fuzzy narrower term relations. The full fuzzy ontology is then pruned by eliminating unnecessary relations in the second stage.

3.1. Building Fuzzy Ontology from Fuzzy Narrower Term Relation

A fuzzy ontology can be constructed by first calculating the membership values of two *NT* relations for each pair of two distinct terms (e.g., $\mu_{NT}(t_i, t_j)$ and $\mu_{NT}(t_j, t_i)$). A set of tests is then applied to select an *NT* relation that will be incorporated in the fuzzy ontology. The selection process at this stage will eliminate redundant, less meaningful and unrelated term relations.

3.1.1 Redundant Term Relation Elimination

For each pair of terms t_i and t_j, we can have $\mu_{NT}(t_i, t_j)$ and $\mu_{NT}(t_j, t_i)$ where it is highly likely that $\mu_{NT}(t_i, t_j) \neq \mu_{NT}(t_j, t_i)$. Suppose μ_{NT}(*fuzzy logic, fuzzy controller*)=0.3 and μ_{NT}(*fuzzy controller, fuzzy logic*)=0.8. These two instances of *NT* relation express basically the same concept instances but with different degree of truth. The two concept instances are valid in the sense that concept with higher membership degree is closer to the concept truth. By rounding the membership values of above fuzzy concepts to the nearest two-value truth (e.g, *true* or *false*), for example *NT*(*fuzzy logic, fuzzy controller*) is *false* (negative concept instance) and *NT*(*fuzzy controller, fuzzy logic*) is *true* (positive concept instance), the validity of the two concepts is more obvious. Both concept instances are equivalent.

The above example demonstrates that whenever one computes $\mu_{NT}(t_i, t_j)$ and $\mu_{NT}(t_j, t_i)$, one can get a redundant information. Eliminating one of the term relations will not reduce the information conveyed but will reduce by half the size of the storage needed to maintain the same amount of information.

In constructing an ontology, we retain the fuzzy `narrower-than` relation that has higher membership value, and delete the relation with lower membership degree. This decision strategy will choose a positive concept instance if one of the membership values is far apart than another (e.g., as example above), or the strategy will choose a stronger relation if the two membership values are close to each other (e.g., 0.8 and 0.9).

3.1.2 Less Meaningful Term Relation Elimination

After redundant term relations are removed, many potential less meaningful information intact. Consider the following example: μ_{NT}(*fuzzy logic, money market*)=0.01 and μ_{NT}(*money market, fuzzy logic*)=0.005. These membership values could exist if the co-occurrence between *fuzzy logic* and *money market* is very small while the individual frequency occurrence of each term is high.

Redundant information elimination process will delete NT(*money market, fuzzy logic*) and retain NT(*fuzzy logic, money market*). Although NT(*fuzzy logic, money market*) is stronger than NT(*money market, fuzzy logic*), NT(*fuzzy logic, money market*) can be considered as a negative concept instance (e.g., by rounding the truth value of the relation as above). Its inverse, BT(*money market, fuzzy logic*), is also a negative concept due to Equations 2 and 3. Negative concept instances, however, might confuse the meaning of term relations in the ontology. Therefore, the stronger NT relation between two term relations whose membership value is small should not be included in the ontology of term associations. The exclusion of this type of term relation can be done by applying an α-cut value to the stronger term relation.

3.1.3 Unrelated Term Relation Elimination

The relations between two distinct terms cannot be established if both terms never co-occur so that their membership values will be 0. It is obvious that unrelated terms should also not be considered during the ontology creation. These terms will be automatically excluded by applying the α-cut as described above.

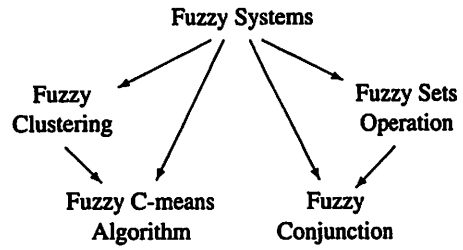


Figure 1. An ontology with excessive relations.

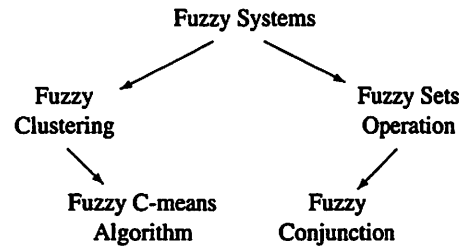


Figure 2. An ontology with a better taxonomy.

3.2. Pruning Fuzzy Ontology

In the first stage of fuzzy ontology construction, the elimination of an NT relation is based on an analysis between two NT relations generated by two distinct terms. Although the relation between two terms in the ontology is strong enough, the resulting ontology is still large and may contain unnecessary relations. More specifically, the most general terms might contain links to many or almost all other narrower terms. Figure 1 gives an example illustrating this situation. Terms such as *Fuzzy Clustering*, *Fuzzy C-means Algorithm*, *Fuzzy Sets Operation* and *Fuzzy Conjunction* definitely have narrower meaning than *Fuzzy Systems*. However, terms *Fuzzy C-means Algorithm* and *Fuzzy Conjunction* are too far to be directly connected in the taxonomy. A better taxonomy for these terms is illustrated by Figure 2.

The second stage of fuzzy ontology creation attempts to reduce the excessive relations by conducting an analysis over the set of relations involving more than two distinct terms. For each $NT(t_i, t_j)$, a search procedure is performed to find an indirect path connecting terms t_i and t_j (e.g., the sequence of $NT(t_i, t_{m1})$, $NT(t_{m1}, t_{m2})$ ... $NT(t_{mn}, t_j)$). Let P be a set of NT relations representing the indirect path for $NT(t_i, t_j)$, and NT(P) represents an alternate NT relation of (t_i, t_j) through the indirect path . The membership degree of NT(P) can be defined as the minimum membership value of NT relations in P. The idea of redundant information elimination as used during the first stage of on-

tology construction can now be applied to determine whether or not $NT(t_i, t_j)$ should be removed. If P for a given $NT(t_i, t_j)$ exists and the $NT(t_i, t_j)$ relation is not stronger than NT(P), then $NT(t_i, t_j)$ relation should be removed from the ontology description. Unlike in the first stage that removes $NT(t_j, t_i)$ whenever $NT(t_i, t_j)$ is stronger, during the second stage however, all NT relations in P remain in the ontology description if $NT(t_i, t_j)$ is stronger than NT(P).

3.3. Algorithm

The following algorithm summarizes the ontology creation described above.

1. **Definition and Initialization.**
   - $T = \{t_1, t_2 \ldots t_m\}$ is a set of distinct terms extracted from a collection C.
   - $0 \leq \alpha \leq 1$ is the α-cut.
   - `Ontology` = {} is an empty ontology description.
2. **First stage.** For each $t_i, t_j \in T$ and $t_i \neq t_j$
   - Calculate $\mu_{NT}(t_i, t_j)$ and $\mu_{NT}(t_j, t_i)$ using Equation 1.
   - Select $NT(t_p, t_q)$ subject to
     - $(t_p, t_q) =$ $\arg\max\{\mu_{NT}(t_i, t_j), \mu_{NT}(t_j, t_i)\}$
     - $NT(t_p, t_q)_\alpha$ or $\mu NT(t_p, t_q) \geq \alpha$
   - Add $\langle NT(t_p, t_q), \mu_{NT}(t_p, t_q)\rangle$ into `Ontology`.
3. **Second stage.** For each $NT(t_i, t_j) \in$ `Ontology`
   - Find P = $\{NT(t_i, t_{m1}),\ NT(t_{m1}, t_{m2}) \cdots NT(t_{mn}, t_j)\}$.
   - $(t_p, t_q) = \arg\min_{NT(t_i,t_j)\in P}\{\mu_{NT}(t_i, t_j)\}$
   - If $\mu_{NT}(t_i, t_j) \leq \mu_{NT}(t_p, t_q)$ then remove $NT(t_i, t_j)$ from `Ontology`.

The `Ontology` now contains a taxonomy description of fuzzy *NT* relations. The corresponding taxonomy description for fuzzy *BT* relations then can be explicitly generated or implicitly inferred using *NT-BT* equivalent relation as defined by Equations 2 and 3.

4. Using Fuzzy Ontology for Query Refinement

Let `Ontology` be an ontology description of *narrower* and *broader* terms, and q be a user's query term. Define two sets `NT-Set` and `BT-Set` that contain a list of narrower terms and a list of broader terms, respectively, which are obtained from ontology description `Ontology` given the query term q. The query refinement mechanism then will ask a user to select one of terms in either `NT-Set` or `BT-Set`. Let the selected term or new term be t_{NT} if it comes from `NT-Set` or t_{BT} if it is from `BT-Set`. How this new term is formulated in the new query and how it affects the search results will be discussed below.

Given t_{NT}, one can construct a new query q' by simply replacing the old query by t_{NT} (e.g., $q' = t_{NT}$) and use q' as the next search query. In this way, the query refinement method will help shifting search focus to a more specific context. One can also construct q' by conjunctively adding t_{NT} to the old query (e.g., $q' = q \wedge t_{NT}$). The effect of conjunctively incorporating the selected query refinement term is to narrow down the previous search results by focusing to the more specific context while still maintaining the context of original query. This method will be very useful if the original search results are very large. Finally, one can disjunctively add the new term with the old query (e.g., $q' = q \vee t_{NT}$). The effect is obvious; the search results will become broader by including the context from both the original query and the new term. However, if $\mu_{NT}(t_{NT}, q)$ is very high or near to 1.0, disjunctively concatenating the new term does not have a significant effect in the search results.

Unlike t_{NT}, t_{BT} has more restrictions in its use for new query reformulation. Using either $q' = t_{BT}$ or $q' = q \vee t_{BT}$ will broaden the search results but the search results from either method does not make any significant difference especially in a case where $\mu_{NT}(t_{BT}, q)$ is high. Conjunctive operator cannot be used to reformulate the new query with t_{BT} because the effect of this operator is rather counter intuitive with the intended use of t_{BT}.

5. Query Refinement in PASS Search Engine

We have implemented and incorporated the fuzzy ontology of *narrower* and *broader* terms for query refinement in PASS search engine. This section presents the overview of the system, how we use the fuzzy ontology in the system, and partial results of the constructed ontology based on the content of collection currently maintained in PASS.

5.1. PASS Overview

PASS is a domain-specific search engine providing abstracts of papers from mostly IEEE Transac-

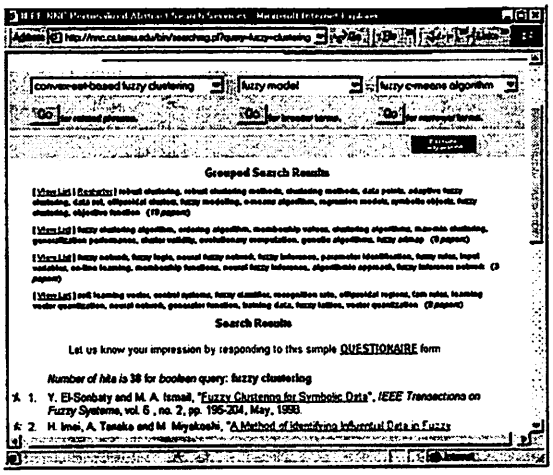


Figure 3. PASS display during search activity.

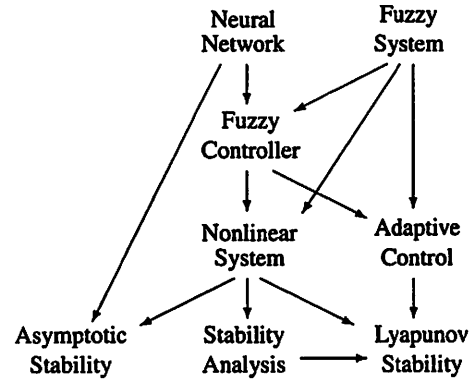


Figure 4. Partial ontology of broader terms generated by PASS search engine.

tions sponsored by the IEEE Neural Network Council (NNC). PASS is currently accessible through `http://nnc.cs.tamu.edu`.

Figure 3 shows a screen shot of PASS output in response to a user's query. In addition to displaying search results returned by the standard keyword-based retrieval, the system gives a set of key-phrases lists for query refinement. If requested by the user, PASS also provides suggestions to a list of most relevant abstracts related to the user's query based on collaborative filtering. The system's suggestions are indicated by the star icons to the left of article' titles. The system can group the search results according to the similarity of their content upon the user's request.

By clicking one of the hyperlinks in the search results, the abstract of the corresponding article title will be displayed. PASS will recommend a list of related articles under the abstract body, which is given by using content-based filtering. PASS also identifies key-phrases that occur in the abstract body and provides hyperlinks including the author-supplied key-phrases, if available, to search abstract based on these key-phrases. These hyperlinks can be viewed as another means of providing query refinement that is based on the content of article of interest. The authors of article are also hyper-linked, allowing users to search all articles written by the corresponding authors.

5.2. Incorporating Fuzzy Ontology in PASS

Fuzzy ontology of *broader* and *narrower* terms in PASS is used as a source of knowledge for PASS to generate *narrower* and *broader* terms suggestions during user search activity. If a user selects one of terms suggested, PASS will use the new term as a new search term.

The ontology is created from PASS collection that contains 584 abstracts of research papers. One of the contributing factors in constructing a high quality, automatically built ontology is the selection of appropriate terms. Specifically, terms used to build the ontology should be relevant to the target domain. These terms in PASS are acquired automatically by using a knowledge-based terms extractor, which combined the use of limited control vocabulary and natural language processing to extract new important terms in the target domain. From current collection, PASS can automatically extract 3443 technical terms.

Figure 4 illustrates a very small portion of the ontology of broader terms generated by PASS where the full ontology involves all identified technical terms (3443 terms). The corresponding ontology for narrower terms is similar to the figure except that the arrows point to opposite directions.

6. Related Work

Clark et *al.* built an *Expert Locator* to find a human expert related to a given query term [5]. Their query refinement strategy incorporated in the system is based on a semantic network. The use of semantic network to develop a model process of query refinement is also studied in earlier work [6]. Similar to PASS approach, the query refinement in *Expert Locator* also uses the notions of broader and narrower terms to describe the association between two terms. Despite the similarity in the knowledge structure, PASS's ontology and Expert Locator's semantic net differ in their construction process. First, PASS's ontology is created automatically from the scratch based on information found in its collection, while the semantic net in *Expert Locator* is created from an existing (hand-crafted) thesaurus. In the ontology construction process, PASS also builds a thesaurus-like matrix based on narrower term relation. However, the

final ontology in PASS is obtained by pruning the thesaurus, while Expert Locator's semantic net is obtained by enhancing the thesaurus (e.g., creating a new link).

HyPursuit is a hierarchical network search engine that cluster search results for browsing [7]. The system employs query refinement mechanism using broader and narrower terms derived from a thesaurus. The thesaurus is automatically constructed based on the term frequency of co-occurrence on document clusters level. The broader or narrower terms relationship is then established between the high- and low-frequency terms according to the similarities between the distribution functions of the term frequency. *HyPursuit* also suggests *collocated* terms in its query refinement mechanism. These terms are selected from the highest weighted terms in a cluster that matches with the given query.

Vèlez et *al.* studied two generic query refinement algorithms [8]. The first algorithm, DM, extracts terms from the top *n* ranked documents retrieved using original query. Terms are then weighted and the top *m* highest weighted terms are selected as a set of term suggestions. The second method, RMAP, uses DM algorithm to pre-compute the set of *m* term suggestions for every term in the corpus offline. Given a query term, the RMAP algorithm will select terms for query refinement by looking up and ranking the pre-processed term suggestions.

Prior works related to query refinement are *query expansion* [9] and *relevance feedback* [10]. Query expansion is an automatic process of expanding a user's original query with related terms from thesauri [11] or documents found in the top list of retrieval results [12]. The relevance feedback, on the other hand, modifies the original query using terms obtained from the user's document relevance feedback.

7. Conclusions

The notion of *narrower* term relation adopted by PASS and the technique developed to prune the resulting excessive narrower terms list has been shown to work well. Additionally, the relations between the list of *narrower & broader* terms and a given term intuitively make sense. The query refinement generated by the technique is thus undoubtedly very promising to be useful to help users find the information they need.

References

[1] C. Fellbaum, *WordNet, an Electronic Lexical Database*, MIT Press, 1998.

[2] H. Chen, B. Schatz, T. Yim, and D. Fye, "Automatic Thesaurus Generation for an Electronic Community System, " *Journal of American Society for Information Science*, 46(3), pp. 175-193, 1995.

[3] G. Ruge, "Experiment on Linguistically-based Term Associations," *Information Processing and Management*, 28(3), pp. 317-332, 1992.

[4] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Boston, Kluwer Academic Publisher, 1990.

[5] P. Clark, J. Thompson, H. Holmback and L. Duncan, "Exploiting a Thesaurus-based Semantic Net for Knowledge-based Search," *Proc. of the 17th National Conference on AI / 12th Conference on Innovative Application of AI*, Menlo Park, CA, 2000, pp 988-995.

[6] H. Chen and V. Dhar, "Online Query Refinement on Information Retrieval System: A Process Model of Searcher/System Interaction," *Proceedings of the thirteenth international conference on Research and development in information retrieval*, 1990, pp 115 - 133.

[7] Weiss et *al.*,"HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering," *Proc. of the Seventh ACM Conference on Hypertext*, Washington, DC, March 1996.

[8] Vèlez et *al.*,"Fast and Effective Query Refinement," *Proc. of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, Philadelphia, PA, pp 6-15.

[9] M. Mitra, A. Singhal and C. Buckley, "Improving Automatic Query Expansion," In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 206 - 214). Melbourne, Australia: ACM Press.

[10] J.J. Rocchio, "Relevance Feedback in Information Retrieval," In *G. Salton, The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313-323), 1971, Englewood Cliffs, NJ: Prentice-Hall.

[11] G. Salton, "Another Look at Automatic Text-Retrieval System," *Communication of ACM*, 29(7):648-656, July 1986.

[12] J. Xu and W.B. Croft, "Query Expansion using Local and Global Document Analysis, " In *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 4-11, Zuricch, Switzerland, August, 1996

A Framework Approach for B2B Test Automation

Danis Yadegar
Arsin Corporation
Email: dyadegar@arsin.com

Abstract

All too often the IT Software Quality Department is a backwater of under-funded outdated manual processes that largely fail to provide the precise metrics associated with testing efforts in other engineering disciplines. But this situation is poised for rapid change. Today, as companies rush to deliver tightly integrated B2B collaborative e-commerce systems, the cost of failure is dramatically increasing – with the potential for a single failure to adversely affect multiple firms. And the weakest links are often the software that is behind the Corporate firewall.

This presentation will focus on the unique demands B2B (both supply and demand chain) integration makes on a Corporation's testing practices. A conceptual taxonomy of B2B failure types will be presented, followed by a description of a testing framework that provides a layered architecture for implementing test strategies associated with a wide range of B2B failure types – including functionality testing, performance assessment, security verification, and fault tolerance. This framework also provides a model for dealing with application that support multiple presentations layers – including Voice/VXML, lite wireless, traditional HTML clients, XML-based peer-to-peer services, etc.). The framework presented here abstracts these interface types and provides a model for test case, data, and framework services (e.g., event logging) sharing across many interface types.

Operations Research and Management Science Applications of Fuzzy Theory *

I.BURHAN TÜRKŞEN
President IFSA
Director, Information / Intelligent Systems Laboratory
Mechanical and Industrial Engineering
University of Toronto
Toronto, Ontario, M5S 3G8
CANADA
turksen@mie.utoronto.ca

Abstract: - Operations Research, OR, and Management Science, MS, methodologies attempt to aid "human decision-making" for the improvement of various performance indecies in operation and control of complex "humanistic systems". A historical review of Zadeh's papers and fuzzy system applications reveal that OR, and MS applications were started early in the development of fuzzy systems, even before the advent of fuzzy control applications. Furthermore the development of fuzzy systems applications continued in OR and MS in the shadow of fuzzy control. It is forecasted that the novel application of fuzzy systems will appear more frequently in OR and MS for managerial decision support and control in the areas of strategic and tactical planning, resource allocation, scheduling, inventory control, logistics, health care and financial planning, etc., under the headings of mathematical programming, quality control, network analysis and control, consumer preference analysis, client credit worthiness, financial portfolio analysis, medical diagnosis, internet and network analysis and design, etc. In particular it is forecasted that Type 2 fuzzy knowledge representation and reasoning will be a key development area in the new millennium. Type 2 fuzzy system models expose uncertainties and risks associated with the real life system behaviours and help management to make better decision for complex humanistic systems. In the new millennium, we need to conduct further research on human information processing capabilities in order to comprehend the impact of fuzzy theory, CWW and perception on OR and MS.

Key-Words: - Humanistic Systems, Decision-Making, Operation Research, Management Science, Industrial Engineering, Fuzzy System Models.

1 Introduction

In order to understand, the development of the "Operations Research, OR, and Management Science, MS, Applications of Fuzzy Theory", it is essential to comprehend the message that Professor Zadeh has been attempting to communicate to us. Naturally what follows is my personal perspective based on my re-reading his papers and listening and re-assessing his many lectures in numerous conferences. I believe his main thesis has been that, in humanistic systems, human reasoning and decision making is not just "measurement" based, as we are taught through out our academic education, rather "perception" based. Furthermore, our knowledge and decisions are communicated between humans with words of natural languages which strongly suggest "Computing with Words", CWW. It is, I believe, this thesis, that resided in the deep recesses of his genius and commenced with "Fuzzy Sets" in 1965 and came to surface toward the beginning of this Millennium in "Toward a Perception-Based Theory... " (Zadeh, 2000).

In R. Hodge's (2000) words " `Fuzzy Logic` was born out of Zadeh's acute sense of different logic(s) inherent in human language and thought" and " his ... concern for the strengths as well as weaknesses of natural languages in scientific thought..." Zadeh uses `fuzzy` "to apply to categories of language or thought, not to the nature of (physical, mechanistic) phenomena". His use of "fuzzy" is an "example of his genius with language" with a background in " Indo-European languages", i.e., Russian and Iranian, and Turki languages, i.e., Azeri-Turkish. Deep roots

* Supported in Part by Natural Sciences and Engineering Research Council of Canada, Nortel Networks and Information Intelligence Co.

of 'fuzzy' logic can be found in R.Hodges(2000) paper. For example, Hodge states "...'fuzzy' comes

from a word (fusus) that refers ... to energies, particles as well as liquids: to a world with unstable outlines, a world in flux. The family of English words that descend from it reflects the range of meanings of fusus. They include 'infuse', 'con-fuse' and 'transfuse' from 'melted or joined', 'diffuse' from 'spread out, extended' and 'profuse' and 'effuse' from 'pour out in abundance'. All these words are formed by the addition of a prefix to 'fusus' to limit or constrain the fuzzy range of meanings of fusus to a more specific (but still somewhat fuzzy) meaning."

Zadeh came, I believe, to realized the limitations placed on scientific thought processes by the classical paradigms while he was working on his publications that appeared prior to 1965, e.g., "Linear Systems Theory-The State Space Approach"(1963) "Frequency Analysis"(1950), "Wiener's theory of prediction"(1950), "Sample-Data Systems"(1952), etc.

His concern therefore with human decision-making processes in scientific thought brought him in contact and initially in conflict with the defenders of Probability theory and later with OR and MS. [I remember OR Conferences where Professor Zadeh was an invited guest speaker addressing large audiences. (A further personal note: I was introduced to Professor Zadeh in summer 1970 at NATO Conference on OR Education that was held in Istanbul, Turkey, by R. Machol, then the President of ORSA, Operation Research Society of America.)]

It is in this context, I would like to say first a few words about OR and MS and then trace Zadeh's works and attempt to show his impact on OR and MS and then review some of the essential contribution of the pioneers who contributed to OR and MS applications of Fuzzy theory.

Briefly, it can be said that Professor Zadeh initially thought that fuzzy theory would impact human decision making processes and therefore what appeared to be a starting point for this would be OR and MS. Because OR and MS began to introduce mathematical models and solutions to humanistic systems at the end of WWII.

It should be recalled that "humanistic" engineering approach had started with Fredrick W. Taylor in the late 1800's. He is often called "the father of scientific management". Taylor was the founder of a school that included C. Barth, H.L. Grant, H. Emerson, and Frank and Lillian Gilbreth. They have designed tools, equipment and systems mathematically that took into account human limitation and capabilities that are physical and physcological.

When we trace the successes of fuzzy logic, we find that initially it started to impact OR and MS investigations; but later the successes in fuzzy control realized in Japan and later in Europe and North America have shadowed its impact on Decision Sciences. Thus the fuzzy logic applications in management decision support area were fewer in comparison to many successes in fuzzy control.

2 OR and MS Application

OR and MS studies started with the application of mathematical models to strategic and tactical military operations related to military decision making processes, during WWII. Later, OR and MS studies became an essential component of the curriculums in Departments of Industrial Engineering, Operations Research, OR, and Management Sciences, MS, and Systems Management Engineering at the Universities starting in the '50s.

There are good many operational issues that are investigated in OR and MS with the purpose of discovering and disseminating knowledge about planned, coordinated and controlled activities of people, machines, materials, money, energy and information. Some of these include planning, allocation, and distribution of resources; analysing, scheduling and controlling these activities; analysis and control of the quality of goods. Some typical examples are planning production of goods, spare parts, inventory control, scheduling processes, tasks and orders, materials, information, energy, and capital; analysis, planning and controlling waiting lines, forecasting and predicting demand, marketing and assessment of consumer and/or client preferences and demand patterns, etc. These operational and logistic issues need to be dealt with in every activity of every day human life by executive and managerial decision-making in manufacturing, process and production industries, in

healthcare systems, in government agencies, in financial institution, etc. Some of the essential methodologies that are utilized in OR and MS are: Linear and non-linear programming, probability, statistics, and stochastic process, mathematics, i.e., set and logic theories, algebra and calculus, etc.

Let us next review some of Professor L.A. Zadeh's messages. In his seminal paper "Fuzzy sets" (1985) he states:"...fuzzy sets...may prove to have a much wider scope of applicability, particularly in the fields of pattern classification and information processing."

It is known that the first sceptics of fuzzy sets were probability theorist which included some OR and MS theorist. They have confused the agenda of fuzzy sets and the agenda of probability theory and, I believe, they were somehow threatened by fuzziness.

At any rate, Professor Zadeh responded with his paper on "Probability Measures of Fuzzy Events" (1968) "showing how the notion of a fuzzy event can be given a precise meaning in the context of fuzzy sets".

Ten years later, Professor Zadeh gave further insights with his interpretation of probability theory when he introduced the possibility theory in "Fuzzy Sets as a Basis for a Theory of Possibility"(1978). In that thesis, he advocates that "...when our main concern is with the meaning of information – rather than with its measure (in Wiener and Shannon sense of the statistical theory of communication)- the proper framework for information analysis is possibilistic rather than probabilistic in nature...".

It can be said that the first impact of fuzzy theory was demonstrated in "Decision-making in a Fuzzy Environment" by R.E.Bellman and L.A.Zadeh(1970) where it was stated that "By decision-making in a fuzzy environment is meant a decision process in which the goals and/or the constraints, but not necessarily the system under control are fuzzy in nature". Further, it is stated that "The use of these concepts is illustrated by examples involving multi-stage decision processes in which the system under control is either deterministic or stochastic".

This may be considered the forerunner of the OR and MS applications of fuzzy theory [Kacprzyk (1982), Kacprzyk and Yager (1985).

Professor Zadeh next published his very important work "Outline of a new Approach to the Analysis of Complex Systems and Decision Processes"(1973). He described this "approach...(to be)...a substantive departure from the conventional quantitative techniques of system analysis"...(which) "has three main distinguishing features:1) Use of so called 'linguistic' variables in place of or in addition to numerical variables; 2) characterization of simple relations between variables by fuzzy conditional statements; and 3) characterization of complex relations by fuzzy algorithms."

This work was a landmark paper. On the basis of the ideas proposed in this paper Mamdani and Assilian (1981) developed first fuzzy control model. This then lead to the industrial applications of fuzzy control.

In 1975, Professor Zadeh, in his celebrated paper, "Concept of a Linguistic Variable...", states: "One of the fundamental tenets of modern science is that a phenomenon can not be claimed to be well understood until it can be characterized in quantitative terms". He further states "Unquestionably...(this has) proved to be highly effective in dealing with mechanistic systems, that is, with inanimate systems whose behaviour is governed by laws of mechanics, physics, chemistry and electromagnetism. Unfortunately, the same cannot be said about humanistic systems, which-so far at least-have proved to be rather impervious to mathematical analysis and computer simulation"..."It may be argued, as we have done in...(previous writings), that the ineffectiveness of computers in dealing with humanistic systems is a manifestation of what might be called the principle of incompatibility- a principle which asserts that high precision is incompatible with high complexity"..."In retreating from precision in the face of overpowering complexity, it is natural to explore the use of what might be called linguistic variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language"..."What is...important, ..., is that by use of so-called extension principle, much of the existing mathematical apparatus of systems analysis can be adapted to the manipulation of linguistic variables. In this way, we may be able to develop an approximate calculus of linguistic variables which could be of use in wide variety of practical applications". This may be considered the essential

message for the beginning of fuzzy systems applications in OR and MS. It is at about this juncture we begin to see works on "optimization of fuzzy Systems", "fuzzy mathematical programming "(Zimmermann, 1978), etc. The years between 1965 and 1975 may be considered incubation years where essential basic works on fuzzy mathematics, cognitive and decision process were being to be developed by Kaufmann(1975), Zadeh, Fu, Tanaka, Shimura(1975), Neogita and Ralescu, (1975).

Next, Zadeh published "A Theory of Approximate Reasoning"(1979) where he states that "Informally, by approximate or, equivalently, fuzzy reasoning, we mean the process or processes by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. Such reasoning, is, for the most part, qualitative rather than quantitative in nature, and almost all of it falls outside of the domain of applicability of classical logic."

In 1983, Zadeh published "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems" where he states "...the convensional approaches to the management of uncertainty in expert systems are intrinsically inadequate because they fail to come to grips with the fact that much of the uncertainty in such systems is possibilistic rather than probabilistic in nature. As an alternative, it is suggested that a fuzzy-logic-based computational framework be employed to deal with both possibilistic and probabilistic uncertainty within a single conceptual System".

During this period, we begin to see substantial works on "decision making and expert systems" and their applications to OR and MS, e.g., Zimmermann, 1987; Negoita, 1981, 1983. There were naturally many additional works on fuzzy mathematics and fuzzy optimization, as well, measurement of membership functions came forward at this period, e.g., D.Dubois and H.Prade, 1980; Negoita and Stefanescu, 1982; Gupta and Sanchez, 1982; Norwich and Türkşen, 1981, 1982; Kaufmann and Gupta, 1985.

In these pioneering works we observe investigations on: membership functions, fuzzy relations, fuzzy logic and inference, classification and similarity measures, expert systems, medical diagnosis, psychological measurements and human behaviour, fuzzy clustering algorithms, individual and group decision-making in fuzzy environments, fuzzy mathematical programming, multi-criteria decision-making, and decision support systems.

Two years later, Professor Zadeh published "Syllogistic Reasoning in Fuzzy Logic and its Application to Usuality and Reasoning with Dispositions" where he states: "Fuzzy logic may be viewed as a generalization of multi-valued logic in that it provides a wider range of tools for dealing with uncertainty and imprecision in knowledge representation, inference and decision analysis". In between these last two papers, he published "A theory of Commonsense Knowledge" where he states "...The conventional knowledge representation techniques based on the use of predicate calculus and related methods are not well-suited for the presentation of commonsense knowledge because the predicates in propositions which represent commonsense knowledge do not, in general, have crisp denotations"..."More generally, the applicability of predicate calculus and related logic systems to the representation of common sense knowledge reflects the fact that such systems make no provision for dealing with uncertainty. Thus, in predicate logic, for example, a proposition is either true or false and no gradations of truth or membership are allowed". It is important to note the separation of "gradation of truth or membership" in Zadeh's words. Unfortunately, research in this area is rather limited, e.g., Narazaki and Türkşen, (1994).

In the nineties, we have been told again and again that fuzzy logic plays a central role in human reasoning and Computing With Words(Zadeh, 1996), and that it is a key for the enhancement of scientific reasoning in management support systems as decision-making aids. An example of a work in this direction is Computing With Words(Wang, 20001). More recently, Professor Zadeh(1999) stated: "There are three basic concepts that underly human cognition: granulation, organization and causation"..."The theory of fuzzy information granulation (TFIG) is inspired by the ways in which humans granulate information and reason with it."..."The point of departure in TFIG is the concept of a generalized constraint."..."The principle modes of generalization in TFIG are fuzzification (f. generalization); granulation (g-generalization); and fuzzy granulation (f.g-generalization)". Again research in these areas appears to be non-existent in general. Although some of these concepts appear in

some papers in special application, e.g., Shanahan(2000).

In more recent years, we find there are quite a few fuzzy theory applications in OR and MS. Examples of these are "Quality Control and Maintenance", "Ecological Modeling and Data Analysis" "Fuzzy Logic and Possibility Theory in Biomedical Engineering", "...Computer Aided Medical Decision Systems", "Strategic Planning", "Decision and Planning in R&D", "Production Planning and Scheduling", "Fuzzy Sets Methodologies in Actuarial Sciences", "Fuzzy Sets in Human Factors and Ergonomics", "...Software Methodology and Design Tools"(Zimmermann, 1999). As well, we find: "Retrieving Information", " Decision-Making", "Designing and Optimization" (Dubois, Prade and Yager, 1997), "Scheduling Under Fuzziness" (Slowinski, Hapke, 2000), "Optimization and Decision" (FSS, Vol. 119, No 1, 2001).

Thus, we observe that there is a resurgence of fuzzy theory applications in OR and MS. However, we need to work on further developments of fuzzy theory in particular on Type 2 fuzzy knowledge representation and reasoning. This is more acutely needed in the development of humanistic decision making domains which Professor Zadeh have been urging us to direct our attention over the last thirty five years or so.

Over the years: it should be noted that the sceptics of fuzzy theory have been asking two questions.

(i) If fuzziness is to deal with imprecision why are the membership functions so precise?

(ii) Why do fuzzy theory use the same formulas of the two-valued theory for "AND", "OR", "IMP", etc.?

Type 2 theory responds positively to these questions: (i) Type 2 membership exposes uncertainty in the acquisition of membership functions and as well as a representation of perception with information granulation (Burillo and Bustince, 1995, 1996; Bilgic and Türkşen, 1997, 2000; Norwich and Türkşen, 1981, 1982, 1984; Karnik and Mendel, 1998, 1999, 2000; Liang and Mendel, 2000); (ii) Type 2 reasoning brings to surface the increase in uncertainty in the combination of uncertain concepts identified by words by the application of Fuzzy Disjunctive Canonical Forms, FDCF and Fuzzy Conjunctive Canonical Forms, FCCF.(Türkşen, 1999, 2001; Gerhrk, Walker and Walker, 2000; Resconi and Türkşen, 2001). Thus in Type 2 theory we can represent uncertainty more effectively and expose risks associated with decision making and hence provide a more effective tool for managerial decision making in OR and MS.

3 Conclusions

It is reasonable to conclude that in the new millennium there will be many more applications of fuzzy theory in OR and MS and other domains of "humanistic systems". For "Management of uncertainty is an intrinsically important issue in the design of ... systems because much of the information in the knowledge base ... is imprecise, incomplete or not totally reliable" (Zadeh, 1983). As well "The conventional ... use(s) of predicate calculus ... are not well suited for the representation of commonsense knowledge because the predicates in propositions... do not, ...have crisp denotations. ...Most Frenchmen are not tall can not be represented as a well-formed formula in predicate calculus..." (Zadeh, 1984). Furthermore, Zadeh observes "... in its traditional sense, computing involves ...manipulation of numbers and symbols. By contrast, humans employ mostly words in computing and reasoning, arriving at conclusions expressed in words from premises expressed in a natural language or having the form of mental perceptions" (Zadeh, 1996).

It is to be observed naturally, that in this information age of telecommunications, we rely heavily on natural languages to express our thoughts, our perceptions and our decisions in management of corporations as well as in our everyday activities.

Decisions on operational, financial, healthcare, and environmental systems, etc., are complex and data we have to rely on are often imprecise or appear to be unrelated even though we have access to large data bases.

We may have access to terabytes or more data stored in data warehouses, but to analyse them efficiently and effectively, we need to use fuzzy data mining and fuzzy system modeling techniques. The forerunners of these are OR and MS applications.

In this regard, we can surmize the importance of CWW and Perceptions by observing the essential levels and elements of information processing in Human Communications. (Table 1)

Table 1. Levels of Information Processing in Human Communication

LEVEL	INFORMATION PROCESSING	EXAMPLES
I Senses	Symbol	Sound, line, color
II Recognition	Signal	Language, contours (shape)
III Understanding (of micro knowledge)	Local meaning	Meanings of words, single objects
IV Understanding (of macro knowledge)	Global meaning	Meanings of sentences, complex objects
V Understanding (of emotions and intentions)	Impression, conception	Association, imagination, the arts, personality

References:

[1] R.E. Bellman and L.A. Zadeh, "Decision-Making in a Fuzzy Environment", *Management Science*, 17:4 (1970) 141-164.

[2] T. Bilgic, I.B. Türkşen, "Measurement-Theoretical Frameworks in Fuzzy Theory", in: *Fuzzy Logic in Artificial Intelligence*, A. Ralescue, T. Martin (eds.), Springer-Verlag, Berlin (1997), 552-565.

[3] T. Bilgic, I.B. Türkşen, "Elicitation of Membership Functions: How Far Can Theory Take us?", *Proceedings of Fuzzy-IEEE '97*, July 1-5, 1997, Barcelona, Spain, Vol. III (1997), 1321-1325.

[4] T. Bilgic, I.B. Türkşen, "Measurement of Membership Functions: Theoretical and Empericial Work", in: *Handbook of Fuzzy Theory*, D. Dubois and H. Prade (eds.) (2000).

[5] P. Burillo, H. Bustince, "Intuitionistic fuzzy relations Part I", *Mathware Soft Comput.* 2 (1995) 5-38.

[6] P. Burillo, H. Bustince, "Entropy on Intuitionistic Fuzzy Sets and on Fuzzy Sets and on Interval Valued Fuzzy Sets", *Fuzzy Sets Syst.* 78 (1996) 305-316.

[7] H. Bustince, P. Burillo, "Interval Valued Fuzzy Relations in a Set Structures", *J. Fuzzy Math.* 4 (1996) 765-785.

[8] D. Dubois and H. Prade, "Fuzzy Sets and Systems: Theory and Applications, *Academic Press*, New York, (1980).

[9] D. Dubois, H. Prade and R.Yager, Fuzzy Information Engineering, Wiley, New York, (1997).

[10] D.Dubois and H.Prade, (eds) Special Issue on Optimizatin and Decision, Fuzzy Sets and Systems, 119, 1, (2001).

[11] M. Gerhrk, C. Walker and E. Walker, Fuzzy Normal Forms and Truth Tables, Proceedings of JCIS- 2000, Feb. 27-March3, 2000 Atlantic City, New Jersey, 211-214.

[12] M.M. Gupta and E. Sanchez (eds), "Approximate Reasoning in Decision Analysis", North-Holland, New York, (1982).

[13] R. Hodge, "Key Terms in Fuzzy Logic Deep Roots and New Understanding", *University of Western Sydney*, Australia (Private Communications) (2000).

[14] N.N. Karnik, Type 2 Fuzzy Logic Systems, *PhD Dissertation*, University of Southern California, Los Angeles, CA. 1998.

[15] N.N. Karnik and J.M. Mendel, Type 2 Fuzzy Logic Systems: Type-Reduction, *Presented at 1998 IEEE SMC Conference*, San Diego, CA, October, 1998.

[16] N.N. Karnik and J.M. Mendel, Introduction to Type 2 Fuzzy Logic Systems, *Presented at1998 IEEE FUZZ Conference*, Anchorage, AK. May, 1998.

[17] N.N. Karnik, J.M. Mendel, and Q. Liang, Type 2 Fuzzy Logic System, *IEEE Trans. Fuzzy Systems*. vol. 7, no. 5, Oct. 1999.

[18] A. Kaufmann, "Introduction to the Theory of Fuzzy Subsets", Academic Press, New York, Vol.1 (1975).

[19] A. Kaufmann and M.M. Gupta, "Introduction to Fuzzy Arithmetic: Theory and Applications", Van Nostrand Reinhold, New York, (1985).

[20] Q. Liang and J.M. Mendel, Interval Type 2 Fuzzy Logic System, *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, 7-10 May 2000, San Antonio, Texas, 328-333.

[21] Q. Liang and J.M. Mendel, Decision Feedback Equalizers for Non-linear Time Varying Channels using Type 2 Fuzzy Adaptive Filters, *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, 7-10 May 2000, San Antonio, Texas, 883-888.

[22] E.H. Mamdani, "Applications of Fuzzy Algorithms for Control of Simple Dynamical Plant", *Proceedings of IEEE*, 121 (1976) 1585-1588.

[23] E.H. Mamdani, "Advances in the Linguistic Synthesis of Fuzzy Controllers", *J. Man-Mach. Studies*, 8 (1976) 669-678.

[24] E.H. Mamdani, "Applications of Fuzzy Logic to Approximate Reasoning Using Linguistic Systems", *IEEE Trans. Comput.*, 26 (1977) 1182-1191.

[25] E.H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, (in) *Fuzzy Reasoning and Its Applications* (eds) E.H. Mamdani and B.R. Gaines, Academic Press, New York, (1981) 311-323.

[26] H. Narazaki and I.B. Türkşen, "An Integrated Approach for Syllogistic Reasoning and Knowledge Consistency Level Maintennance", IEEE Trans on SMC, 24, 4 (1994) 548-563.

[27] C.V. Negoita and D.A. Ralescu, , "Applications of Fuzzy Sets to Systems Analysis", *Birkhauser*, Basel and Stuttgart, (1975).

[28] C.V. Negoita and D.A. Ralescu, , "Applications of Fuzzy Sets to Systems Analysis", *Birkhauser*, Basel and Stuttgart, (1975).

[29] C.V. Negoita, "Fuzzy Systems", *Abacus*, Tunbridge Wells, UK, (1981).

[30] C.V. Negoita and A.C. Stefanescu, "On Fuzzy Optimization", in Gupta and Sanchez (eds), *Approximate Reasoning in Decision Analysis*, North Holland, New York, (1982) 247-250.

[31] C.V. Negoita, "Expert Systems and Fuzzy Systems", *Benjamin/Cummings*, Menlo Park, Cal.,(1985).

[32] A.M. Norwich and I.B. Türkşen, "Measurement and Scaling of Membership Functions", in: Lasker, G.E. (ed.), *Proceedings of International Conference on Applied Systems and Cybernetics*, Vol. VI, Acapulco, Mexico, Dec.1980, Pergamon Press, New York, (1981) 2851-2858.

[33] A.M. Norwich and I.B. Türkşen, "Stochastic Fuzziness", in: M.M. Gupta and E.E. Sanchez (eds.), *Approximate Reasoning in Decision Analysis*, North Holland, Amsterdam, (1982) 13-22.

[34] A.M. Norwich and I.B. Türkşen, "The Fundamental Meausrement of Fuzziness", in: R.R. Yager (ed.), *Fuzzy Sets and Possibility Theory*. Pergamon Press, New York, (1982) 49-60.

[35] A.M. Norwich and I.B. Türkşen,"The Construction of Membership Functions", in: R.R. Yager (ed.), *Fuzzy Sets and Possibility Theory*. Pergamon Press, New York, (1982) 61-67.

[36] A.M. Norwich and I.B. Türkşen, "Stochastic Fuzziness", in: M.M. Gupta and E.E. Sanches (eds.), *Fuzzy Information and Decision Processes*. North Holland, Amsterdam, (1982) 13-22.

[37] A.M. Norwich and I.B. Türkşen, "A Model for the Measurement of Membership and the Consequences of its Empirical Implementation", *Fuzzy Sets and Systems*, 12(1984), 1-25.

[38] G. Resconi and I.B. Türkşen, "Canonical Forms of Fuzzy Truthoods by Meta-Theory Based Upon Modal Logic", *Information Sciences* (2001).

[39] J.G. Shanahan, Soft Computing for Knowledge Discovery: Introducing Cartesian Granule Features, Kluwer, Boston, (2000).

[40] R. Slowinski and M. Hapke, Scheduling Under Fuzziness, *Physica-Verlag*, Hiedelbeg, (2000).

[41] I.B. Türkşen, "Theories of Set and Logic with Crisp or Fuzzy Information Granules", *J.of Advanced Computational Intelligence*, 3,4(1999) 264-273.

[42] I.B. Türkşen, "Computing with Descriptive and Veristic Words: Knowledge Representation and Reasoning", *in: Computing With Words*, P.P. Wang(ed.), Wiley, New York, (2001)(to appear).

[43] P.P. Wang(ed.), Computing With Words, Wiley, New York (2001) (to appear).

[44] L.A. Zadeh, "Frequency Analysis of Variable Networks", *Proc.of IRE*, (1950) 291-299.

[45] L.A. Zadeh and J. R.Ragazzini, "An Extension of Wieners Theory of Prediction", *J.Appl.Phys.*, 21 (1950) 645-655.

[46] L.A. Zadeh and J. R.Ragazzini, "The Analysis of Sampled-Data Systems", *Applications and Industry (AIEE)*, 1 (1952) 224-234.

[47] L.A. Zadeh and C.A. Desoer, "Linear System Theory-The State Space Approach", *Mc Graw-Hill*, New York, (1963).

[48] L.A. Zadeh, "Fuzzy Sets", *Information and Control Systems*, Vol.8, (1965) 338-353.

[49] L.A. Zadeh, "Probability Measures of Fuzzy Events", *J.Math. Analysis and Appl.*, 10 (1968) 421-427

[50] L.A. Zadeh, K-S Fu, K. Tanaka, M. Shimura, "Fuzzy Sets and Their Applications to Cognitive and Decision Progresses", Academic Press, New York, (1975).

[51] L.A. Zadeh, "The Concept of a Linguistic Variable and its Application to Approximate Reasoning", Part 1, 2, and 3, *Information Sciences*, 8 (1975) 199-249, 301-357 and 9 (1976) 43-80.

[52] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility", Fuzzy Sets and Systems, (1978) 3-28.

[53] L.A. Zadeh, "A Theory of Approximate Reasoning", *in J. Hay, D. Michie, and L.I. Mirkulich (eds) Machine Intelligence*, Halstead Press, New York, Vol. 9 (1979) 149-194.

[54] L.A. Zadeh, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems", *Fuzzy Sets and Systems*, 11 (1983) 199-227.

[55] L.A. Zadeh, "A Theory of Common Sense Knowledge", *in H.J. Skala, S. Termini, and E. Trillas (eds), Aspects of Vagueness*, Dodrecht: D.Riedel, (1984) 257-296.

[56] L.A. Zadeh, "Syllogistic reasoning in Fuzzy Logic and its Application to Usuality and Reasoning with Dispositions", *IEEE-Trans.SMC*, 15 (1985) 754-763.

[57] L.A. Zadeh and J. Kacprzyk (eds), "Fuzzy Logic for the Management of Uncertainty", *John Wiley and Sons*, New York, (1992).

[58] L.A. Zadeh, "Fuzzy Logic = Computing With Words", *IEEE-Trans on Fuzzy Systems*, 4, 2(1996), 103-111.

[59] L.A. Zadeh, "From Computing with Numbers to Computing with Words—From Manipulation of Measurements to Manipulation of Perceptions", *IEEE-Trans on Curciuts and Systems*, 45 (1999) 105-119.

[60] L.A. Zadeh, "Toward a Perception-Based Theory of Probabilistic Reasoning", Key note address; *Fourth International Conference on Applications of Fuzzy Systems and Soft Computing*, June 27-29, Siegen, Germany, (2000).

[61] H.J. Zimmermann, "Fuzzy Programming and Linear Programming with Several Objective Functions", *FSS*, 1 (1978) 45-55.

[62] H.J. Zimmermann, Fuzzy Sets, Decision Making, and Expert Systems, Kluwer, Boston, (1987).

[63] H.J. Zimmermann(ed.), Practical Applications of Fuzzy Technologies, *Handbook of Fuzzy Set Series*, Kluwer, Boston, (1999).

Multistage Fuzzy Personalization – Making Rulebases Significantly Easier to Maintain

Dipl.-Inform. Gero Presser[1]
Department of Computer Science I,
University of Dortmund
D-44221 Dortmund, Germany
Gero.Presser@QuinScape.de

Abstract
As the Internet is emerging from a seller's market to a buyer's market, it is becoming quite hard to attract customers. Probably, the most promising technique is to address customers with the content (and/or products) they are really interested in. This is the key idea of personalization – to dynamically tailor content based on information on the user's profile. Obviously, the underlying techniques cannot only be applied to personalize websites but in almost any scenario where there is repeated contact to a certain class of users.

In practical applications, rule-based personalization has become the most important technique (apart from techniques used for recommender systems which we do not consider here). Basically, simple IF-THEN rules are used to tailor the content, for example, "IF Customer bought a Siemens Phone last week THEN show all Siemens accessories".

In this paper we sketch a rule-based approach to personalization which makes use of *fuzzy* IF-THEN rules. However, it is very important to note that we do not only replace the crisp rules by fuzzy rules but suggest to also replace the (commonly used) one-stage inference process by a *multistage* inference. Only the combination of these two extensions really helps in increasing the maintainability of the rule-based personalization scheme.

The *multi*stage inference allows the computation of interim values which makes the rulebase easier to read and to state. In particular it helps in defining the rulebase in a modular fashion such that parts of it can be reused in other settings. Hence rulebases can be separated into different parts which can be compared to subroutines (that may be used for the computation of interim values).

The *fuzzification* of the rules makes it possible to handle vague concepts as they are frequently used in marketing issues. And – this is surely a point that cannot be overstressed – personalization is mainly a *marketing* question! Fuzzified rules are of particular relevance when the input variables of the respective rule are output variables of a prior inference and hence fuzzy values.

Apart from a description of the motivation and the model, we shortly introduce two applications of the fuzzy personalization model: These are the personalization of *chatterbots* (where fuzzy rules are used to control the "mood" of the softbot) as well as the personalization of *newsletters* (where fuzzy rules are used to dynamically assemble a newsletter out of some given text modules based on the profile of the respective customer).

1. Introduction

During the last decade, the amount of pages available in the World Wide Web (WWW) has definitely been exploding. According to the most recent study of the internet software consortium (isr, http://www.isc.org), approximately 110,000,000 Hosts have been connected to the Internet in January 2001.

From the standpoint of a commercial website, the problem with the web is that it is drowning in other websites making it quite hard to attract customers in the internet. In addition – since for almost any need

[1] Gero Presser is co-founder of QuinScape GmbH, a German software company that develops innovative products for personalization. More information are available on the website http://www.QuinScape.de.

nowadays there is a plenty of websites fulfilling this specific need – there is a serious and still growing problem to keep customers coming back.

To say it with other words, the internet has emerged from a seller's market to a buyer's market making it hard for sellers to attract customers.

2. The role of Personalization (p13n[2])

One of the most outstanding methods for increasing customer loyalty is personalization which can be used for implementing the one-to-one marketing paradigm (see [2]). Personalization means matching the content to the user's personality or a group of user's preferences. For example, a user who has purchased an item from the store is likely to be interested in information about that item such as promotions and new arrivals.

The term personalization is used in quite a broad sense which is sometimes misleading. Its meaning reaches from simple personal greetings (e.g., "Hello Peter Stone", as can be found on almost any website) and manually adjustable content (see for example MyYahoo!, http://www.yahoo.com and MyZDnet, http://www.zdnet.com) to automatically matched content like in the case of amazon.com. To make things even more complicated, many companies (in particular, in the Content-Management industry) speak of personalization when the user only sees the menu options that he is allowed to use.

We think it is very important to clearly distinguish between personalization and *customization*. There are other discriminations as well, but this seems to be – in our view – the most serious point. Customization means, that the user himself manually adjusts the content in order to match his needs. Whenever a site ZZZ offers something named MyZZZ, it is probably the possibility to customize the content of ZZZ. For a quite astonishing comment on the relevance of customization compared to personalization see [7]. (Most users seem to prefer customization!)

[2] It is quite common to abbreviate the term "internationalization" by i18n (an "i" followed by 18 characters followed by an "n"). Hence it would be straightforward to abbreviate "personalization" by *p13n* which is – however – not very common.

In this paper we solely investigate *personalization*. In our view, personalization means that the content is automatically tailored (probably based on the user's profile) and can hence its meaning can be clearly distinguished from customization.

Numerous techniques are being used for personalization such as demographic profiling, collaborative filtering and clickstream analysis (see for example [8], [9], and [11]).

Personalization has emerged from a "nice feature" to an important aspect for the overall acceptance of a website. Gartner Group expects, that in 2003 more than 80% of all major websites make use of personalized content to address users in a more efficient way (cf. [4]).

3. Best Practice: Rule-based Personalization

As there are many different species of personalization, there are many different techniques as well. One of the leading edge technologies certainly is rule-based personalization as it is implemented, e.g., in the Dynamo Web Server (art technology group, http://www.atg.com) or the One-to-One product suite (BroadVision, http://www.broadvision.com). For a short description and overview concerning different personalization techniques with main focus on the rule-based approach see [1].

The idea of rule-based personalization is to utilize specific information about individual visitors to precisely tailor the content displayed using a rulebase consisting of IF-THEN rules. Frequently, these rules are called *business rules*.

The business rules are usually manipulated using some rule editor and they are evaluated by a rule engine. Certainly, the most common rule engine is ILOG Rules (and its Java-pendant ILOG JRules) by ILOG (see http://www.ilog.com), which is – for example – utilized in the Dynamo Web Server and is also available as a Personalization-Add-On to the VIP Content Manager, the Content Management System of Gauss Interprise (see http://www.gauss-interprise.com).

4. Basic Principle

Rule-based personalization delivers content based on the user profiles and a set of rules defined by the web site designers. By capturing the needs, interests, preferences and motivations of the individuals that visit a website and applying business rules against that information, one can provide these visitors with information and recommendations that are of particular relevance to them. Hence, it is possible to make them aware of interesting promotions, offer them discounts based on their purchasing history, and take advantage of opportunities to cross-sell and up-sell without offending them, e.g.

- IF Customer bought a Siemens Phone last week THEN show all Siemens accessories.
- IF Customer traveled to Germany within the last year THEN send notification of German travel promotions.

In real world applications the rulebase (which represents the business logic) is maintained using some rule editor as can be found in almost any state-of-the-art product suite. (However, as stated before, in most commercial products the *same* underlying rule engine, namely ILOG Rules, is being used.)

5. Fuzzy Rules – Where and Why?

The rule-based approach towards personalization definitely has many advantages: any business rule has a clear interpretation and the approach is quite flexible allowing almost any kind of content-tailoring.

Its most serious problem probably is, that as the number of rules increases, it becomes hard to maintain the rulebase. You can do almost anything using rules but this can lead to almost any degree of complexity.

So, why are we proposing to use fuzzy rules? Does this really help in making the rulebase easier to manage?

Well, the answer to the last question certainly is "Yes and No". The key point here is that we do not only want to substitute the crisp business rules used for personalization by fuzzy rules. In fact, we propose to use additional fuzzy layers to compute interim values that can be used in the premises of other personalization rules. The combination of these two aspects *can* lead to rulebases that are easier to understand and hence easier to maintain as well.

The motivation for this approach is very easy to explain: Personalization rules are used to map specific profiles to specific content. If we would have complete knowledge concerning the profile of the customers, the definition of business rules would be an easy task. For example, we could include a rule like

> "IF customer is most interested in football THEN show all items related to football"

However – as You probably can imagine – in general we do *not* have complete information on what a customer is most interested in! Nevertheless, what we can actually do is *approximate* his interest in football (a fuzzy value!) by means of a fuzzy rulebase (and a suitable inference mechanism) using only variables that can directly be observed (or variables that have been stored in a database). Since fuzzy rules allow the formalization of vague concepts and are therefore more natural to human operators, they can significantly help in making the personalization logic easier to formalize as well as to maintain.

Recapitulating, we propose to use a multistage fuzzy system for tailoring the content. Obviously, this is a direct extension of the classical rule-based approach to personalization.

6. Model

The underlying model is almost straight-forward: There is a set I of variables that can directly be observed. In addition, there are fuzzy rulebases RB_1, ..., RB_n. A rulesbase RB_i (where $1 \le i \le n$) consists of a finite number of rules which are of the form

> IF X_1 is L_1 AND ... AND X_m is L_m
> THEN Y_j is L

where L_1, ..., L_m, L are linguistic terms. Let O_i be the set of output variables used in rulebase RB_i. All input variables X_1, ..., X_m used in RB_i must either be observable variables (i.e., in I) or output variables of a rulebase RB_k where $k < i$.

The computation is done using some (usually fixed) inference mechanism. Though there are many alternatives (e.g., the t-norm used as well as the

Figure 1: Chatterbot "Perseus"

defuzzification method), these aspects are only peripheral in this work.[3] Given the values of the observable variables, successively the inference for the rulebases $RB_1, \ldots, RB_n$ can be done and eventually the personalization can be done based on the output variables.

The output variables Y_j are fuzzy values that are mostly reused as input variables to rulebases in the upcoming stages. In practical settings, a rulebase is frequently used to compute some interim value with a clear interpretation which is then used in further rulebases as an input. Hence, it can be compared to some kind of a subroutine. For example, the interest of a given customer in some given theme can be computed using a rule base. This computed (fuzzy-)value can be applied in the premise of another IF-THEN rule in the next stage to decide, whether or not a special offering should be presented to the respective customer.

Therefore, the usage of a multistage inference helps in separating different aspects of the rulebase and hence in making it reusable. In particular, these smaller portions are easier to define and to maintain (which implies a far better debugging process if the result is not as it was expected or intended to be).

[3] In our view, these are details of *any* fuzzy inference process that are well-understood by experts of fuzzy logic and that need not be explained in this paper. Our main intention here is to sketch an important field for the application of fuzzy logic in the internet (and content-driven applications), not its details.

7. Prototype

We have prototypically implemented the sketched method in order to validate its practical relevance. The setting we used – the personalization of chatterbots and newsletters – may look somewhat odd at first glance, but these are definitely important fields for personalization methods.

A *chatterbot* is a softbot that is capable of doing simple conversation in natural language. Chatterbots are frequently used to guide users through a website or simply to entertain them or perform some news announcement. In our case we utilized a technology developed by QuinScape – the so-called *ReceptoBot-Technology*, see [12] – which allows to easily construct a chatterbot out of predefined components. See Figure 1 for a screenshot (in German).

The personalization of chatterbots is quite a new aspect which definitely is not "standard". However, it seems to be very important for the overall acceptance of chatterbots, that they have some kind of memory and are capable of finding (i.e., learning) the right way to communicate with a specific person. Hence, they need be personalized.

We utilized the fuzzy personalization approach to maintain the "mood" of the chatterbot as well as to fine-tune its reactions based on the observed conversation-history. This was done using different fuzzy rulebases and a three-stage fuzzy inference. The output computed by the rulebases are fuzzy values and so are some of the inputs. This makes the definition of the fuzzy rulebase quite easy since rules like

> IF interest-of-user-in(football) is High
> AND mood-of-user is Friendly

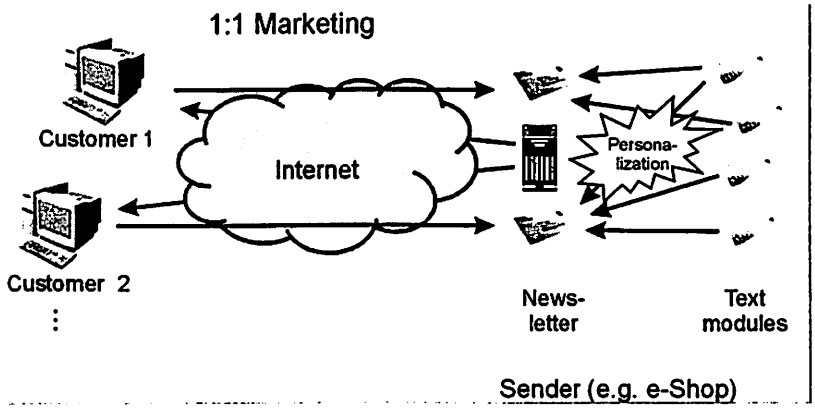


Figure 2: Personalization of Newsletters

AND current-theme-belongs-to(smalltalk) is Probable
THEN change-to-theme(football) is Promising

have a quite natural interpretation. We do not explain any details here, however it is probably interesting to know that all fuzzy-values *interest-of-user-in(football)*, *mood-of-user* as well as *current-theme-belongs-to(smalltalk)* are computed by rulebases on prior stages.

Another quite promising application for the fuzzy personalization scheme is the *personalization of newsletters*. Though the internet is changing from a pure pull- to a mixture of a pull- and push-medium, it is by far not trivial to design a successful newsletter. Experience tells that the content of the newsletter should be desired, personal, and relevant (cf. [13]).

We utilized our approach to easily construct personalized newsletters based on some user profile (see [10]). The key idea is to dynamically assemble the text modules for the "personal" newsletter and to use fuzzy rulebases (and inference) to control this assembling-process. The scheme of this process is depicted in Figure 2.

8. Summary

In this paper we have introduced fuzzy personalization which is an extension to the classical rule-based approach to personalization. The key idea is to use fuzzy IF-THEN rules (instead of crisp rules) *and* to utilize a multistage fuzzy inference.

Since the approach is quite "natural" for persons familiar with fuzzy-logic, we have omitted technical details and concentrated on a pure description.

We have used the proposed approach in practical settings – namely the personalization of chatterbots and newsletters – where it turned out that the flexibility of the approach is quite helpful and in particular its clear interpretation strongly helps in keeping the "knowledge bases" easy to maintain even by non IT-users.

9. References

[1] C. Allen, *Personalization– yesterday, today, and tomorrow*, personalization.com, Soapbox, 2000.

[2] D. Amor. *The E-business (R)evolution*. Prentice Hall, Upper Saddle River, NJ, 1999.

[3] J. Fink and A. Kobsa. "A review and analysis of commercial user modeling servers for personalization on the world wide web," *User Modeling and User-Adapted Interaction*, Vol. 10, 2000, pp 209-249.

[4] Gartner Group, *Do you know what 'personalization' means?*, Research Note, 2000.

[5] A. Kandel, editor. *Fuzzy Expert Systems*. CRC Press, Boca Raton, FL, 1991.

[6] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, Upper Saddle River, NJ, 1995.

[7] P.F. Nunes and A. Kambil. "Personalization? No Thanks." *Havard Business Review*, Vol. 4, 2001, pp 32-34.

[8] D.M. Pennrock and E. Horvitz. "Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach," *Proceedings of the IJCAI Workshop on Machine Learning for Information Filtering*, 1999.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. „Analysis of recommendation algorithms for e-commerce," *ACM Conference on Electronic Commerce*, 2000, pp. 158-167.

[10] G. Presser, "Personalization of Newsletters Using Multistage Fuzzy Inference," *Proceeding of the 6th Fuzzy Days*, to appear.

[11] J. Schafer, J. Konstan, and J. Riedl. „Recommender systems in elecotrnic commerce," *Proceedings of the ACM Conference on Electronic Commerce (EC-99)*, 1999, 158-166.

[12] T. Biskup, "Agent-Based Site Relationship Management: Eliza's Grandchildren Get A Real Job," *Proceedings of the IAWTIC'2001*, to appear.

[13] B. J. Pine II. *Mass Customization*. Havard Business School, Boston, 1993.

Dynamic Knowledge Representation for E-Learning Applications

M. E. S. Mendes, L. Sacks
Department of Electronic and Electrical Engineering, University College London
London, UK
{mmendes,lsacks}@ee.ucl.ac.uk

Abstract
In this paper, we present an approach to organize E-Learning materials according to knowledge domains, by means of fuzzy clustering analysis. A new modified version of the Fuzzy C-Means clustering algorithm has been derived to employ a non-Euclidean metric, which is common in traditional information retrieval systems. The preliminary trials with this modified algorithm show that it performs better than the original one, when used for clustering text documents.

1. Introduction

Information resources play an important role in almost every Internet service, and especially in E-Learning (Internet-based teaching and learning). Functionalities like information search and retrieval are a must. But, in the case of E-Learning, such functionalities need to be complemented by taking into account issues like, for instance, learning objectives, pedagogical approaches and learner profile. Thus, there is a context for retrieving information. It is necessary to define which learning materials are relevant to a particular user, who wants to learn about a particular subject.

To determine which of the materials available in the courseware database are relevant, two components should be considered. Firstly, the set of subjects associated to each material should be identified. Thus, there needs to be a way to classify and organize materials in terms of knowledge domains. Secondly, the learning context should be identified, since the learning goals and pedagogical models may impact on the way materials are structured and consequently, on the definition of relevant links.

In this paper we present our approach to obtain the first input for the computation of relevance. The subsequent sections are organized as follows. The CANDLE project is briefly overviewed in section 2. In section 3 issues associated with the representation and retrieval of learning materials in the project's context are presented. The argument for employing fuzzy clustering to organize learning materials and to build a domain knowledge representation is presented in section 4. In section 5, we present the background for our experiments with fuzzy clustering applied to text documents, which are then detailed in section 6. In section 7 we present our conclusions and further research issues.

2. The CANDLE Project

Presently, there are many initiatives around the world working on the development of Internet-based teaching and learning applications, both for education and training purposes. One of those initiatives is the CANDLE[1] project, which is a European Commission shared cost research project under the IST fifth framework programme. Its main focus is on the delivery of courseware, for the telematics domain, over the Internet.
The project's major long-term objectives focus on: -

(a) helping educators to be increasingly successful;
(b) increasing the flexibility of the learning process, to accommodate various pedagogical approaches and flexible usages of the learning materials;
(c) developing a suitable technical framework to allow learning materials to be sharable and reusable, enabling the rapid development and deployment of new courseware.

So, CANDLE puts a strong emphasis on the learning materials, both in terms of representation and retrieval, for allowing the flexibility and reusability required.

3. Representation and Retrieval of E-Learning Materials

In CANDLE, learning materials are described by metadata and represented in XML (eXtensible Markup Language), following the current paradigm of semantic interoperability among networked information resources. Metadata is simply descriptive information about resources like: title, authors, conditions of use, keywords, technical requirements, etc. Its fundamental role is to enhance the process of information retrieval, by providing rich and machine "understandable" representations.

[1] Collaborative And Network Distributed Learning Environment: http://www.candle.eu.org/

The project is developing its own educational metadata scheme, which specializes IEEE's LOM (Learning Object Metadata) specification [1]. The main purposes of using metadata in CANDLE's context are: -

(a) to aid in the construction of new courseware, re-using already existent materials;
(b) and to enhance the navigation and exploration of the learner.

Among other metadata fields, a category to classify the course materials, according to a pre-defined taxonomy of the telematics domain knowledge has been defined. So, whenever new material is created, its author is asked to select a set of key words from that taxonomy. Additionally, the author is asked to assign a weight to each of the selected key words to quantify their significance to the material being tagged. With this approach, metadata can be used to discover relevant material based on its *knowledge space* location.

The abstract *knowledge space* can be built in several ways. A sophisticated approach to do so is to develop an ontology of the domain, which defines a set of concepts (equivalent to the taxonomic key words) and set of relations between those concepts. So, once learning material is tagged with specific concepts, the relationships defined in the ontology can be used to locate related courseware. This facilitates the search of material both in the authoring and learning scenarios. In the former case, an author might start from one point in the ontology and follow the appropriate links to locate material with the relationships required for his course. More important for the learner the ontology can be used for navigation and possibly automated location of content – by following the relationship links relevant content can be located.

The key question with this approach is which ontology to use. Two problems can be foreseen. On the one hand different experts in a given field are likely to disagree on the correct ontology. On the other hand, in fields like engineering or telematics, an ontology can change through time as the fields develop. Thus, several ontological frameworks need to be available or a dynamic ontology creation process needs to exist.
This leads to an approach supporting a process of discovering the underlying knowledge structure and generating the relationships between learning materials that are part of the courseware database. We are exploring the use of fuzzy clustering to build such dynamic knowledge representation.

4. Fuzzy Clustering for Knowledge Representation

The common goal of clustering methods is to group data elements according to some similarity measure so that related elements are placed in the same cluster. This makes possible the discovery of unobvious relations and structures in data sets.
Information retrieval is one of the many application areas that make use of cluster analysis. Document clustering has long been applied to enhance the process of search and retrieval based on the so-called *cluster hypothesis* [2]. Document clustering has also been used as a post-retrieval tool for browsing large document collections [3].
Our objective is to discover the knowledge-relations that may exist between learning material, based on their metadata descriptions. This can be seen as a document clustering problem. A suitable algorithm may be applied for organizing the XML documents and for discovering hidden relations between them.

Agglomerative hierarchical clustering algorithms are perhaps the most popular for document clustering [4]. These methods have the advantage of providing a hierarchical organization of the document collection, but they are slower than partitional methods, like the K-Means algorithm. For this reason the latter is also very popular. Both methods generate hard clusters, in the sense that each document is assigned to one and only one cluster. Considering our objectives, a method capable of generating fuzzy clusters would the most suitable. The arguments that support this statement are the following:

(a) The first one concerns the representation of the *knowledge space*. We pointed out previously that a major problem of using an ontology was to be able to define the correct representation of the domain knowledge. Since knowledge is such an abstract thing every attempt to represent it will be to some extent uncertain. Thus, fuzzy document relations are more likely to represent the "true" knowledge structure than crisp ones.

(b) Another reason has to do with the way learning materials are classified. The selection of weighted key words represents the author's best attempt to define the subjects associated each material. But this tagging process is intrinsically imprecise, firstly because authors may differ in their exact understanding of the key words (which may be

occasionally ambiguous) and secondly, the assigned weights express a subjective opinion.

The theory of fuzzy sets provides the mathematical means to deal with uncertainty [5]. Fuzzy clustering brings together the ability to find unobvious relations and structures in data sets with the ability to cope with uncertainty. The following sections present some background and report on the document clustering experiments in which fuzzy clustering techniques have been applied.

5. Background for the Clustering Experiments

In order to apply a clustering algorithm to a document collection it is necessary to have suitable document representations. In the well-known Vector Space Model (VSM) of information retrieval [6] each document is represented by a set of indexing terms in the form of a k-dimensional vector:

$$x_i = [w_{i1}\ w_{i2} \ldots w_{ik}] \tag{5.1}$$

where k is the total number of terms and w_{ij} represents the weight (or significance) of term j in document x_i. The weights w_{ij} can be obtained in various ways. Usually an automatic indexing procedure finds the indexing terms associated with each text document and the weights are then computed as a function of the term frequencies. A term weighting system that has proved to perform well considers the term frequency, the inverse document frequency (that is, the term specificity within the document collection) and a factor of document length normalization [7]:

$$w_{ij} = \frac{f_{ij} \cdot \log(N/n_j)}{\sqrt{\sum_{t=1}^{k} (f_{it} \cdot \log(N/n_t))^2}} \tag{5.2}$$

where f_{ij} is the frequency of term j in document i, n_j is the number of documents that contain term j and N is the total number of documents.

A similar vector representation can be obtained for CANDLE's learning materials from their metadata descriptions. In this case, the indexing terms (key words from the taxonomy) and the associated weights are assigned manually by authors.

The weighted document vectors are suitable to be processed by a clustering algorithm. Regardless of the algorithm, documents will be grouped according to their similarities. Hence, it is necessary to choose an appropriate similarity measure for the document space. A familiar function that is used in the VSM to compare document vectors with query vectors is the based on inner product of the two vectors [7]:

$$S(x_\alpha, x_\beta) = \sum_{j=1}^{k} w_{\alpha j} \cdot w_{\beta j} = x_\alpha \cdot x_\beta^T \tag{5.3}$$

Since x_α and x_β are normalized weighted vectors the similarity function exhibits the following properties:

$$0 \le S(x_\alpha, x_\beta) \le 1, \forall_{\alpha,\beta} \tag{5.4}$$

$$S(x_\alpha, x_\alpha) = 1, \forall_\alpha \tag{5.5}$$

Some clustering algorithms group data elements based on dissimilarities or distances. A dissimilarity function can be obtained from the similarity measure defined in (5.3) by an appropriate transformation:

$$D(x_\alpha, x_\beta) = 1 - S(x_\alpha, x_\beta) = 1 - \sum_{j=1}^{k} w_{\alpha j} \cdot w_{\beta j} \tag{5.6}$$

Both the document vector representation and the similarity function introduced above were applied in our document clustering experiments. Next, we present the clustering techniques that were used.

5.1 The Fuzzy C-Means Algorithm

In section 4 we presented the arguments that supported the use of a clustering algorithm capable of generating a fuzzy output. Thus, we had to select one such algorithm.

The Fuzzy C-Means (FCM) [8] is one of the most popular fuzzy clustering methods. It generalizes the hard K-Means, by producing a fuzzy partition of the data space, as it is required in our case. We decided to use this algorithm in our experiments, for its simplicity and for being the fuzzy extension of a technique that is common for document clustering. Furthermore, a study presented in [9] indicates that the FCM can perform at least as well as the traditionally used agglomerative hierarchical clustering method.

The algorithm is summarized as follows. Given a data set with N elements each represented by k-dimensional feature vector, the FCM takes as input a $(N \times k)$ matrix $X=[x_i]$. It requires the prior definition of the final number of clusters c $(1<c<N)$, the choice of the fuzzification parameter m $(m>1)$ and the selection of a distance function $\|\cdot\|$, the most common being the Euclidean norm:

$$d_{i\alpha}^2 = \|x_i - v_\alpha\|^2 = \sum_{j=1}^{k} (x_{ij} - v_{\alpha j})^2 \tag{5.7}$$

The algorithm runs iteratively to obtain the cluster centers – $V=[v_\alpha]$: $(c \times k)$ – and a partition matrix – $U=[u_{\alpha i}]$: $(c \times N)$ – which contains the membership of each data element in each of the c clusters.

Both the cluster centers and the partition matrix are computed optimizing the following objective function:

$$J_m(U,V)=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^m d_{i\alpha}{}^2=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^m\|x_i-v_\alpha\|^2 \quad (5.8)$$

The FCM algorithm starts with a random initialization of the partition matrix subject to the following constraints:

1. $u_{\alpha i}\in[0,1],\ \forall_{\alpha\in\{1,\ldots,c\}}\ \forall_{i\in\{1,\ldots,N\}}$ (5.9)

2. $\sum_{\alpha=1}^{c}u_{\alpha i}=1,\ \forall_{i\in\{1,\ldots,N\}}$ (5.10)

3. $0<\sum_{i=1}^{N}u_{\alpha i}<N,\ \forall_{\alpha\in\{1,\ldots,c\}}$ (5.11)

At each iteration, the cluster centers and the grades of membership are updated according to (5.12) and (5.13) respectively:

$$v_\alpha=\frac{\sum_{i=1}^{N}u_{\alpha i}{}^m\cdot x_i}{\sum_{i=1}^{N}u_{\alpha i}{}^m} \quad (5.12)$$

$$u_{\alpha i}=\frac{1}{\sum_{\beta=1}^{c}\left(\frac{d_{i\alpha}{}^2}{d_{i\beta}{}^2}\right)^{1/(m-1)}}=\frac{1}{\sum_{\beta=1}^{c}\left(\frac{\|x_i-v_\alpha\|}{\|x_i-v_\beta\|}\right)^{2/(m-1)}} \quad (5.13)$$

The algorithm ends when a termination criterion is met or the maximum number of iterations is achieved.

5.2 The Modified Fuzzy C-Means Algorithm

The Euclidean norm, which is frequently applied in the FCM algorithm, is not the most suitable for comparing document vectors. This statement can be supported by the following example. Let us suppose that we have two documents x_A and x_B that are indexed with a set k of terms T. Let us also assume that most of the terms in T, say k', appear neither in x_A nor in x_B. Let us also assume that x_A and x_B have no terms in common. Since the two document vectors agree in k' dimensions in which they both have zero term weights, their Euclidean distance will be relatively small, when in fact x_A and x_B are totally dissimilar.

So, the problem with the Euclidean norm is that the non-occurrence of the same terms in both documents is treated in the same way as the co-occurrence of terms.

A suitable dissimilarity function for document vectors was introduced in (5.6). For the previous example this function results in the maximum value possible, that is $D(x_A,x_B)=1$, indicating total dissimilarity.

Thus, we decided to apply the dissimilarity measure as the metric for clustering documents, using the Fuzzy C-Means approach. The modified objective function is similar to (5.8), but now the norm $\|\cdot\|^2$ is replaced by the function defined in (5.6):

$$J_m(U,V)=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^m D_{i\alpha}=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^m\left(1-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right) \quad (5.14)$$

As the expression used to update of the clusters centers (5.12) was obtained considering the Euclidean distance we had to derive a new expression to work with the new metric. In order to use the dissimilarity measure, such that property (5.5) holds, the cluster centers need to be normalized. Therefore, we had to introduce the following constraint:

$$S(v_\alpha,v_\alpha)=\sum_{j=1}^{k}v_{\alpha j}\cdot v_{\alpha j}=\sum_{j=1}^{k}v_{\alpha j}{}^2=1,\forall\alpha \quad (5.15)$$

It can be proved that minimizing (5.14) with respect to $u_{\alpha i}$ leads to a similar result as in (5.13), but now $d_{i\alpha}{}^2$ and $d_{i\beta}{}^2$ are replaced by $D_{i\alpha}$ and $D_{i\beta}$,. The expression for $u_{\alpha i}$ is:

$$u_{\alpha i}=\frac{1}{\sum_{\beta=1}^{c}\left(\frac{D_{i\alpha}}{D_{i\beta}}\right)^{1/(m-1)}}=\frac{1}{\sum_{\beta=1}^{c}\left(\frac{1-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}{1-\sum_{j=1}^{k}x_{ij}\cdot v_{\beta j}}\right)^{1/(m-1)}} \quad (5.16)$$

To minimize (5.14) with respect to v_α we applied the method of the Lagrange multipliers to introduce the constraint (5.15), obtaining:

$$L=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^m\left(1-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right)+\sum_{\alpha=1}^{c}\lambda_\alpha\left(\sum_{j=1}^{k}v_{\alpha j}{}^2-1\right) \quad (5.17)$$

Then,

$$\frac{\partial L}{\partial v_\alpha}=-\sum_{i=1}^{N}u_{\alpha i}{}^m x_i+2\lambda_\alpha v_\alpha=0\Leftrightarrow v_\alpha=\frac{1}{2\lambda_\alpha}\cdot\sum_{i=1}^{N}u_{\alpha i}{}^m x_i \quad (5.18)$$

By applying the constraint (5.15) we get

$$\sum_{j=1}^{k}v_{\alpha j}{}^2=\left(\frac{1}{2\lambda_\alpha}\right)^2\cdot\sum_{j=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}{}^m x_{ij}\right)^2=1\Leftrightarrow$$

$$\Leftrightarrow\frac{1}{2\lambda_\alpha}=\sqrt{\frac{1}{\sum_{j=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}{}^m x_{ij}\right)^2}} \quad (5.19)$$

Replacing $\frac{1}{2\lambda_\alpha}$ in (5.18) we obtain

$$v_\alpha = \sum_{i=1}^{N} {u_{\alpha i}}^m x_i \cdot \sqrt{\frac{1}{\sum_{j=1}^{k}\left(\sum_{i=1}^{N} {u_{\alpha i}}^m x_{ij}\right)^2}} \qquad (5.20)$$

The new modified FCM runs similarly to the original FCM, differing only on the expressions used to update v_α and to calculate the distances.

5.3 Fuzziness of the Document Clusters

It is known that increasing values of m lead to a fuzzier partition matrix. For the reasons presented in section 4, the more fuzzy the results, the more flexible will be the use of the discovered document relations. However, there needs to be a compromise between the amount of fuzziness and capability to obtain good clusters and reason from those relations. If all documents end up with the same membership in every cluster, the conclusion will be that they are all equally related to each other.

A simple cluster validity measure that indicates the closeness of a fuzzy partition to a hard one is the Partition Entropy (*PE*), which is defined as [8]:

$$PE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}\log_a(u_{\alpha i}) \qquad (5.21)$$

The possible values of *PE* range from 0 – when U is *hard* – to $\log_a(c)$ – when every data element has equal membership in every cluster ($u_{i\alpha} = 1/c$).

6. Experiments with Fuzzy Clustering

The aim of our experiments was to investigate whether or not fuzzy clustering was suitable for our purposes. We carried out several trials to assess and compare the performance of the FCM applying different metric concepts: the Euclidean distance and the dissimilarity function. This section reports on our experiments.

6.1 Data Set Description

The process of populating CANDLE's database with learning materials has just recently started. As we had to simulate CANDLE database, we decided to work with a familiar collection of text document.

We selected a set of RFC text documents (that describe standard protocols and policies of the Internet). Each of the documents was automatically indexed with keywords from an existing taxonomy [10]. Document vectors as in (5.1) were generated and organized as rows of a ($N \times k$) matrix, where N=67 was the collection size and k=465 was the total number of indexing terms.

We manually created a clustering benchmark based on our knowledge of the documents' contents, complemented by the indexing information found in [10]. The benchmark indicated that the RFCs could be distributed into 6 fairly homogeneous clusters although some of the documents could have been attributed to more than one cluster [11].

6.2 Experimental Results

In our first trials the objective was to analyze if the FCM algorithm would be able to generate a good partition of the document collection. We fixed the number of clusters in 6 (as our benchmark indicated) and we ran the algorithm applying both the Euclidean distance (FCM-ED) and the dissimilarity function (FCM-DF). For each case we tried several values of the fuzzification parameter – $m \in [1.1,2.5]$. We fixed the conversion threshold to 10^{-4} and the maximum number of iterations to 300. For the FCM-DF trials we created a document matrix of term weights ($X_1=[w_{ij}]$) using (5.2). For the FCM-ED trials we also generated a matrix of term frequency counts ($X_2=[f_{ij}]$).

To be able to compare the results with the reference clustering we used the maximum membership criterion to generate hard clusters from the fuzzy ones. Initially we set m=1.1 so that the results would be close to the hard case. When X_1 was used as input, both FCM-ED and FCM-DF performed quite well generating clusters with a high degree of match with the benchmark. We found out that the FCM-ED performed poorly when X_2 was used even for such a low value of m, ~86% of the documents ending up in the same cluster. We also noticed that in this case for increasing values of m the execution times increased exponentially. When X_1 was used as input, the computation times were fairly stable for increasing values of m, both with the FCM-ED and the FCM-DF. This result is shown in Figure 1. From the plot we see that the FCM-DF converges faster than FCM-ED for $m \leq 1.3$ and slower for $m \geq 1.4$. Although this suggests that for increasing fuzziness the FCM-ED has lower execution times, these times just decrease because the maximum fuzziness has been achieved. In Figure 2 evidence of this is presented. We can observe that for $m \geq 1.4$ the partition entropy is maximal.

An important remark is that the FCM-DF successfully obtains fuzzier partitions. For higher values of m the matching between benchmark and hardened clusters is lower, but the partitions generated are still fairly good.

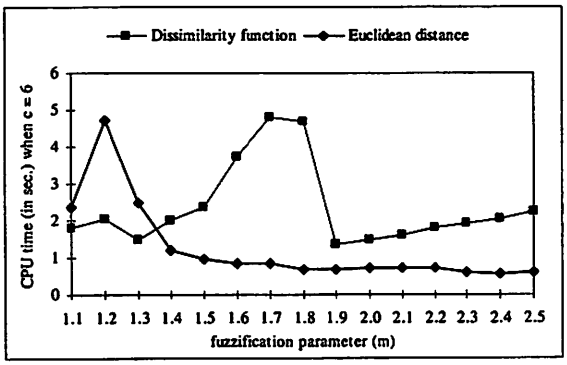


Figure 1. Comparison of the computation times for increasing values for *m*, with *c* set to 6 clusters, using X_1 as input data

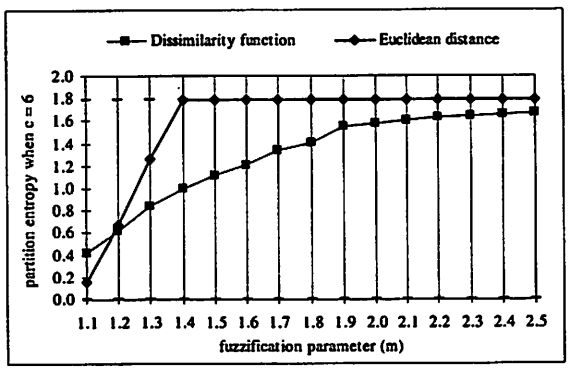


Figure 2. Comparison of the partition entropy for increasing values for *m*, with *c* set to 6 clusters, using X_1 as input data

The decrease verified on the execution times when *m* goes from 1.8 to 1.9 is due to the fact that the partition entropy has increased. This is not very evident from the plot but in fact, when *m*=1.8 around 28% of the documents still have maximum membership ≥0.5 and only 7% of them have maximum membership close to $1/c \cong 0.17$. But when m=1.9, the first statistic decreases to 18% and the second one increases to 24%.

7. Conclusions and Future Work

In this paper, we presented an approach to dynamically represent knowledge domains through the discovery of fuzzy relationships between E-Leaning materials. We proposed a new modified version of the fuzzy c-means clustering algorithm that employed a dissimilarity function common in traditional information retrieval systems. Our experiments with the RFC document collection showed that the FCM algorithm produces poor results for term frequency vectors, but when normalized weighted vectors are used the FCM successfully approximates the reference clusters. We also verified that with the Euclidean distance, good partitions were generated, but only for low values of *m*, whereas with the dissimilarity function higher degrees of fuzziness were acceptable without compromising the quality of the clusters. This is an important result that answers our requirements regarding knowledge-based organization of CANDLE's learning materials.

In the near future, our research will address issues regarding the incremental update of the fuzzy clusters to deal with new document arrivals in the database. A hierarchical organization of the learning materials based on a nested refinement of the fuzzy partitions will also be investigated.

8. Acknowledgments

This work has been supported by *Fundação para a Ciência e a Tecnologia* through the PRAXIS XXI scholarship programme.

9. References

[1] IEEE LOM Working Group. "Draft Standard for Learning Object Metadata," Nov. 2000.

[2] C. J. van Rijsbergen. *Information Retrieval.* Second Edition, Butterworth, London, 1979.

[3] D. R. Cutting, D.R. Karger, J.O. Pedersen, J. W. Tukey. "Scatter/Gather: a cluster-based approach to browsing large document collections," SIGIR'92, 1992, pp. 318-329.

[4] P. Willett. "Recent trends in hierarchical document clustering: a critical review," Information Processing and Management, Vol. 24, No. 5, 1988, pp. 577-597.

[5] L. A. Zadeh. "Fuzzy Sets," Information and Control, Vol. 8, 1965, pp. 338-353.

[6] G. Salton, J. M. McGill. *Introduction to modern information retrieval.* McGraw-Hill, New York, 1983.

[7] G. Salton, J. Allan, C. Buckley. "Automatic structuring and retrieval of large text files," Communications of the ACM, Vol. 37, No. 2, Feb. 1994, pp. 97-108.

[8] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[9] D. H. Kraft, J. Chen, A. Mikulcic. "Combining Fuzzy Clustering and Fuzzy Inference in Information Retrieval," FUZZ IEEE 2000, Vol. 1, 2000, pp. 375-380.

[10] L. Wheeler. *IETF RFC Index.* Available at: http://www.garlic.com/~lynn/rfcietf.htm

[11] M. E. S. Mendes, L. Sacks. "Assessment of the Performance of Fuzzy Cluster Analysis in the Classification of RFC Documents," Proc. of The London Communications Symposium, 2000, London.

EMERGENCE OF WEB-CENTRIC VIRTUAL ORGANIZATIONS

-A Fuzzy-Evolutionary Approach-

Mihaela Ulieru and Silviu Ionita
Electrical and Computer Engineering Department
The University of Calgary
Alberta, CANADA
ulieru@enel.ucalgary.ca
http://isg.enme.ucalgary.ca

Abstract
An approach to the emergence of web-Centric virtual organizations is introduced, within the broader *Holonic Enterprise* © framework. The first part of the paper presents the key concepts of this new e-Business model that merges latest results obtained by the Holonic Manufacturing Systems (HMS) Consortium with newly developed standards for platform interoperability released by the Foundation for Intelligent Physical Agents (FIPA), to enable the development of global collaborative e-Commerce/e-Business applications. In the second part of the paper we introduce a fuzzy-evolutionary approach useful to the identify collaborative partners in web-Centric virtual organizations such as the Holonic Enterprise. It consists of two parts. First the Fuzzy Relevancy Evaluation of the goal of each individual partner with respect to the global goal of the emergent organization is determined. Then the partners are selected via a dynamic evolutionary search in the cyberspace regarded as open domain. Implemented at the inter-enterprise level of the Holonic Enterprise, this approach enables clustering of organizations into emergent partnerships with common objectives.

Keywords. Web-centric Virtual Organizations; Collaborative Information Ecosystem; Fuzzy Relevancy Evaluation; Dynamic Evolutionary Search.

1. Introduction

A holonic enterprise [1] is a holarchy of collaborative enterprises, where the enterprise is regarded as a holon. (Here the term enterprise is used in a broad, generic manner: entity, system, 'thing', agent). The term holon was coined by Artur Koestler [2] to denominate entities that exhibit simultaneously both autonomy and cooperation capabilities which demand balance of the contradictory forces that define each of these properties on a behavioral level. One main characteristic of a holon is its multiple granularity [1]manifested through replication into self-similar structures at multi-resolution levels. This heterarchical decomposition turns out into a nested hierarchy of fractal entities – named holarchy. A holonic enterprise has three levels of granularity, Fig. 1:

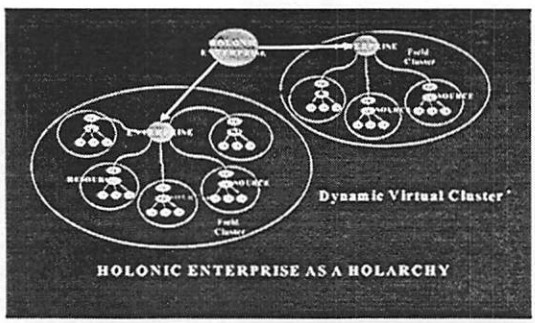


Fig 1: Dynamic Virtual Clustering Pattern in the Holonic Enterprise

1. 1 Global inter-enterprise collaborative level

At this level several holon-enterprises cluster into a collaborative holarchy to produce a product or service. Traditionally this level was regarded as a mostly static chain of customers and suppliers. In the holonic enterprise the supply chain paradigm is replaced by

[1] The term Holonic Enterprise was introduced by Dr. Mihaela Ulieru on April 3, 2001 at the FIPA Product Design and Manufacturing meeting in London, UK.

the collaborative holarchy paradigm (Fig. 1). The dynamic collaborative holarchy can cope with unexpected disturbances through on-line re-configuration of the open system it represents. It provides on-line order distribution across the available partners as well as deployment mechanisms that ensure real-time order error reporting and on-demand order tracking.

1. 2 Intra-enterprise level

Once each enterprise has undertaken responsibility for the assigned part of the work, it has to organize in turn its own internal resources to deliver on time. Planning and dynamic scheduling of resources at this level enable functional reconfiguration and flexibility via (re)selecting functional units, (re)assigning their locations, and (re)defining their interconnections. Re-configuration of schedules to cope with new orders or unexpected disturbances (e.g. when a machine breaks) is enabled through re-clustering of the agents representing the actual resources of the enterprise. The main criteria for resource (re)allocation when (re)configuring the schedules are related to cost minimization achieved via multi-criteria optimization.

1. 3 Machine (physical agent) level

This level is concerned with the distributed control of the physical machines that actually perform the work. To enable agile manufacturing through the deployment of self-reconfiguring control elements, each machine is cloned as an agent that abstracts those parameters needed for the configuration of the holonic control system managing the distributed production.

2. Patterns of Holonic Collaboration

The common mechanisms that characterize the collaborative information ecosystem created by the three levels of a holonic enterprise follow the following design patterns for adaptive multi-agent systems [3] (Fig. 2). The overall architecture of the Holonic Enterprise builds on the Metamorphic Architecture Pattern that replicates at all levels.

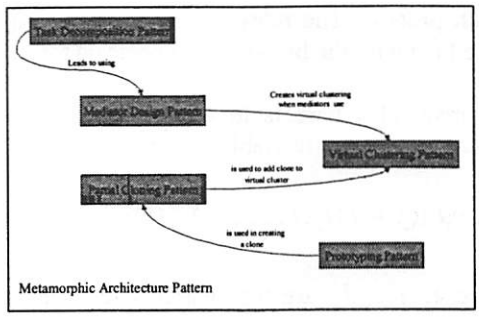


Fig 2: Pattern Interaction within the metamorphic Architecture

- **Metamorphic Architecture Pattern.** This pattern works by synergetic integration of two other patterns:
- **Dynamic Virtual Clustering** configured to minimize cost and enabling for flexible, re-configurable structures. At all levels of the holonic enterprise, task propagation occurs by a process of virtual cluster (or holarchy) formation. This pattern is facilitated by the general layered architecture of the holonic enterprise.
- **Mediator Agent Pattern** supporting the decision-making process that creates and (re)-configures the dynamic virtual clusters of collaborative entities.
- **Partial Cloning Pattern.** This pattern defines which of the enterprise's characteristics (attributes and functionality) we need to abstract into agents at each level when modeling the holonic enterprise as a collaborative multi-agent system.

In the sequel we propose a fuzzy-evolutionary approach that implemented within the mediator can optimize the dynamic virtual clustering process.

3. A Fuzzy Measure for Collaborative Partnering

In this Subsection we introduce a fuzzy approach capable to capture relevancy of the individual goals of possible collaborative partners with respect to the global goal of the emergent organization, by appropriate modeling of the linguistic ambiguity. The search problem can be defined as a mapping of the search indexes (term frequency occurrence; position of the key term in the document and its linkage to other documents, etc) onto the relevancy measure *r*, Fig. 3 [5]. The search criteria are fixed and predefined according to user interests in the beginning of the

search process. The relevancy evaluation results in a ranked list with the best-fit documents on top.

The main idea here is to express the informational relevancy as a multivariable function:

$$relevancy = r(i_1, i_2, ..., i_n) \quad (1)$$

where $i_1, i_2, ...i_n$ are the search indexes used in the informational evaluation of the document. To optimize the search means to optimize the function r (1). It is not trivial to find an adequate expression for the relevancy function to be able to get its value for each particular document.

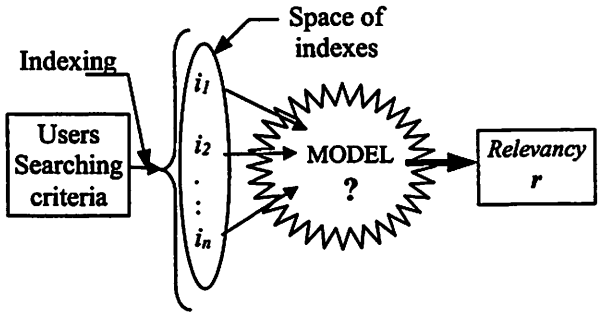


Fig. 3: Search Model

By establishing the fuzzy model [4] connecting the fuzzy sets of indexes and the weighted relevancy of the documents found a search strategy is determined as an aggregate of all indexes. The strategy for indexes aggregation is, Fig.3:

IF (i_1 =<*label*> AND i_2=<*label*> AND . . . i_7=<*label*>) THEN r =<*label*>

4. An Evolutionary Dynamic Search Strategy for Finding Collaborators in Cyberspace

Besides refining the search by accommodating the intrinsic ambiguity of natural language, the fuzzy model method has no other advantage as search strategy relatively to the existing search engines which sort the information by searching a predefined, static domain

In this subsection we introduce an iterative, incremental search strategy that expands the search domain over time. For this we use the property of global optimizer inherent in genetic algorithms. Our construction is based on the observation that the search process in a set of documents is analogous to the genetic selection of the most relevant ones in a population of documents.

The genetic operators **pm** – mutation probability and **pc** – crossover probability in the probabilistic model that generates the new structures (genotypes) in the evolutionary processes can be defined for a population of documents in a dynamic, open search domain. In this way the most relevant documents with respect to the search indexes will be naturally selected as 'best' through the evolutionary process. The intrinsic property of global optimizer that the evolutionary process exhibits leads naturally to increase the gradient of the relevancy (1), Fig. 3 as it searches for the 'best' solution in each 'generation' of documents. This analogy is detailed on a complex case study in [5]. Now both steps of the search (the relevancy evaluation and the ranking of documents) can be done concurrently, Fig. 4 as the search domain can expand continuously and new documents can enter it at each iteration/'generation'. This gives the search a dynamic context.

To achieve this we shall use the general property of Genetic Algorithms (GAs) to find the global optimum as the best search results [6].

We propose a model based on the following assumptions:

- The search is an *evolutionary process* in the space of indexes aiming to optimize the document relevancy as an objective function.

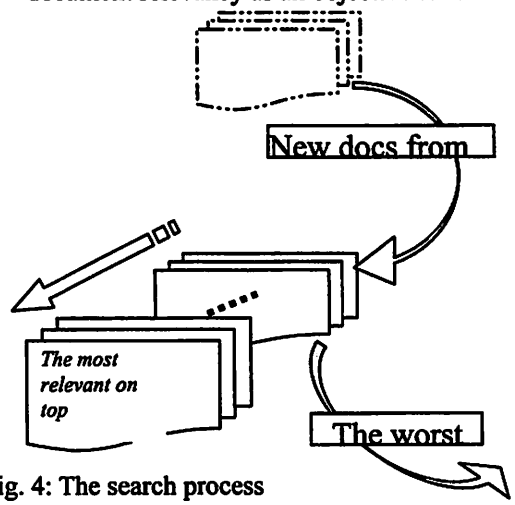


Fig. 4: The search process

The search domain is open (expanding after each search) but relevancy estimation and documents ranking are made on a partial domain (a predefined population) each time.

- At each iteration (i.e. new domain expansion) the documents are reached according to their location in the current partial search domain. The lowest relevant documents will be replaced with newfound ones in the extended domain.
- The final result is obtained when the entire defined population contains the most relevant documents.
- The results reflect always the best partial relevancy ranking and depend on when the user stops the continuous incremental spiral search process in the practically infinite cyberspace. The user has at each moment the global optimum with respect to the number of iterative partial searches that are driven by the local optimal relevancy ranking on each partial domain.

With these assumptions the search problem could be described as follows:

(a): Inside a finite data space search for the most relevant P documents according to a predefined goal. The group of found documents represents the initial *population* consisting of P members. They are the phenotype. Any evaluated document has a numerical index evaluating its relevancy with respect to the search context.
(b): As the search space expands the members of P are changing continuously, those members with low indexes being eliminated and by this making room for new members found to have higher relevancy indexes.
(c): For each document/member its set of indexes constitutes the genotype. They are numeric (words repetition rate, number of words) and represented as binary strings. Theoretically the index *term frequency* is defined e.g. on the interval 0% to 100% that means in binary from 000000 to 1100100, therefore seven bits maximum. *Concatenating* the binary domains for all seven indexes we need 49 bits; therefore in this case the chromosome will have 49 bits length.
(d) The initial population evolves by reproduction based on the two major genetic operators: *mutation* and *crossover* which are probabilistic parameters p_m and p_c. Each chromosome of the population (i.e. relevancy index) will be randomly affected. The isomorphic consideration of genetic operators in the context of information search process interprets mutation and crossover operators as modeling the probability of finding keywords inside the partial domain considered at each iteration.
(e) The population's evolution generated from the previous search is controlled by the *selection* mechanism [6]. This is possible by defining a certain *evaluation function* as a selection criterion. In essence this function is the model "?" in Fig. 3 reflecting the relevancy of the document.

The essence of this evolutionary search process stems from the recursive modification of the chromosomes of the concatenated indexes in each generation while monitoring the evaluation function. With each iteration all members of the current generation are compared with each other. The best results are placed on the top and the worst are replaced with the new members. The subsequent iteration resumes this process on the partially renewed population (for details see [5]).

The link between the evaluation function and the relevancy is made by the search query criterion. E.g. if we want to find the documents focused on a concept we must establish the values for each indexes as associate descriptors to our query. For example the query about "fuzzy semantics" could be done as: $< \textit{fuzzy semantics} \ / i_1 = 80\ \%, i_2 = 10, ..., > .$ An evaluation function based on the distances between the current indexes i_k^c and the query's indexes i_k^* is:

$$F_k = \left| i_k^c - i_k^* \right|, \quad k = 1,2,...,n \qquad (2)$$

The optimum criterion about relevancy is determined at each step as min(F_k) among all members of that generation.

5. Simulation Results

5.1. Fuzzy Relevancy Evaluation

We will evaluate the fuzzy relevancy searching on N=100 documents using the following three search indexes [5]:

- Term frequency $i_1 = \left(\frac{n_{term}}{n_{total}} \cdot 100 \right)$; where n_{term} is the number of times the searched term was matched exactly in the text.
- The frequency of occurrence of different forms for the term $i_5 = \left(\frac{n_{tdf}}{n_{total}} \cdot 100 \right)$; where n_{tdf} is the number of different forms of term found.

- The synonyms' frequency $i_6 = \left(\frac{n_{syn}}{n_{total}} \cdot 100 \right)$; where n_{syn} is the number of synonyms of term found.

The simulation[2] of this model was made considering random values for parameters n_{term}, n_{tdf}, n_{syn}. The sets of indexes for the first ten of the N documents are given in Table 1. The last column contains the relevancy of the document evaluated by the fuzzy model.

Table 1. The partial numerical results of fuzzy model of relevancy

Doc. No.	Index 1 i_1	Index 2 i_5	Index 3 i_6	Relevancy r
1	8.3394	0.8656	13.015	5.7751
2	0.0234	3.1849	1	2.2641
3	7.8670	10.479	8.5389	6.0416
4	1.7032	6	1.2014	3.5807
5	3.8987	5.0108	6.5011	5.4230
6	14.019	5.3589	6.4499	***9.1351***
7	8	10.921	0.2833	7.1046
8	6.2830	4	14.909	5.9999
8	0.1740	12.568	0	3.9381
10	5.7781	9	2.4127	5.0728
	10.231	14.502	6.2926	
	7	1	10.854	
		2.1745	2	
		3.4055		

Fig.5. illustrates the different relevancy degrees according to the strategy of search for the whole set documents.

5.2. Evolutionary Search on a Dynamic Domain

For the same case and the same kind of indexes

we now use a search query with specified goal as follows:

$$i_1^* = 10\% \ , \ i_2^* = 5\% \ , \ i_3^* = 3\% \qquad (3)$$

Thus inside a finite data space it searches for the most relevant $N=100$ documents according to the specified goal as above. In each generation the population of N documents receives randomly new members (documents). Their random nature can be assigned the genetic operators mutation and crossover. This simulation was made with $p_m=1\%$ and $p_c=20\%$. We assumed the maximum domain of any index was 15%. The new members will have the current informational indexes i_k^c , $k=1,2,3$. The selection of the best member of a generation (the most relevant document) is made with criteria: $\min \left| i_1^c - 10 \right|$, $\min \left| i_2^c - 5 \right|$, $\min \left| i_3^c - 3 \right|$. We ran this model by adapting a more general program for automatic location problem [7].

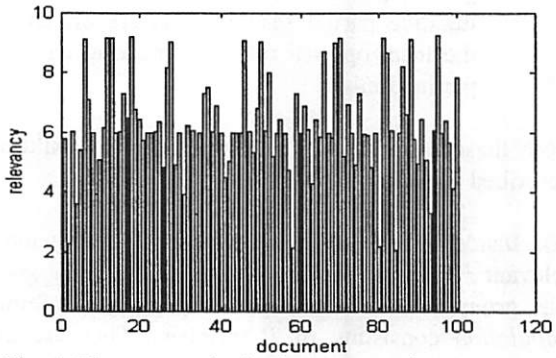


Fig. 5: The non-ranked relevancy marks

Fig. 6 shows the mean of the evaluation function for the three indexes of relevancy on successive generations. It is clear how they all converge towards the best generation i.e. the set of best documents matching the search criteria.

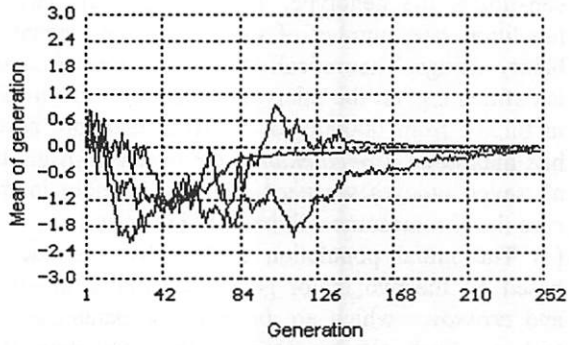


Fig. 6. The mean values of search indexes convergence towards the targeted values

[2] The Fuzzy toolbox of MatLab was used to run the model.

6. Conclusions

In the context of a global, web-driven economy, searching for collaborative partners may be reduced to a search for relevant keywords in the web pages of different organizations. The search methods introduced in this paper endow the holonic enterprise [1] with a mechanism for clustering the best partners to solve a particular task. On one side the fuzzy relevancy of the individual goals of each candidate organization with respect to the overall goal of the collaborative cluster is evaluated. On the other side this is done as a continuous, evolutionary process expanding the search domain until an optimal organization emerges, capable to solve the problem in the most efficient manner. The evolutionary selection of the optimal partners in the collaborative cluster in an open, ever-expanding domain facilitates the natural evolution of the virtual organization towards an optimal holonic structure.

References

[1] Mihaela Ulieru, Scott Walker and Robert Brennan, "Holonic Enterprise as a Collaborative Information Ecosystem", Proc. Workshop: Holons, Autonomous and Cooperative Agents for the Industry, Montreal, Canada, May 20, 2001 (AA 2001).

[2] Koestler, A. (1960), The Ghost in the Machine, Arcana Press, NY.

[3] Shu, Sudong and D.H. Norrie (1999) "Patterns for Adaptive Multi-Agent Systems in Intelligent Manufacturing", Proc. of the 2nd International Workshop on Intelligent Manufacturing Systems (IMS'99), Leuven, Belgium, pp. 67-74, September 22-24, 1999.

[4] Bezdek, J. *Fuzzy Models: What are they and why,* Editorial, IEEE Transactions of Fuzzy Systems, **Vol.1**, 1/1993.

[5] Ionita, S., Ulieru, M. (2001) A Fuzzy-Evolutionary Approach to Dynamic Search on an Open Ever-Expanding Domain in the Cyberspace, IEEE Transactions on Fuzzy Systems, Special Issue: Data Mining and Knowledge Discovery, December 2001 (submitted).

[6] Michalenicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, 1992.

[7] Ionita, S. *Genetic Algorithms for Control Engineering Applications* (in Romanian), ECIT-97, Pitesti, Romania, 21-22 Nov. 1997, p. 77-84.

Panel Discussions

Panel Discussion: TUESDAY, AUGUST 14

Search Engine and Queries: Challenges and Road Ahead

Humans have a remarkable capability (perception) to perform a wide variety of physical and mental tasks without any measurements or computations. Familiar examples of such tasks are: playing golf, assessing wine, recognizing distorted speech, and summarizing a story. The question is whether a special type information retrieval processing strategy can be designed that build in human perception.

One of the problems that Internet users are facing today is to find the desired and relevant information correctly and effectively in an environment that the available information, the repositories of information, indexing, and tools are all dynamic. Even though some tools were developed for a dynamic environment, they are suffering from "too much" or " too little" information retrieval. Some tools return too few resources and some tool returns too many resources.

To solve the above problem to some extend, it is important to use user-defined queries to retrieve useful information according to certain measures. Even though techniques exist for locating exact matches, finding relevant partial matches might be a problem. It may not be also easy to specify query requests precisely and completely - resulting in a situation known as a imprecise-querying. It is usually not a problem for small domains and structured database, but for large unstructured repositories such as World Wide Web and distributed databases; a request specification becomes a bottleneck. Thus, a flexible retrieval algorithm is required, allowing for imprecise or query specification and search.

"In my view, among the many ways in which fuzzy logic may be employed, there are two that stand out in importance. The first is search. Another, and less obvious, is deduction in an unstructured and imprecise environment." Professor Lotfi A. Zadeh, (Father of Fuzzy Logic), Berkeley-California, 2001.

"We need a semantic web which will provide guarantees, and about which one can reason with logic. (A fuzzy system might be good for finding a proof -- but then it should be able to go back and justify each deduction logically to produce a proof in the unifying HOL language, which anyone can check). Any real SW system will work not by believing anything it reads on the web but by checking the source of any information. (I wish people would learn to do this on the Web as it is." Tim Barnes-Lee (Father of WWW), Cambridge-Massachusetts, 1999.

"For example, World Wide Web search engines have become the most heavily- used online services, with millions of searches performed each day. Their popularity is due, in part, to their ease of use. While searches may retrieve thousands of hits, finding relevant partial matches might be a problem. The estimated user of wireless devices is estimated 1 billion within 2003 and 95 % of all wireless devices will be Internet enabled within 2005." BISC-FLINT program, Berkeley-California, 2001.

Moderators:

Rebecca Roberts; KQED TV
Masoud Nikravesh; University of California, Berkeley

Panelists:

Lee C. Giles (USA)
giles@ist.psu.edu
(814) 865-7884

Mori Anvari (USA)
Anvari@cs.berkeley.edu
510-643-4519

Marti Hearst (USA)
hearst@sims.berkeley.edu
510-642-8016

Jim Gray (USA)
Gray@Microsoft.com
415-778-8222 fax -8210

Trevor Martin (UK)
Trevor.Martin@bris.ac.uk
Tel. +44 117 928 8200

Panel Discussion: WEDNESDAY, AUGUST 15

Internet and Academia: Challenges

Moderators:

Rebecca Roberts; KQED TV
Masoud Nikravesh; University of California, Berkeley

Panelists:

Fernando Gomide (Brazil)
gomide@dca.fee.unicamp.br
+55 (19) 788-3782

John Meech (Canada)
jam@mining.ubc.ca
(604) 822-3984
(604) 943-0306

Elie Sanchez (France)
elie.sanchez@medecine.univ-mrs.fr

Tomohiro Takagi (Japan)
takagi@cs.meiji.ac.jp

Mihaela Ulieru (Canada)
ulieru@enel.ucalgary.ca
(403) 220-8616

Ronald Yager (USA)
ryager@iona.edu
yager@panix.com
(212) 249-2047

Lotfi A. Zadeh (USA)
zadeh@cs.berkeley.edu
(510) 642-4959

Panel Discussion: THURSDAY, AUGUST 16

Soft Computing: Past, Present, Future

Internet applications such as search engines and financial decisions have become the corners stone of online services, with hundreds of millions of transactions performed each day and billions of financing decisions made each year serving hundreds of billion dollars industry. Their importance and popularity are due, in part, to their power and ease of use. However, finding decision-relevant and query-relevant information from incomplete, imprecise, perception-based, and inconsistent data is a challenging problem, which has to be addressed. Soft computing provides technologies for dealing with imprecise information using fuzzy logic, neuro-computing, evolutionary computation, and other intelligent system technologies.

The main focus of this panel discussion will be on issues related to soft computing and the Internet, including search engines, web crawlers, web queries, web mining, decision support and risk analysis, and applications to e-commerce and e-business. The panel will provide an open forum so that a distinguished panel of experts discuss and exchange their thoughts, concerns and ideas about the past and present of soft computing. The panel will also explore the future challenges, share solutions and discuss research directions and applications of soft computing for the future.

Moderators:

Rebecca Roberts; KQED TV
Masoud Nikravesh; University of California, Berkeley

Panelists:

Jim Baldwin (UK)
Jim.Baldwin@bris.ac.uk
Tel. +44 117 928 7753

Hamid Berenji (USA)
berenji@iiscorp.com
(408) 730-8345

Ebrahim Mamdani (UK)
e.mamdani@ic.ac.uk

Tomohiro Takagi (Japan)
takagi@cs.meiji.ac.jp

Burhan Turksen (Canada)
turksen@mie.utoronto.ca

Ronald Yager (USA)
ryager@iona.edu
yager@panix.com
(212) 249-2047

Lotfi A. Zadeh (USA)
zadeh@cs.berkeley.edu
(510) 642-4959

Panel Discussion: FRIDAY, AUGUST 17

Internet and Industry

At the dawn of the new millennium, we can expect dramatic increase in the use of intelligent systems in the internet applications, since we have to deal with an increasing amount of data, that is mainly unstructured, dynamic and designed for human access. Therefore, it is usually hard to extract relevant information automatically. These aspects will be reflected in the subjects treated at this panel discussion.
The main purpose of the discussion panel is to draw the attention of the academic community as well as the industrial community to the fundamental importance of specific Internet-related problems. This issue is critically significant about problems that center on search and deduction in large, unstructured and distributed knowledge bases, and the use of intelligent techniques in e-business and B2B applications. The panel will provide a unique opportunity for the academic and corporate communities to address new challenges, share solutions, and discuss research directions for the future.

Moderators:

Rebecca Roberts; KQED TV
Masoud Nikravesh; University of California, Berkeley

Panelists:

B. Azvine (BT)
ben.azvine@bt.com
+44 (1473) 605466

J. Shanahan (Xerox)
James.Shanahan@xrce.xerox.com
+33-4.76.61.51.13

T. Cowden (Sonalysis)
cowden@sonalysts.com
800-526-8091 x419

M. Shan (HP)
mcshan@exch.hpl.hp.com
650-857-7158

B. Hodjat (Dejima)
Babak.Hodjat@dejima.com
(408) 535 0850 x 4515

D. Yadegar (Arsin)
dyadegar@arsin.com
(408) 653-2020