Neural Machine Translation with Imbalanced Classes

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 the effect of class imbalance on NMT. We analyze the effect of vocabulary sizes on NMT performance and reveal an explanation for 'why' certain vocabulary sizes are better than others.

Introduction

NLP tasks such as sentiment analysis BIBREF0, BIBREF1, spam detection, etc., are modeled as classification tasks where instances are independently classified. Tasks such as part-of-speech tagging BIBREF2, and named entity recognition BIBREF3 are some examples for sequence tagging in which tokens are classified into tags within the context of sequences. Similarly, we can cast neural machine translation (NMT), an example of a natural language generation (NLG) task, as a form of classification task where tokens are classified within an autoregressor (see Section SECREF2) .

Since the parameters of ML classification models are estimated from training data, certain biases in the training data affect the final performance of model. 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 SECREF32). With neural networks, the imbalanced learning is mostly targeted to computer vision tasks; NLP tasks are underexplored BIBREF4. Word types in natural

language models follow a Zipfian distribution, i.e. in any natural language corpus, we observe that a few types are extremely frequent and the vast number of others lie on the long tail of infrequency. The Zipfian distribution thus causes two problems to the classifier based NLG systems:

Open-ended Vocabulary: Treating each word type in the vocabulary as a class of ML classifier does not cover the entire vocabulary, because the vocabulary is open-ended and classifiers model a finite set of classes only.

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 degradation in the performance BIBREF4.

Subwords obtained through e.g. byte pair encoding (BPE) BIBREF5 addresses the open-ended vocabulary problem by using only a finite set of subwords. Due to the benefit and simplicity of BPE, it is rightfully part of the majority of current NMT models. However, the choice of vocabulary size used for BPE is a hyperparameter whose effect is not well understood. In practice, BPE vocabulary choice is either arbitrary or chosen from several trial-and-errors.

Regarding the problem of imbalanced classes, steedman-2008-last 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 SECREF2). For the best performance of the classifier, we argue that the balanced class distribution is desired, and describe a method to measure class imbalance in a Zipfian distribution (Section SECREF6). For the best performance of the autoregressor, we argue that it is desired to have shorter sequences (Section SECREF7). In Section SECREF8, we describe how BPE vocabulary relates with the desired settings for both classifier and autoregressor. Our experimental setup is described in Section SECREF3, followed by the analysis of results in Section SECREF4 that offers an explanation with evidence for why some vocabulary sizes are better than others. Section SECREF5 uncovers the impact of class imbalance, particularly the discrimination on classes based on their frequency. Section SECREF6 provides an overview of the related work, followed by a conclusion in Section SECREF7.

Classifier based NLG

Machine translation is commonly defined as the task of transforming sequences from the form $x = x_1 x_2 x_3 ... x_m$ to $y = y_1 y_2 y_3 ... y_n$, where x is from source language X and y is from target language Y respectively. NMT accomplishes the translation objective using artificial neural networks.

There are many variations of NMT architectures with a varied range of differences (Section SECREF30), 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 a pair of 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 FIGREF4.

Autoregressor R, BIBREF6 being the main component of the NMT model, has many implementations based on various neural network architectures: RNNs such as LSTM and GRU, CNN, and Transformer (Section SECREF30). For any given time step t, R transforms the input context consisting of $y_{<t}, x_{1:m}$ into a hidden state vector as $h_t = R(y_{<t}, x_{1:m})$.

Classifier C is the same across all architectures. It maps h_t to a probability distribution $P(y_j | h_t) \forall y_j \in V_Y$, where V_Y is the vocabulary of Y. Intuitively, C scores h_t against an embedding of every class type, then transforms those arbitrarily ranged scores into a probability distribution using the SoftMax normalizer. In machine learning, input to classifiers such as C is generally described as features that are either hand-engineered or automatically extracted using neural networks. In this high-level view of NMT architecture, R is a neural network that serves as an automatic feature extractor for C.

Classifier based NLG ::: 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 a poor recall of infrequent classes.

When a model is used in a domain mismatch scenario, i.e. where a test set's distribution does not match the training set's distribution, 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 learning algorithm may appear with high frequency in the newer domain. koehn2017sixchallenges showed empirical evidence of poor generalization of NMT to out-of-domain datasets.

In other classification tasks, where each instance is classified independently, methods such as up-sampling the infrequent classes and down-sampling frequent classes are used. In NMT, since the classification is done within the context of sequences, it is possible to accomplish the objective of balancing by altering the lengths of sequences. This phenomenon of achieving balance by altering the sequence lengths is indirectly achieved by, e.g., BPE subword segmentation BIBREF5.

Quantification of Zipfian Imbalance: The class imbalance of an observed distribution of training classes is quantified as Divergence (D) from a balanced (uniform) distribution. Divergence is measured using a simplified version of Earth Mover Distance, in which the total cost for moving a probability mass between any two bins (analogous to class types) is the sum of the total mass moved. Since any mass moved out of one bin is moved into another, we divide the total per-bin 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:

The range of D is $0 \le D \le 1$, and we argue that a lower value of D a desired setting for C.

Classifier based NLG ::: Shorter Sequences for Autoregressor

Every autoregressive model is an approximation, some maybe better than others, but no model is a perfect one. Therefore, there is a non-zero probability of an error at each time step. The total error accumulated along the sequence 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 koehn2017sixchallenges. If sequence encoders such as BPE subwords can reduce the steps in the sequences, this indirectly reduces the errors in language generation by imperfectly approximated autoregressors.

We summarize sequence lengths using Mean Sequence Length, $\mu $, computed trivially as the arithmetic mean of the lengths of target language sequences after encoding them:

We argue that a smaller $\mu $ is a desired setting for R.

Classifier based NLG ::: Choosing the Vocabulary Size Systematically

BPE vocabulary is learned using a greedy and iterative algorithm BIBREF5. The BPE learning algorithm starts with characters as its initial vocabulary. In each iteration, it greedily selects a pair of the most frequent types (either characters or subwords) that co-occur, and replaces them with a newly created compound type. During segmentation, BPE splitting is performed left-to-right with greedily selecting the longest matched code in the vocabulary. These operations have an effect on both D and $\mu $.

Effect of BPE on $\mu $: BPE segmentation in comparison to word segmentation, expands rare words into two or more subwords, thus increases the sequence length. In comparison to character segmentation, BPE groups frequent characters as subwords thus reduces the length. BPE vocabulary size is more general that the words and characters are special cases that are attained at the two extremes BIBREF7. It can be used to create sequences that are long as character sequences (undesired for R), or short as word sequences (desired for R).

Effect of BPE on D: Whether viewed as a merging of frequent subwords into a relatively less frequent compound, or splitting of rare words into relatively frequent subwords, it alters the class distribution by moving the probability mass of classes. Hence, by altering class distribution, it also alters D.

Figure FIGREF9 shows the relation between the BPE vocabulary size on both D and $\mu $. A smaller vocabulary of BPE, after merging a few extremely frequent pairs, has smallest D which is a desired

setting for C, but at the same point $\mu $ is large and undesired for R. When BPE vocabulary is set to a large one, the effect is reversed i.e. D is large and unfavorable to C while $\mu $ small and favorable to R. As seen with evidence in Section SECREF4, there exists optimal vocabulary size of BPE that achieve the best setting for both C and R. Hence, BPE vocabulary size is not arbitrary since it can be tuned to reduce D while keeping $\mu $ short enough as well.

For a comparison, word and character segmentation have no influence on $\mu $. However, the trim size of word and character vocabulary has an effect on class imbalance D and Out-of-Vocabulary (OOV) tokens and is presented in Figures FIGREF9 and FIGREF9, respectively. The summary of word, character, and BPE with respect to D and $\mu $ is presented in Table TABREF10.

Experimental Setup

We perform NMT experiments using the base Transformer architecture BIBREF8. A common practice, as seen in vaswani2017attention's experimental setup, is to learn BPE vocabulary jointly for the source and target languages, which facilitates three-way weight sharing between the encoder's input, the decoder's input, and the decoder's output embeddings (classifier's class embeddings) BIBREF9. To facilitate fine-grained analysis of source and target vocabulary sizes and their effect on class imbalance, our models separately learn source and target vocabularies; weight sharing between the encoder's and decoder's embeddings is thus not possible. For the target language, however, we share weights between the decoder's input embeddings and the classifier's class embeddings.

Experimental Setup ::: Dataset

We use the publicly available Europarl v9 parallel data set for training German (De) and English (En) languages. We use 1.8M sentences of this corpus and build models in English to German and vice versa.

To segment initial words (i.e. before any subword processing) we use the Moses word tokenizer and detokenizer. We evaluate with the NewsTest2013 and NewsTest2014 datasets from the WMT 2014 news translation track.

Experimental Setup ::: Hyperparameters

Our Transformer NMT model has 6 layers in each of the encoder and decoder, 8 attention heads, 512 hidden vector units, and feed forward intermediate size of 2048. We use label smoothing at 0.1. We use the Adam optimizer BIBREF10 with a controlled learning rate that warms up for 8,000 steps followed by the decay rate recommended for training Transformer models. All models are trained for 100,000 optimizer steps. Mini-batch size per step is no more than 4,200 tokens. We group mini-batches into sentences of similar lengths to reduce padding tokens per batch BIBREF8. We trim sequences longer than 512 time steps. The average training time per experiment is 10Hrs on Nvidia 1080Ti GPUs. For inference (i.e decoding the test sets), we use checkpoint averaging of the last 5 states each, saved at 1000 optimizer steps apart, and a beam size of 4.

Analysis

We use character, word, and BPE subword encoding with various vocabulary sizes to analyze the effect of D and $\mu $. Each experiment is run twice and we report the mean of BLEU scores in Table TABREF15. The BLEU scores were computed using SacreBLEU BIBREF11 . All results are in Table TABREF15. We observe the following:

Experiments #1 and #2 use a word vocabulary, while #3 and #4 use a BPE vocabulary. The results show that with BPE, increasing the vocabulary size at this range reduces BLEU. Experiment #3 with a vocabulary as large as $64k$ BPE types even fails to reach the comparable Word model's (#1) BLEU

score, which raises the need for a systematic understanding of `Why BPE model reduced BLEU when vocabulary increased from $32k$ to $64k$?'. With increase in BPE vocabulary, $\mu $ is reduced which is favorable to R. An explanation is that the D increased which is unfavorable to C. For Word models, there is an effect of OOVs along with D, and it is beyond the scope of this work.

Experiments #3, #4, #5, #6 show that with BPE, decreasing the vocabulary indeed improves BLEU. Hence the larger BPE vocabulary such as $32k$ and $64k$ are not the best choice.

Experiments #7, #8, #9 and #10 with comparison to #6 showed that reducing vocabulary too much also negatively affects BLEU. Though Experiment #9 with $1k$ target vocabulary has the lowest D favoring the C, in comparison to others, the BLEU is still lower than the others. An explanation for this reduction is that $\mu $ is higher and unfavorable to R. Hence a strictly smaller vocabulary is not the best choice either.

By comparing #6 with #11, we see that, both have the same target vocabulary of $8k$, hence the same D and $\mu $, however, the source vocabulary differs from $8k$ to $32k$. Even though #11 had more imbalanced source types than #6, it has no adverse effect on BLEU. Therefore, imbalance on source vocabulary is not meaningful since source types are not the classes of C. Increasing the source vocabulary and hence rows in embeddings matrix is a simple way of increasing parameters of NMT model without hurting the BLEU.

Experiments #6 and #12 have differences in BLEU that is more significant than the previous pair (#6, #11). Here, both have the same $8k$ as source vocabulary, but the target differs from $8k$ to $32k$ which lead to noticeable differences in D and $\mu $. Even though #12 has more parameters in the target embeddings matrix, and smaller $\mu $ than #6, the BLEU is noticeably lower. An explanation we offer is that the $32k$ target types became classes and raised the class imbalance D, leading to a

reduction in the performance of C. This argument holds on both the directions of De-En and En-De. Thus, the class imbalance problem exists in NMT.

Measuring Classifier Bias due to Imbalance

In a typical classification setting with imbalanced classes, the classifier learns an undesired bias based on frequencies. Specifically, a biased classifier overclassifies frequent classes, leading to over recall but poor precision of frequent words, and underclassifies rare classes, leading to poor recall of rare words. An improvement in balancing the class distribution, therefore, debiases in this regard, leading to improvement in the precision of frequent classes as well as recall of infrequent classes. BLEU focuses only on the precision of classes; except for adding a global brevity penalty, it is ignorant to the poor recall of infrequent classes. Therefore, the numbers reported in Table TABREF15 capture only a part of the improvement from balanced classes. 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 on all our experiments.

Measuring Classifier Bias due to Imbalance ::: Class Frequency Bias Measurement

We measure frequency bias using the Pearson correlation coefficient, $\rho $, between class rank and class performance, where for performance measures we use precision and recall. We rank classes 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 P for a class similar to the unigram precision in BLEU and extend its definition to the unigram recall R. For the sake of clarity, consider a test dataset T of N pairs of parallel sentences, $(x^{(i)}, y^{(i)})$ where x and y are source and reference sequences respectively. We use single reference $y^{(i)}$ translations for this analysis. For each $x^{(i)}$, let $h^{(i)}$ be the translation hypothesis from an MT model.

Let the indicator $\mathbb {1}_k^{a}$ have value 1 iff type c_k exists in sequence a, where a can be either hypothesis $h^{(i)}$ or reference $y^{(i)}$. The function $count(c_k, a)$ counts the times token c_k exists in sequence a; $match(c_k, y^{(i)}, h^{(i)})$ returns the times c_k is matched between hypothesis and reference, given by $min\lbrace count(c_k, y^{(i)}), count(c_k, h^{(i)})\rbrace $

Let $P_k^{(i)}$ and $R_k^{(i)}$ be precision and recall of c_k on a specific record $i \in T$, given by:

Let P_k, R_k be the expected precision and recall for c_k over the whole T, given by:

The Pearson correlation coefficients between F_k vs. P_k, and F_k vs. R_k are reported in Table TABREF15 as $\rho _{F, P}$ and $\rho _{F, R}$ respectively.

Measuring Classifier Bias due to Imbalance ::: Analysis of Class Frequency Bias

A classifier that does not discriminate classes based on their frequencies is the one that exhibits no correlation between class rank vs precision and class rank vs recall. However, in the top rows of Table TABREF15 where larger vocabularies such as $64k$ are used, we make two observations:

$\rho _{F, P}$ is strong and positive. This is an indication that frequent classes have relatively less precision than infrequent classes. If the rank increases (i.e frequency is decreases), precision increases

in relation to it, leading to $\rho _{F, P} > 0$.

$\rho _{F, R}$ is strong and negative. This is an indication that frequent classes have relatively higher recall than infrequent classes. If the rank increases, recall decreases in relation to it, leading to $\rho _{F, R} < 0$.

Figure FIGREF26, as a visualization of Table TABREF15, shows a trend that the correlation (i.e. frequency bias) is lower with smaller vocabulary sizes. However, there still exists some correlation in $\rho _{F, R}$ since the class imbalance, $D > 0$.

Related Work

We categorize the related work into the subsections as following:

Related Work ::: NMT architectures

Several variations of NMT models have been proposed and refined: sutskever2014seq2seq, cho2014learning introduced recurrent neural network (RNN) based encoder-decoder models for sequence-to-sequence translation learning. bahdanau2014nmtattn introduced the attention mechanism and luong2015effectiveAttn proposed several variations that became essential components of many future models. RNN modules, either LSTM BIBREF12 or GRU BIBREF13, were the popular choice for composing encoder and decoder of NMT. The encoder used bidirectional information, but the decoder was unidirectional, typically left-to-right, to facilitate autoregressive generation. gehring2017CNNMT showed used convolutional neural network (CNN) architecture that outperformed RNN models. vaswani2017attention proposed another alternative called Transformer whose main components are feed-forward and attention networks. There are only a few models that perform non-autoregressive NMT

BIBREF14, BIBREF15. These are focused on improving the speed of inference and the generation quality is currently sub-par compared to autoregressive models. These non-autoregressive models can also be viewed as a token classifier with a different kind of feature extractor whose strengths and limitations are yet to be theoretically understood. Analyzing the non-autoregressive component, especially its performance with longer sequences, is beyond the scope of this work (however, an interesting direction).

Related Work ::: Bye Pair Encoding subwords

sennrich-etal-2016-bpe introduced byte pair encoding (BPE) as a simplified way for solving OOV words without using back-off models. They noted that BPE improved the translation of not only the OOV words, but also some of rare in-vocabulary words. In their work, the vocabulary size was arbitrary, and large as $60k$ and $100k$.

morishita-etal-2018-improving viewed BPE more generally in the sense that both character and word vocabularies as two special cases of BPE vocabulary. Their analysis was different than ours in a way that they viewed BPE with varied vocabulary sizes as hierarchical features which were used in addition to a fixed BPE vocabulary size of $16k$ on the target language. DBLP:journals/corr/abs-1810-08641 offer an efficient way to search BPE vocabulary size for NMT. kudo-2018-subword used BPE segmentation as a regularization by introducing sampling based randomness to the BPE segmentation. For the best of our knowledge, no previous work exists that analyzed BPE's effect on class imbalance or answered `why certain BPE vocabularies are better than others?'.

Related Work ::: Class Imbalance

The class imbalance problem has been extensively studied in classical ML BIBREF16. In the medical

domain Maciej2008MedicalImbalance found 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, and Sudre2017GeneralizedDice have investigated the sensitivity of various loss functions. Johnson2019SurveyImbalance surveyed imbalance learning with neural networks and reported that the effort is mostly targeted to computer vision tasks. buda-etal-2018-imbalance-cnn provided a definition and quantification method for two types of class imbalance: step imbalance and linear imbalance. Since natural languages are Zipfian, where the class imbalance is neither single stepped nor linear, we defined a divergence measure in Section SECREF6 to quantify it.

Conclusion

Envisioning NMT models as a token classifier with an autoregressor helped in analysing the weaknesses of each component independently. The class imbalance was found to cause bias in the token classifier. We showed that BPE vocabulary size is not arbitrary, and it can be tuned to address the class imbalance and sequence lengths appropriately. Our analysis provided an explanation why BPE encoding is more effective compared to word and character models for sequence generation.

Even though BPE encoding indirectly reduces the class imbalance compared to words and characters, it does not completely eliminate it. The class distributions after applying BPE contain sufficient imbalance for biasing the classes, and affecting the recall of rare classes. Hence more work is needed in directly addressing 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.