

A Study of Generalization Metrics for Natural Language Processing: Correlational Analysis and a Simpson's Paradox

Raguvir Kunani

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-71

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-71.html>

May 11, 2022



Copyright © 2022, 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 some of the many people who supported me throughout my academic journey. Thank you to my advisor Professor Joseph Gonzalez for opening the door for me (and countless others) to research opportunities at Berkeley. Thank you to Yaoqing Yang for collaborating closely with me on this work for the last year and for being an fantastic research mentor for the last 2 years. Thank you to Professor Michael Mahoney for feedback and suggestions on this work. Thank you to all the outstanding Berkeley professors I had the privilege of learning from for helping me build a strong technical foundation and inspiring me to dream big. Finally, thank you to my friends and family for the unparalleled and unconditional love and support throughout my time at Berkeley.

**A Study of Generalization Metrics for Natural Language Processing:
Correlational Analysis and a Simpson's Paradox**

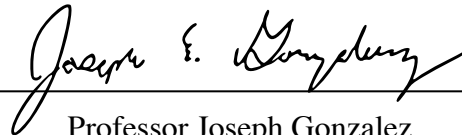
by Raguvir Kunani

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:



Professor Joseph Gonzalez
Research Advisor

5/11/22

(Date)

* * * * *



[Michael Mahoney \(May 12, 2022 00:38 EDT\)](#)

Professor Michael Mahoney
Second Reader

5/12/22

(Date)

**A Study of Generalization Metrics for Natural Language Processing:
Correlational Analysis and a Simpson's Paradox**

by

Raguvir Kunani

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master's of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Joseph Gonzalez, Chair

Professor Michael Mahoney

Spring 2022

Abstract

**A Study of Generalization Metrics for Natural Language Processing:
Correlational Analysis and a Simpson's Paradox**

by

Raguvir Kunani

Master's of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Joseph Gonzalez, Chair

A predictive model's utility lies in its ability to generalize to data it has not seen. Unfortunately, it is difficult to reliably measure a model's ability to generalize to unseen data since it requires reasoning about the model's interactions with unknown environments. Generalization of deep learning models has been the subject of extensive study for years, but there has been a recent increase in the exploration of generalization metrics to predict the generalization of deep learning models.

While prior work in generalization metrics has been dominated by computer vision, in this work, we conduct one of the first analyses of generalization metrics in natural language processing (NLP). We study 36 generalization metrics spanning various motivations/theories with the goal of understanding the degree to which each metric is appropriate for use in predicting the generalization of models common in NLP. We particularly focus on shape metrics (generalization metrics derived from the shape of the empirical distribution of eigenvalues of weight correlation matrices) and are among the first to consider out-of-distribution generalization when evaluating the effectiveness generalization metrics.

We find that shape metrics are a promising category of generalization metrics, as they are the best metrics among those we consider at predicting generalization performance throughout training and show characteristics of being "ideal" generalization metrics. Interestingly, many of the generalization metrics we consider exhibit a behavior reminiscent of the Simpson's paradox when related to generalization performance. Moreover, the generalization metrics we consider are generally robust to changes in data distribution. However, there are signs this robustness is limited.

Acknowledgments

I would like to thank some of the many people who supported me throughout my academic journey. Thank you to my advisor Professor Joseph Gonzalez for opening the door for me (and countless others) to research opportunities at Berkeley. Thank you to Yaoqing Yang for collaborating closely with me on this work for the last year and for being an fantastic research mentor for the last 2 years. Thank you to Professor Michael Mahoney for feedback and suggestions on this work. Thank you to all the outstanding Berkeley professors I had the privilege of learning from for helping me build a strong technical foundation and inspiring me to dream big. Finally, thank you to my friends and family for the unparalleled and unconditional love and support throughout my time at Berkeley.

Contents

Contents	i
List of Figures	ii
List of Tables	iii
1 Introduction	1
2 Background and Prior Work	2
2.1 Natural Language Processing	2
2.2 Generalization and Generalization Metrics	3
2.3 Heavy-Tailed Self-Regularization Theory	4
3 Approach	6
3.1 In-Distribution Generalization vs. Out-of-Distribution Generalization	6
3.2 Generalization Performance vs. Generalization Gap	7
3.3 Data-Dependent Generalization Metrics vs. Data-Independent Generalization Metrics	8
3.4 Focus on Shape Metrics	8
4 Experiments and Results	9
4.1 Overview of Generalization Metrics	9
4.2 Datasets and Models	9
4.3 Predicting Generalization Performance throughout the Training Process	11
4.4 Predicting Trends in Generalization Performance	15
5 Conclusion and Future Work	21
Bibliography	23
A Generalization Metrics Details	27
B Experimental Setup Details	28

List of Figures

2.1	Transformer architecture	3
2.2	Examples of common distributions fitted to the ESDs.	5
4.1	To assess how well a generalization metric predicts generalization performance throughout training, we compute the rank correlation between the metric curve (red) and the generalization performance (orange) at each epoch. Which generalization performance curve we use differs if we are considering ID generalization or OOD generalization.	12
4.2	Shape metrics outperform other types of metrics at predicting generalization performance throughout training. Each boxplot is a distribution of the rank correlation between in-distribution BLEU score and the corresponding generalization metric over 150 Transformer models. The shape metrics (except <code>PL_alpha</code>) are computed by fitting a truncated power law distribution (Appendix A) to the ESDs.	13
4.3	Shape metrics outperform other metrics at predicting out-of-distribution generalization throughout training.	14
4.4	<code>TPL_alpha</code> exhibits characteristics of an “ideal” generalization metric.	16
4.5	<code>TPL_alpha</code> behaves closer to an “ideal” generalization metric than <code>PL_alpha</code>	16
4.6	The overall <code>alpha</code> vs. generalization performance trend is opposite for <code>TPL_alpha</code> compared to <code>PL_alpha</code>	17
4.7	When categorized by the learning rate used for training, there are significant differences in the trends between each metric and generalization performance.	18
4.8	The same trend exists for both in-distribution and out-of-distribution generalization.	19
4.9	<code>log_prod_of_fro</code> increases as a function of depth, but the generalization performance does not change.	20

List of Tables

4.1	Overview of the generalization metrics we consider. See [46] for the formal definitions of these metrics.	10
B.1	Models trained for Section 4.3. This is the same experimental setup as [46]. . . .	29

Chapter 1

Introduction

The goal of machine learning – like all predictive models – is to develop a model that generalizes from training data to new data. A model that is not able to make predictions on new data is of little use. Yet, although generalization is the core goal of machine learning, the current understanding of generalization in deep learning models is limited. Despite a vast body of literature on the topic, generalization in deep learning is still not well understood due to the inherent complexity of the problem.

Understanding the generalization of models is fundamentally difficult since it requires reasoning about the model’s interactions with unknown environments. When one adds the complexity introduced by deep learning, understanding generalization becomes significantly more subtle. Traditionally, generalization was quantified by deriving bounds based on assumptions about the model and data (e.g. VC theory). However, such generalization bounds are not easily derived for deep neural networks [8]. As the scale and intricacies of deep learning models and the data used to train them continually change, it is increasingly harder to formulate a comprehensive set of assumptions to inform a practical generalization bound. Moreover, the prevalence of pre-trained models being used on tasks they were not specifically trained for adds yet another dimension to generalization (out-of-distribution generalization) that theories of generalization in deep learning must handle. Despite the generalization of deep learning models not being well understood, the ubiquity of deep learning models in applications that affect our daily lives continues to grow. Therefore, it is imperative to continue the search for a satisfying theory of generalization in deep learning to prevent unwanted consequences of deep learning models.

One approach to understanding generalization is through the framework of *generalization metrics*, which aim to predict how well a model will generalize to new data. While there has been a recent rise in studying generalization metrics for deep learning, these works overwhelmingly focus on computer vision (CV). This work is among the first to instead examine generalization metrics for NLP. Additionally, our work is among the first to use generalization metrics to predict out-of-distribution generalization.

Chapter 2

Background and Prior Work

2.1 Natural Language Processing

Natural language processing (NLP) refers to the set of statistical techniques and methods used for automatic analysis of natural language (i.e. human language). Modern NLP is dominated by neural models that learn representations of natural language used for downstream tasks such as sentiment classification, machine translation, and question answering.

The evolution of modern NLP techniques can be understood from analyzing machine translation as a motivating example. In contrast to other areas of deep learning, machine translation presents the challenge of handling variable-length inputs since not all sentences are the same length. To address this issue, machine translation was cast as a sequence-to-sequence problem, where recurrent neural networks [45] are commonly used. However, recurrent neural networks struggle with long input sequences [30] due to vanishing/exploding gradients. This was a major problem for translation which often spans multiple sentences. As a response, long short-term memory (LSTM) models were designed to handle longer input sequences [15]. LSTMs were a breakthrough in sequence-to-sequence modeling and are still used in practice today for non-NLP sequence-to-sequence applications.

As more languages were demanded from translation systems, it became impractical to implement pairwise translation between languages. Encoder-decoder architectures were designed to address this problem by condensing all necessary information from the input (in the case of translation, a sentence in the source language) into a fixed-size vector, which in principle can be decoded into any target language. However, this fixed-size vector turned out to be an information bottleneck for natural language sentences, even when it was generated from an LSTM model. This led to the development of attention-based decoders [2] and finally the foundation for current state-of-the-art NLP models, the Transformer

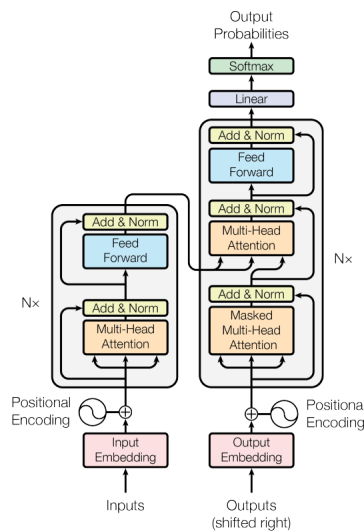


Figure 2.1: Transformer architecture

[42]. Intuitively, attention mechanisms allow decoders to “pay attention” to certain parts of the input sequence when decoding each step of the output sequence, which aligns well with the idea that certain words in a target language sentence correspond more to some words in a source language sentence than other words. The Transformer utilizes attention in an architecture (Figure 2.1) that empirically performs extremely well on many NLP tasks.

Modern state-of-the-art NLP models used in practice such as BERT [7] and RoBERTa [21] build on the Transformer architecture and design various training techniques to further extract performance. These models have a massive number of parameters and therefore require a large amount of training data (often web-scale) to learn language representations. As a result, training modern NLP models from scratch is inaccessible to most practitioners due to limited compute resources. Moreover, labeled data for each potential NLP task is scarce, rendering traditional supervised learning impractical. Therefore, modern NLP models are pre-trained on large, diverse corpora of unlabeled text using self-supervised learning [14]; these pre-trained models are used as foundational building blocks in NLP systems that achieve state-of-the-art performance on a wide variety of downstream tasks.

2.2 Generalization and Generalization Metrics

Generalization refers to a model’s ability to generalize from the training data distribution to an unseen data distribution. A model is said to generalize if its performance on data from an unseen distribution is similar to its performance on data from training data distribution. Concretely, if we consider the canonical supervised learning setup in which a

model \mathcal{M} is trained on a dataset $\mathcal{D}_{\text{train}}$ sampled from a data distribution \mathcal{P} , then the model’s generalization refers to its tendency to perform well on \mathcal{P} despite only being trained on $\mathcal{D}_{\text{train}}$. In addition to this type of generalization – known as *in-distribution* generalization – there is also *out-of-distribution* generalization, which characterizes a model’s ability to perform well on a distribution $\mathcal{P}' \neq \mathcal{P}$. Out-of-distribution generalization has received increasing attention recently due to the prevalence of pre-trained models as building blocks to be used on tasks they were not specifically trained for. Although in-distribution and out-of-distribution generalization are similar, out-of-distribution is considered separately because good in-distribution generalization does not guarantee out-of-distribution generalization [13].

While the generalization of statistical models is explained by VC theory and related methods [41, 12], it has been commonly observed that these methods are not fit for understanding the generalization of deep neural networks [28]. As a result, researchers have explored many new theories to explain generalization in deep learning [33, 3, 32, 11, 1, 31, 43, 22]. At the core of understanding generalization is the notion of a *generalization metric* (sometimes also called a generalization/complexity measure), a quantity that is designed to possess a monotonic relationship with generalization. Informally, this means an “ideal” generalization metric should order the models with respect to their ability to generalize. Formally, an “ideal” generalization metric μ should satisfy

$$\forall m_1, m_2 \in \mathbb{M}, G(m_1) > G(m_2) \implies \mu(m_1) > \mu(m_2)$$

where \mathbb{M} is a set of models and G represents the generalization of a model. Note that since the requirement for μ to be “ideal” is it must have a monotonic relationship with generalization, the signs in the equation above can be flipped. In addition, it is possible (and common) for μ to have access to more than just the model.

Generalization metrics are a great tool in studying generalization because they can both validate the theory behind the metric and be used in practice as a predictive tool. As such, there is interest to focus on using generalization metrics to predict the generalization of deep learning models [19].

2.3 Heavy-Tailed Self-Regularization Theory

Heavy-Tailed Self-Regularization (HT-SR) theory involves analyzing the empirical spectral densities of the correlation matrices of a deep neural network’s weight matrices (henceforth referred to as ESDs). The ESD is a histogram of the eigenvalues of a correlation matrix of a weight matrix. Concretely, if \mathbf{W} is a weight matrix, then $\mathbf{X} = \mathbf{W}^\top \mathbf{W}$ is the corresponding correlation matrix. The ESD of \mathbf{X} is the histogram of its eigenvalues. HT-SR theory is motivated by (1) the empirical observation that the weight matrices of

well-trained deep neural networks have spectral densities that are heavy-tailed¹ [24, 29, 23] and (2) the common practice within statistical physics of modeling strongly-correlated systems with heavy-tailed distributions [39].

HT-SR theory asserts that the ESDs organically follow a heavy-tailed distribution as an artifact of the optimization process [24]. Moreover, it claims that the shape of the ESDs holds information about the quality of the model [27]. In order to extract the shape of the ESDs, one has to fit a particular distribution (specified a priori) to the ESDs and use information from the shape of the fitted distribution as a proxy for the true shape of the ESDs. However, there are a few challenges with fitting a heavy-tailed distribution to the ESDs:

1. It is hard to know a priori which distribution to fit to the ESDs. Some commonly used distributions are shown in Figure 2.2.
2. It is numerically difficult to fit heavy-tailed distributions to the ESDs [25, 6, 26]. Thankfully, the Transformer-based models common in modern NLP have many large linear layers, enabling more accurate fitting [46].
3. The ability of the fitted distribution to extract useful information about the shape of the ESDs is sensitive to the choice of fitted distribution and the quality of the fit [46].

Overcoming these challenges is the subject of current work (including this one).

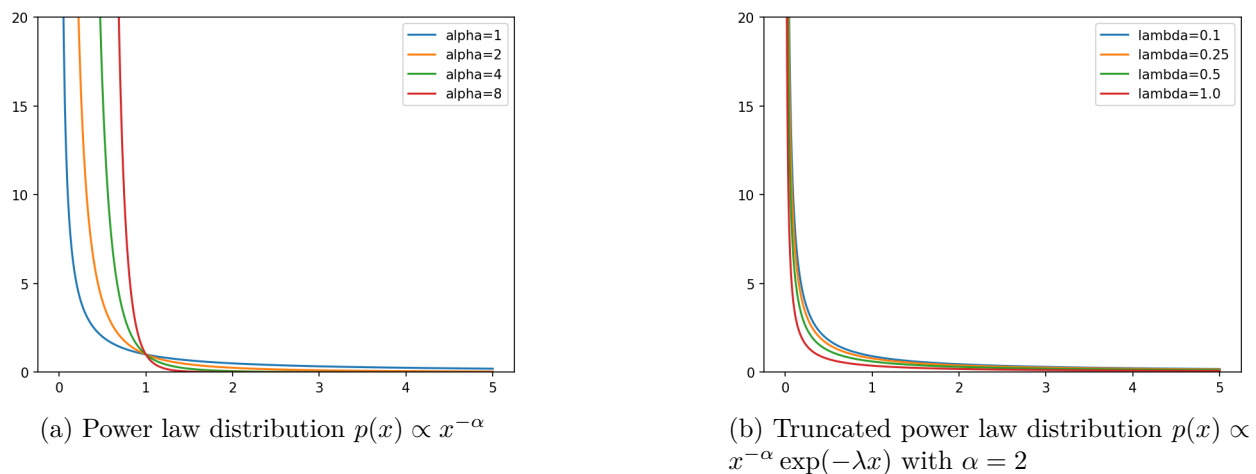


Figure 2.2: Examples of common distributions fitted to the ESDs.

¹Heavy-tailed distributions are probability distributions with a relatively large amount of probability mass very far from the mean.

Chapter 3

Approach

We compare various generalization metrics based on

- their ability to predict generalization performance throughout the training process (Section 4.3)
- their ability to predict trends in the generalization performance of fully trained models (Section 4.4)

The first scenario is interesting to consider because it captures the idea that loss landscapes in NLP are particularly complex and therefore understanding the quality of the training process is desirable [47]. The second scenario more closely resembles common ways of evaluating generalization metrics [20], but we are among the first to study generalization metrics for NLP.

There are many experimental design choices when evaluating generalization metrics, including:

- What type of generalization is considered?
- What quantity is the generalization metric predicting?
- What information does the generalization metric have access to?

In the following sections, we discuss where our approach falls within each of the above questions.

3.1 In-Distribution Generalization vs. Out-of-Distribution Generalization

As discussed in Chapter 2, generalization can take on different forms. Whereas in-distribution generalization refers to a model’s ability to generalize from training data to

test data drawn from the same distribution (in expectation), out-of-distribution generalization refers to a model’s ability to generalize to data drawn from a different distribution than the training data distribution. In-distribution generalization has been the focus of prior work in generalization metrics, and although generalization metrics for in-distribution generalization is still an ongoing open research area, we highlight the importance of also studying generalization metrics for out-of-distribution generalization given the dominance of pre-trained models in today’s NLP landscape (Chapter 2).

Thus, we consider both in-distribution and out-of-distribution generalization in our analysis. To our knowledge, we are among the first to study generalization metrics for out-of-distribution generalization within NLP.

3.2 Generalization Performance vs. Generalization Gap

Prior work in generalization metrics, which has been dominated by computer vision, has focused on using generalization metrics to predict the generalization gap. In NLP, predicting the generalization gap may not be appropriate because it is nearly impossible to train modern NLP models to convergence due to the massive size of the training data and the inherent noisiness present in natural language. Therefore, the generalization gap does not carry the same meaning in NLP as in other areas of deep learning because NLP models often do not even fully learn the training data (i.e. the gap between training error and test error cannot be attributed to differences in the test data alone since there are still portions of the training data the model has yet to learn).

There is also a practical reason to prefer predicting generalization performance over the generalization gap. Imagine a scenario in which one needs to choose between 2 models for a task. Suppose it is known that Model A has 2% training error, Model B has 5% training error, and an ideal generalization metric predicts that Model B has a lower generalization gap than Model A. Despite having full knowledge of the training performance of both models and having an ideal generalization metric, this is still not enough information to know which is the better model [46]. The core issue is that a generalization metric’s value can only be interpreted in a *relative* context, so if one wants to choose between 2 models based on which model has a better generalization performance, the generalization metric must predict generalization performance rather than generalization gap.

Thus, we predict generalization performance directly as opposed to generalization gap in our analysis.

3.3 Data-Dependent Generalization Metrics vs. Data-Independent Generalization Metrics

Compared to other domains, the training data used for modern NLP models is often inaccessible to users due to the scale of the data. Therefore, it is preferable to study generalization metrics which predict generalization without needing access to data. While we include data-dependent generalization metrics in our study, we pay special attention to the data-independent generalization metrics due to their relative ease of application compared to data-dependent generalization metrics. While one may expect that access to data is crucial to predicting generalization, recent work suggests that access to data is not needed [27].

3.4 Focus on Shape Metrics

While we compare many generalization metrics in our analysis, we pay special attention to shape metrics derived from HT-SR theory for a few reasons.

1. Shape metrics are data-independent, which allows them to more easily be applied in practice (Section 3.3).
2. HT-SR Theory has been shown to be effective in measuring the quality of the training process [16, 17]. Gaining insight into the quality of the training process is particularly important in NLP, where the loss landscapes are uniquely complex [47].
3. Shape metrics are computed by fitting heavy-tail distributions to ESDs. This idea seems to be a natural fit for NLP, where actual data often follow heavy-tail distributions [10].

Chapter 4

Experiments and Results

4.1 Overview of Generalization Metrics

We study 36 generalization metrics spanning various motivations/theories (presented in Chapter 2) with the goal of understanding the degree to which each metric is appropriate for use in predicting the generalization of models common in NLP. A summary of all metrics we study is captured in Table 4.1.

4.2 Datasets and Models

We use the following datasets commonly used as benchmarks for neural machine translation [35, 42, 38, 9] for our experiments:

- **IWSLT** [5]: A dataset of TED talk transcripts in multiple languages. We utilize IWSLT for machine translation on English-German sentence pairs (around 200K pairs).
- **WMT14** [4]: A dataset of news articles in multiple languages. We utilize WMT14 for machine translation on English-German sentence pairs (around 4.5 million pairs). We henceforth refer to this dataset as WMT.

We measure generalization performance in terms of BLEU [36] – the most commonly used metric to evaluate machine translation performance.

We study the generalization of Transformer [42] models, a foundational model in modern NLP. We use an open-source implementation¹ of the Transformer model which reproduces the results from the Transformer paper and follow the training setup from the Transformer paper. See Appendix B for details of the model.

¹<https://github.com/gordicaleksa/pytorch-original-transformer>

Table 4.1: Overview of the generalization metrics we consider. See [46] for the formal definitions of these metrics.

Metric Name	Proposed by	Scale/Shape	Data Requirements
l2	-	Scale	Data-independent
l2_dist	-	Scale	Data-independent
param_norm	[18]	Scale	Data-independent
fro_dist	[18]	Scale	Data-independent
log_norm	[24]	Scale	Data-independent
log_sum_of_fro	[18]	Scale	Data-independent
log_spectral_norm	[25]	Scale	Data-independent
dist_spec_init	[18]	Scale	Data-independent
log_prod_of_fro	[18]	Scale	Data-independent
log_sum_of_spec	[18]	Scale	Data-independent
log_prod_of_spec	[18]	Scale	Data-independent
path_norm	[33]	Scale	Data-independent
mp_softrank	[24]	Shape	Data-independent
stable_rank	[24]	Shape	Data-independent
alpha	[24]	Shape	Data-independent
exponent	[44]	Shape	Data-independent
ks_distance	[24]	Shape	Data-independent
tail_mean_vec_entropy	[44]	Shape	Data-independent
bulk_mean_vec_entropy	[44]	Shape	Data-independent
entropy	[24]	Shape	Data-independent
rand_distance	[44]	Shape	Data-independent
alpha_weighted	[24]	Hybrid	Data-independent
log_alpha_norm	[25]	Hybrid	Data-independent
inverse_margin	[18]	Scale	Data-dependent
log_prod_of_spec_over_margin	[3, 37]	Scale	Data-dependent
log_sum_of_spec_over_margin	[3, 37]	Scale	Data-dependent
log_prod_of_fro_over_margin	[3, 37]	Scale	Data-dependent
log_sum_of_fro_over_margin	[3, 37]	Scale	Data-dependent
path_norm_over_margin	[33]	Scale	Data-dependent
pacbayes_init	[34]	Scale	Data-dependent
pacbayes_orig	[34]	Scale	Data-dependent
pacbayes_flatness	[34]	Scale	Data-dependent
pacbayes_mag_init	[18]	Scale	Data-dependent
pacbayes_mag_orig	[18]	Scale	Data-dependent
pacbayes_mag_flatness	[18]	Scale	Data-dependent

4.3 Predicting Generalization Performance throughout the Training Process

In this experiment, our goal is to evaluate the generalization metrics’ ability to predict generalization performance throughout the training process. As discussed in Chapter 3, we consider 2 types of generalization: in-distribution (ID) generalization and out-of-distribution (OOD) generalization; generalization performance refers to both of these types. We will distinguish between the types of generalization when appropriate.

Experimental Setup

We train Transformer models in 50 different settings by varying the dataset, number of samples used for training, network depth, learning rate, and dropout. We choose these variables to vary to study the effectiveness of generalization metrics in the face of changes to the data (varied dataset and number of samples), the model size (varied depth), and the optimization process (varied learning rate and dropout). A summary of all settings is provided in Table B.1. We train 3 Transformer models per setting (3 different seeds). This is the same experimental setup as [46].

We utilize Spearman’s rank correlation coefficient [40] (henceforth referred to as rank correlation) to measure how well a generalization metric predicts a model’s generalization performance. Rank correlation (r_s) is defined as the Pearson correlation coefficient (ρ) between the rank variables. Formally,

$$r_s(X, Y) = \rho(R(X), R(Y)) = \frac{\text{cov}(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}$$

As the name suggests, the rank variable $R(X)$ transforms X into ranks (relative position of each value of X). Intuitively, rank correlation measures how well the relationship between two variables can be described by a monotonic function. In our experiments, X is the generalization performance of a model at each epoch and Y is the value of a generalization metric of a model at each epoch. See Figure 4.1 for a visual depiction. A high (in magnitude) rank correlation means that the generalization metric is useful in predicting generalization performance throughout training, whereas a low rank correlation means that the metric is not.

Results

Shape metrics outperform other metrics

Overall, shape metrics are the best generalization metrics among the metrics we consider at predicting the generalization performance throughout training (Figure 4.2). There are a few “scale” metrics (norm-based metrics) that also seem to predict generalization

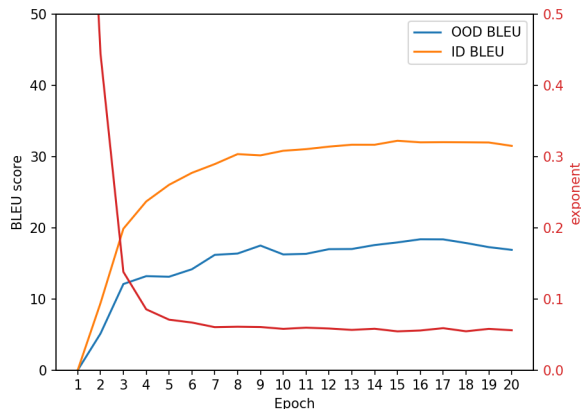


Figure 4.1: To assess how well a generalization metric predicts generalization performance throughout training, we compute the rank correlation between the metric curve (red) and the generalization performance (orange) at each epoch. Which generalization performance curve we use differs if we are considering ID generalization or OOD generalization.

performance relatively well (namely `inverse_margin`), but shape metrics dominate the metrics that perform well. A careful reader might notice that many of the scale metrics near the bottom of Figure 4.2 have a higher (in magnitude) median rank correlation than the shape metrics, which could indicate that those metrics are better at predicting generalization performance than the shape metrics. While it is true that some scale metrics have a higher median rank correlation than the shape metrics, this interpretation ignores an important part of the distribution of rank correlation: the width. A generalization metric which produces a rank correlation distribution of low width demonstrates the metric is often well-correlated with generalization performance despite differences in the training settings, indicating the generalization metric is indeed predicting generalization performance and not some other confounding variable. When the width of the rank correlation distribution is wide, we cannot be sure the metric is predicting generalization performance. Since the width of the rank correlation distributions is lower for the shape metrics, we say that the shape metrics are better at predicting generalization performance.

Another interesting observation is that `PL_alpha` (A.1) predicts generalization performance much better than `TPL_alpha` (A.2), yet `exponent` predicts generalization performance better than `PL_alpha` (Figure 4.2). This is a great example of the delicate nature of fitting distributions to the ESDs (Section 2). This result suggests that an exponential distribution could be a good candidate to fit to the ESDs, which we leave for future work (Chapter 5).

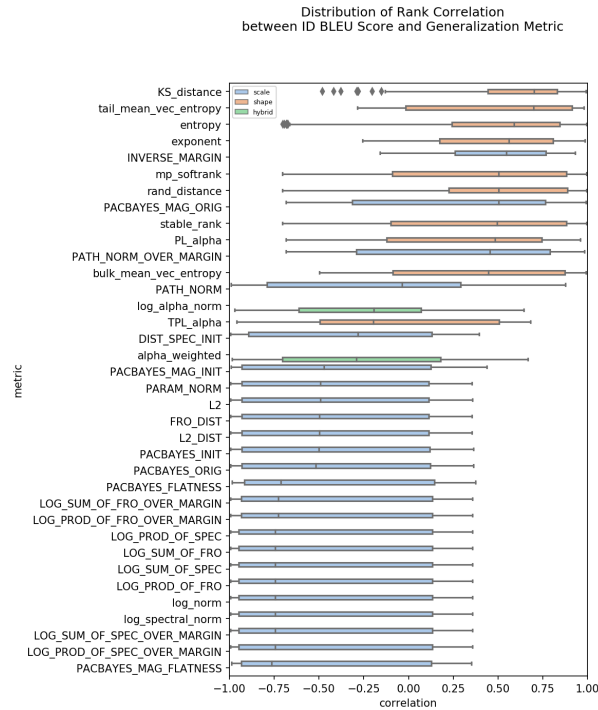
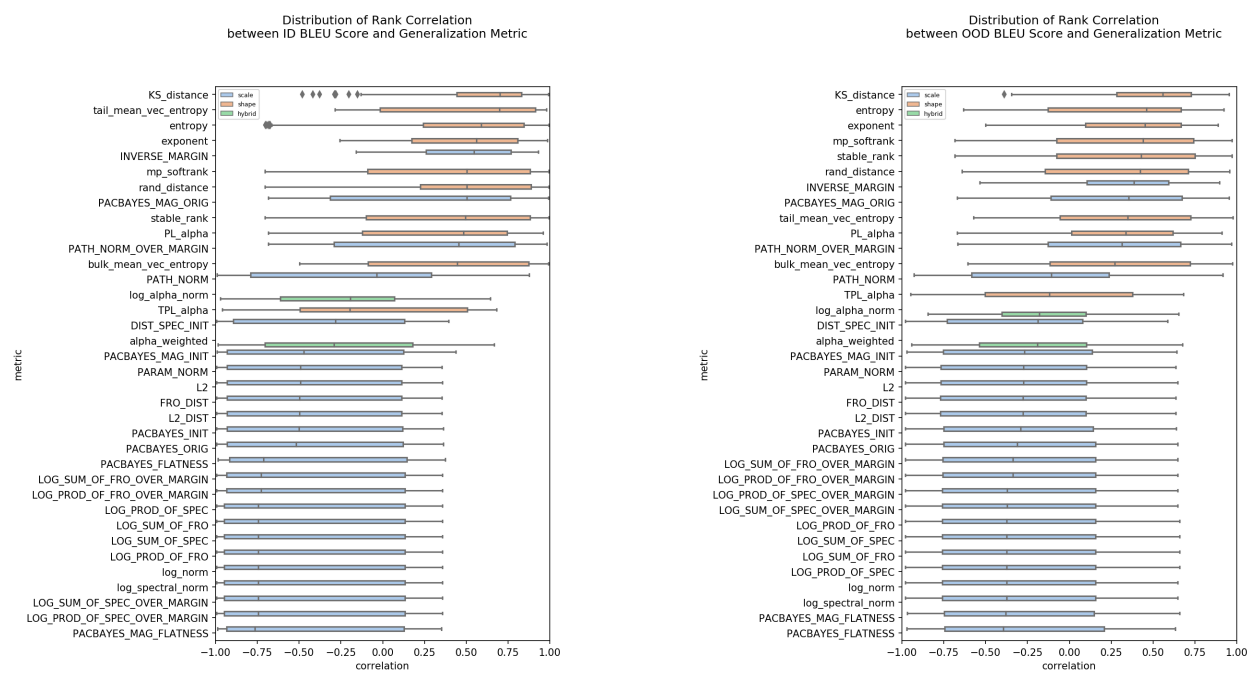


Figure 4.2: Shape metrics outperform other types of metrics at predicting generalization performance throughout training. Each boxplot is a distribution of the rank correlation between in-distribution BLEU score and the corresponding generalization metric over 150 Transformer models. The shape metrics (except `PL_alpha`) are computed by fitting a truncated power law distribution (Appendix A) to the ESDs.

In-Distribution vs. Out-of-Distribution Generalization

Like in-distribution generalization, shape metrics are the best generalization metrics among those we consider at predicting out-of-distribution generalization performance throughout training (Figure 4.3), which is an encouraging result as it suggests the relative performance of these generalization metrics is robust to changes in data distribution. However, the rank correlations themselves are noticeably lower (in magnitude) for OOD generalization than ID generalization which indicates the robustness of the generalization metrics to changes in data distribution is limited. It is worth noting that the difference in rank correlation from ID to OOD generalization is less for data-independent metrics, which may be a sign that they are more robust than the data-dependent metrics.



(a) Distribution of rank correlation between ID BLEU score and each generalization metric

(b) Distribution of rank correlation between OOD BLEU score and each generalization metric

Figure 4.3: Shape metrics outperform other metrics at predicting out-of-distribution generalization throughout training.

4.4 Predicting Trends in Generalization Performance

In this experiment, our goal is to evaluate the generalization metrics’ ability to predict trends in the generalization performance of fully trained models. As in Section 4.3, generalization performance refers to both in-distribution (ID) and out-of-distribution (OOD) generalization and we will distinguish between ID and OOD when appropriate.

Experimental Setup

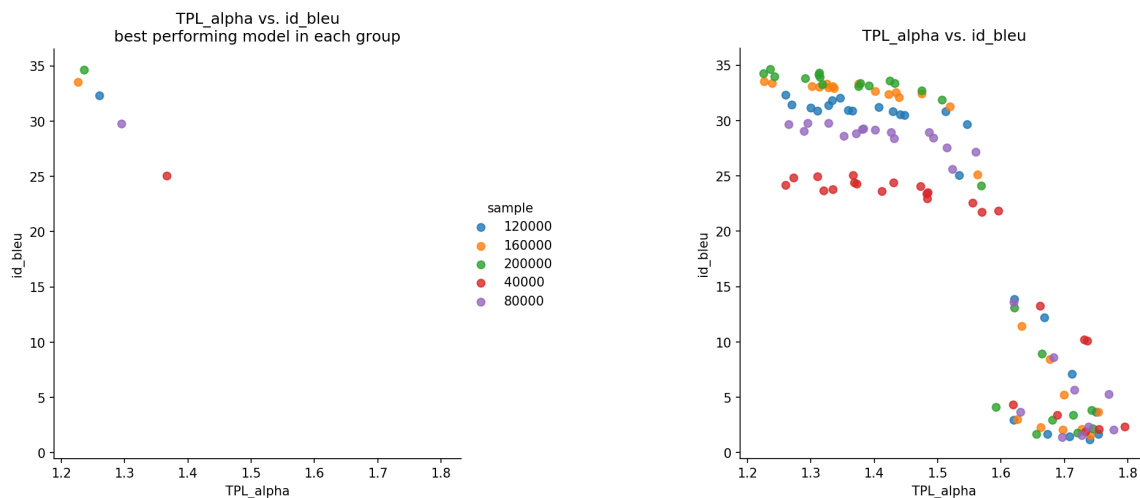
We train Transformer models in 125 different settings by varying the number of samples used for training, network depth, and learning rate. We choose these variables to vary to study the effectiveness of generalization metrics in the face of changes to the data (varied number of samples), the model size (varied depth), and the optimization process (varied learning rate). A summary of all settings is provided in Appendix B. For each model, we evaluate its BLEU score on a test set from the dataset it was trained on – called in the in distribution (ID) dataset – and its BLEU score on a dataset which it was not trained on – called the out of distribution (OOD) dataset. In our experiments, the ID dataset is IWSLT and the OOD dataset is WMT. We consider the same metrics as in Section 4.3.

Results

“Ideal” Generalization Metrics

There are some metrics which exhibit characteristics of “ideal” generalization metrics. We show `TPL_alpha` as an example of such a metric in Figure 4.4 below. An “ideal” generalization metric ranks models in the same order as their generalization performance (see Section 2) for more discussion on “ideal” generalization metrics), which would manifest in Figure 4.4 as a linear trend between the best performing models within each group.

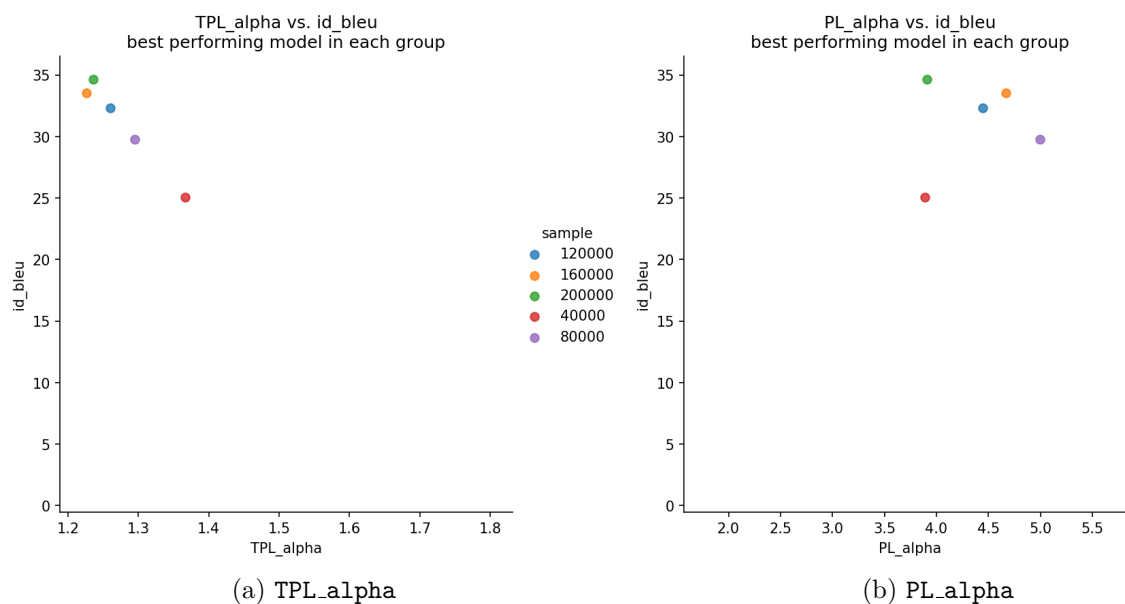
While it is promising that `TPL_alpha` exhibits characteristics of an “ideal” generalization metrics, it is important to keep in mind that shape metrics are sensitive to the distribution used to fit the ESDs. In comparison to `TPL_alpha`, `PL_alpha` does not show as strong of a correlation with generalization performance (Figure 4.5). Moreover, the overall trend between `alpha` and generalization performance is the opposite for `TPL_alpha` and `PL_alpha` (Figure 4.6).



(a) There is a strong linear association between `TPL_alpha` and in-distribution test BLEU score. Each point on this scatter plot represents the best performing model with each group, where each group consists of all models trained with a specific number of samples (meaning the depth and learning rate are varied within each group).

(b) This scatter plot shows `TPL_alpha` vs. in-distribution test BLEU score for all 125 models, colored by the number of samples each model was trained on. Generally, as the number of samples increases, so does the BLEU score. The best-performing points from the figure to the left are chosen from this scatter plot.

Figure 4.4: `TPL_alpha` exhibits characteristics of an “ideal” generalization metric.



(a) `TPL_alpha`

(b) `PL_alpha`

Figure 4.5: `TPL_alpha` behaves closer to an “ideal” generalization metric than `PL_alpha`.

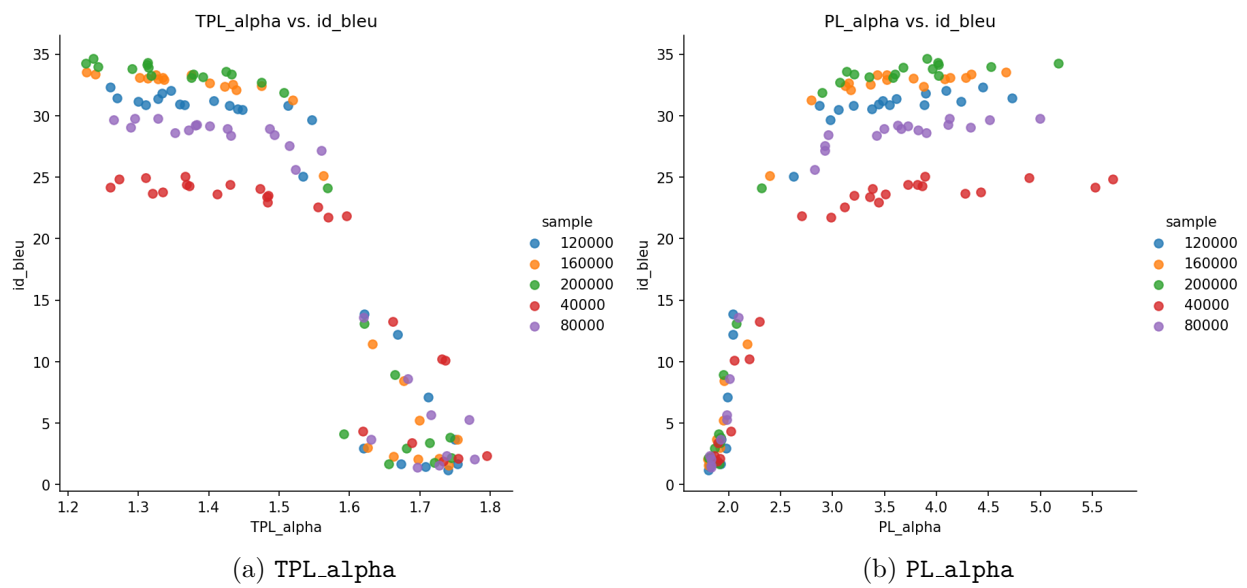


Figure 4.6: The overall alpha vs. generalization performance trend is opposite for TPL_alpha compared to PL_alpha.

Simpson’s Paradox in Well-Trained vs. Poorly-Trained Models

When models are categorized by the learning rate used during training, we observe significant differences in the linear association between each metric and generalization performance. Figure 4.7, shows examples of the main types of trends we observe. In many cases, these trends are reminiscent of a Simpson’s paradox. We are yet to explore what causes these differences, but observe that the well-trained models (models achieving relatively high generalization performance) follow the same trend while the poorly-trained models follow the opposite (or otherwise different) trend. One hypothesis is there is a true trend between each generalization metric and generalization performance, but this true trend only holds for well-trained models. This hypothesis is reasonable because poorly trained models can search vastly different parts of the loss landscape in comparison to well-trained models (especially in NLP where the loss landscape is known to be particularly complex [47]), thus making it hard for generalization metrics to capture their behavior. With this hypothesis, we can interpret the different subgroups trends in Figure 4.7 as different “stages” of model quality, where the generalization metric becomes better calibrated (i.e. approaches its true relationship with generalization performance) as the model is more well-trained. We leave exploration of this hypothesis to future work.

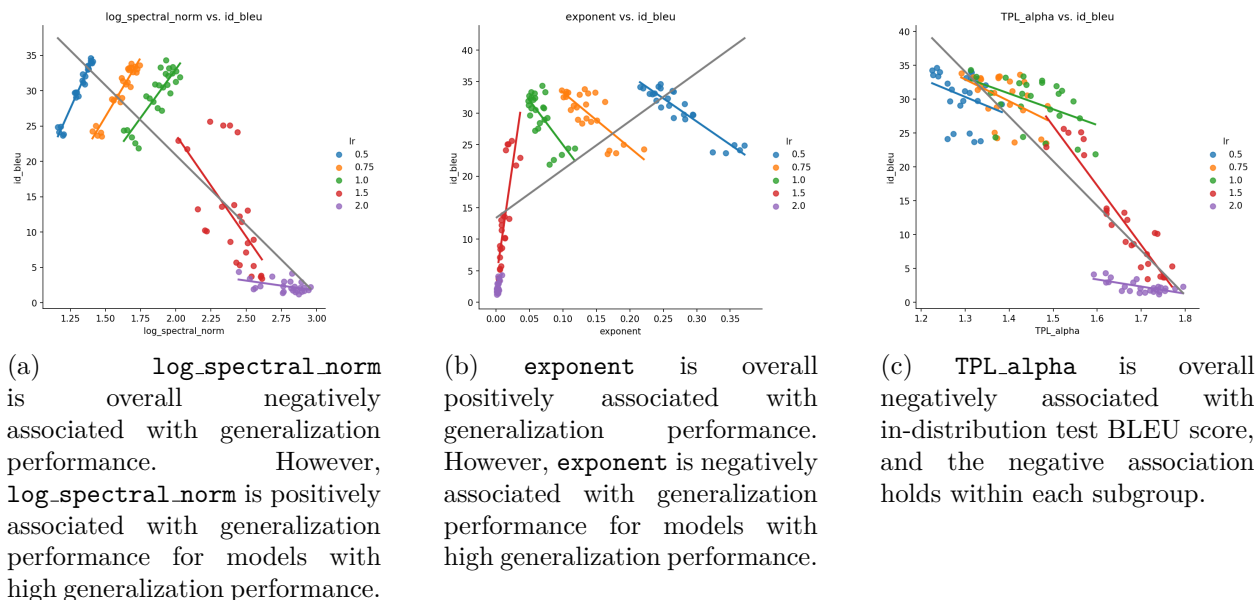


Figure 4.7: When categorized by the learning rate used for training, there are significant differences in the trends between each metric and generalization performance.

In-Distribution vs. Out-of-Distribution Generalization

For all metrics considered, the observed trends do not differ from ID to OOD generalization (see Figure 4.8 for an example). The only change is OOD generalization performance is generally lower than ID generalization performance, which is not surprising as the distribution of OOD data is more different than the distribution of ID test data to the distribution of training data. This is an encouraging result, as it indicates that the generalization metrics we consider are robust to changes in data distribution.

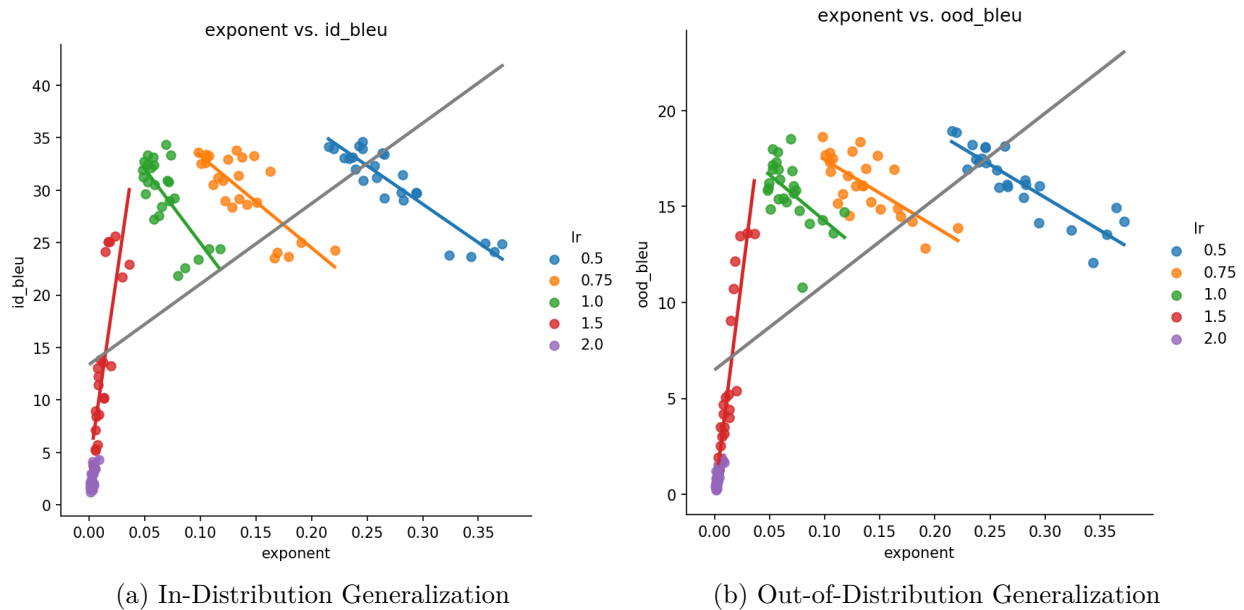


Figure 4.8: The same trend exists for both in-distribution and out-of-distribution generalization.

Effect of Network Depth

This section briefly discusses an observation that is not particularly relevant to our analysis but is interesting to point out. Overall, network depth does not seem to affect generalization performance much. However, some generalization metrics – especially those involving products – are sensitive to network depth/number of parameters (see Figure 4.9 for an example), indicating “engineering issues” can arise from improper applications of such metrics.

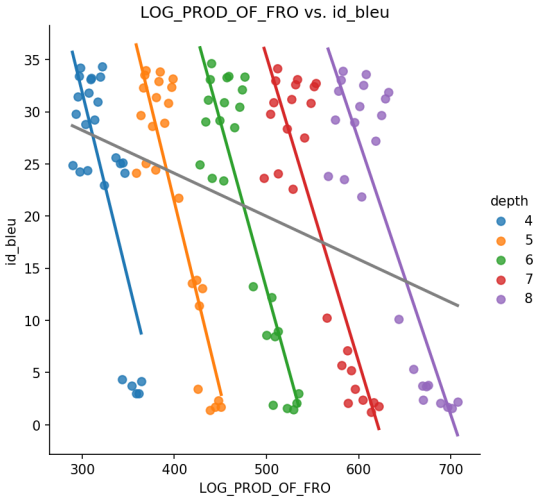


Figure 4.9: log_prod_of_fro increases as a function of depth, but the generalization performance does not change.

Chapter 5

Conclusion and Future Work

As deep neural networks become increasingly common in applications affecting our everyday lives, it is important we gain a deeper understanding of the properties that allow these models to perform as well as they do. While there are many ways to understand the behavior of deep neural networks, the framework of generalization metrics is a particularly useful way of dissecting deep neural network behavior since a generalization metric can validate the theory behind the metrics and can be used as a predictive tool in practice.

In this thesis, we build on prior work in generalization metrics to conduct one of the first analyses of generalization metrics for NLP models. We consider both in-distribution and out-of-distribution generalization and also pay special attention to shape metrics, or those metrics derived from fitting heavy-tail distributions to the ESDs of weight matrices of deep neural networks. Our main findings are:

1. Shape metrics are a promising category of generalization metrics. They are the best metrics among those we consider at predicting generalization performance throughout training (Figure 4.2) and show characteristics of being “ideal” generalization metrics (Figure 4.4). This is an encouraging result since shape metrics have many benefits (Section 3.4).
2. The generalization metrics we consider are generally robust to changes in data distribution. The trends between generalization metrics and generalization performance hold for both in-distribution and out-of-distribution generalization (Figure 4.8). However, there are signs that this robustness is limited, which we encourage future work to examine.
3. The practicality of generalization metrics may be dependent on the quality of the model in consideration. Although our results are preliminary, we observe signs that generalization metrics tend to show different trends with generalization performance in well-trained versus poorly-trained models (Section 4.4). These trends are reminiscent of a Simpson’s paradox.

There are several interesting opportunities for future work, in addition to extending our analysis with more models/NLP tasks:

- **Fitting different distributions to the ESDs:** As mentioned throughout our work, shape metrics are sensitive to the choice of distribution fit to the ESDs. In our work we saw that fitting a power law vs. a truncated power law both yield promising results, but there are signs that other distributions might be even better (Section 4.3). For example, there has already been work exploring fitting an exponential distribution $p(x) \propto \exp(-\lambda x)$ to the ESDs [46].
- **Modeling OOD transforms:** Our work explores the relationship between generalization metrics and OOD generalization performance by training a model on one dataset and measuring OOD generalization performance on another dataset with the same task. However, this is a simplistic way of modeling OOD transforms. There is a vast space of opportunity for more thorough experimental setup to explore OOD generalization. For example, one idea we considered (but ultimately did not pursue due to compute restrictions) is to create a *spectrum* of OOD datasets with varying amounts of OOD data to examine how generalization metrics’ performance change as a function of “dataset dissimilarity”.
- **Design of experimental setup:** Designing a way to measure the effectiveness of a generalization metric is as important as designing the generalization metric itself. It is not a straightforward problem since the goal is to establish causality between generalization metrics and generalization performance. Prior work has explored setups towards capturing causal relationships [18], but more work is needed.

Bibliography

- [1] Sanjeev Arora et al. *Stronger generalization bounds for deep nets via a compression approach*. 2018.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.
- [3] Peter Bartlett, Dylan Foster, and Matus Telgarsky. “Spectrally-normalized margin bounds for neural networks”. In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 6241–6250.
- [4] Ondřej Bojar et al. “Findings of the 2014 workshop on statistical machine translation”. In: *Proceedings of the ninth workshop on statistical machine translation*. 2014, pp. 12–58.
- [5] Mauro Cettolo et al. “Report on the 11th iwslt evaluation campaign, iwslt 2014”. In: *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*. Vol. 57. 2014.
- [6] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. “Power-law distributions in empirical data”. In: *SIAM review* 51.4 (2009), pp. 661–703.
- [7] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019.
- [8] Gintare Karolina Dziugaite et al. “In search of robust measures of generalization”. In: *Advances in Neural Information Processing Systems* 33 (2020).
- [9] Sergey Edunov et al. “Understanding Back-Translation at Scale”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 489–500.
- [10] Vitaly Feldman. “Does learning require memorization? a short tale about a long tail”. In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020, pp. 954–959.

- [11] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. *Size-Independent Sample Complexity of Neural Networks*. 2017.
- [12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2003.
- [13] Dan Hendrycks et al. *Pretrained Transformers Improve Out-of-Distribution Robustness*. 2020.
- [14] Dan Hendrycks et al. “Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty”. In: *Advances in Neural Information Processing Systems*. 2019.
- [15] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* (1997).
- [16] Liam Hodgkinson and Michael W Mahoney. “Multiplicative noise and heavy tails in stochastic optimization”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4262–4274.
- [17] Liam Hodgkinson et al. “Generalization Properties of Stochastic Optimizers via Trajectory Analysis”. In: *arXiv preprint arXiv:2108.00781* (2021).
- [18] Yiding Jiang et al. “Fantastic Generalization Measures and Where to Find Them”. In: *International Conference on Learning Representations*. 2019.
- [19] Yiding Jiang et al. *NeurIPS 2020 Competition: Predicting Generalization in Deep Learning*. 2020.
- [20] Yiding Jiang et al. “NeurIPS 2020 Competition: Predicting Generalization in Deep Learning”. In: *arXiv preprint arXiv:2012.07976* (2020).
- [21] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).
- [22] Philip M. Long and Hanie Sedghi. *Generalization bounds for deep convolutional neural networks*. 2019.
- [23] Charles H Martin and Michael W Mahoney. “Heavy-tailed Universality predicts trends in test accuracies for very large pre-trained deep neural networks”. In: *Proceedings of the 20th SIAM International Conference on Data Mining*. 2020.
- [24] Charles H Martin and Michael W Mahoney. “Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning”. In: *Journal of Machine Learning Research* 22.165 (2021), pp. 1–73.
- [25] Charles H Martin and Michael W Mahoney. *Post-mortem on a deep learning contest: a Simpson’s paradox and the complementary roles of scale metrics versus shape metrics*. Tech. rep. Preprint: arXiv:2106.00734. 2021.
- [26] Charles H Martin and Michael W Mahoney. *Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior*. Tech. rep. Preprint: arXiv:1710.09553. 2017.

- [27] Charles H Martin, Tongsu Serena Peng, and Michael W Mahoney. “Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data”. In: *Nature Communications* 12.1 (2021), pp. 1–13.
- [28] Charles H. Martin and Michael W. Mahoney. *Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior*. 2018.
- [29] Charles H. Martin and Michael W. Mahoney. *Traditional and Heavy-Tailed Self Regularization in Neural Network Models*. 2019.
- [30] Michael C Mozer. “Induction of Multiscale Temporal Structure”. In: *Advances in Neural Information Processing Systems*. 1991.
- [31] Vaishnavh Nagarajan and J. Zico Kolter. *Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience*. 2019.
- [32] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. “A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks”. In: *International Conference on Learning Representations*. 2018.
- [33] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. “Norm-based capacity control in neural networks”. In: *Conference on Learning Theory*. PMLR. 2015, pp. 1376–1401.
- [34] Behnam Neyshabur et al. “Exploring Generalization in Deep Learning”. In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 5947–5956.
- [35] Myle Ott et al. “Scaling Neural Machine Translation”. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. 2018, pp. 1–9.
- [36] Kishore Papineni et al. “Bleu: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.
- [37] Konstantinos Pitas, Mike Davies, and Pierre Vandergheynst. “Pac-bayesian margin bounds for convolutional neural networks”. In: *arXiv preprint arXiv:1801.00171* (2017).
- [38] Dinghan Shen et al. “A simple but tough-to-beat data augmentation approach for natural language understanding and generation”. In: *arXiv preprint arXiv:2009.13818* (2020).
- [39] Didier Sornette. *Critical phenomena in natural sciences: chaos, fractals, selforganization and disorder: concepts and tools*. Springer Science & Business Media, 2006.
- [40] C. Spearman. “The Proof and Measurement of Association between Two Things”. In: *The American Journal of Psychology* (1904).
- [41] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

- [42] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.
- [43] Colin Wei and Tengyu Ma. *Data-dependent Sample Complexity of Deep Neural Networks via Lipschitz Augmentation*. 2019.
- [44] *WeightWatcher*. <https://github.com/CalculatedContent/WeightWatcher>. 2018.
- [45] Ronald J. Williams and David Zipser. “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks”. In: *Neural Computation* (1989).
- [46] Yaoqing Yang et al. “Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data”. In: *arXiv preprint arXiv:2202.02842* (2022).
- [47] Yaoqing Yang et al. “Taxonomizing local versus global structure in neural network loss landscapes”. In: *Thirty-Fifth Conference on Neural Information Processing Systems*. 2021.

Appendix A

Generalization Metrics Details

We use the same definitions and follow the same method of calculating each metric as in [46]. The only difference is that this work explores multiple ways of computing `alpha`:

- `PL_alpha`: The value of `alpha` when computed by fitting a power law (PL) distribution to the ESDs.

$$p(x) \propto x^{-\alpha}, \quad x_{\min} < x < x_{\max} \quad (\text{A.1})$$

- `TPL_alpha`: The value of `alpha` when computed by fitting a truncated power law (TPL) distribution to the ESDs.

$$p(x) \propto x^{-\alpha} \exp(-\lambda x), \quad x_{\min} < x < x_{\max} \quad (\text{A.2})$$

Appendix B

Experimental Setup Details

For all experiments, we use a Transformer model with 8 attention heads and an embedding dimension of 512. We train with the inverse square-root learning rate and 10% label smoothing. We train each model for 20 epochs. When calculating the ESDs of the weight matrices, we treat the query, key and value matrices as separate weight matrices.

Given the embedding dimension d_e , step number t , number of warm-up steps t_w , the formula for the inverse square-root learning rate schedule is the following.

$$\text{Learning Rate} = d_e^{-0.5} \cdot \min(t^{-0.5}, t \cdot t_w^{-1.5}).$$

In our experiments, we multiply this learning rate by a constant factor, shown in the “Learning rate” column of Table B.1.

Table B.1: Models trained for Section 4.3. This is the same experimental setup as [46].

Dataset	Number of samples	Learning rate	Network depth	Dropout	Number of training epochs
IWSLT	10K	1	6	0.1	20
IWSLT	10K	1	6	0.0	20
IWSLT	20K	1	6	0.1	20
IWSLT	20K	1	6	0.0	20
IWSLT	40K	1	6	0.1	20
IWSLT	40K	1	6	0.0	20
IWSLT	80K	1	6	0.1	20
IWSLT	80K	1	6	0.0	20
IWSLT	160K	1	6	0.1	20
IWSLT	160K	1	6	0.0	20
IWSLT	160K	0.75	6	0.1	20
IWSLT	160K	0.75	6	0.0	20
IWSLT	160K	0.5	6	0.1	20
IWSLT	160K	0.5	6	0.0	20
IWSLT	160K	0.375	6	0.1	20
IWSLT	160K	0.375	6	0.0	20
IWSLT	160K	0.25	6	0.1	20
IWSLT	160K	0.25	6	0.0	20
IWSLT	160K	1	5	0.1	20
IWSLT	160K	1	5	0.0	20
IWSLT	160K	1	4	0.1	20
IWSLT	160K	1	4	0.0	20
IWSLT	160K	1	3	0.1	20
IWSLT	160K	1	3	0.0	20
IWSLT	160K	1	2	0.1	20
IWSLT	160K	1	2	0.0	20
WMT	160K	1	6	0.1	20
WMT	160K	1	6	0.0	20
WMT	320K	1	6	0.1	20
WMT	320K	1	6	0.0	20
WMT	640K	1	6	0.1	20
WMT	640K	1	6	0.0	20
WMT	1.28M	1	6	0.1	20
WMT	1.28M	1	6	0.0	20
WMT	1.28M	0.75	6	0.1	20
WMT	1.28M	0.75	6	0.0	20
WMT	1.28M	0.5	6	0.1	20
WMT	1.28M	0.5	6	0.0	20
WMT	1.28M	0.375	6	0.1	20
WMT	1.28M	0.375	6	0.0	20
WMT	1.28M	0.25	6	0.1	20
WMT	1.28M	0.25	6	0.0	20
WMT	1.28M	1	5	0.1	20
WMT	1.28M	1	5	0.0	20
WMT	1.28M	1	4	0.1	20
WMT	1.28M	1	4	0.0	20
WMT	1.28M	1	3	0.1	20
WMT	1.28M	1	3	0.0	20
WMT	1.28M	1	2	0.1	20
WMT	1.28M	1	2	0.0	20

For the experiments in Section 4.4, we vary the following hyperparameters to create 125 settings:

- **Number of Samples:** takes on values {40K, 80K, 120K, 160K, 200K}
- **Network Depth:** takes on values {4, 5, 6, 7, 8}. These values correspond to the number of Transformer encoder/decoder layers.
- **Learning Rate:** takes on values {0.5, 0.75, 1.0, 1.5, 2.0}. These values correspond to the constant factor multiplying the learning rate used for training (see discussion of learning rate above).

All models are trained with dropout 0.1 on IWSLT. The OOD dataset is WMT.