Finding the Optimal Vocabulary Size for Neural Machine Translation

Thamme Gowda and Jonathan May
Information Sciences Institute
University of Southern California
{tg, jonmay}@isi.edu

Abstract

We cast neural machine translation (NMT) as a classification task in an autoregressive setting and analyze the limitations of both classification and autoregression components. Classifiers are known to perform better with balanced class distributions during training. Since the Zipfian nature of languages causes imbalanced classes, we explore its effect on NMT. We analyze the effect of various vocabulary sizes on NMT performance on multiple languages with many data sizes, and reveal an explanation for *why* certain vocabulary sizes are better than others.[1]

1 Introduction

Natural language processing (NLP) tasks such as sentiment analysis (Maas et al., 2011; Zhang et al., 2015) and spam detection are modeled as classification tasks, where instances are independently labeled. Tasks such as part-of-speech tagging (Zeman et al., 2017) and named entity recognition (Tjong Kim Sang and De Meulder, 2003) are examples of structured classification tasks, where instance classification is decomposed into a sequence of per-token contextualized labels. We can similarly cast neural machine translation (NMT), an example of a natural language generation (NLG) task, as a form of structured classification, where an instance label (a translation) is generated as a sequence of contextualized labels, here by an autoregressor (see Section 2).

Since the parameters of modern machine learning (ML) classification models are estimated from training data, whatever biases exist in the training data will affect model performance. Among those biases, *class imbalance* is a topic of our interest. Class imbalance is said to exist when one or more classes are not of approximately equal frequency in data. The effect of class imbalance has been extensively studied in several domains where classifiers are used (see Section 6.3). With neural networks, the imbalanced learning problem is mostly targeted to computer vision tasks; NLP tasks are under-explored (Johnson and Khoshgoftaar, 2019).

Word types in natural language models resemble a Zipfian distribution, i.e. in any natural language corpus, we observe that a type's rank is roughly inversely proportional to its frequency. Thus, a few types are extremely frequent, while most of the rest lie on the long tail of infrequency. Zipfian distributions cause two problems in classifier-based NLG systems:

1. **Unseen Vocabulary:** Any hidden data set may contain types not seen in the finite set used for training. A sequence drawn from a Zipfian distribution is likely to have a large number of rare types, and these are likely to have not been seen in training.
2. **Imbalanced Classes:** There are a few extremely frequent types and many infrequent types, causing an extreme imbalance. Such an imbalance, in other domains where classifiers are used, has been known to cause undesired biases and severe performance degradation (Johnson and Khoshgoftaar, 2019).

The use of *subwords*, that is, decomposition of word types into pieces, such as the widely used Byte Pair Encoding (BPE) (Sennrich et al., 2016) addresses the open-ended vocabulary problem by ultimately allowing a word to be represented as a sequence of characters if necessary. BPE has a single hyperparameter named *merge operations* that governs the vocabulary size. The effect of this hyperparameter is not well understood. In practice, it is either chosen arbitrarily or via trial-and-error (Salesky et al., 2018).

[1] Tools, configurations, system outputs, and analyses are at https://github.com/thammegowda/005-nmt-imbalance

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3955–3964

Regarding the problem of imbalanced classes, Steedman (2008) states that "the machine learning techniques that we rely on are actually very bad at inducing systems for which the crucial information is in rare events." However, to the best of our knowledge, this problem has not yet been directly addressed in the NLG setting.

In this work, we attempt to find answers to these questions: *'What value of BPE vocabulary size is best for NMT?'*, and more crucially an explanation for *'Why that value?'*. As we will see, the answers and explanations for those are an immediate consequence of a broader question, namely *'What is the impact of Zipfian imbalance on classifier-based NLG?'*

The contributions of this paper are as follows: We offer a simplified view of NMT architectures by re-envisioning them as two high-level components: a *classifier* and an *autoregressor* (Section 2). We describe some of the desired settings for the classifier (Section 2.1) and autoregressor (Section 2.2) components. In Section 2.3, we describe how vocabulary size choice relates to the desired settings for the two components. Our experimental setup is described in Section 3, followed by an analysis of results in Section 4 that offers an explanation with evidence for *why* some vocabulary sizes are better than others. Section 5 uncovers the impact of class imbalance, particularly frequency based discrimination on classes.[2] Section 6 provides an overview of related work, and in Section 7 we recommend a heuristic for choosing the BPE hyperparameter.

2 Classifier based NLG

Machine translation is commonly defined as the task of transforming sequences from the form $x = x_1x_2x_3...x_m$ to $y = y_1y_2y_3...y_n$, where x is in source language X and y is in target language Y. There are many variations of NMT architectures (Section 6.1), however, all share the common objective of maximizing $\prod_{t=1}^{n} P(y_t|y_{<t}, x_{1:m})$ for pairs $(x_{1:m}, y_{1:n})$ sampled from a parallel dataset. NMT architectures are commonly viewed as encoder-decoder networks. We instead re-envision the NMT architecture as two higher level components: an autoregressor (R) and a token classifier (C), as shown in Figure 1.

Autoregressor R, (Box et al., 2015) being the most complex component of the NMT model, has many implementations based on various neural network architectures: recurrent neural networks (RNN) such as long short-term memory (LSTM) and gated recurrent unit (GRU), convolutional neural networks (CNN), and Transformer (Section 6.1). At time step t, R transforms the input context $y_{<t}, x_{1:m}$ into hidden state vector $h_t = R(y_{<t}, x_{1:m})$.

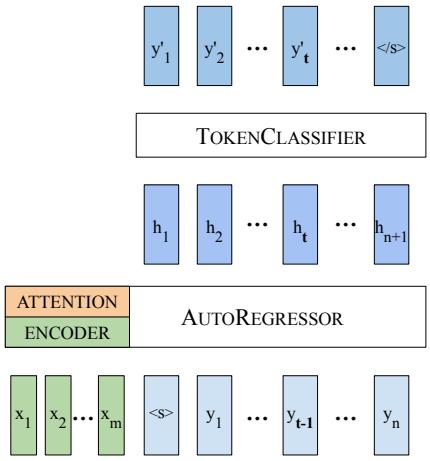


Figure 1: The NMT model re-envisioned as a token classifier with an autoregressive feature extractor.

Classifier C is the same across all architectures. It maps h_t to a distribution $P(y_j|h_t)\forall y_j \in V_Y$, where V_Y is the vocabulary of Y. In machine learning, input to classifiers such as C is generally described as features that are either hand-engineered or automatically extracted. In our high-level view of NMT architectures, R is a neural network that serves as an automatic feature extractor for C.

2.1 Balanced Classes for Token Classifier

Untreated, class imbalance leads to bias based on class frequencies. Specifically, classification learning algorithms focus on frequent classes while paying relatively less importance to infrequent classes. Frequency-based bias leads to poor recall of infrequent classes (Johnson and Khoshgoftaar, 2019).

When a model is used in a *domain mismatch* scenario, i.e. where test and training set distributions do not match, model performance generally degrades. It is not surprising that frequency-biased classifiers show particular degradation in domain mismatch scenarios, as types that were infrequent in the training distribution and were ignored by the learning algorithm may appear with high frequency in the new domain. Koehn and Knowles (2017) showed empirical evidence of poor generalization of NMT to out-of-domain datasets.

[2] In this work, 'type' and 'class' are used interchangeably.

In other classification tasks, where each instance is classified independently, methods such as up-sampling infrequent classes and down-sampling frequent classes are used. In NMT, since classification is done within the context of sequences, it is possible to accomplish the objective of balancing by altering sequence lengths. This can be done by choosing the level of subword segmentation (Sennrich et al., 2016).

Quantification of Zipfian Imbalance: We use two statistics to quantify the imbalance of a training distribution:

The first statistic relies on a measure of **Divergence** (D) from a balanced (uniform) distribution. We use a simplified version of Earth Mover Distance, in which the total cost for moving a probability mass between any two classes is the sum of the total mass moved. Since any mass moved *out of* one class is moved *into* another, we divide the total per-class mass moves in half to avoid double counting. Therefore, the imbalance measure D on K class distributions where p_i is the observed probability of class i in the training data is computed as:

$$D = \frac{1}{2}\sum_{i=1}^{K} |p_i - \frac{1}{K}|; \quad 0 \leq D \leq 1$$

A lower value of D is the desired setting for C, since the lower value results from a balanced class distribution. When classes are balanced, they have approximately equal frequencies; C is thus less likely to make errors due to class bias.

The second statistic is **Frequency at 95th% Class Rank** ($F_{95\%}$), defined as the least frequency in the 95^{th} percentile of most frequent classes. More generally, $F_{P\%}$ is a simple way of quantifying the minimum number of training examples for at least the$P$th percentile of classes. The bottom $(1 - P)$ percentile of classes are overlooked to avoid the noise that is inherent in the real-world natural-language datasets.

A higher value for $F_{95\%}$ is the desired setting for C, as a higher value indicates the presence of many training examples per class, and ML methods are known to perform better when there are many examples for each class.

2.2 Shorter Sequences for Autoregressor

Every autoregressive model is an approximation; some may be better than others, but no model is perfect. The total error accumulated grows in proportion to the length of the sequence. These accumulated errors alter the prediction of subsequent tokens in the sequence. Even though beam search attempts to mitigate this, it does not completely resolve it. These challenges with respect to long sentences and beam size are examined by Koehn and Knowles (2017).

We summarize sequence lengths using **Mean Sequence Length**, μ, computed trivially as the arithmetic mean of the lengths of *target* language sequences after encoding them: $\mu = \frac{1}{N}\sum_{i=1}^{N} |y^{(i)}|$ where $y^{(i)}$ is the ith sequence in the training corpus of N sequences. Since shorter sequences have relatively fewer places where an imperfectly approximated autoregressor model can make errors, a smaller μ is a desired setting for R.

2.3 Choosing the Vocabulary Size Systematically

BPE (Sennrich et al., 2016) is a greedy iterative algorithm often used to segment a vocabulary into useful *subwords*. The algorithm starts with characters as its initial vocabulary. In each iteration, it greedily selects the most frequent type bigram in the training corpus, and replaces the sequence with a newly created compound type. Once the subword vocabulary is learned, it can be applied to a corpus by greedily segmenting words with the longest available subword type. These operations have an effect on D, $F_{95\%}$, and μ.

Effect of BPE on μ: BPE expands rare words into two or more subwords, lengthening a sequence (and raising μ) relative to simple white-space segmentation. BPE merges frequent-character sequences into one subword piece, shortening a sequence (and lowering μ) relative to character segmentation. Hence, the sequence length of BPE segmentation lies in between the sequence lengths obtained by white-space and character-only segmentation methods (Morishita et al., 2018).

Effect of BPE on $F_{95\%}$ and D: Whether BPE is viewed as a merging of frequent subwords into a relatively less frequent compound, or a splitting of rare words into relatively frequent subwords, BPE alters the class distribution by moving the probability mass of classes. Hence, by altering the class distribution, BPE also alters both $F_{95\%}$ and D. The BPE hyperparameter controls the amount of probability mass moved between subwords and compounds.

Figure 2 shows the relation between number of BPE merges (i.e. the BPE hyperparameter), and both D and μ. When few BPE merge operations are performed, we observe the lowest value of D, which is a desired setting for C, but at the same point μ is large and undesired for R (Section 2). When a large number of BPE merges are performed, the effect is reversed, i.e. we observe that D is large and unfavorable to C while μ is small and favorable to R. In the following sections we describe our experiments and analysis to locate the optimal number of BPE merges that achieves the right trade-off for both C and R.

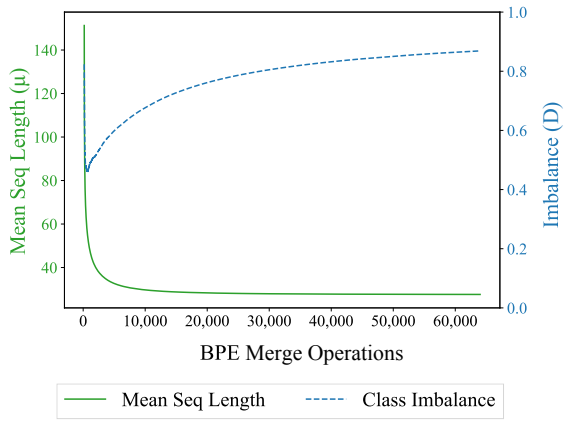


Figure 2: Effect of BPE merge operations on mean sequence length (μ) and class imbalance (D).

3 Experimental Setup

Our NMT experiments use the base Transformer model (Vaswani et al., 2017) on four different target languages at various training data sizes, described in the following subsections.

3.1 Datasets

We use the following four language pairs for our analysis: English→German, German→English, English→Hindi, and English→Lithuanian. To analyze the impact of different training data sizes, we randomly sub-select smaller training corpora for English↔German and English→Hindi languages. Statistics regarding the corpora used for validation, testing, and training are in Table 1. The datasets for English↔German, and English→Lithuanian are retrieved from the News Translation task of WMT2019 (Barrault et al., 2019).[3] For English→Hindi, we use the IIT Bombay Hindi-English parallel corpus v1.5 (Kunchukuttan et al., 2018). English, German, and Lithuanian sentences are tokenized using SacreMoses.[4] Hindi sentences are tokenized using IndicNlpLibrary.[5]

The training datasets are trivially cleaned: we exclude sentences with length in excess of five times the length of their parallel counterparts. Since the vocabulary is a crucial part of this analysis, we exclude all sentence pairs containing URLs.

3.2 Hyperparameters

Our model is a 6 layer Transformer encoder-decoder that has 8 attention heads, 512 hidden vector units, and a feed forward intermediate size of 2048, with GELU activation. We use label smoothing at 0.1, and a dropout rate of 0.1. We use the Adam optimizer (Kingma and Ba, 2015) with a controlled learning rate that warms up for 16K steps followed by the decay rate recommended for training Transformer models (Popel and Bojar, 2018). To improve performance at different data sizes we set the mini-batch size to 6K tokens for the 30K-sentence datasets, 12K tokens for 0.5M-sentence datasets, and 24K for the remaining larger datasets (Popel and Bojar, 2018). All models are trained until no improvement in validation loss is observed, with a patience of 10 validations, each done at 1,000 update steps apart. Our model is implemented using PyTorch and run on NVIDIA P100 and V100 GPUs. To reduce padding tokens per batch, mini-batches are made of sentences having similar lengths (Vaswani et al., 2017). We trim longer sequences to a maximum of 512 tokens after BPE. To decode, we average the last 10 checkpoints, and use a beam size of 4 with length penalty of 0.6, similar to Vaswani et al. (2017).

Since the vocabulary size hyperparameter is the focus of this analysis, we use a range of vocabulary sizes that include character vocabulary and BPE operations that yield vocabulary sizes between 500 and 64K types. A common practice, as seen in Vaswani et al. (2017)'s setup, is to jointly learn BPE for both source and target languages, which facilitates three-way weight sharing between the encoder's input, the decoder's input, and the output (i.e. classifier's class) embeddings (Press and Wolf, 2017). However, to facilitate fine-grained analysis of vocabulary sizes and their effect on class imbalance, our models separately learn source and target vocabularies; weight sharing between the encoder's

[3] http://www.statmt.org/wmt19/translation-task.html

[4] https://github.com/alvations/sacremoses

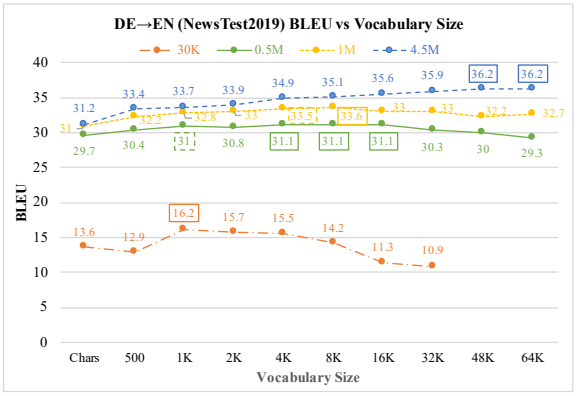
[5] https://github.com/anoopkunchukuttan/indic_nlp_library

Languages	Training	Sentences	EN Toks	XX Toks	Validation	Test
DE→EN EN→DE	Europarl v10	30K	0.8M	0.8M	NewsTest18	NewsTest19
	WMT13CommonCrawl	0.5M	12.9M	12.2M		
	NewsCommentary v14	1M	25.7M	24.3M		
		4.5M	116M	109.8M		
EN→HI	IITB Training	0.5M	8M	8.6M	IITB Dev	IITB Test
		1.3M	21M	22.5M		
EN→LT	Europarl v10	0.6M	17M	13.4M	NewsDev19	NewsTest19

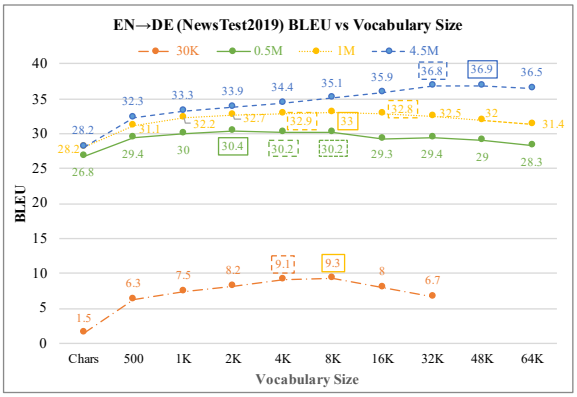
Table 1: Training, validation, and testing datsets, along with sentence and token counts in training sets. We generally refer to dataset's sentence size in this work.

and decoder's embeddings is thus not possible. For the target language, however, we share weights between the decoder's input and the classifier's class embeddings.

4 Results and Analysis



(a) DE→EN BLEU on NewsTest2019



(b) EN→DE BLEU on NewsTest2019

Figure 3: EN↔DE NewsTest2019 BLEU as a function of vocabulary size at various training set sizes. Only the large dataset with 4.5M sentences has its best performance at a large vocabulary; all others peak at an 8K or smaller vocabulary size.

BLEU scores for DE→EN and EN→DE experiments are reported in Figures 3a and 3b respectively. Results from EN→HI, and EN→LT are combined in Figure 4. All the reported BLEU scores are obtained using SacreBLEU (Post, 2018).[6]

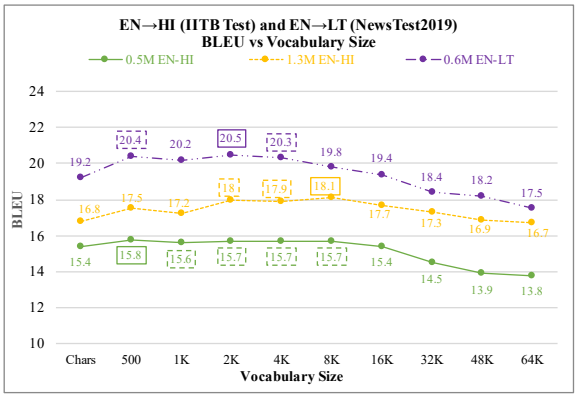


Figure 4: BLEU on EN→HI IITB Test and EN→LT NewsTest2019 as a function of vocabulary size. These language pairs observed the best BLEU scores in the range of 500 to 8K vocabulary size.

We make the following observations: smaller vocabulary such as characters have not produced the best BLEU for any of our language pairs or dataset sizes. A vocabulary of 32K or larger is unlikely to produce optimal results unless the data set is large e.g. the 4.5M DE↔EN sets. The BLEU curves as a function of vocabulary sizes have a shape resembling a hill. The position of the peak of the hill seems to shift towards a larger vocabulary when the datasets are large. However, there is a lot of variance in the position of the peak: one extreme is at 500 types on 0.5M EN→HI, and the other extreme is at 64K types in 4.5M DE→EN.

Although Figures 3 and 4 indicate *where* the optimal vocabulary size is for these chosen language pairs and datasets, the question of *why* the peak is where it is remains unanswered. We visualize μ, D, and $F_{95\%}$ in Figure 5 to answer that question,

[6] `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.6`

and report these observations:

1. Small vocabularies have a relatively larger $F_{95\%}$ (favorable to classifier), yet they are sub-optimal. We reason that this is due to the presence of a larger μ, which is unfavorable to the autoregressor.
2. Larger vocabularies such as 32K and beyond have a smaller μ which favors the autoregressor, yet rarely achieved the best BLEU. We reason this is due to the presence of a lower $F_{95\%}$ and a higher D being unfavorable to the classifier. Since the larger datasets have many training examples for each class, as indicated by a generally larger $F_{95\%}$, we conclude that bigger vocabularies tend to yield optimal results compared to smaller datasets in the same language.
3. On small (30K) to medium (1.3M) data sizes, the vocabulary size of 8K seems to find a good trade-off between μ and D, as well as between μ and $F_{95\%}$.

There is a *simple heuristic* to locate the peak: the near-optimal vocabulary size is where sentence length μ is small, while $F_{95\%}$ is approximately 100 or higher.

BLEU scores are often lower at larger vocabulary sizes—where μ is (favorably) low but D is (unfavorably) high (Figure 5). This calls for a further investigation that is reported in the following section.

5 Measuring Classifier Bias Due to Imbalance

In a typical classification setting with imbalanced classes, the classifier learns an undesired bias based on frequencies.

A balanced class distribution debiases in this regard, leading to improvement in the precision of frequent classes as well as recall of infrequent classes. However, BLEU focuses only on the *precision* of classes; except for adding a global brevity penalty, it is ignorant of the poor recall of infrequent classes.

Therefore, the BLEU scores shown in Figures 3a, 3b and 4 capture only a part of the improvements and biases. In this section we perform a detailed analysis of the impact of class balancing by considering both precision *and* recall of classes.

We accomplish this in two stages: First, we define a method to measure the bias of the model for classes based on their frequencies. Second, we track the bias in relation to vocabulary size and class imbalance, and report DE→EN, as it has many data points.

5.1 Frequency Based Bias

We measure frequency bias using the Pearson correlation coefficient, ρ, between class rank and class performance, where for performance measures we use precision and recall. Classes are ranked based on descending order of frequencies in the training data encoded with the same encoding schemes used for reported NMT experiments. With this setup, the class with rank 1, say F_1, is the one with the highest frequency, rank 2 is the next highest, and so on. More generally, F_k is an index in the class rank list which has an inverse relation to class frequencies. We define precision as P_k for class k similar to the unigram precision in BLEU and extend its definition to the unigram recall as R_k (See Appendix A for detail). The Pearson correlation coefficients between class rank and precision ($\rho_{F,P}$), and class rank and recall ($\rho_{F,R}$) are reported in Figure 6. In datasets where D is high, the performance of classifier correlates with class rank. Such correlations are undesired for a classifier.

5.2 Analysis of Class Frequency Bias

An ideal classifier is one that does not discriminate classes based on their frequencies, i.e. one that exhibits no correlation between $\rho_{F,P}$, and$\rho_{F,R}$. However, we see in Figure 6 that:

1. $\rho_{F,P}$ is positive when the dataset has high D; i.e if the class rank increases (frequency decreases), precision increases in relation to it. This indicates that frequent classes have relatively less precision than infrequent classes. The bias is strongly positive on smaller datasets such as 30K DE→EN, which gradually diminishes if the training data size is increased or a vocabulary setting is chosen to reduce D.
2. $\rho_{F,R}$ is negative, i.e., if the class rank increases, recall decreases in relation to it. This is an indication that infrequent classes have relatively lower recall than frequent classes.

Figure 6 shows a trend that frequency based bias measured by correlation coefficient is lower in settings that have lower D. However, since D is non-zero, there still exists non-zero correlation between

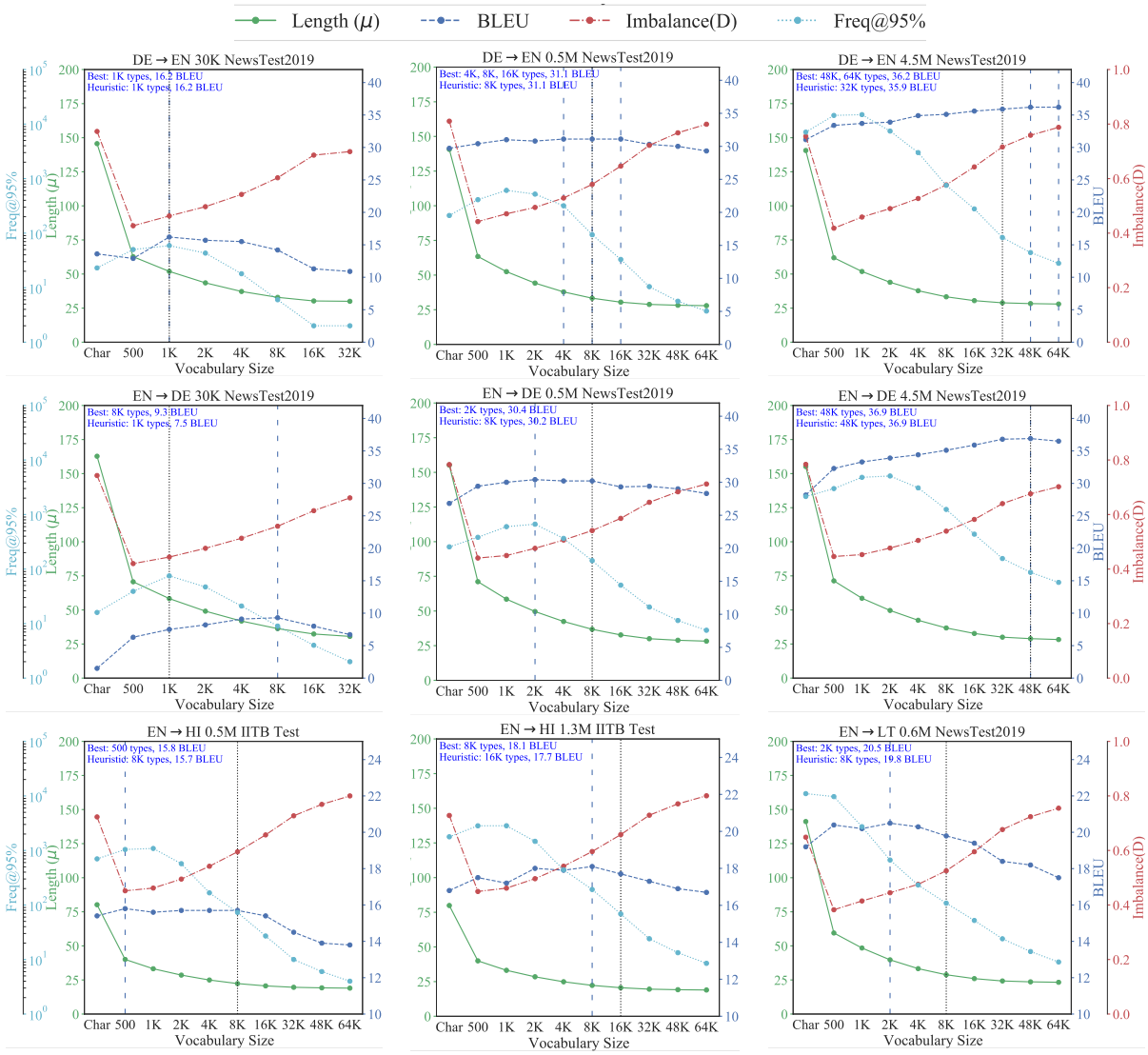


Figure 5: Visualization of sequence length (μ) (lower is better), class imbalance (D) (lower is better), frequency of 95^{th} percentile class ($F_{95\%}$) (higher is better; plotted in logarithmic scale), and test set BLEU (higher is better) on all language pairs and training data sizes. DE↔EN of 1M resembles resembles DE↔EN of 0.5M and is provided in Appendix B along with visualizations on validation sets. The vocabulary sizes that achieved highest BLEU are indicated with dashed vertical lines, and the vocabulary our heuristic selects is indicated by dotted vertical lines.

recall and class rank ($\rho_{F,R}$), indicating the poorer recall of low-frequency classes.

6 Related Work

6.1 NMT Architectures

Several variations of NMT models have been proposed and refined: Sutskever et al. (2014) and Cho et al. (2014b) introduce the RNN-based encoder-decoder model. Bahdanau et al. (2015) introduce the attention mechanism and Luong et al. (2015) propose several variations that became essential components of many future models. RNN modules, either LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014a), have been popular choices for composing NMT encoders and decoders. The encoder uses bidirectional information, but the decoder is unidirectional, typically left-to-right, to facilitate autoregressive generation. Gehring et al. (2017) use a CNN architecture that outperforms RNN models. Vaswani et al. (2017) propose the **Transformer**, whose main components are feed-forward and attention networks. There are only a few models that perform non-autoregressive NMT (Libovický and Helcl, 2018; Gu et al., 2018). These are focused on improving the speed of inference; generation quality is currently sub-par compared to autoregressive models. These non-autoregressive models can also be

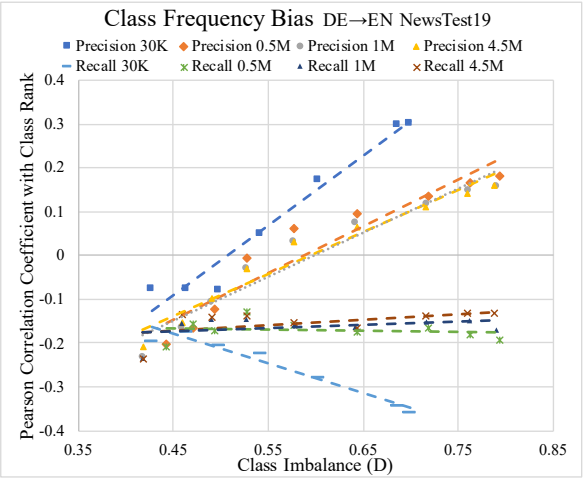


Figure 6: Correlation analysis on DE→EN shows that NMT models suffer from frequency based class bias, indicated by non-zero correlation of both precision and recall with class rank. Reduction in class imbalance (D), as shown by the horizontal axis, generally reduces the bias as indicated by the reduction in magnitude of correlation.

viewed as token classifiers with a different kind of feature extractor, whose strengths and limitations are yet to be theoretically understood.

6.2 BPE Subwords

Sennrich et al. (2016) introduce BPE as a simplified way to solve out-of-vocabulary (OOV) words without having to use a back-off dictionary for OOV words. They note that BPE improves the translation of not only the OOV words, but also some rare in-vocabulary words. The analysis by Morishita et al. (2018) is different than ours in that they view various vocabulary sizes as hierarchical features that are used in addition to a fixed vocabulary. Salesky et al. (2018) offer an efficient way to search BPE vocabulary size for NMT. Kudo (2018) use BPE as a regularization technique by introducing sampling based randomness to the BPE segmentation. To the best of our knowledge, no previous work exists that analyzes BPE's effect on class imbalance.

6.3 Class Imbalance

The class imbalance problem has been extensively studied in classical ML (Japkowicz and Stephen, 2002). In the medical domain Mazurowski et al. (2008) find that classifier performance deteriorates with even modest imbalance in the training data. Untreated class imbalance has been known to deteriorate the performance of image segmentation. Sudre et al. (2017) investigate the sensitivity of various loss functions. Johnson and Khoshgoftaar (2019) survey imbalance learning and report that the effort is mostly targeted to computer vision tasks. Buda et al. (2018) provide a definition and quantification method for two types of class imbalance: *step imbalance* and *linear imbalance*. Since the imbalance in Zipfian distribution of classes is neither single-stepped nor linear, we use a divergence based measure to quantify the imbalance.

7 Conclusion

Envisioning NMT as a token classifier with an autoregressor helps in analysing its weaknesses. Our analysis provides an explanation of *why* text generation using BPE vocabulary is more effective compared to word and character vocabularies, and *why* certain BPE hyperparameters are better than others. We show that the number of BPE merges is not an arbitrary hyperparameter, and that it can be tuned to address the class imbalance and sequence length problems. Our recommendation for Transformer NMT is to *use the largest possible BPE vocabulary such that at least 95% of classes have 100 or more examples in training*. Even though certain BPE vocabulary sizes indirectly reduce the class imbalance, they do not completely eliminate it. The class distributions after applying BPE contain sufficient imbalance for inducing the frequency based bias, especially affecting the recall of rare classes. Hence more effort in the future is needed to directly address the Zipfian imbalance.

Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract # FA8650-17-C-9116, and by research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-1000. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, Air Force Laboratory, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. 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*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249 – 259.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Maciej A. Mazurowski, Piotr A. Habas, Jacek M. Zurada, Joseph Y. Lo, Jay A. Baker, and Georgia D. Tourassi. 2008. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2):427 – 436. Advances in Neural Networks Research: IJCNN '07.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. Improving neural machine translation by incorporating hierarchical subword features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 618–629, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2018. Optimizing segmentation granularity for neural machine translation. *CoRR*, abs/1810.08641.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mark Steedman. 2008. On becoming a discipline. *Computational Linguistics*, 34(1):137–144.

Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, Cham. Springer International Publishing.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press.