Meaningless yet Meaningful: Morphology Grounded Subword-level NMT

Tamali Banerjee
IIT Bombay
Mumbai, India
tamali@cse.iitb.ac.in

Pushpak Bhattacharya
IIT Bombay
Mumbai, India
pb@cse.iitb.ac.in

Abstract

We explore the use of two independent subsystems, namely Byte Pair Encoding (BPE) and Morfessor as basic units for subword-level neural machine translation (NMT). We have shown that for linguistically distant language-pairs Morfessor-based segmentation algorithm produces significantly better quality translation than BPE. However, for close language-pairs BPE-based subword-NMT may translate better than Morfessor-based subword-NMT. We have proposed a combined approach of these two segmentation algorithms Morfessor-BPE (M-BPE) which outperforms these two baseline systems in terms of BLEU score. Our results are supported by experiments on three language-pairs: English-Hindi, Bengali-Hindi and English-Bengali.

1 Introduction

Subword-level NMT is an NMT approach that can tackle OOV problem. In order to train an NMT (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015) model for a language-pair, the size of vocabularies for source and target languages should be constant. But in reality, the vocabulary of a natural language is open. Some words in test data may be absent in system vocabulary. NMT model cannot interpret the semantics of these OOV words. So, translation quality deteriorates as the number of unseen (rare) words increases (Sutskever et al., 2014).

OOV words are mainly of three types described in Table 1. The first type of OOV words needs transliteration. But for translating the second type of OOV words, we need to look deeper. A word based NMT system treats '*house*' and '*houses*' as two completely different words, which limits the coverage of vocabulary. Morphological analyzers tackle this problem by segmenting '*houses*' as '*house*' and '*s*'. This way it can cover many words and their inflections too. The third type of OOV words are dealt by leveraging **lexical similarity** between language-pairs. Lexically similar languages share many words (cognates, loan words) with similar spelling, pronunciation and meaning. Subword-level approaches are effective ways for translation of such shared words.

Type	Example
Named entities	'दिल्ली' *Delhi*
Compound words and inflected words	'*moonlight*', '*examined*'
Rare words in reality	'*serendipity*'

Table 1: Types of OOV words with example.

A character n-gram of a word is called a **subword**. It may or may not be meaningful. On the other hand, a **morpheme** is the smallest grammatical meaningful unit of a language. If we segment *'houses'* as *'hou'*+*'ses'*, then *'hou'* and *'ses'* will be meaningless subwords. But if we segment *'houses'* as *'house'*+*'s'*, then *'house'* and *'s'* will be subwords as well as morphemes.

2 Related work

A word can be segmented as BPE (Sennrich et al., 2016), orthographic syllable (Kunchukuttan and Bhattacharyya, 2016), character (Chung et al., 2016; Costa-jussà and Fonollosa, 2016), Huffman encoding (Chitnis and DeNero, 2015). In our experiment we show that, for translation between linguistically close language-pair BPE subword segmentation is suitable, whereas for transla-

Proceedings of the Second Workshop on Subword/Character LEvel Models, pages 55–60
New Orleans, Louisiana, June 6, 2018. ©2018 Association for Computational Linguistics

tion between linguistically distant language-pair Morfessor-based segmentation is suitable. Our proposed subword segmentation approach utilizes benefit of both BPE and Morfessor (Creutz and Lagus, 2006; Smit et al., 2014; Grönroos et al., 2014) and performs well for both linguistically close and distant language-pairs.

3 BPE algorithm

BPE (Gage, 1994) is originally a data compression technique. The main idea behind BPE is- *"Find the most frequent pair of consecutive two character codes in the text, and then substitute an unused code for the occurrences of the pair."* (Shibata et al., 1999)

3.1 BPE as subword unit

BPE works as subword segmentation method for both NMT (Sennrich et al., 2016) and SMT (Kunchukuttan and Bhattacharyya, 2017). In this method, two vocabularies are used: **training vocabulary** and **symbol vocabulary**. Words in training vocabulary are character-sequences followed by an end-of-word symbol. At first, all characters are added to symbol vocabulary. This step is followed by adding the most frequent symbol bigram to the vocabulary, and all its occurrences are replaced by a new symbol (merged symbol bigram). This step is repeated for a number of times, which is a hyperparameter.

Starting from character level as the number of merge operations is increased, primarily frequent character-sequences and then full words are also added as a single symbol. So, the number of merge operations balances between the NMT model vocabulary size and the length of training sentences. Symbol '@@' is used here to indicate the places of segmentations.

3.2 Hyperparameter selection of BPE

Higher number of merge operations adds almost all words to symbol vocabulary. It will prevent the NMT system to translate on subword level segmentation of words.

Using BPE subword segmentation, the average length of sentences is increased as words are broken into subwords. Larger the sentence size, more difficult it becomes for NMT to learn well from them (Bahdanau et al., 2015). So, proper tuning of this hyperparameter is needed. Higher number of merge operations makes the elements more word-like. Lower number of merge operations makes the elements more character-like, where sometimes character-to-character mappings add transliterated words in the translation output.

3.3 Comparison of BPE segmentation with Morfessor

The goal of morphological analyzers such as Morfessor is to segment a word into its **morphs**, the surface forms of morphemes. Comparison between BPE subword segmentation and Morfessor is described below.

- BPE is a greedy approach. Morfessor takes highest probable segmentation of words and deals with local optima by removing and adding word tokens. So, Morfessor produces more acceptable morphological segmentation than BPE.

- Main advantage of BPE is solving OOV problem in two ways: i) some segmentations are almost morphological segmentation, and ii) some segmentations are nearly character-level segmentations. As a result, OOV words are either transliterated or produce partially correct translations. But in absence of some morphs in the dictionary, Morfessor does not produce character-level segmentations. In such cases, it faces OOV problem.

Our Morfessor-based segmentation algorithm takes all the valid words from the corpora and passes these through morfessor. After getting their morphological segmentation, we replace them in data at their respective places. Like BPE, '@@' is used here to indicate the places of segmentation. That means while decoding from subwords we need to join subwords having '@@' signs with next subword.

4 Our approach

The idea behind our proposed combined approach M-BPE comes from comparing BPE and Morfessor. The hypothesis of this approach is- *"Words should be segmented into real morphs. After that, segmentation of*

morphs into subwords may be beneficial to handle open vocabulary." Words can be morphologically segmented by using Morfessor. BPE will be helpful for OOV morphs of type 1 and 3 described in section 1. Work-flow of this approach is described below.

Step 1: Use Morfessor on the set of all words from the dataset.

Step 2: Find and replace all occurrences of these words with their segmented form (symbol '**' is used to keep information of segmenting positions). For example- '*googling*' will be segmented into two morphs '*googl***' and '*ing*'.

Step 3: Learn and apply BPE on that morph-segmented data. Use symbol '@@' for these segmentations. For example, this may segment the word '*googl***' as '*go@@*', '*og@@*' and '*l***', if '*googl***' is rarely occurring word in data. It will not merge already segmented subwords followed by symbol'**', because it'll treat already segmented subwords as different elements.

Step 4: Replace symbol '**' with the symbol '@@'. Finally, the word '*googling*' will become '*go@@ og@@ l@@ ing@@*'.

4.1 Hyperparameter selection of M-BPE

With increasing average number of elements per sentence, performance of an NMT model degrades (Bahdanau et al., 2015). Using the same number of merge operations for both BPE and M-BPE produces a higher number of elements per sentence in case of M-BPE than BPE because the Morfessor part of M-BPE increases the number of elements of a sentence before applying the BPE part on it. In order to get a fair comparison between BPE and M-BPE, we have adjusted their hyperparameter in such a way that average numbers of elements per sentence after segmentation become almost same. So, for maintaining that criterion, here we have kept the number of merge operations of M-BPE higher than that of BPE.

5 Experimental setup

There are three systems of subword segmentation in our experiment, namely- BPE, Morfessor and M-BPE. We have used subword-nmt[1] for BPE segmentation, Flatcat (Grönroos et al., 2014) and NLP Indic Library[2] for producing morphological segmentation of English and Indian words.

5.1 Datasets

We have used data from English-Hindi (En-Hi), English-Bengali (En-Bn) and Bengali-Hindi (Bn-Hi) language-pairs from health and tourism domain multilingual parallel Indian Language Corpora Intitiative (ILCI) corpus (Jha, 2010). We clean and tokenize the training corpus. English data was tokenized using the Stanford tokenizer (Klein and Manning, 2003) and then true-cased using *truecase.perl* provided in MOSES toolkit[3]. For Hindi and Bengali data, we tokenized using NLP Indic Library (Kunchukuttan et al., 2014). Then parallel sentences were divided into three parts for training, testing and tuning/validation. For each language-pair, we have 44,777 sentence-pairs in training data, 1,000 sentence-pairs in tuning data and 2,000 sentence-pairs in test data.

5.2 System details

After tokenization, words of source sentences are broken into subwords using a segmentation algorithm. NMT system receives a sequence subwords of a sentence as input and produces the output of a subword-sequence in target language. Then, subwords are combined to produce words in order to get an actual sentence in target language. We have used NEMATUS (Sennrich et al., 2017) as an attention-based encoder-decoder NMT system in our experiment.

6 Results and discussion

The example given below shows the difference among three segmentations:

Example:

Word level: focusing your mind

BPE level: foc@@ us@@ sing your mind

Morfessor level: focus@@ ing your mind

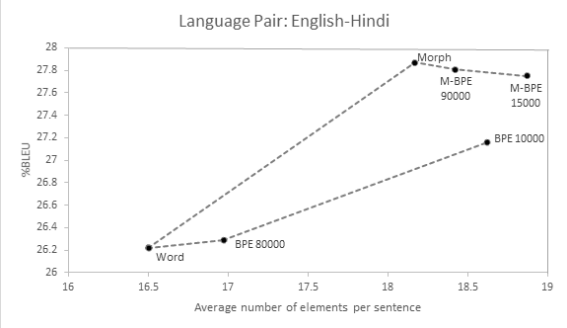
M-BPE level: foc@@ us@@ ing your mind

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

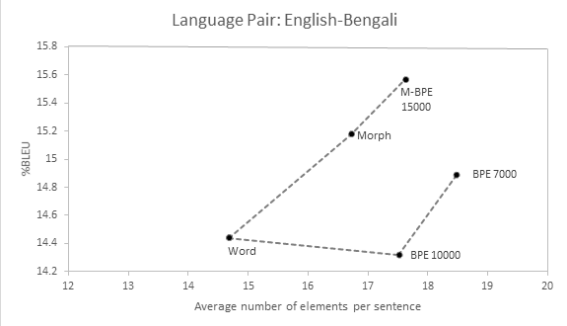
[2]anoopkunchukuttan.github.io/indic_nlp_library/

[3]http://www.statmt.org/moses/

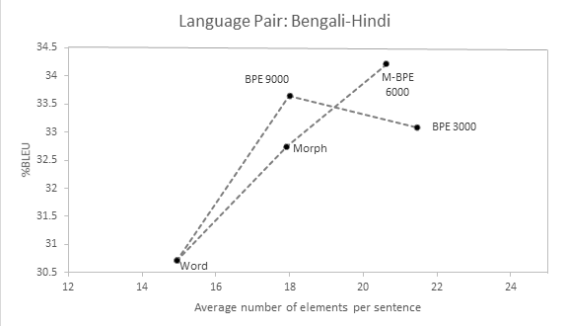
Fig 1 shows changes in BLEU scores (Papineni et al., 2002) when we train NMT models using sentences with increasing average number of elements (by tuning hyperparameters). Here, two paths indicate two different approaches of segmentation: i) from word level to BPE level, ii) from word level to M-BPE level via Morfessor level.



(a) Pair: English-Hindi



(b) Pair: English-Bengali



(c) Pair: Bengali-Hindi

Figure 1: Translation accuracies of NMT systems for various translation units (BLEU scores reported).

Table 2 compares among word-level, Morfessor-level, BPE-level and M-BPE level NMT output accuracies for three language-pairs. Tuned numbers of merge operations of BPE and M-BPE, for Bn-Hi, are 3k and 6k. In case of En-Hi, these are 10k and 90k respectively, and for En-Bn these are 7k and 15k respectively. Translation between lexically close language-pairs like Bn-Hi has more character-to-character mappings than En-Hi. For that reason, Bn-Hi language-pair needs a lower value of hyperparameter than English-Hindi.

Pair	W	M	BPE	M-BPE
Bn-Hi	30.71	32.74	33.09	**34.21**
En-Hi	26.22	**27.87**	27.16	27.81
En-Bn	14.44	15.18	14.89	**15.57**

Table 2: Translation accuracies for various translation units (BLEU scores reported). The reported scores are:- W: word-level, M: Morfessor-level, BPE: BPE-level, M-BPE: M-BPE-level. The values marked in bold indicate the best score for a language pair.

Some findings from the results are listed below.

- For En-Hi and En-Bn language-pairs, Morfessor produces better quality translation than BPE.
- For Bn-Hi language-pair, BPE is capable of producing better translation than Mofessor as segmentation algorithm.
- M-BPE can maintain translation quality for all language-pairs.

In case of Bn-Hi language-pair, BPE helps in improvement of baseline (word-level) translation quality. But in case of En-Hi and En-Bn, it fails to show a considerable amount of improvement. En-Hi and En-Bn language-pairs are quite different from each other in terms of syntactical (word-order, morphology) and lexical similarities. Bengali and Hindi are much closer to each other in comparison with En-Hi and En-Bn. This property of Bn-Hi language-pair helps their translation model to figure out mappings between source and target n-grams. Grammatical rules of languages may not be revealed due to morphologically wrong segmentations. But it hardly affects Bn-Hi translation quality because of their syntactic similarities. Moreover, small subwords add transliterated words in output which is favorable for Bn-Hi translation.

In case of En-Hi and En-Bn, translation models do not easily find mappings between

source and target random subwords. It will be useful, only if these subwords are real morphs. For these language-pairs, correct segmentation of word is necessary, not only for getting an accurate translation of word, but also for understanding its grammar (word order and function words). En-Hi and En-Bn language-pairs do not have much lexical similarity; small meaningless subwords do not help in that case; these can even degrade the translation quality.

M-BPE can segment words correctly. It can also produce small subwords by further segmenting morphs. So, by tuning its hyper-parameter, we can make it suitable for all language pairs, i.e. linguistically close and linguistically distant language-pairs.

7 Conclusion and future work

As a subword segmentation algorithm, M-BPE outperforms baseline BPE in case of both lexically close and distant language-pairs. However, when compared with baseline Morfessor, improvement due to M-BPE depends on lexical closeness. For lexically close language-pair the improvement is significant. In that case, meaningless BPE subwords play a meaningful role in improving translation quality. Future investigation will be focused on the automatic tuning of hyperparameter for M-BPE.

Acknowledgments

We would like to thank Anoop Kunchukuttan, Kevin Patel and Ajay Anand Verma for their valuable inputs during discussion. We also thank the reviewers for their feedback.

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *In Proceedings of the International Conference on Learning Representations (ICLR)*.

Rohan Chitnis and John DeNero. 2015. Variable-length word encodings for neural translation models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 2088–2093.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1693–1703.

Marta R Costa-jussà and José AR Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 357–361.

Mathias Creutz and Krista Lagus. 2006. Morfessor in the morpho challenge. In *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*. Citeseer, pages 12–17.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.* 12(2):23–38. http://dl.acm.org/citation.cfm?id=177910.177914.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, Mikko Kurimo, et al. 2014. Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *COLING*. pages 1177–1185.

Girish Nath Jha. 2010. The tdil program and the indian langauge corpora intitiative (ilci). In *LREC*.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 423–430.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2016. Orthographic syllable as basic unit for smt between related languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1912–1917.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2017. Learning variable length units for smt between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. pages 14–24.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014. Sata-anuvadak: Tackling multiway translation of indian languages. *pan* 841(54,570):4–135.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 311–318.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain, pages 65–68. http://aclweb.org/anthology/E17-3017.

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)*. volume 1, pages 1715–1725.

Yusuke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. Byte pair encoding: A text compression scheme that accelerates pattern matching.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, Mikko Kurimo, et al. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April 26-30, 2014*. Aalto University.

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.