Advanced Techniques of Machine Translation Assignment 3

Tiancheng Hu, Yuchen Jiang
{tianhu, yucjiang}@student.ethz.ch
Olat ID: tianhu, yucjia

November 16, 2021

Our github repo: https://github.com/pitehu/atmt

1 Experimental Setup

We decide to use BPE [1] and online BPE dropout [2] as the two strategies to experiment with. BPE lies somewhere in between character level and word level encoding. It enables open-vocabulary machine translation that enables operations learned on the training set to be applicable to never-seen-before words on the testing set. Besides, it compresses the most frequent string sequences such that one may save memory and compute on vocabulary and embedding. It is also beautiful as something you get "for free" - it does not involve any architecture change or changes in the training procedure or additional training data. BPE can be added on top of other strategies. It has been shown to improve performance on various MT settings and is widely adopted. It only adds one hyperparameter, which we study in detail. BPE dropout is also something we get "for free". It addresses some of the issues of BPE - the exact same word could have drastically different encoding based on the context and related words may never share any representation at all. In a sense, BPE dropout is like data augmentation, as we randomly drop merge operations so that hopefully similar words would at least share representations in some version of the training data. It is conceptually simple and elegant and has been shown to be very effective. It also only adds one parameter and we look into the effect of different values of the merging probability.

We implement BPE using this repository provided by Prof. Senrich's[1]. We change the data preprocessing pipeline. Specifically, before binarizing the data and building a dictionary, we add the BPE step. We cap the dictionary size to the default value of 4,000 in all our experiments and vary the number of merge operations as well as the BPE dropout probability. We use two separate vocabularies for English and French, not a joint vocabulary as recommended by [1], as we want to compare with word-level models with unshared vocabularies. We only use BPE dropout during training but not testing. We use the default training parameters provided in the starting repo for all our experiments. We use the default evaluation setup as well. To ensure a fair comparison, we run the baseline model on our machine, instead of using the checkpoint provided. For models with BPE, during evaluation, we first combine the BPE-encoded prediction back into normal prediction with the help of Prof. Senrich's repository.

[1] https://github.com/rsennrich/subword-nmt

2 Results

All the parameter combinations we try and the corresponding results in terms of BLEU score and N-Gram Precisions can be found in Table 1. When we change the number of BPE merges, we do not consider BPE dropout at all. We then fix the number of BPE merges to be 1500 and vary the BPE Dropout Probability.

Additionally, we show the resulting BLEU score as a function of the number of BPE merges and BPE Dropout Probability in Figure 1.

We find that using BPE does improve performance, but only when a suitable number of merges is used. We can then get a 3 BLEU improvement. The BLEU score first goes up, reaches the highest point then goes down with the number of BPE merges n increasing. We hypothesize that n controls the vocabulary size and hence change the model from something similar to a character-level model (when n is small) into a word-level model (when n is large). The number of merges can roughly control the resulting vocabulary size. Notice that, however, to ensure a fair comparison of different number of merges, we cap the vocabulary size to 4000. This may explain the worse performance when n is large, as we may map too many words into UNK due to vocabulary size. Another factor to consider is the small dataset size, with only 10000 sentences. Because of this fact, using a large vocabulary size may cause underfitting.

The BLEU score also goes up and then goes straight down as the BPE dropout probability p increases. Notice that using a very small p seems to actually hurt the performance compared to not using BPE dropout at all. This is somewhat different from what we expect: we expect that the performance with any BPE dropout with p below a certain value would outperform the model without BPE dropout. One possible explanation is the following: since we use the same default training parameters for all of our experiments, the early stopping parameter is set at 3 epochs. It could cause some model training to stop too early. And because we are only able to run all experiments once, we cannot say for sure. As p increases further, the performance drops precipitously. This may be due to too much randomness being injected into the training data, so much so that the model is unable to recognize the underlying patterns. If we have a larger training set, this could be less of an issue.

We also look at some of the translation results qualitatively. Overall, even the best system has many clearly wrong translations or sentences that is grammatical but do not make any sense. All systems seem to suffer from the problem of repeating. An example is “I never figured a word, but I’d be disappointed for a sentence in a sentence of the sentence of a sentence in the sentence of the sentence.” We do think the BPE systems have relatively fewer repetitions.

We do think the systems with higher BLEU perform better in manual inspection. For BPE systems, we can clearly see the artifacts of BPE - there is a number of word-looking gibberish in the translation here and there. We also look at the translation of named entities. One ground truth English sentence is “Linda will be here.” The baseline translation is “The company is be here.” We believe this is because of the small vocabulary size and training data size, the word name Linda is not in the vocabulary. When n is small with no BPE dropout, the BPE models translate Linda into “Lindon”, “Linese”, “Linating” and “Lindindent”. We can clearly see the artifacts of BPE. “Lin” seems to be a subword unit that is kept intact in all these cases. When n gets larger, the BPE models suffer the same issue as the word level model. Interestingly, when we use BPE dropout, we see “Linda” in some cases being translated into “Layla”, suggesting that the BPE dropout does cause the merging to be different.

3 Lessons

In this assignment, we tried out BPE and BPE dropout in order to improve our MT system. We are able to improve the system performance by nearly 7 BLEU, which is rather large. This tells us that even in a setting with a limited amount of data, we could still squeeze out performance by various tricks. In our work, we only considered BPE and BPE dropout but image if we also use the autoencoding strategy as well as backtranslation and carefully tune the hyperparameters of the model and the training process.

At the begining, we made a small yet destructive mistake in our BPE implementation which caused our first few training runs to be futile. It took us quite a long time to pinpoint where the mistake was. In the future, we will make sure every component of a complex model to be correct before we actually start training. Also, we will try to start earlier.

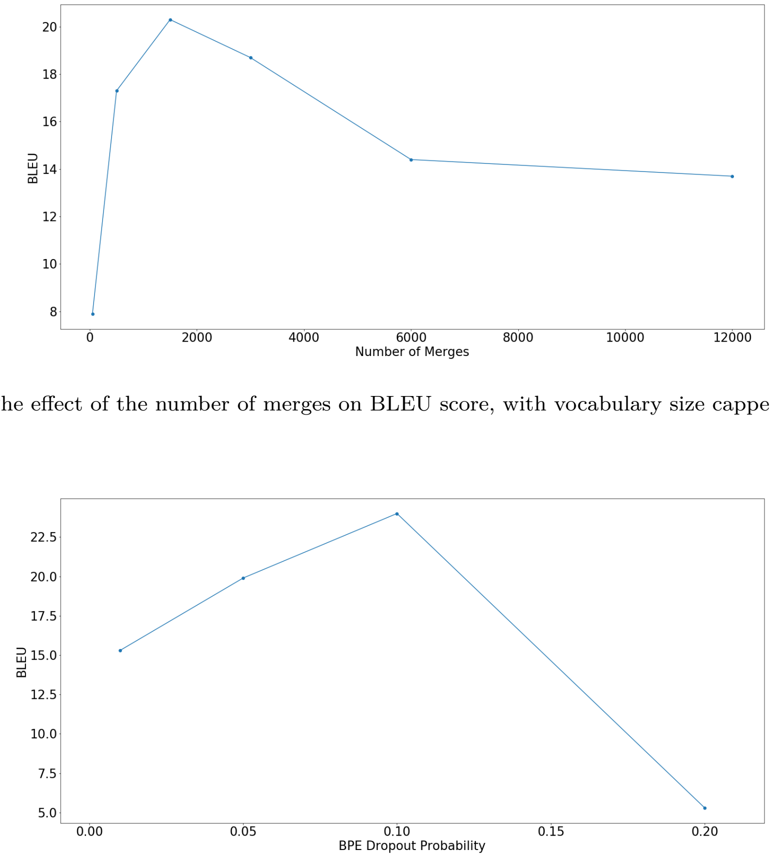
# BPE Merge	BPE Dropout	BLEU	N-gram Precision			
Baseline		17.3	46.8	23.1	12.5	6.7
50	0	7.9	33.3	11.1	4.9	2.2
500	0	17.3	46.3	22.7	12.4	6.8
1500	0	**20.3**	**48.4**	**25.4**	**15.2**	**9.0**
3000	0	18.7	47.5	24.1	13.8	7.8
6000	0	14.4	41.7	19.5	10.1	5.2
12000	0	13.7	41.0	19.0	9.6	4.7
1500	0	20.3	48.4	25.4	15.2	9.0
1500	0.01	15.3	42.8	20.5	10.9	5.8
1500	0.05	19.9	48.8	24.9	14.7	8.8
1500	0.10	**24.0**	**55.7**	**29.5**	**18.3**	**11.1**
1500	0.20	5.3	30.2	8.5	3.2	1.0

Table 1: The BLEU Scores and N-Gram Precisions of our MT system with different BPE and BPE dropout parameters

References

[1] R. Sennrich, B. Haddow, and A. Birch, “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)*, (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, Aug. 2016.

[2] I. Provilkov, D. Emelianenko, and E. Voita, “BPE-dropout: Simple and effective subword regularization,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 1882–1892, Association for Computational Linguistics, July 2020.



(a) The effect of the number of merges on BLEU score, with vocabulary size capped at 4000

(b) The effect of BPE Dropout Probability on BLEU score, with Number of Merges fixed at 1500 and vocabulary size capped at 4000

Figure 1: Effect of Number of Merges and BPE Dropout Probability on BLEU score