Pinyin as Subword Unit for Chinese-Sourced Neural Machine Translation

Jinhua Du[†‡], Andy Way[†]

[†]ADAPT Centre, School of Computing, Dublin City University, Ireland
[‡]Accenture Labs, Dublin, Ireland
{jinhua.du, andy.way}@adaptcentre.ie

Abstract. Unknown word (UNK) or open vocabulary is a challenging problem for neural machine translation (NMT). For alphabetic languages such as English, German and French, transforming a word into subwords is an effective way to alleviate the UNK problem, such as the Byte Pair encoding (BPE) algorithm. However, for the stroke-based languages, such as Chinese, aforementioned method is not effective enough for translation quality. In this paper, we propose to utilize Pinyin, a romanization system for Chinese characters, to convert Chinese characters to subword units to alleviate the UNK problem. We first investigate that how Pinyin and its four diacritics denoting tones affect translation performance of NMT systems, and then propose different strategies to utilise Pinyin and tones as input factors for Chinese–English NMT. Extensive experiments conducted on Chinese–English translation demonstrate that the proposed methods can remarkably improve the translation quality, and can effectively alleviate the UNK problem for Chinese-sourced translation.

1 Introduction

In recent years, NMT has made impressive progress [8, 3, 20, 1, 4]. The state-of-the-art NMT model employs an encoder–decoder architecture with an attention mechanism, in which the encoder summarizes the source sentence into a vector representation, and the decoder produces the target string word by word from vector representations, and the attention mechanism learns the soft alignment of a target word against source words [1]. NMT systems have outperformed the state-of-the-art SMT model on various language pairs in terms of translation quality [13, 2, 7, 22, 21, 5].

The translation of rare words is not only an open problem for statistical machine translation (SMT), but also for NMT. Current NMT systems take a fixed vocabulary for the input and output sequences, and the rare words in the data are denoted as a symbol "UNK", which will make the translation inaccurate and disfluent to some extent. The vocabulary of neural models is typically limited to 30,000–50,000 words, but translation is an open-vocabulary problem, especially for languages with productive word formation processes, such as agglutination and compounding. In these cases, translation models require mechanisms that go below the word level [19].

Recent work has been done in improving the generalisation capability of NMT for open vocabulary [14, 6, 11, 12, 19, 10]. For example, translation of the out-of-vocabulary (OOV) words can be regarded as a post-processing step as in SMT, i.e. keeping the

OOVs in the hypothesis, and then using a bilingual dictionary to obtain translations of these OOVs. The deficiency of this back-off dictionary method is that it needs extra knowledge or resources to alleviate the OOV problem. However, for some low-resource languages or domains, it it not feasible.

Byte pair encoding is an effective way to segment a word into subwords for alphabetic languages such as English, German and French, and it does not rely on external resources [19]. However, it is not that straightforward for the stroke-based languages, such as Chinese. For Chinese-sourced NMT systems, word is often used as the basic unit in the input sequence. However, word-level unit for a large-scale data set, compared to the character-level and subword-level units, will bring a data sparsity problem in terms of rare words, i.e. many name entities, date, time and numbers occur infrequently, resulting in a very huge vocabulary. Therefore, in NMT these infrequent words are, accordingly, represented as an "UNK" token. Intuitively, if we can transform Chinese characters into alphabetic compositions, then we can easily employ the BPE algorithm to convert Chinese words into subwords and may alleviate the rare words problem.

Chinese Pinyin, literally means "spelled sounds", is the official romanization system for Standard Chinese in mainland China, Malaysia, Singapore and Taiwan.[1] The system includes four diacritics denoting tones. Pinyin without tone marks is used to spell Chinese names and words in languages written with the Latin alphabet, and also in certain computer input methods to enter Chinese characters.

An example of Chinese characters with their corresponding Pinyin and English translations is shown below.[2]

Pinyin:	mā	má	mǎ	mà	ma
Character:	妈	麻	马	骂	吗
Tone:	First	Second	Third	Fourth	Neural
English:	mother	hemp	horse	scold	question particle

In this example, we can see that in the row of "Pinyin", the letters are same, but the tones are different, which indicates that the pronunciation for each Chinese character in the row of "Character" is different. Tones are essential for correct pronunciation of Mandarin syllables.

Normally, the tone is placed over the letter that represents the syllable nucleus except the "Neural" tone. Explanations for tones are:

- The first tone (Flat or High Level Tone) is represented by a macron (¯) added to the pinyin vowel;
- The second tone (Rising or High-Rising Tone) is denoted by an acute accent (´);
- The third tone (Falling-Rising or Low Tone) is marked by a caron (ˇ);
- The fourth tone (Falling or High-Falling Tone) is represented by a grave accent (`);
- The fifth tone (Neutral Tone) is represented by a normal vowel without any accent mark.

[1] https://en.wikipedia.org/wiki/Pinyin

[2] The source of the example is: https://en.wikipedia.org/wiki/Pinyin

From this example we can see that Chinese characters and words can be converted into alphabetic forms using Pinyin with or without tones, so the BPE algorithm can be applied like alphabetic languages. In this paper, we explore different ways to utilize Pinyin as the subword unit converter for Chinese-sourced NMT, namely character-level Pinyin without tones (ChPy), character-level Pinyin with tones (ChPyT), word-level Pinyin without tones (WdPy), and word-level Pinyin with tones (WdPyT). Furthermore, we propose to use Pinyin as an input factor for a standard word-level NMT system (fac. NMT), and use tones as the input factor for the "WdPy" NMT system (fac. WdPy), respectively. Extensive experiments conducted on Chinese→English NIST translation task show that 1) using Pinyin to replace Chinese characters/words can significantly reduce the vocabulary size, resulting in a significant decrease of UNK symbols in translations; 2) WdPyT and factor-based Pinyin NMT systems can significantly improve translation quality compared to the standard word-level NMT system.

The main contributions of this work include:

- We extensively investigate different use of Pinyin as subword units for Chinese-sourced NMT systems.
- We propose to integrate Pinyin or tones as input factors to augment NMT systems.
- We provide a qualitative analysis on the translation results.

The rest of the paper is organised as follows. In Section 2, related work to the open vocabulary problem is introduced. Section 3 describes the attentional encoder–decoder framework for NMT, and introduces the factored NMT. In Section 4, we detail the proposed different Pinyin-based NMT frameworks. In Section 5, we report the experimental results on Chinese→English NIST task. Section 6 concludes and gives avenues for future work.

2 Related Work

The work on the open vocabulary problem for NMT can be roughly categorised into three categories:

- UNK in post-processing: This is a traditional way that is usually used in SMT to handle OOVs in the translation, e.g. using a back-off dictionary to translate OOVs. Different from SMT, NMT does not have a hard alignment between the source and target words, so the UNKs in the translation are not strictly aligned to those in the source sequence.
- UNK in pre-processing: in this scenario, the unknown words in the source-side input are substituted by semantically similar words or paraphrases. However, it is not guaranteed that a proper substitution can be acquired from the limited in-vocabulary words. This method is not only applicable for alphabetic languages, but also for stroke-based languages. Splitting words into subwords is another effective way to pre-process source-side sentences, such as the BPE.
- UNK in decoding: in this scenario, the UNKs are dynamically processed during decoding. For example, a word-character combined model can be used to recover target UNK by a character-based model if the input word is an OOV. Another methodology is to manipulate a large-scale target vocabulary by selecting a subset to speed up the decoding and alleviate the UNK problem.

Regarding the first category, Luong et al. propose a back-off dictionary method to handle OOVs in the translation [14]. They first train an NMT system augmented by the output of a word alignment algorithm, allowing the NMT system to emit, for each OOV word in the target sentence, the position of its corresponding word in the source sentence. Then a post-processing step is used to translate every OOV word using a dictionary. Their experiments on the WMT'14 English→French translation task show a substantial improvement of up to 2.8 BLEU points over a standard NMT system.

In terms of the second category, Li et al. propose a substitution-translation-restoration method [11]. The rare words in a testing sentence are replaced with similar in-vocabulary words based on a similarity model learnt from monolingual data in the substitution step. In translation and restoration steps, the sentence will be translated with a model trained on new bilingual data with rare words replaced, and finally the translations of the replaced words will be substituted by those of original ones. Experiments on Chinese-to-English translation demonstrate that the proposed method can significantly outperform the standard attentional NMT system.

Sennrich et al. propose a variant of byte pair encoding for word segmentation in the source sentences, which is capable of encoding open vocabularies with a compact symbol vocabulary of variable-length subword units [19]. This method is simpler, and more effective than using a back-off translation model. Experiments on the WMT'15 translation tasks English→German and English→Russian show that the BPE-based subword models significantly outperform the back-off dictionary baseline.

With respect to the third category, extensive work has been done on using very large target vocabulary for NMT [6, 15, 10]. The basic idea is to select a subset from a large-scale target vocabulary to produce a target word during the decoding process. Their experiments on different language pairs show that the proposed methods can not only speed up the translation, but also alleviate the UNK problem. Luong and Manning propose a word-character solution to achieving open vocabulary NMT [12]. A hybrid system is built to translate mostly at the word level and consult the character components for rare words, i.e. a character-based model will be used to recover the target UNK if the input word is an OOV. On the WMT'15 English→Czech translation task, the proposed hybrid approach outperforms systems that already handle unknown words.

3 Neural Machine Translation

3.1 Attentional NMT

The basic principle of an NMT system is that it can map a source-side sentence $\mathbf{x} = (x_1, \ldots, x_m)$ to a target sentence $\mathbf{y} = (y_1, \ldots, y_n)$ in a continuous vector space, where all sentences are assumed to terminate with a special "end-of-sentence" token $< eos >$. Conceptually, an NMT system employs neural networks to solve the conditional distributions in (1):

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y_i|y_{<i}, x_{\leq m}) \tag{1}$$

We utilise the NMT architecture in [1], which is implemented as an attentional encoder-decoder network with recurrent neural networks (RNN).

In this framework, the encoder is a bidirectional neural network [20] with gated recurrent units [3] where a source-side sequence $\mathbf{x}$ is converted to a one-hot vector and fed in as the input, and then a forward sequence of hidden states $(\overrightarrow{h}_1, \ldots, \overrightarrow{h}_m)$ and a backward sequence of hidden states $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_m)$ are calculated and concatenated to form the annotation vector h_j. The decoder is also an RNN that predicts a target sequence $\mathbf{y}$ word by word where each word y_i is generated conditioned on the decoder hidden state s_i, the previous target word $y_{i`1}$, and the source-side context vector c_i, as in (2):

$$p(y_i|y_{<i}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \tag{2}$$

where g is the activation function that outputs the probability of y_i, and c_i is calculated as a weighted sum of the annotations h_j. The weight α_{ij} is computed as in (3):

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{ik})} \tag{3}$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

is an alignment model which models the probability that the inputs around position j are aligned to the output at position i. The alignment model is a single-layer feedforward neural network that is learned jointly through backpropagation.

3.2 Factored NMT

Factored NMT, introduced in [18], represents the encoder input as a combination of features as in (4):

$$\overrightarrow{h}_j = g(\overrightarrow{W}(\Big\|_{k=1}^{|F|} E_k x_{jk}) + \overrightarrow{U}\overrightarrow{h}_{j-1}) \tag{4}$$

where $\|$ is the vector concatenation, $E_k \in \mathbb{R}^{m_k \times K_k}$ are the feature-embedding matrices, with $\sum_{k=1}^{|F|} m_k = m$, and K_k is the vocabulary size of the k_{th} feature, and $|F|$ is the number of features in the feature set F [18].

In factored NMT, the features can be any form of knowledge which might be useful to NMT systems, such as POS tags, lemmas, morphological features and dependency labels as used in [18]. In our work, we use Pinyin or tones as the input factor to augment NMT (c.f. Section 4).

4 Pinyin for Chinese-Sourced Subword NMT

Chinese Pinyin, as a romanization system for Standard Chinese, is often used for teaching purpose or computer input means. Four tones, namely the first, second, third and

fourth tones with a neural tone are used to distinguish different characters/words with different pronunciations.

By transforming Chinese characters to Pinyin forms, the BPE method to encode words into subwords can be directly applied. In order to investigate what a role of tones play for Pinyin-based NMT systems, we set up four different configurations, namely:

- ChPy: the NMT system takes the character-level Pinyin without tones as input. The character-level NMT indicates that we first segment a Chinese sentence into a character sequence, and then convert each Character into its Pinyin form without the tone.
- ChPyT: the NMT system takes the character-level Pinyin with tones as input. In some sense, the tone is helpful to disambiguate the character in a context.
- WdPy: the NMT system takes the word-level Pinyin without tones as input. The word-level NMT indicates that we first segment a Chinese sentence into a word sequence, and then convert each word into its Pinyin form without the tone.
- WdPyT: the NMT system takes the word-level Pinyin with tones as input.

An example is shown below in terms of these four settings.

Chinese:	许多球队以纪律和组织战来降低风险
English:	many teams try to reduce risks through discipline and organization
Characters:	许 多 球 队 以 纪 律 和 组 织 战 来 降 低 风 险
ChPy:	xu duo qiu dui yi ji lv he zu zhi zhan lai jiang di feng xian
ChPyT:	xǔ duō qiú duì yǐ jì l ǜ hé zǔ zhī zhàn lái jiàng dī fēng xiǎn
Words:	许多 球队 以 纪律 和 组织战 来 降低 风险
WdPy:	xuduo qiudui yi jilv he zuzhizhan lai jiangdi fengxian
WdPy(BPE):	xuduo qiudui yi jilv he zu@@ zhizhan lai jiangdi fengxian
WdPyT:	xǔduō qiúduì yǐ jìl ǜ hé zǔzhīzhàn lái jiàngdī fēngxiǎn
WdPyT(BPE)	xǔduō qiúduì yǐ jìl ǜ hé zǔzhī@@ zhàn lái jiàngdī fēngxiǎn

In this example, we can see that:

- BPE can be easily applied to either character-level or word-level Pinyin sequence, either with or without tones, which can reduce the OOVs in the data.
- The BPE algorithm encodes an infrequent word "zuzhizhan" ("组织战" in Chinese and "organization" in English) in WdPy to subwords "zu@@" and "zhizhan" in WdPy(BPE), which represent "组" and "织战", respectively. However, this segmentation is not correct in terms of semantic meaning because "织战" is not a meaningful subword. We infer that this is caused by the homophone "zhizhan", i.e. the same Pinyin might correspond to different word forms. For example, "zhizhan" could be Chinese word "之战" ("fight" in English), "只占" ("has only" in English) etc. Therefore, the word Pinyin without tones brings more ambiguities for translation.
- For the WdPyT, we can see that the word "zǔzhīzhàn" ("组织战" in Chinese and "organization" in English) is encoded to subwords "zǔzhī@@" and "zhàn" in WdPyT(BPE). The subword "zǔzhī@@" indicates "organization" in English and "zhàn" represents "fight" in English. This segmentation is meaningful and correct, so the tone is indeed helpful to disambiguate Pinyin forms.

We also utilize Pinyin and tones as input factors for NMT systems, namely 1) word-level Pinyin without tones as the input factor of Chinese words for a standard NMT system; 2) tones as the input factor for the WdPy NMT system.

An example to illustrate these two factored NMT systems is shown below.

Chinese:	在本届世足赛大放异彩
English:	dazzles at the world cup
factored NMT:	在|zai 本|ben 届|jie 世足赛|shizusai 大放异彩|dafangyicai
WdPy(BPE)	zai ben jie shizusai dafangyicai
factored WdPy:	zai|4 ben|3 jie|4 shizusai|4-2-4 dafangyicai|4-4-4-3
factored WdPy(BPE):	zai|O|4 ben|O|3 jie|O|4 shi@@|B|4-2-4 zusai|E|4-2-4 dafang@@|B|4-4-4-3 yicai|E|4-4-4-3

In this example, "factored NMT" (fac. NMT) represents that the encoder takes the Chinese word and its Pinyin without tones as input. In the row of factored NMT, the factor on the left of the vertical bar "|" represents a Chinese word, and the factor on the right of "|" indicates the corresponding Pinyin of the word. We expect that the Pinyin can provide extra information to the word to further improve translation performance. WdPy(BPE) indicates that the BPE algorithm is applied to the word-level Pinyin NMT without tones. "factored WdPy" (fac. WdPy) represents that the tone is integrated into the encoder of WdPy NMT as an input factor. The left of the vertical bar "|" is the Pinyin of a word, and the right is its corresponding tone. "factored WdPy(BPE)" represents that the BPE algorithm is applied to the fac. WdPy. We can see that:

- by applying BPE, we have one more factor set {O, B, E} where "O" indicates a non-subword, "B" indicates the beginning of subwords, and "E" represents the end of subwords.
- the infrequent words "shizusai" and "dafangyicai" are segmented into subwords "shi@@" (literally "world") and "zusai" (literally "football game"), and "dafang@@" (literally "demonstrate") and "yicai" (literally "splendor"), respectively. From English translations of these subwords, we can see that rare words are possible to be translated by the segmentation.
- in WdPy(BPE), the Pinyin word "shizusai" and "dafangyicai" are not segmented, but they are segmented in factored WdPy(BPE), so we infer that tones are helpful to segment rare words into subwords.

5 Experiments

5.1 Experimental Settings

For Chinese→English task[3], we use 1.4M sentence pairs extracted from LDC ZH–EN corpora as the training data, and NIST 2004 current set as the development/validation set that contains 1,597 sentences, and NIST 2005 current set as the test set that contains 1,082 sentences. There are four references for each Chinese sentence.

[3] In the rest of the paper, we use ZH and EN to denote Chinese and English, respectively.

The baseline NMT system takes the Chinese word sequence as input without any Pinyin information, which is also defined as the standard NMT system. We use Nematus [17] as the NMT system, and set minibatches of size 80, a maximum sentence length of 60, word embeddings of size 600, and hidden layers of size 1024. The vocabulary size for input and output is set to 45K. The models are trained with the Adadelta optimizer [23], reshuffling the training corpus between epochs. We validate the model every 10,000 minibatches via BLEU [16] scores on the validation set and save the model every 10,000 iterations.

As in [18], for factored NMT systems, in order to ensure that performance improvements are not simply due to an increase in the number of model parameters, we keep the total size of the embedding layer fixed to 600.

We use a Python tool to convert Chinese characters/words into Pinyin.[4]

All results are reported by case-insensitive BLEU scores and statistical significance is calculate via a bootstrap resampling significance test [9].

5.2 Statistics

The source-side vocabulary sizes of different Pinyin-based NMT systems and factored NMT systems are shown in Table 1.

SYS	baseline	ChPy	ChPyT	WdPy	WdPyT	fac. NMT	fac. WdPy
V-size	185,029	15,872	19,697	97,918	114,067	185,029/144,584	50,589/9,700
Ratio(%)	–	8.6	10.65	52.92	61.65	–/78.14	27.34/5.24

Table 1. Vocabulary sizes of the source-side training data from different NMT systems

In Table 1, all NMT systems except the baseline and factored NMT (fac. NMT) are applied the BPE algorithm. "Ratio(%)" indicates the percentage of the vocabulary size of a Pinyin-based NMT system over that of the baseline.We can see that vocabulary sizes of all Pinyin-based NMT systems, namely the ChPy, ChPyT, WdPy, WdPyT and fac. WdPy are significantly reduced. The reduction of the vocabulary size also indicates the decrease of rare words.

5.3 Experimental Results

Table 2 shows the results of different NMT systems.
From Table 2, we can see that:

- ChPy and ChPyT are significantly worse than the baseline. We analyse that this is 1) due to the significantly longer sequences caused by character-level units; 2) due to the smaller vocabulary that introduces more ambiguities to the Pinyin characters.

[4] https://github.com/mozillazg/python-pinyin

SYS	Validation	Test
baseline	35.49	31.76
ChPy	27.10	21.72
ChPyT	32.30	27.29
WdPy	35.31	31.63
WdPyT	**36.74***	**32.51***
fac. WdPy	**35.92***	**32.14***
fac. NMT	**37.45***	**33.18***

Table 2. Results of different NMT systems "*" indicates translation performance is significantly better.

- WdPy is comparable to the baseline in terms of BLEU. However, the vocabulary size of WdPy is only 52.92% of that of the baseline. The results from ChPy, ChPyT and WdPy give us an inspiration: if we add extra factors to disambiguate the Pinyin, we might further improve the translation quality. Thus, we propose the fac. WdPy to verify this intuition.
- WdPyT significantly improves translation performance by 1.25 (35.49→36.74) BLEU points on the validation set, and 0.75 (31.76→32.51) BLEU points on the test set, respectively, compared to the baseline. However, the vocabulary size of WdPyT is only 61.65% of that of the baseline. The result shows that the word-level Pinyin with tones can not only reduce the vocabulary size or rare words, but also improve system performance.
- fac. WdPy significantly outperforms the baseline by 0.38 (31.76→32.14) BLEU points on the test set, and significantly improves 0.51 (31.63→32.14) BLEU points on the test set compared to WdPy, which shows that tones can provide extra useful information to disambiguate the Pinyin word to further improve translation quality.
- fac. NMT significantly improves 1.96 (35.49→37.45) BLEU points on the validation set, and 1.42 (31.76→33.18) BLEU points on the test set, respectively, compared to the baseline. The results show that Pinyin as an input factor for the standard NMT is indeed helpful.

5.4 Analysis

Beside reporting the BLEU scores, we also examine the influence of Pinyin on the UNK issue in translations. Table 3 shows the change of UNK symbols from different systems.

SYS	baseline	ChPy	ChPyT	WdPy	WdPyT
UNKs	17,597	634,82	13,832	1,128	550
Ratio(%)	–	-360	21.40	93.59	96.87

Table 3. Number of UNK symbols in the translation of different NMT systems

In Table 3, "Ratio" indicates the reduction rate of the number of UNK symbols in a Pinyin-based NMT system over that of the baseline. From Table 3, we can see that:

- ChPy produces more UNK symbols in the translation. The reason is that the serious ambiguity issue caused by the smaller vocabulary size makes the NMT system produce many continuous UNK sequences.
- ChPyT reduces the number of UNKs in translations due to the constraint of tones on character-level Pinyin to disambiguate the units.
- WdPy and WdPyT significantly reduce the UNK symbols in translations.

6 Conclusion

In this paper we propose a subword transformation solution for Chinese-sourced NMT, i.e. use Chinese Pinyin to convert Chinese characters/words into subword units. Subsequently, the BPE algorithm is directly applied to reduce the number of rare words. Furthermore, we propose two factored NMT, one of which uses tones as the input factor for word-level Pinyin NMT, and the other of which integrates word-level Pinyin without tones as input factor to a standard word sequence-based NMT system. We observe from experiments on Chinese→English NIST task that 1) Pinyin as subword unit can indeed significantly reduce rare words. However, it can also introduce more ambiguities. 2) tones can, on the one hand, keep the vocabulary size of a Pinyin-based NMT in a reasonable scale, on the other hand, it can achieve comparable (WdPy) or better (WdPyT) translation performance. 3) using Pinyin or tones as input factors can improve translation quality compared to the baseline which shows that they can provide extra information to disambiguate the input units.

As to future work, we expect more experiments on more effective factors to further improve translation performance, and we will explore the feasibility of Pinyin in the Chinese-targeted NMT systems.

Acknowledgement. We would like to thank the reviewers for their valuable and constructive comments. This research is supported by the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106), and by SFI Industry Fellowship Programme 2016 (Grant 16/IFB/4490).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd International Conference on Learning Representations. pp. 1–15. San Diego, USA (2015)
2. Bentivogli, L., Bisazza, A., Cettolo, M., Federico, M.: Neural versus phrase-based machine translation quality: a case study. In: Proceedings of the EMNLP. pp. 257–267. Austin, Texas, USA (2016)
3. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the EMNLP. pp. 1724–1734. Doha, Qatar (2014)
4. Du, J., Way, A.: Neural pre-translation for hybrid machine translation. In: In Proceedings of MT Summit XVI, vol.1: Research Track. pp. 27–40. Nagoya, Japan (2017)

5. Du, J., Way, A.: Pre-reordering for neural machine translation: Helpful or harmful? The Prague Bulletin of Mathematical Linguistics (108), 171–182 (2017)
6. Jean, S., Kyunghyun Cho, R.M., Bengio, Y.: On using very large target vocabulary for neural machine translation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. pp. 1–10. Beijing, China (2015)
7. Junczys-Dowmunt, M., Dwojak, T., Hoang, H.: Is neural machine translation ready for deployment? A case study on 30 translation directions. In: Proceedings of the IWSLT. Tokyo, Japan (2016)
8. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In: Proceedings of the EMNLP. pp. 1700–1709. Seattle, Washington, USA (2013)
9. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proceedings of the EMNLP. pp. 388–395. Barcelona, Spain (2004)
10. L'Hostis, G., Grangier, D., Auli, M.: Vocabulary selection strategies for neural machine translation. In: arXiv:1610.00072 (2017)
11. Li, X., Zhang, J., Zong, C.: Towards zero unknown word in neural machine translation. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16). pp. 2852–2858 (2016)
12. Luong, M.T., Manning, C.D.: Achieving open vocabulary neural machine translation with hybrid word-character models. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 1054 – 1063. Berlin, Germany (2016)
13. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the EMNLP. pp. 1412–1421. Lisbon, Portugal (2015)
14. Luong, T., Sutskever, I., Le, Q., Vinyals, O., Zaremba, W.: Addressing the rare word problem in neural machine translation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. pp. 11–19. Beijing, China (2015)
15. Mi, H., Wang, Z., Ittycheriah, A.: Vocabulary manipulation for neural machine translation. In: In Proceedings of Annual Meeting of the Association for Computational Linguistics. Berlin, Germany (2016)
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: A method for automatic evaluation of machine translation. In: Proceedings of the ACL. pp. 311–318. Philadelphia, USA (2002)
17. Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A.V.M., Mokry, J., Nădejde, M.: Nematus: a toolkit for neural machine translation. In: arXiv:1703.04357 (2017)
18. Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. In: Proceedings of the First Conference on Machine Translation. pp. 83–91. Berlin, Germany (2016)
19. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the ACL. pp. 1715–1725. Berlin, Germany (2016)
20. Sutskever, I., Vinyals, O., , Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 2014 Neural Information Processing Systems. pp. 3104–3112. Montreal, Canada (2014)
21. Toral, A., Sánchez-Cartagena, V.M.: A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In: Proceedings of the EACL. Valencia, Spain (2017)
22. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., et al., M.N.: Google's neural machine translation system: Bridging the gap between human and machine translation. In: arXiv:1609.08144 (2016)
23. Zeiler, M.D.: Adadelta: An adaptive learning rate method. In: CoRR, abs/1212.5701 (2012)