

The Narrative Explorer

Natalia Bilenko



Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-63

<http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-63.html>

May 12, 2016

Copyright © 2016, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

I would like to thank my Masters advisor, Maneesh Agrawala, for his guidance and advice throughout the project. The work described here started as a class project for the Visualization class taught by Maneesh Agrawala, and I am grateful for his support in developing the project further. I would like to thank my second reader, Alyosha Efros, for his advice. I also thank Asako Miyakawa, who was my collaborator at the beginning of this project. Finally, I thank the EECS department and staff.

The Narrative Explorer

Natalia Y. Bilenko

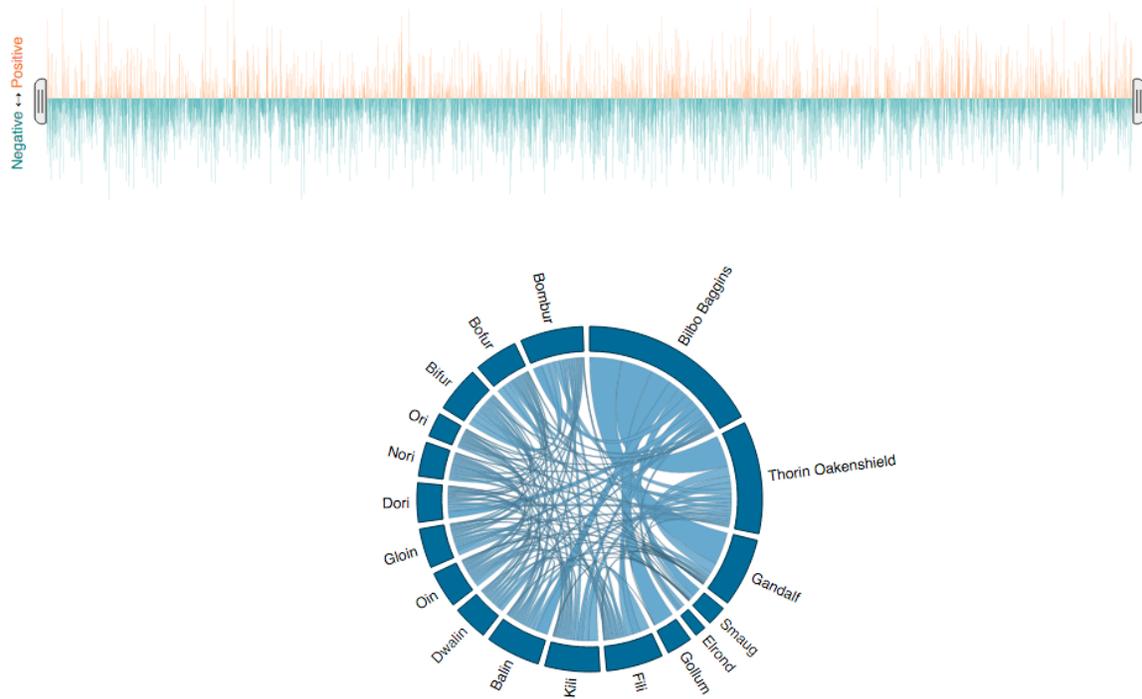


Fig. 1: The Narrative Explorer visualization for *The Hobbit* by J.R.R. Tolkien.

Abstract—The Narrative Explorer is an interactive web application for text visualization and analysis. The texts are visualized in the Narrative Explorer in two ways. The affective content of the text, quantified as the sentiment polarity of the sentences, is represented by a bar graph timeline. The co-occurrence of characters or concepts in the text is represented by a chord diagram. These visual representations are linked and can be examined interactively. The Narrative Explorer allows the user to upload and visualize a new text or select an existing text from the database for visualization. The Narrative Explorer is intended for a variety of audiences, from humanities scholars and students engaged in exploratory text analysis to non-specialist readers interested in an analytical perspective on a familiar text.

Index Terms—Text analysis, narrative, visualization

1 INTRODUCTION

Digital texts are a form of complex and high-dimensional data. In many fields of research, including digital humanities, computational linguistics, and artificial intelligence, scholars use text analysis and visualization tools to derive meaning from texts. Despite its numerous applications, text visualization presents many challenges. Texts are already a form of visual representation of information. Condensing texts and summarizing them visually is difficult without significant information loss. Underlying structure and metadata associated with texts are highly variable and cannot be captured by simple models. Additionally, there are few agreed upon conventions for visual representation of textual information in comparison to numeric data.

Text visualization as a form of literary analysis can be particularly challenging. Questions asked by humanities scholars, students, and even non-specialists who seek a deeper understanding of a literary text can require complex text processing and metadata collection. Manual parsing and tagging of texts for answering analytical questions is tedious and time-consuming. Automated text analysis methods are often unsuitable for more complex tasks. When automated methods are

appropriate, they can be daunting for scholars in qualitative fields due to technical complexity.

In this report, we present the Narrative Explorer, a tool for text exploration that allows the user to visualize the text in an interactive way. The user is able to select from a library of available visualizations that are included with the application. The user is also able to create a new visualization by uploading and analyzing a text file. The Narrative Explorer combines visualizations of two aspects of the text structure, the sentiment timeline and character co-occurrences.

The sentiment timeline is represented by a bar graph where each bar represents a sentence in the text in chronological order. The sentiment polarity of each sentence is classified as positive or negative. The sentiment polarity and the probability of the rating are visualized as the height of each bar in the sentiment timeline. The co-occurrences of characters or other concepts in the text are represented as a radial chord diagram. Each chord corresponds to co-occurrences of two characters or words. The sentiment timeline and the co-occurrence graph are interactively linked. The appearances of each character in the co-

occurrence graph can be visualized on the sentiment timeline. Portions of the sentiment timeline can be selected to limit the character co-occurrence visualization to that portion of the text.

The Narrative Explorer is intended for humanities scholars, students, and non-specialist readers. In research settings, it allows for exploratory questions to be probed in an engaging way. While a comprehensive research investigation might require more advanced text analysis tools, the Narrative Explorer can be used to guide sensemaking at early stages of research. The Narrative Explorer allows scholars and students to ask questions about co-occurrences of characters or concepts in texts, track emotional narratives of characters through the story, and explore how character relationships change throughout the story.

For recreational readers, the Narrative Explorer can provide a new perspective on a familiar text. Unlike many existing visualizations of texts that are limited to a particular text, the Narrative Explorer allows readers to interactively add and visualize texts of their choice. Exploring character co-occurrences and aspects of the emotional narrative in a story can be a rewarding and engaging experience for the readers. This process can also serve as an accessible introduction to text analysis for non-specialists.

This report summarizes the Narrative Explorer and its usage. After a brief summary of relevant work in Section 2, we give an overview of the Narrative Explorer and its functionality in Section 3. In Section 4, we describe the implementation, focusing on the technical details. Finally, we describe future directions for the development of the Narrative Explorer in Section 5.

2 RELEVANT WORK

As digital media and digital text collections continue to grow, there is an increasing need for text analysis and visualization. In the past decade, many visualization techniques and tools have emerged with a particular focus on texts, document collections, and literary works. We focus on those that have inspired the Narrative Explorer.

One of the goals of text visualization is to summarize a complex text visually and to give the user a snapshot of a set of features in the text in an easily navigable representation. Many tools take an overview approach for the purpose of comparing documents or distinguishing individual documents in a collection. One early influential example of a text visualization tool with such an approach is TileBars [9], created to distinguish retrieved documents in a search user interface. Literature Fingerprinting [12] is a more recent project that represents literary texts in a collection as unique visual fingerprints, created based on a variety of literary analysis features of the texts.

Literary texts and their features can serve as a fruitful subject for data artists. Two notable examples of artistic text visualizations are Writing Without Words [22] and (En)tangled World Bank [21] by Stefanie Posavec. These projects focus on specific texts, *On the Road* by Jack Kerouac and *The Origin of Species* by Charles Darwin. The resulting visualizations are not only beautiful, but convey complex information about the structure of the texts. However, the underlying data are obtained by painstaking manual labeling of the relevant features, making these visualizations singular in their application.

Some text visualization tools go beyond the overview approach and delve into analysis by examining the text at different levels of abstraction. Two examples of visual tools that allow for different levels of text analysis are the VarifocalReader [13] and WordSeer [18]. The VarifocalReader is a tool for analysis of large text documents that uses focus and context techniques in combination with automatic text analysis tools. WordSeer is a tool for comparing text structure and document metadata across document collections that supports in-depth text analysis on a large scale.

The Narrative Explorer provides a visual overview of the text in the form of the sentiment timeline and the co-occurrence graph. It also supports more detailed analysis by allowing the user to explore appearances of a specific character or concept and to track co-occurrences of characters or concepts in subsets of the text.

Many text visualization tools focus on affective content of texts. For example, Liu et al. (2003) [14] use a timeline to display the domi-

nant emotion present at each point in a text as a method for describing the affective structure. Mohammad (2011) [16] takes a similar approach by visualizing the prevalence of multiple emotions throughout a text. Stor-e-Motion [31] is a tool for tracking affective content in data streams, such as Twitter, and uses a timeline visualization as well. Rose Plots [7] is a technique that takes a gestalt approach to sentiment content by showing the deviation of the affect scores for a particular text from the average scores for a collection of text documents. The Narrative Explorer adopts the timeline technique and the use of color for representing sentiment. For simplicity of visual representation, the sentiment timeline represents a single dimension of affective content: the sentiment polarity.

Understanding the context and co-occurrence patterns of concepts or characters in a text is a frequent analytical task. Some visualization tools summarize the context of word occurrences, such as Wordgraph [25] and Word Tree [30]. Other tools visualize co-occurrence of characters and key concepts, such as Phrase Nets [29], TextArc [19], and Character Flower [24]. Poetry visualization tools Poemage [15] and rule-based poetry visualization [1] display connections between semantic and phonetic elements throughout poems.

ScriptThreads [10] tracks character co-occurrences in movie screenplays by visualizing them as intersecting timelines. ScriptThreads draws inspiration from a web comic XKCD that humorously visualized intersecting character narratives for popular movies [17]. Character co-occurrences in *Les Misérables* by Victor Hugo are the subject of a matrix visualization by Mike Bostock [3].

The Narrative Explorer combines the co-occurrence graph and timeline approaches for tracking characters and concepts throughout the story. Appearances of characters or concepts can be highlighted in the sentiment timeline. Co-occurrences of characters or concepts in the whole document or a subset of the text are displayed in a radial chord diagram.

3 OVERVIEW

The Narrative Explorer is a web application tool for text visualization. The visual layout of the Narrative Explorer is shown in Figure 1 and an annotated layout is shown in Figure 2. The Narrative Explorer layout includes two main areas. At the top of the page is the control panel area and below is the visualization area.

The control panel area includes five buttons that allow the user to manage the current display by selecting a new visualization, uploading a new text, editing the character list, saving the current visualization, or exporting the data. There is an *About* button in the top right corner of the page that brings up information about the visualization and interaction instructions.

The visualization area includes the title for the current visualization at the top, which can be edited by clicking on it. Below the title is the sentiment timeline bar graph. At the bottom of the visualization is the co-occurrence chord diagram that displays co-occurrences between characters in the text.

The users begin their interaction with the Narrative Explorer in one of two ways: by selecting a previously prepared visualization from the existing library of texts or by creating a new visualization for an uploaded text. After the visualization is either loaded or created using these methods, the user can interactively explore two elements of the story: the sentiment trajectory of the text and the character co-occurrences.

3.1 Initializing the visualization

In the first view of the Narrative Explorer, the user sees two buttons in the control panel area: *Select visualization* and *Upload new text*. This view is shown in Figure 3. The user selects one of these buttons to start using the Narrative Explorer. After a visualization is displayed, either of these buttons can be used to restart the interaction flow and display a new visualization.

3.1.1 Selecting an existing visualization

To explore a visualization that is already in the database, the user clicks the *Select visualization* button in the control panel area. The

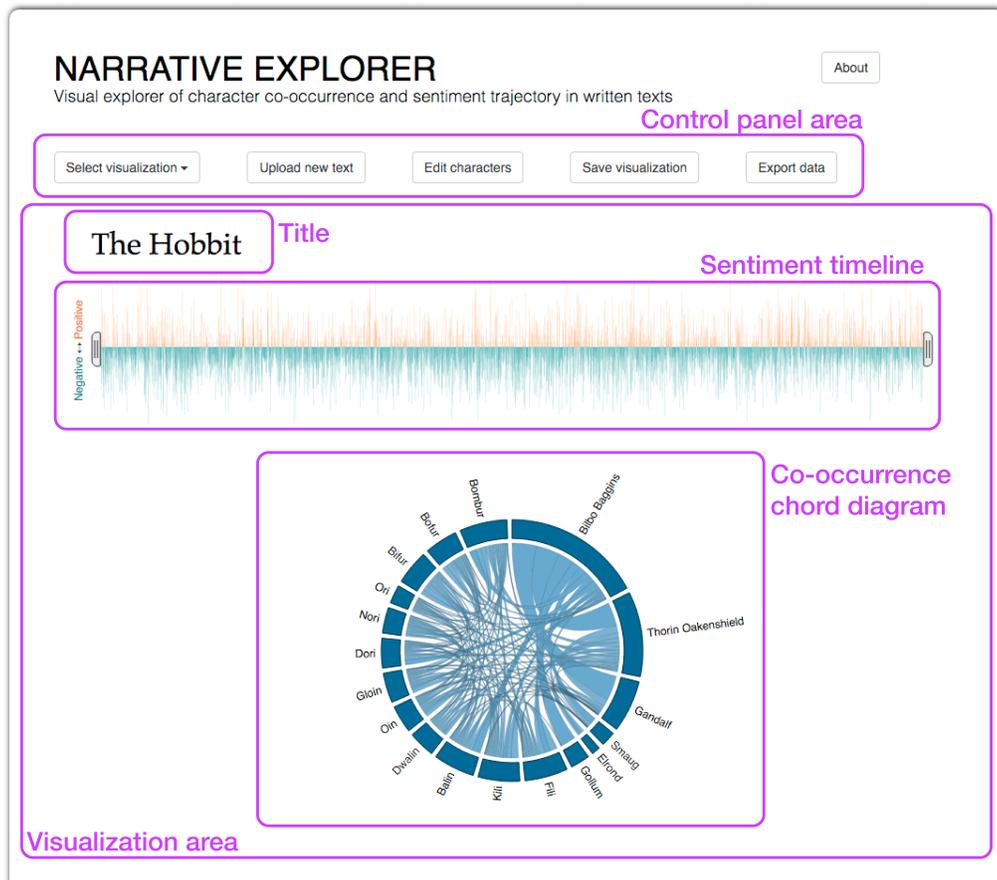
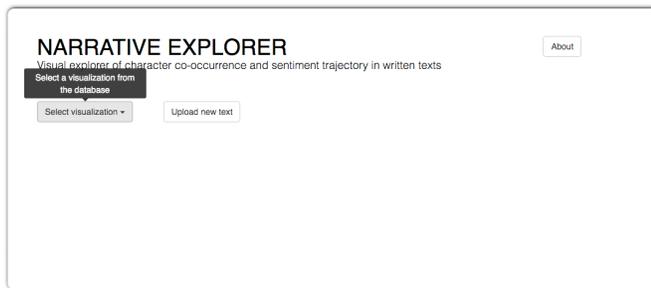
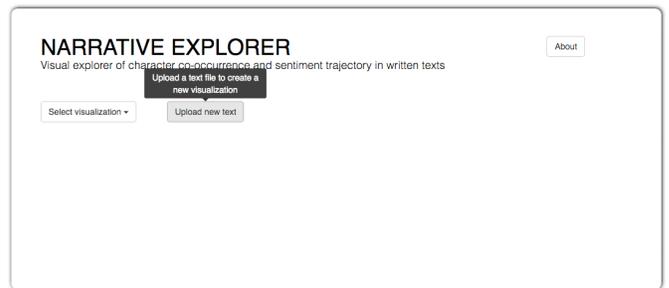


Fig. 2: The full view of the Narrative Explorer for *The Hobbit* by J.R.R. Tolkien, annotated with the main interaction areas. The control panel area at the top of the page includes the buttons for managing the current display of the Narrative Explorer. The visualization area below contains the editable title, the sentiment timeline, and the chord diagram of character co-occurrences.



(a) Select a visualization



(b) Upload a new text

Fig. 3: Two views of the Narrative Explorer that the user first sees upon coming to the page. The user initializes the interaction with the Narrative Explorer either by (a) selecting a visualization from those available on the server or (b) uploading a new text to create a new visualization.

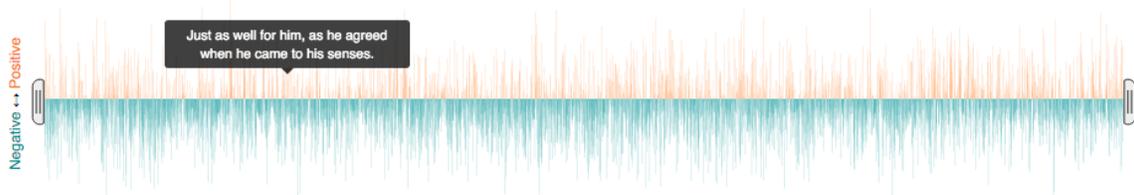


Fig. 4: The sentiment timeline section of the Narrative Explorer for *The Hobbit* by J.R.R. Tolkien. Each sentence is represented by a bar. Sentences with a positive sentiment rating are represented by orange bars above the x-axis. Sentences with a negative sentiment rating are represented by teal bars below the x-axis. Hovering over each bar displays the text of the sentence.

button reveals a drop-down menu with the names of existing texts that have been visualized previously and saved on the server. When the user selects a text, the corresponding data are loaded from the database. These data include the title of the visualization, the text of all sentences, and the corresponding sentiment rankings (probability that each sentence has a positive or negative sentiment). The data also include a list of characters, their appearances in the text, and co-occurrences of the characters within a five-sentence sliding window.

3.1.2 Creating a new visualization

To upload a new text and create a new visualization, the user clicks the *Upload new text button*. The button opens a modal dialogue that allows the user to select and upload a new text file from their computer. After the file is uploaded, its contents is processed automatically in order to create the visualization. The text is read and split into sentences. Each sentence is then split into words, which are tagged with the corresponding parts of speech. Each sentence is scored with a probability that its sentiment is positive or negative using a previously trained sentiment classifier stored on the server. A named entity recognition algorithm is used to find the list of twenty most commonly occurring characters in the text. The appearances of these characters in the text are detected. The character co-occurrences are computed within a five-sentence window.

3.2 Exploring the visualization

After the visualization is loaded or computed using one of the methods described in Section 3.1, it is displayed to the user. The user sees two graphical representations of the text: a bar graph at the top of the visualization area that represents the sentiment timeline and a chord diagram below that represents character co-occurrences in the text. The two graphs are interactively linked.

3.2.1 Sentiment timeline

The sentiment timeline is a bar graph, where each bar corresponds to one sentence in the text. The probability that each sentence has a positive or negative sentiment is plotted on the y-axis of the graph. Sentences that are more likely to be positive are represented by orange bars above zero. Sentences that are more likely to be negative are represented by teal bars below zero. The height of each bar corresponds to the probability of the sentiment rating. The user is able to read the text of each sentence by hovering over the bar. See Figure 4 for an example of the sentiment timeline.

3.2.2 Character co-occurrences

Character co-occurrences are plotted in a chord diagram below the sentiment timeline. The characters are represented as ring segments, with the name of the character displayed next to the segment. The co-occurrences between the characters in the text are represented by chords linking the ring segments. The thickness of each chord corresponds to the relative number of co-occurrences of the two characters relative to the numbers of co-occurrences between the other characters. Hovering the mouse over one of the characters fades the co-occurrences that do not include that character, shown in Figure 5. Hovering the mouse over one of the chords displays the text “Connections between X and Y”, where X and Y are the names of the co-occurring characters represented by that chord.

3.2.3 Interaction between the sentiment timeline and the co-occurrence graph

The sentiment timeline and the character co-occurrence graph are linked together interactively. The appearances of each character in the text can be highlighted in the sentiment timeline by clicking on the character’s name or corresponding ring segment in the co-occurrence graph. When the character’s name is clicked, the font color for that name in the co-occurrence graph transitions from black to orange. The bars in the sentiment timeline that correspond to sentences in which that character appears remain at full opacity. The bars that correspond to sentences in which that character does not appear are partially faded

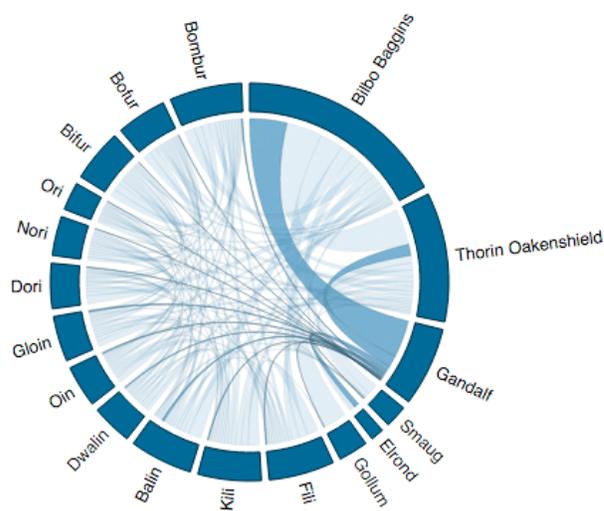


Fig. 5: The character co-occurrence diagram of the Narrative Explorer for *The Hobbit* by J.R.R. Tolkien. Each character is represented by a segment of the ring. The co-occurrences of the characters are represented by the chords linking them. Hovering over a character’s name or ring segment (in this case, Gandalf) highlights the co-occurrences for only that character by fading the others.

out. Figure 6a shows the Narrative Explorer with one of the characters selected by clicking.

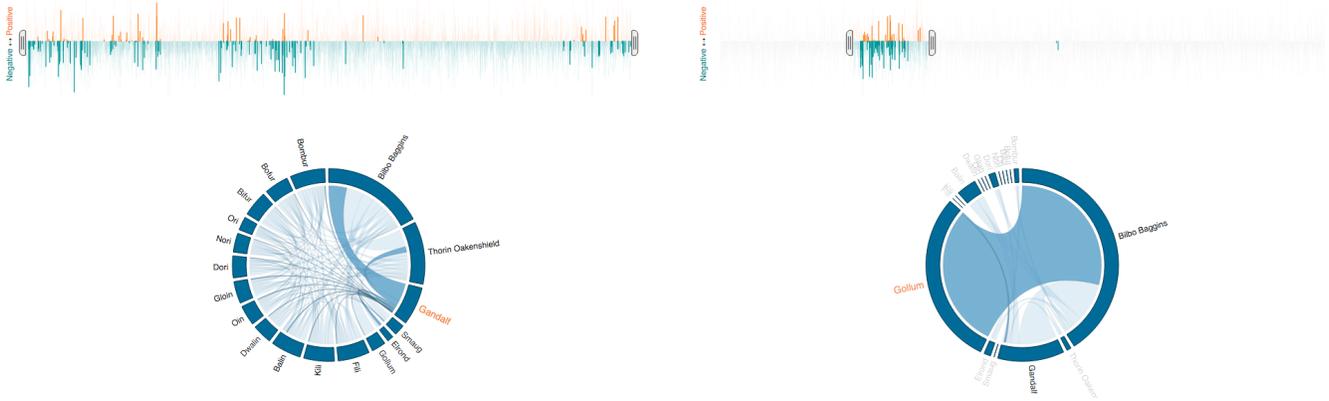
The sentiment timeline graph includes brush handles that can be used to select a portion of the text for which character co-occurrences are displayed. The handles are set by default to select the full text. Brushing is accomplished by clicking and dragging the brush handles independently or by clicking and dragging the region between the brush handles, which retains the length of the selected segment. To indicate the selected region, the color fill of the bars in the sentiment timeline that correspond to sentences outside of the selection is set to gray. After the brush handles are moved, the co-occurrence chord diagram is updated to reflect character co-occurrences in the selected portion of the text. Figure 6b shows the Narrative Explorer with a subset of the text selected and one of the characters selected by clicking.

3.3 Editing the visualization

The list of characters displayed in the character co-occurrence graph in the Narrative Explorer is editable by the user. Editing the character list for visualizations loaded from the database can be valuable when the user is interested in exploring the co-occurrences of characters other than those that were chosen by the visualization creators. When creating new visualizations, this feature is useful for correcting the results of automatic text processing. The named entity recognition algorithm used in automatic processing of the text is imperfect and sometimes produces unwanted results that have to be altered by the user.

For example, when processing the text for *How the Grinch Stole Christmas* by Dr. Seuss, in addition to the characters such as Grinch and Santa, the algorithm detects temporal entities such as Christmas and Tomorrow, as well as some common words such as Would and Stuffed. These mistakenly detected entities can be removed from the list by the user.

The named entity recognition algorithm may also miss characters that may be **important** to the user, but less commonly occurring. For ease of interaction, we limited the output of the algorithm to twenty most commonly occurring characters in the text, which may be insufficient for texts with a large number of characters. To compensate for this issue, characters can be added to the list. The total number of characters



(a) Clicking the character name (in this case, Gandalf) highlights the appearances of that character in the sentiment timeline.

(b) Selecting portions of the text in the sentiment timeline by using brush handles constrains the character co-occurrence graph to co-occurrences for that portion of the text.

Fig. 6: Two views of the Narrative Explorer for *The Hobbit* by J.R.R. Tolkien. The views show how the interactions for the sentiment timeline and the character co-occurrence graph are linked together.

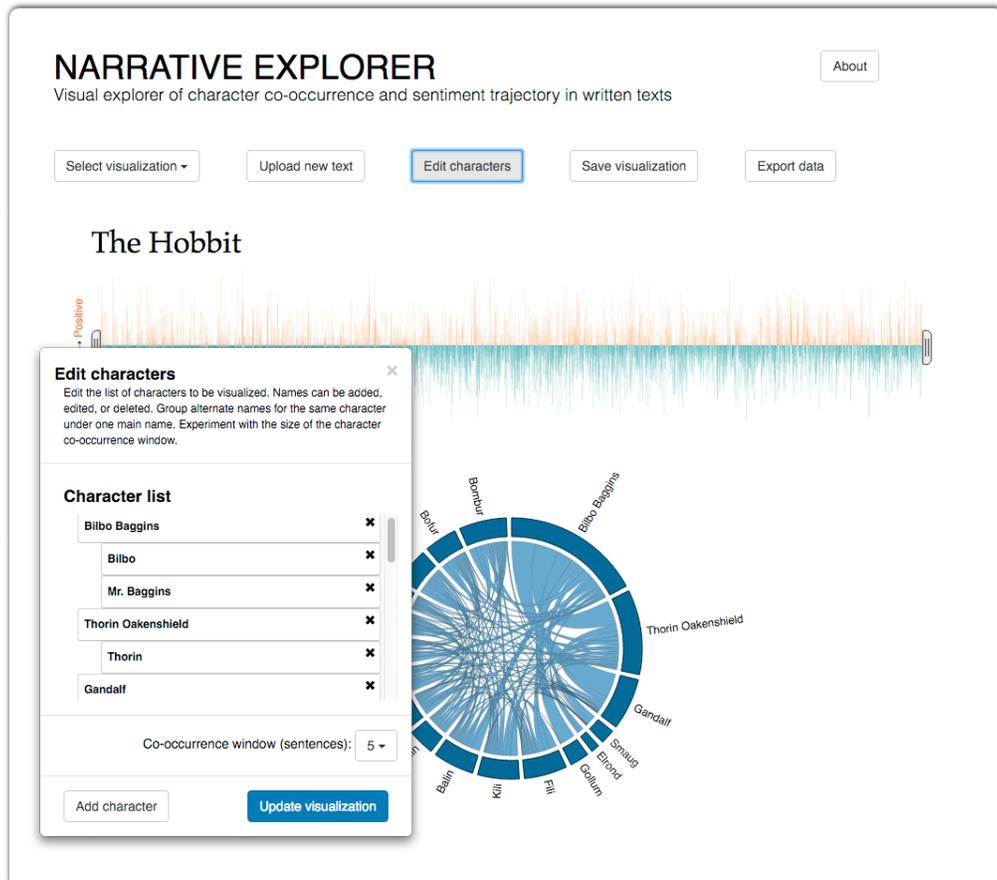


Fig. 7: The Narrative Explorer visualization for *The Hobbit* by J.R.R. Tolkien. Pressing the *Edit characters* button in the control panel area opens a panel with an editable list of characters. The names of the characters can be changed, added, or deleted. Different names for the same character can be grouped together.

that can be added is unlimited.

Texts often have multiple names for the same character. For example, in *How the Grinch Stole Christmas*, the same character is referred to as Santa, Santy Claus, and Saint Nick. These names can be grouped together, so that all instances of these names count as appearances of the same character.

Editing the character list can also be useful when the user is inter-

ested in appearances and co-occurrences of concepts or specific words in the text, instead of or in addition to characters. For example, the user might want to track symbols or motifs throughout the story or to connect specific characters to notable phrases or topics. In this case, the user may want to replace the list of characters with a list of keywords relevant for the analysis.

To edit the character list, the user clicks the *Edit characters* button

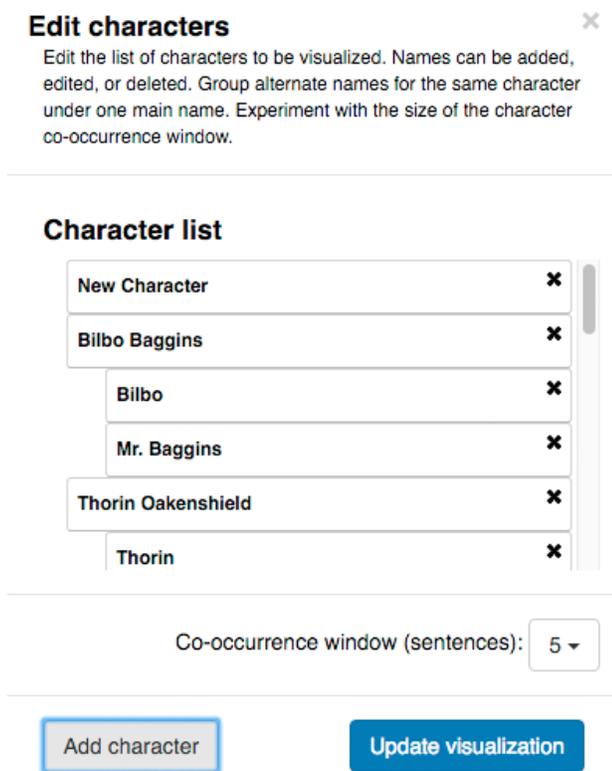


Fig. 8: A view of the *Edit characters* panel for the Narrative Explorer visualization of *The Hobbit* by J.R.R. Tolkien. Characters can be added to the character list by pressing the *Add character* button. When the button is clicked, a character named “New Character” is added to the beginning of the list. That name can be then edited by clicking on it.

in the control panel area. This button brings out a previously hidden panel in the lower left of the screen. This view is shown in Figure 7. The *Edit characters* panel contains a brief set of instructions and an editable list of characters. The user can delete a character’s name from the list by clicking the cross icon on the right of that name. Clicking on the character’s name allows the user to edit it.

To group alternate names for the same character, a character’s name can be dragged to group it under the primary name for that character. For example, the name “Mr. Baggins” can be dragged under the name “Bilbo Baggins” to count all appearances of “Mr. Baggins” together with the appearances of “Bilbo Baggins”. The primary name at the top level of the character list hierarchy will be displayed in the character co-occurrence diagram.

The *Add character* button in the lower left of the panel allows the user to add a new character to the list. The new name is added to the beginning of the list with the text “New Character” as the name. This view is shown in Figure 8.

In addition to editing the character list, the *Edit characters* panel allows the user to edit the sliding window used to compute the co-occurrence of characters in the text. By default, two characters are defined as co-occurring at a specific point in the text if they both appear within a sliding window of five sentences centered at that point. The drop-down menu labeled “Co-occurrence window” allows the user to specify a different sliding window, from 3 sentences up to 21 sentences.

After editing the character list and choosing the character co-occurrence window, the user clicks the *Update visualization* button in the *Edit characters* panel to update the visualization with the selected parameters and hide the panel. To close the *Edit characters* panel without updating the visualization, the user can click the cross

icon in the top right of the panel or the *Edit characters* button in the control panel area of the Narrative Explorer.

3.4 Saving and exporting the visualization

The user has two options for preserving the visualization data in the Narrative Explorer. If the user wishes to save the visualization on the server and have it available in the drop-down menu for later selection, the user clicks the *Save visualization* button on the control panel. Clicking this button commits the visualization data to the database from which the users can select prepared visualizations.

The user can also export the data for further analysis in JSON format by clicking the *Export data* button. Clicking this button reveals a panel with a text area that contains the data for the currently displayed visualization in JSON format. This view is shown in Figure 9. The user can copy the data from the text area to the clipboard by clicking the *Copy data to clipboard* button at the bottom right of the panel. The user can then paste the data to a text editor. Clicking the *Copy data to clipboard* button or the cross icon at the top right of the panel closes the panel.

Exporting the data may be beneficial if the user wants to use it for another analysis. The user is also able to save the data as a JSON file and upload it to the Narrative Explorer using the *Upload new text* button. The Narrative Explorer is configured to detect that a JSON file is uploaded instead of a text file. If the data contained in the JSON file are consistent with the visualization data, the visualization is created from these data.

Saving or exporting the data completes the interaction flow of the Narrative Explorer. The user is able to restart the process at any point by selecting an existing visualization from the database or by uploading a text file and creating a new visualization.

4 IMPLEMENTATION

The Narrative Explorer is an open-source implementation, and the code can be downloaded via GitHub (https://github.com/nbilenko/narrative_explorer/). The web application was implemented in Python using the Flask web framework [8]. Using Flask allowed us to integrate tools for data storage, text analysis, and the visual interface. The Werkzeug library in Python [32] was used to manage WSGI utilities.

4.1 Data storage

Data storage was organized using a PostgreSQL database [23]. We chose PostgreSQL due to its simplicity of usage and the flexibility of its data models. These aspects proved convenient at early stages of the application design. The Python library SQLAlchemy [5] was used to access the database. SQLAlchemy is an Object Relational Mapper that allowed us to define the data models in Python and interface with the data directly from the Flask framework.

4.2 Text analysis

Text analysis was done using a combination of custom code and the Natural Language Toolkit (NLTK) in Python [2]. NLTK is a widely used platform for text analysis and contains a large number of libraries, tools, and corpora. During automatic processing of uploaded text, we used NLTK tools to parse the text into sentences and words, and to tokenize words with part of speech labels.

We used a custom classifier to perform sentiment analysis. We used the Scikit-learn machine learning library [20] to test several types of models and chose logistic regression due to lower rates of overfitting to training data. We then trained the classifier on a database of labeled sentences. The trained classifier was saved on the server. For each uploaded text, the classifier was loaded in NLTK and used to predict the sentiment rating of each sentence.

Character detection was done using a named entity recognition (NER) algorithm in NLTK. A three-category NER algorithm was used, and the entities that were identified as belonging to the “Person” category were selected. Twenty most commonly occurring characters were

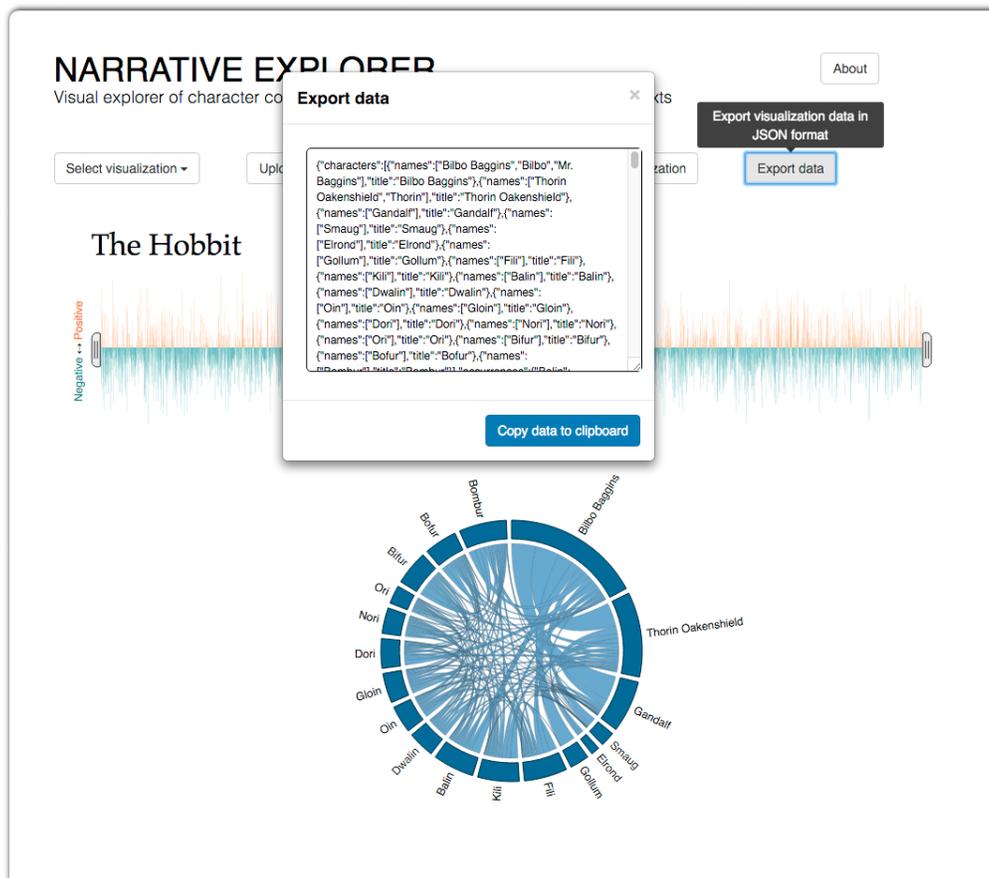


Fig. 9: The Narrative Explorer visualization for *The Hobbit* by J.R.R. Tolkien. Pressing the *Export data* button in the control panel area brings out a panel with the visualization data in JSON format. The data can be copied to the clipboard if the user wishes to analyze them further.

chosen from these entities and included in the character list. Appearances and co-occurrences of characters in the text were computed using custom code.

4.3 Visual interface

The visual interface was implemented in JavaScript. The Data-Driven Documents (d3.js) [4] library was used to create the sentiment timeline and the co-occurrence chord diagram. JQuery and jQuery UI [27] were used to manage interactions and interface elements. The Tipy plugin for jQuery [6] was used to create tooltips displayed on mouse hover. The nestedSortable plugin for jQuery [26] was used to create the editable character list. The Twitter Bootstrap library [28] was used to create the interaction styles of the buttons and menus. Corresponding CSS libraries for these JavaScript and jQuery libraries and plugins were used to manage the style of the visualization elements. AJAX tools were used to communicate between the server and the client side of the application. The selection menu for choosing among the existing visualizations was populated from the database using HTML templating managed by the Jinja2 engine in Python [11].

5 CONCLUSION AND FUTURE WORK

The Narrative Explorer is an interactive web application for exploratory text visualization. In this report, we demonstrate how the Narrative Explorer can be used to explore texts via two types of text analysis: by examining the trajectory of the sentiment ratings of sentences throughout the story and the co-occurrences of characters in the text. The application allows users to create new visualizations for uploaded texts or to examine or edit available visualizations included in the application. The users are able to export the visualization data and contribute their visualizations to the database.

One important further step in the development of the Narrative Explorer would be to perform a user study in order to systematically incorporate feedback based on the application usage. Based on preliminary feedback, we expect that both scholars and non-specialist readers would find the Narrative Explorer engaging and helpful. A study would be instrumental in evaluating further development directions and the contribution of the Narrative Explorer.

We hope to make several specific improvements to the Narrative Explorer in the future. One potential change would be in organization of the library of existing visualizations. The visualizations are currently sorted alphabetically by title, but there is no further organization. The visualizations could be grouped into different categories, for example, by genre. Such organization would make it easier to look through the available visualizations as the database grows larger. To improve this task even further, a search bar with search term completion could be added to search the visualizations by name.

Another improvement we plan to make would have the goal of increasing the utility of the Narrative Explorer for exploratory analysis. Text analysis often requires extracting quotes from the text to support the researcher's hypothesis. To enable quote extraction, we could add a panel to Narrative Explorer that would contain the full text. The text would be linked to the sentiment timeline, such that selecting a subset of the text in the timeline would highlight a portion of the text in the panel. Selecting specific characters in the co-occurrence graph would highlight the character names in the text. This improvement would allow the user to easily navigate through the text and extract relevant quotes to support their investigation.

REFERENCES

- [1] A. Abdul-Rahman, J. Lein, K. Coles, E. Maguire, M. Meyer, M. Wynne, C. R. Johnson, A. Trefethen, and M. Chen. Rule-based Visual Mappings

- with a Case Study on Poetry Visualization. *Computer Graphics Forum*, 2013.
- [2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.
- [3] M. Bostock. Les Misrables Co-occurrence. <https://bost.ocks.org/mike/miserables/>. Accessed: 2016-04-01.
- [4] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven Documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.
- [5] R. Copeland. *Essential SQLAlchemy. Mapping Python to Databases*. O'Reilly, Beijing, 2008.
- [6] J. Frame. Tippy jQuery plugin. <http://onehackoranother.com/projects/jquery/tippy>, 2008–2010.
- [7] M. L. Gregory, N. Chinchor, P. Whitney, R. Carter, E. Hetzler, and A. Turner. User-directed Sentiment Analysis: Visualizing the Affective Content of Documents. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, SST '06, pages 23–30, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
- [8] M. Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc., 1st edition, 2014.
- [9] M. A. Hearst. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 59–66, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [10] E. Hoyt, K. Ponto, and C. Roy. Visualizing and Analyzing the Hollywood Screenplay with ScripThreads. *Digital Humanities Quarterly*, 8(4), 2014.
- [11] Jinja Development Team. Jinja2. <http://jinja.pocoo.org/>, 2014.
- [12] D. A. Keim and D. Oelke. Literature fingerprinting: A new method for visual literary analysis. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 115–122, Oct 2007.
- [13] S. Koch, M. John, M. Wrner, A. Mller, and T. Ertl. VarifocalReader - In-Depth Visual Analysis of Large Text Documents. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1723–1732, Dec 2014.
- [14] H. Liu, T. Selker, and H. Lieberman. Visualizing the Affective Structure of a Text Document. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 740–741, New York, NY, USA, 2003. ACM.
- [15] N. McCurdy, J. Lein, K. Coles, and M. Meyer. Poemage: Visualizing the Sonic Topology of a Poem. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):439–448, Jan 2016.
- [16] S. Mohammad. From Once Upon a Time to Happily Ever After: Tracking Emotions in Novels and Fairy Tales. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '11, pages 105–114, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [17] R. Munroe. XKCD. Movie Narrative Charts. <http://xkcd.com/657/>. Accessed: 2016-04-01.
- [18] A. Muralidharan, M. A. Hearst, and C. Fan. WordSeer: A Knowledge Synthesis Environment for Textual Data. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 2533–2536, New York, NY, USA, 2013. ACM.
- [19] W. Paley. TextArc: Showing Word Frequency and Distribution in Text. In *Proceedings of IEEE Symposium on Information Visualization, Poster Compendium*. IEEE CS Press, 2002.
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [21] S. Posavec. (En)tangled Word Bank. <http://www.stefanieposavec.co.uk/-everything-in-between/#/entangled-word-bank/>. Accessed: 2016-04-09.
- [22] S. Posavec. Writing without Words. <http://www.stefanieposavec.co.uk/personal/#/writing-without-words/>. Accessed: 2016-04-09.
- [23] PostgreSQL Global Development Group. PostgreSQL. <http://www.postgresql.org>, 2008.
- [24] T. Regan and L. Becker. Visualizing the Text of Philip Pullman's Trilogy "His Dark Materials". In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 759–764, New York, NY, USA, 2010. ACM.
- [25] P. Riehmman, H. Gruendl, M. Potthast, M. Trenkmann, B. Stein, and B. Froehlich. WORDGRAPH: Keyword-in-Context Visualization for NETSPEAK's Wildcard Search. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1411–1423, Sept 2012.
- [26] M. J. Sarfatti. NestedSortable jQuery plugin. <http://onehackoranother.com/projects/jquery/tippy>, 2010–2016.
- [27] E. Sarrion. *jQuery UI*. O'Reilly Media, Inc., 2012.
- [28] Twitter Bootstrap Development Team. Twitter Bootstrap. <http://twitter.github.com/bootstrap/>, 2011–2016.
- [29] F. van Ham, M. Wattenberg, and F. B. Viegas. Mapping Text with Phrase Nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176, Nov. 2009.
- [30] M. Wattenberg and F. Viegas. The Word Tree, an Interactive Visual Concordance. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1221–1228, Nov 2008.
- [31] A. Weiler, M. Grossniklaus, and M. H. Scholl. The Stor-e-Motion Visualization for Topic Evolution Tracking in Text Data Streams. In J. Braz, editor, *IVAPP 2015 : Proceedings of the 6 th International Conference on Information Visualization Theory and Applications*, pages 29–39. SciTeppres, 2015.
- [32] Werkzeug Development Team. Werkzeug. <http://werkzeug.pocoo.org/>, 2014.