A Comparison on Fine-grained Pre-trained Embeddings for the WMT19 Chinese-English News Translation Task

Zhenhao Li
Department of Computing
Imperial College London, UK
zhenhao.li18@imperial.ac.uk

Lucia Specia
Department of Computing
Imperial College London, UK
l.specia@imperial.ac.uk

Abstract

This paper describes our submission to the WMT 2019 Chinese-English (zh-en) news translation shared task. Our systems are based on RNN architectures with pre-trained embeddings which utilize character and sub-character information. We compare models with these different granularity levels using different evaluating metics. We find that a finer granularity embeddings can help the model according to character level evaluation and that the pre-trained embeddings can also be beneficial for model performance marginally when the training data is limited.

1 Introduction

Neural Machine Translation (NMT) systems are mostly based on an encoder-decoder architecture with attention. Given a sentence x in source language, the model predicts a corresponding output sentence y in target language, which maximizes the conditional probability $p(y|x)$. The attention-based Recurrent Neural Network (RNN) version of this architecture has been a very popular approach to NMT (Bahdanau et al., 2015; Luong et al., 2015). Despite the success of these models, they still suffer from problems such as out-of-vocabulary (OOV) words, i.e. words that have not been seen at training. To alleviate the OOV problem, we follow the methods used in word representation and segment words into smaller units. In some morphorlogically rich languages such as Chinese, a word can be divided into characters and then the characters can be further divided into smaller components called *glyphs*. Both character and glyph might contain semantic information and therefore utilizing such information might help alleviate the OOV problem.

Based on the RNN attention-based model (Bahdanau et al., 2015), we experiment with different granularity levels on the WMT19 Chinese-English (zh-en) news translation shared task. This paper describes our submitted systems with embeddings pre-trained on monolingual corpora. The two submitted systems use pre-trained embeddings enhanced by character and sub-character information respectively. The preprocessing methods include Chinese word segmentation, tokenization, data filtering based on rules and Byte Pair Encoding (BPE). Our baseline model is based on RNNSearch (Bahdanau et al., 2015) operating on word level and we use Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) as encoder and decoder. For character level word embeddings, we use the Character-Enhanced Word Embedding (CWE) proposed by Chen et al. (2015). For the sub-character level embeddings, we use the Joint Learning Word Embedding (JWE) proposed by Yu et al. (2017). We use various metrics, namely BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011), TER (Snover et al., 2006) and CharacTER (Wang et al., 2016) for evaluation.

When compared with our baseline model, the models with pre-trained sub-character level embeddings on monolingual corpus show better performance, achieving an increase of +0.53 BLEU score with the sub-character level embeddings. We ran additional experiments on the character and subcharacter level pre-trained embeddings and found that the use of these embeddings can benefit the model when the training corpus size is limited.

This paper is structured as follows: Section 2 introduces the related work including the model architecture and pre-trained embeddings used in our experiment. In Section 3, data selection and preprocessing methods are described. Section 4 introduces the model architectures and hyperparameter settings. Section 5 shows the evaluation results on models with different granularity levels. Section 6

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 249–256
Florence, Italy, August 1-2, 2019.

shows additional experiments to better understand our models.

2 Related Work

NMT has been an important task in Natural Language Processing. A translation system aims to find the corresponding target sentence $\mathbf{y} = \{y_1, y_2, ..., y_m\}$ given a sentence $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ in source language, in a probabilistic manner, represented as $\max_y P(\mathbf{y}|\mathbf{x})$. Most NMT models are based on the sequence-to-sequence approach, and the RNN-based architecture (Sutskever et al., 2014) with attention (Bahdanau et al., 2015) is a popular version of such an approach. The attention mechanism functions as a dynamic calculation of the context vector. At each decoding step, a probability distribution is calculated based on the current decoder hidden state and all encoder hidden states. This distribution is defined as the attention score, representing the importance of each input token at current decoding time step. The context vector is calculated as a weighted average of all encoder hidden state vector, where the attention score is the weight. With the introduction of attention, the model does not need to rely on a single context vector to represent the whole sentence and thus can better handle long sentences.

In recent years model architectures based on convolutional neural networks (Gehring et al., 2017) and transformers (Vaswani et al., 2017) have shown competitive or better performance than RNN-based architectures. In addition, strategies such as back translation (Sennrich et al., 2016a), reranking (Neubig et al., 2015) and model ensembling have led to improvements in translation quality. In our experiments, we only experiment with RNN architectures and focus on the effect of using character and sub-character level embeddings and only use ensembling for comparison purposes.

We use the CWE model proposed by Chen et al. (2015) and the JWE model proposed by Yu et al. (2017) for pre-trained embeddings training. Both models are based on the word2vec proposed by Mikolov et al. (2013). Based on Continuous-Bag-of-Word (CBOW), the CWE model construct a new word representation by summing the word embeddings with character embeddings (see Eq 1). Chen et al. also proposed a multi-prototype character embeddings where characters are tagged with additional factors, such as position and context cluster, for character disambiguation.

$$x_j = w_j \oplus \frac{1}{N_j} \sum_{k=1}^{N_j} c_k \quad (1)$$

where w_j is the word embeddings and c_k is the embeddings of the k-th character in x_j. $\oplus$ is the composition operator (either addition or concatenation).

The JWE model proposed by Yu et al. (2017) is also based on CBOW and it utilizes character and sub-character level information. They construct a dictionary that maps each Chinese character to its sub-character components. As Figure 1 shows, words together with the characters and sub-character components within the context window are all used to predict the target word. The additional semantic information provided by character and subcharacters are shown to improve over word representation, especially in addressing out-of-vocabulary words.

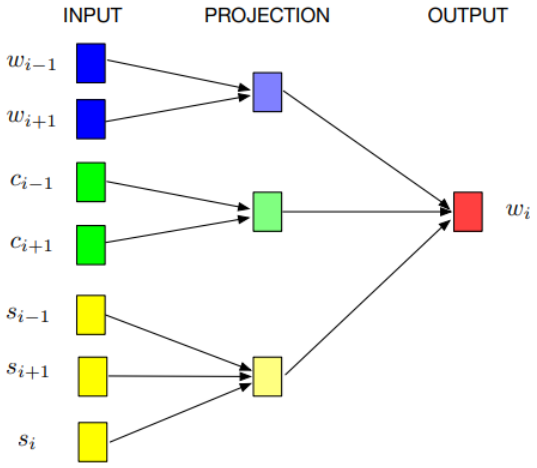

Figure 1: Illustration of JWE embedding taken from (Yu et al., 2017). w_{i-1} and w_{i+1} are context words. c_{i-1} and c_{i+1} represent characters in context words. s_{i-1} and s_{i+1} represent sub-characters of context characters and s_i is the sub-character of target word w_i.

3 Data and Preprocessing

We use all the parallel data provided by WMT for the zh-en translation task, including the News Commentary v14, UN Parallel Corpus V1.0 and the CWMT corpora. In addition, the Common Crawl Corpus from WMT is used as monolingual data to pre-train the embeddings. We use the newsdev2018 and newsdev2017 as validation set

and the newstest2019 as our test data. We tokenize English sentences with the Moses tokenizer (Koehn et al., 2007). On the Chinese side we use Jieba for Chinese word segmentation.[1] The data preprocessing consists of filtering sentences to be added to the parallel training corpus by rules and by alignment score. Following the preprocessing criteria from submissions in previous years (Xu and Carpuat, 2018; Stahlberg et al., 2018; Haddow et al., 2018), we filter the training data based on the following criteria:

- The length of sentences in both languages must be between 4 and 50.
- The maximum length ratio of sentence pairs is 1.3.
- Chinese sentences with no Chinese character are filtered out.
- English sentences with no English character are filtered out.
- Same source and target sentences are removed.
- Sentences should not contains HTML tags.
- Sentence pairs with alignment score above -65 are removed.[2]

The `fast_align` toolkit[3] is used to calculate the alignment score for the parallel data. After the filtering, 10.38M sentence pairs are used as training data. We apply Byte-pair Encoding (BPE) (Sennrich et al., 2016b) with 30,000 merge operations on the English sentences. For Chinese sentences, we segment them into different granularity levels, including words, subwords via BPE and characters. In the character level setting, only Chinese words are separated and each character is treated as a single token. The training texts for models with pre-trained embeddings is the same as baseline, which use words as basic units.

4 Models

4.1 Baseline

The baseline model is based on the bidirectional RNN architecture with attention (Bahdanau et al., 2015). Our models are built with OpenNMT-py (Klein et al., 2017). We follow the hyperparameter setting of Deep RNN from Xu and Carpuat (2018) and use a four-layer LSTM for both the encoder and decoder. The embeddings and hidden layer size are limited to 512. We use the Adam optimizer (Kingma and Ba, 2015) with initial learning rate of 0.0005. We apply label smoothing (Szegedy et al., 2016) and dropout (Srivastava et al., 2014) of 0.1 to avoid overfitting. We use the multi-layer perception (mlp) attention as in (Bahdanau et al., 2015). The batch size is 4096 tokens per batch and the models are selected based on best performance on the validation set. All our models are trained on a GTX 1080Ti GPU.

4.2 Pre-trained Embeddings

We apply pre-trained embeddings to the two submitted systems. The character level and sub-character level pre-trained embeddings are trained with CWE (Chen et al., 2015) and JWE (Yu et al., 2017) respectively. We trained the embeddings on the Common Crawl Corpus provided by WMT19 and fine-tuned them on the task data when training the RNN. The preprocessing for monolingual data includes Chinese word segmentation and removal of non-Chinese characters. Apart from the pre-trained embeddings, the hyperparameters of the two submissions are the same as in the baseline system.

5 Result and Analysis

We use the `CharacTER.py`[4] script for CharacTER score calculation and `multeval`[5] (Clark et al., 2011) to calculate BLEU, METEOR and TER scores. The evaluation results for models on word, subword and character level are presented in Table 1.

The model with BPE applied on both source and target languages (bpe2bpe) achieves higher score than other single models, with an increase of +1.18 BLEU score over the baseline system. The two models (baseline+cwe, baseline+jwe) utilizing character and sub-character information are based on pre-trained embeddings with CWE and JWE as described in Section 2. We use the source training text for the pre-trained embeddings to prevent the introduction of noise. As we can see from the BLEU scores, the model with JWE pre-

[1] https://github.com/fxsjy/jieba

[2] We tried different filter strategies and found this criterion gives a better performance than others.

[3] https://github.com/clab/fast_align

[4] https://github.com/rwth-i6/CharacTER

[5] https://github.com/jhclark/multeval

Granularity	Model	BLEU	METEOR	TER	CharacTER
word	baseline	16.90	**23.0**	**64.0**	0.717
	baseline+cwe	16.59	22.8	64.4	0.716
	baseline+jwe	**16.91**	**23.0**	**64.0**	0.712
subword	bpe2bpe	**18.08**	**24.2**	**62.1**	0.678
	bpe2bpe+cwe	17.97	**24.2**	62.4	**0.677**
char	char2bpe	15.80	22.5	64.9	**0.705**
word	apprentice-c*	16.94	23.0	63.7	0.713
	apprentice-g*	16.54	23.0	63.7	0.717
	apprentice-g(best)	**17.43**	**23.2**	**63.4**	**0.710**
	ensemble(jwe)	**18.16**	23.5	62.9	**0.702**

Table 1: Model performance on different granularity levels. The two models with a star are the official systems submitted to the WMT19 zh-en news translation shared task, where the pre-trained embeddings is trained on extra monolingual data.

trained embeddings shows similar performance to the baseline system while the model with CWE embeddings on character level shows a marginal decrease. The METEOR and TER score presents similar trends to BLEU, whereas from the evaluation of CharacTER scores the introduction of pre-trained embeddings on both character and sub-character levels shows better performance than the baseline.

It can also be seen from the comparison on BPE-based models that the model with CWE embeddings performs slightly worse than the bpe2bpe model, which operates on BPE on both source and target languages. The results according to CharacTER show that finer granularity embeddings can benefit the model in character level evaluations. The char2bpe model shows the worst performance according to BLEU scores, whereas the CharacTER score of this model is higher than that of other word level models. Finally, when we ensemble the baseline and four models with JWE embeddings pre-trained on different iterations, the BLEU score shows an increase of +1.26 BLEU over the baseline.

The two models with stars (apprentice-c and apprentice-g) are our official shared task submissions, with the first one operating on character level and the second, on glyph (sub-character) level. The apprentice-c model uses the CWE pre-trained embeddings while the apprentice-g uses JWE embeddings. For the first, we train the pre-trained embeddings on the monolingual data (Common Crawl) and then fine-tune it on filtered parallel data during the training of RNN models. Note that we did not use back-translation to augment the training data and due to time limit we apply a relatively larger learning rate than previous work to boost training speed, therefore our systems achieve relatively lower score than the previous work (Xu and Carpuat, 2018). The CWE-based model shows a better BLEU score than the baseline model. The lower performance for the apprentice-g model might have resulted from insufficient training epochs for the JWE embeddings. Due to time restrictions, we did not submit the system with the best word embeddings. In the additional experiments after the task deadline, we fine-tuned the models on the best word embeddings version and achieve a higher BLEU score of 17.43 for the apprentice-g(best) model. The CharacTER score for the fine-tuned model is lower than other models except the two with BPE. Generally, the sub-character level models perform better than the word level and character level models.

6 Additional Experiments

6.1 Evaluating Embeddings

We have tried additional experiments to evaluate the effect of character and subcharacter level pre-trained embeddings. Table 2 presents the model performance with respect to the embeddings performance in traditional word similarity and analogy tasks. We use the wordsim-240 and wordsim-297 dataset and the analogy dataset from Chen et al. (2015) for word similarity and analogy evaluation respectively. We use the evaluation script in JWE[6] for both evaluations.

From Table 2, we can see that among all models with JWE pre-trained embeddings, the one with

[6] https://github.com/HKUST-KnowComp/JWE

Model	BLEU	wordsim-240	wordsim-297	analogy
baseline	16.90	/	/	/
baseline+jwe5	16.43	0.4880	0.5833	0.4680
baseline+jwe10	**16.91**	0.5099	0.5985	**0.5293**
baseline+jwe20	16.82	**0.5152**	0.6037	0.5205
baseline+jwe50	16.37	0.5048	**0.6075**	0.4786
baseline+cwe5	16.59	0.4569	0.5769	0.2820
baseline+cwe10	16.47	0.4593	0.5742	0.3585
baseline+cwe20	16.52	0.4610	0.5764	0.3754
baseline+cwe50	16.49	0.4528	0.5765	0.3443

Table 2: Comparison of model performance and word embeddings performance. The evaluation on wordsim-240 and wordsim-297 test set shows Spearman correlation between the pre-trained embedding and human judgements. The performance on analogy indicates accuracy on analogy reasoning in "a:b::c:?" format. The number after the embeddings type represents number of training iterations.

10 iterations performs the best. When the embeddings are trained over 20 iterations, the BLEU score starts to decrease. The same pattern can be found on the CWE-based models. However, the model with 5-iteration embeddings achieves the highest BLEU score among all CWE-based models. From the embeddings performance on the analogy task, excluding the cwe5 model, we find that the embeddings performance correlates with BLEU scores. When comparing the CWE-based models with the JWE-based models, we see that on both translation quality and word embeddings evaluations, the model on finer granularity performs best.

6.2 Effect of Corpus Size

Another experiment was done to compare the effect of pre-trained embeddings on different corpora sizes. We train the word embeddings with best iteration setting and train the RNN model on different corpora sizes. Smaller corpora are created by taking 25% and 50% of the original corpus. Table 3 presents the BLEU scores for models on smaller corpora.

Model/data size	25%	50%	100%
baseline	15.95	15.95	16.90
baseline+cwe5	16.00	15.82	16.59
baseline+jwe10	16.04	15.95	16.91

Table 3: BLEU score with different training data sizes.

It can be seen from Table 3 that with smaller parallel training corpora the introduction of the pre-trained word embeddings has a more marked positive influence. When the dataset is reduced to half, all the three models show a decrease in BLEU score. However, the gap between the baseline and the cwe-based model is smaller. When the dataset is further limited to 25%, both models with pre-trained embeddings perform better than the baseline, whose score does not change. Although it seems that the pre-trained embeddings, even with sub-character level semantic information involved, could only benefit marginally on the whole training data, the introduction of extra semantic information might play a more important role when the parallel training resources are limited.

6.3 Effect of Sentence Length

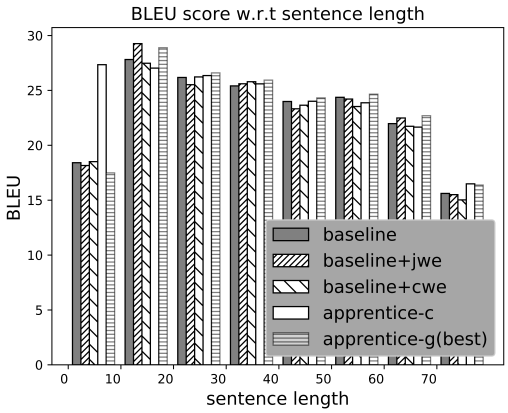

Figure 2: BLEU score of models w.r.t sentence length.

Here we measure the performance of models with varying sentence lengths, as shown in Figure 2. The test set is seperated into 8 subsets based on the sentence lengths and models are evaluated on each subset, the x-axis in Figure 2 represents sentence length intervals. We see that the two models with embeddings trained on a larger

monolingual corpus perform better than the other models in medium-length sentences (between 30 and 50). The apprentice-c model, which uses CWE embeddings operating on character level, greatly outperforms the other models on short sentences with length less than 10. Since the sentence length is short, the tokens in the sentence are mostly composed of one or two characters, thus the model with character-based embeddings has an advantage. Regarding the two models with embeddings trained without extra monolingual data, both models show good performance on medium length sentences but perform poorly on long sentences. The introduction of pre-trained embeddings can increase the models' preference to generate shorter sentences, resulting in the model achieving lower BLEU score on long sentences.

6.4 Analysis of Model Perplexity

In order to understand the effect of pre-trained embedding on target language model, we calculate the model perplexity on the test data with models on different corpus size. The result is represented in Table 4. The model with JWE pre-trained embeddings performs better on all corpus sizes, having a lower perplexity, though the difference is marginal. Similar result as the BLEU evaluation shows that the pre-trained embeddings benefit model performance on smaller corpus sizes.

Model	Perplexity	Corpus size
baseline	2.947	100%
+cwe	3.005	
+jwe	**2.932**	
baseline	3.049	50%
+cwe	3.046	
+jwe	**3.023**	
baseline	2.860	25%
+cwe	2.847	
+jwe	**2.836**	

Table 4: Model perplexity on test set.

6.5 Transformer Models

Besides the RNN model, we also experimented with pre-trained embeddings and the transformer architecture. We follow the hyperparameter setting from Vaswani et al. (2017), limiting the embeddings to 512 dimensions. We compare the transformer models with and without pre-trained embeddings. The results are presented in Table 5.

From the evaluation results on BLEU and CharacTER, the transformer models without pre-trained embeddings show better performance. We find it interesting that the embedding pre-trained with CWE decrease the performance severely, leading to a reduction of -3.85 BLEU score from the model without it. The introduction of finer granularity embeddings might not benefit the transformer performance. We hypothesize that the pre-trained embedding enhanced by character and sub-character infomation might conflict with the fixed positional encoding used in transformer.

Model	BLEU	CharacTER
transformer	17.82	0.692
transformer+cwe	13.97	0.754
transformer+jwe	17.59	0.695

Table 5: BLEU and CharacTER for transformer models.

7 Conclusion

This paper describes our NMT models with pre-trained embeddings operating on character and sub-character levels. We participated in the WMT19 zh-en news translation shared task and submitted two systems with embeddings trained on monolingual corpus. We experimented with the effect of using fine-grained pre-trained embeddings and showed the potential benefit of using them. In additional experiments, we find that using pre-trained embeddings can better benefit the translation models when the parallel training data is limited.

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*.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages

176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. *ArXiv e-prints*.

Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone, and Rico Sennrich. 2018. The university of edinburghs submissions to the wmt18 news translation task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 403–413, Belgium, Brussels. Association for Computational Linguistics.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. 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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 35–41, Kyoto, Japan. Workshop on Asian Translation.

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*, pages 311–318. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Felix Stahlberg, Adri de Gispert, and Bill Byrne. 2018. The university of cambridges machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 508–516, Belgium, Brussels. Association for Computational Linguistics.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 505–510.

Weijia Xu and Marine Carpuat. 2018. The university of maryland's chinese-english neural machine translation systems at wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 539–544, Belgium, Brussels. Association for Computational Linguistics.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 286–291.