Character-level Chinese-English Translation through ASCII Encoding

Nikola I. Nikolov*, **Yuhuang Hu***, **Mi Xue Tan, Richard H.R. Hahnloser**
Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
{niniko, yuhuang.hu, mtan, rich}@ini.ethz.ch

Abstract

Character-level Neural Machine Translation (NMT) models have recently achieved impressive results on many language pairs. They mainly do well for Indo-European language pairs, where the languages share the same writing system. However, for translating between Chinese and English, the gap between the two different writing systems poses a major challenge because of a lack of systematic correspondence between the individual linguistic units. In this paper, we enable character-level NMT for Chinese, by breaking down Chinese characters into linguistic units similar to that of Indo-European languages. We use the Wubi encoding scheme[1], which preserves the original shape and semantic information of the characters, while also being reversible. We show promising results from training Wubi-based models on the character- and subword-level with recurrent as well as convolutional models.

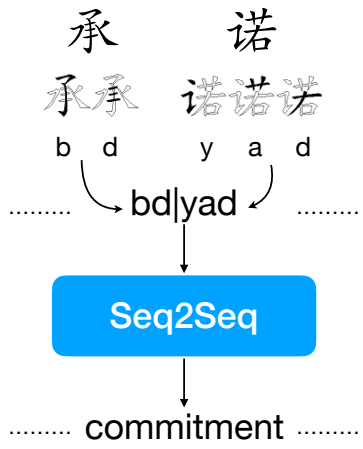


Figure 1: Overview of the **wubi2en** approach to Chinese-to-English translation. A raw Chinese word (‘承诺’) is encoded into ASCII characters (‘bd|yad’), using the Wubi encoding method, before passing it to a Seq2Seq network. The network generates the English translation ‘commitment’, processing one ASCII character at a time.

1 Introduction

Character-level sequence-to-sequence (Seq2Seq) models for machine translation can perform comparably to subword-to-subword or subword-to-character models, when dealing with Indo-European language pairs, such as German-English or Czech-English (Lee et al., 2017). Such language pairs benefit from having a common Latin character representation, which facilitates suitable character-to-character mappings to be learned. This method, however, is more difficult for non-Latin language pairs, such as Chinese-English. Chinese characters differ from English characters, in the sense that they carry more meaning and resemble subword units in English. For example, the Chinese character ‘人’ corresponds to the word ‘human’ in English. This lack of correspondence makes the problem more demanding for a Chinese-English character-to-character model, as it would be forced to map higher-level linguistic units in Chinese to individual Latin characters in English. Good performance on this task may, therefore, require specific architectural decisions.

In this paper, we propose a simple solution to this challenge: encode Chinese into a meaningful string of ASCII characters, using the **Wubi** method (Lunde, 2009) (Section 3). This encoding enables efficient and accurate character-level prediction applications in Chinese, with no changes required to the model architecture (see Figure 1). Our approach significantly reduces the character vocabulary size of a Chinese text, while preserving the shape and semantic information encoded in the Chinese characters.

[1]Code and data available at https://github.com/duguyue100/wmt-en2wubi.

* Equal contribution

We demonstrate the utility of the Wubi encoding on subword- and character-level Chinese NMT, comparing the performance of systems trained on Wubi vs. raw Chinese characters (Section 4). We test three types of Seq2Seq models: recurrent (Cho et al., 2014) convolutional (Gehring et al., 2017) as well as hybrid (Lee et al., 2017). Our results demonstrate the utility of Wubi as a preprocessing step for Chinese translation tasks, showing promising performance.

2 Background

2.1 Sequence-to-sequence models for NMT

Neural networks with Encoder-Decoder architectures have recently achieved impressive performance on many language pairs in Machine Translation, such as English-German and English-French (Wu et al., 2016). Recurrent Neural Networks (RNNs) (Cho et al., 2014) process and *encode* the input sequentially, mapping each word onto a vector representation of fixed dimensionality. The representations are used to condition a *decoder* RNN which generates the output sequence.

Recent studies have shown that Convolutional Neural Networks (CNNs) (LeCun et al., 1998) can perform better on Seq2Seq tasks than RNNs (Gehring et al., 2017; Chen and Wu, 2017; Kalchbrenner et al., 2016). CNNs enable simultaneous computations which are more efficient especially using parallel GPU hardware. Successive layers in CNN models have an increasing receptive field for modeling long-term dependencies in candidate languages.

2.2 Chinese-English translation

Recent large-scale benchmarks of RNN encoder-decoder models (Wu et al., 2016; Junczys-Dowmunt et al., 2016) have shown that translation pairs involving Chinese are among the most challenging for NMT systems. For instance, in Wu et al. (2016) an NMT system trained on English-to-Chinese had the least relative improvement across five other language pairs, measured over the performance of a phrase-based machine translation baseline.

While it is known that the quality of a Chinese translation system can be significantly impacted by the choice of word segmentation (Wang et al., 2015), there has been little work on improving the representation medium for Chinese translation. Wang et al. (2017) perform an empirical comparison on various translation granularities for the Chinese-English task. They find that adding additional information about the segmentation of the Chinese characters, such as marking the start and the end of each word, leads to improved performance over raw character or word translation.

The work that is most related to ours is (Du and Way, 2017), in which they use Pinyin[2] to romanize raw Chinese characters based on their pronunciation. This method, however, adds ambiguity to the data, because many Chinese characters share the same pronunciation.

3 Encoding Chinese characters with Wubi

Wubi (Lunde, 2009) is a shape-based encoding method for inputting Chinese characters on a computer QWERTY keyboard. The encoding is based on the structure of the characters rather than on their pronunciation. Using the method, each raw Chinese character (e.g., "设") can be efficiently mapped to a unique sequence of 1 to 5 ASCII characters (e.g., "ymc"). This feature greatly reduces the ambiguity brought by other phonetic input methods, such as Pinyin.

As an input method, Wubi uses 25 key caps from the QWERTY keyboard, where each key cap is assigned to five categories based on the character's first stroke (when written by hand). Each of the key caps is associated with different character roots. A Chinese character is broken down into its character roots, and a corresponding QWERTY association of the character roots is used to encode a word. For example, the Wubi encoding of '哈' is 'kwgk', and the character roots of this word are 口(k), 人(w), 王(g) and 口(k). To create a one-to-one mapping of every Chinese character to a Wubi encoding during translation, we append numbers to the encodings, whenever one code maps to multiple Chinese characters.

Table 1: Examples of Wubi words and the corresponding Chinese words

English	Chinese	Wubi
Set up	编设	xyna0|ymc
Public property	公共财产	wc|aw|mf|u
Step aside	让开	yh|ga

Applying Wubi significantly reduces the

[2]The official romanization system for Standard Chinese in mainland China.

character-level vocabulary size of a Chinese text (from $> 5,000$ commonly used Chinese characters, to 128 ASCII characters[3]), while preserving its shape and semantic information. Table 1 contains examples of Wubi, along with the corresponding words in Chinese and English.

4 Results

4.1 Dataset

In this work, we use a subset of the English and Chinese parts of the United Nations Parallel Corpus (Ziemski et al., 2016). We choose the UN corpus because of its high-quality, man-made translations. The dataset is sufficient for our purpose: our aim here is not to reach state-of-the-art performance on Chinese-English translation, but to demonstrate the potential of the Wubi encoding on the character level.

We preprocess the UN dataset with the MOSES tokenizer[4], and use Jieba[5] to segment the Chinese sentence into words, following which we encode the texts into Wubi. We use the '|' character as a subword separator for Wubi, in order to ensure that the mapping from Chinese to Wubi is unique. We also convert all Chinese punctuation marks (e.g. '。、《》') from UTF-8 to ASCII (e.g. '.,<>') because they share similar linguistic roles to English punctuations. This conversion additionally decreases the size of the Wubi character vocabulary.

Our final dataset contains 2.1M sentence pairs for training, and 55k pairs for validation and testing respectively (Table 2 contains additional statistics). Note that our procedures are entirely reversible.

Table 2: Statistics of our dataset (mean and standard deviation).

	English	Wubi	Chinese
words per sentence	25.8±11.0	22.9±10.0	22.9±10.0
characters per word	4.9±3.3	4.6±3.3	1.8±0.83
characters per sentence	152.3±67.9	127.1±56.5	63.5±27.6

To investigate the utility of the Wubi encoding, we compare the performance of NMT models on four training pairs: raw Chinese-to-English (*cn2en*) versus Wubi-to-English (*wubi2en*); English-to-raw Chinese (*en2cn*) versus English-to-Wubi (*en2wubi*). For each pair, we investigate three levels of sequence granularity: word-level, subword-level, and character-level. The word-level operates on individual English words (*e.g.* walk) and either raw-Chinese words (*e.g.* 编设) or Wubi words (*e.g.* sh|wy). We limit all word-level vocabularies to the 50k most frequent words for each language. The subword-level is produced using the byte pair encoding (BPE) scheme (Sennrich et al., 2016), capping the vocabulary size at 10k for each language. The character-level operates on individual raw-Chinese characters (e.g. '重'), or individual ASCII characters.

4.2 Model descriptions and training details

Our models are summarized in Table 3, including the number of parameters and vocabulary sizes used for each pair. For the subword- and word-level experiments, we use two systems[6]. The first, *LSTM*, is an LSTM Seq2Seq model (Cho et al., 2014) with an attention mechanism (Bahdanau et al., 2015). We use a single layer of 512 hidden units for the encoder and decoder, and set 512 as the embedding dimensionality. The second system, *FConv*, is a smaller version of the convolutional Seq2Seq model with an attention mechanism from (Gehring et al., 2017). We use word embeddings with dimension 256 for this model. The encoder and the decoder of *FConv* have the same convolutional architecture which consists of 4 convolution layers for the encoder and 3 for the decoder, each layer having filters with dimension 256 and size 3.

For all character-level experiments, we use the fully-character level model, *char2char* from (Lee et al., 2017)[7]. The encoder of this model consists of 8 convolutional layers with max pooling, which produce intermediate representations of segments of the input characters. Following this, a 4-layer highway network (Srivastava et al., 2015) is applied, as well as a single-layer recurrent network with gated recurrent units (GRUs) (Cho et al., 2014). The decoder consists of an attention mechanism and a two-layer GRU, which predicts the output one character at a time. The character embedding dimensionality is 128 for the encoder and

[3] 302 ASCII and special characters such as non-ASCII symbols used in the experiments, see Section 4.

[4] https://github.com/moses-smt

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

[6] We use the fairseq library https://github.com/pytorch/fairseq.

[7] https://github.com/nyu-dl/dl4mt-c2c

Table 3: Model and vocabulary sizes used in our experiments. In brackets, we include the number of embedding parameters for a model (left), or the percentage of vocabulary coverage of the dataset (right).

	No. of model parameters (Embedding)			**Vocab Size (% coverage of dataset)**		
level	char2char	FConv	LSTM	EN	Wubi	CN
word	-	42M (25M)	83M (51M)	50k (99.7%)	50k (99.5%)	50k (99.5%)
subword	-	11M (5.1M)	22M (10.6M)	10k (100%)	10k (100%)	10k (98.7%)
character	69-74M (0.21M-2.81M†)	-	-	302 (100%)	302 (100%)	5183 (100%)

†: 0.21M for wb2en/en2wb (69M in total); 0.77M for cn2en (70M) and 2.81M for en2cn (74M), due to a larger size of the decoder embedding.

Table 4: BLEU test scores on the UN dataset.

	character	**subword**		**word**	
	char2char	FConv	LSTM	FConv	LSTM
wubi2en	**40.55**	38.20	43.06	39.53	43.36
cn2en	39.60	38.20	43.03	39.64	43.67
en2wubi	**36.78**	**36.04**	**39.03**	36.98	39.69
en2cn†	36.13	35.41	38.64	37.25	39.59

†: We convert these translations to Wubi before computing BLEU to ensure a consistent comparison.

512 for the decoder, whereas the number of hidden units is 512 for the encoder and 1024 for the decoder.

We train all models for 25 epochs using the Adam optimizer (Kingma and Ba, 2014). We used four NVIDIA Titan X GPUs for conducting the experiments, and use beam search with beam size of 20 to generate all final outputs.

4.3 Quantitative evaluation

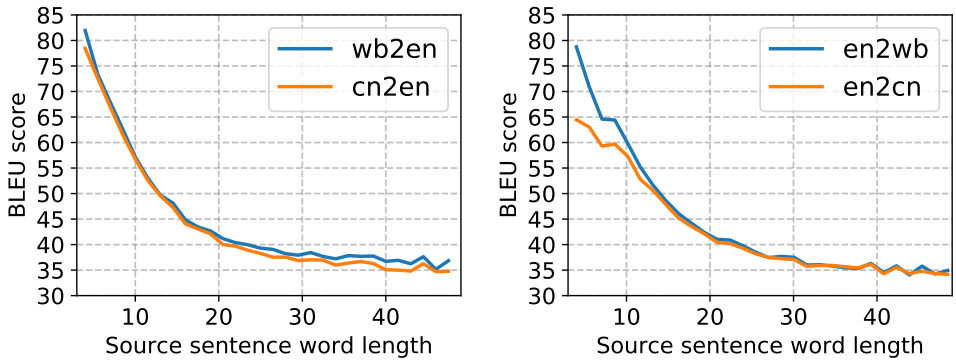
In Table 4, we present the BLEU scores for all the previously described experiments. Before computing BLEU, we convert all Chinese outputs to Wubi to ensure a consistent comparison. This conversion has a one-to-one mapping between Chinese and Wubi, whereas, in the reverse direction, ill-formed Wubi output on the character-level might not be reversible to Chinese.

On the word-level, the Wubi-based models achieve comparable results to their counterparts in Chinese, in both translation directions. *LSTM* significantly outperforms *FConv* across all experiments here, most likely due to its much larger size (see Table 3).

On the subword-level, we observe a slight increase of about 0.5 BLEU when translating from English to Wubi instead of raw Chinese. This increase is most likely due to the difference in the BPE vocabularies: while the English and Wubi BPE rules that were learned cover 100% of the dataset, for Chinese this is 98.7% - the remaining 1.3% had to be replaced by the *unk* symbol under our vocabulary constraints. While the models were capable of compensating for this gap when translating to English, in the reverse direction it resulted in a loss of performance. This highlights one benefit of Wubi on the subword-level: the Latin encoding seems to give a greater flexibility for extracting suitable BPE rules. It would be interesting to repeat this comparison using much larger datasets and larger BPE vocabularies.

Character-level translation is more difficult than word-level, since the models are expected to not only predict sentence-level semantics, but also to generate the correct spelling of each word. Our *char2char* Wubi models outperformed the raw Chinese models with 0.95 BLEU points when translating to English, and 0.65 BLEU when translating from English. The differences are statistically significant ($p = 0.001$ and $p = 0.034$ respectively) according to bootstrap resampling (Koehn, 2004) with 1500 samples. The results demonstrate the advantage of Wubi on the character-level, which outperforms raw Chinese even though it has fewer parameters dedicated for character embeddings (Table 3) and that it has to deal with substantially longer input or output sequences (see Table 2).

In Figure 2, we plot the sentence-level BLEU scores obtained by the *char2char* models on our test set, with respect to the length of the input sentences. When translating from Chinese to En-



(a) Translation from Chinese to English.

(b) Translation from English to Chinese.

Figure 2: Sentence-level BLEU scores obtained by the character-level *char2char* models on our test dataset, plotted with respect to the word length of the source sentences.

glish (Figure 2a) the Wubi-based model consistently outperforms the raw Chinese model, for all input lengths. Interestingly, the gap between the two systems increases for longer Chinese inputs of over 20 words, indicating that Wubi is more robust for such examples. This result could be explained by the fact that the encoder of the *char2char* model is more suitable for modeling languages with a higher level of granularity such as English and German. When translating from English to Chinese (Figure 2b) Wubi still has a small edge, however in this case we see the reverse trend: it performs much better on shorter sentences up to 12 English words. Perhaps, the increased granularity of the output sequence led to an advantage during decoding using beam search.

Interestingly, all the *char2char* models use only a tiny fraction of their parameters as embeddings, due to the much smaller size of their vocabularies. The best-performing LSTM word-level model has the majority of its parameters, 61% or over 50M, dedicated to word embeddings. For the Wubi-based character-level models, the number is only 0.3% or 0.21M. There is even a significant difference between Wubi and Chinese on the character-level, for example, *en2wb* has 12 times fewer embedding parameters than *en2cn*. Thus, although *char2char* performed worse than *LSTM* in our experiments, these results highlight the potential of character-level prediction for developing compact yet performant translation systems, for Latin as well as non-Latin languages.

4.4 Qualitative evaluation

In Table 5, we present four examples from our test dataset that cover short as well as long sentences. We also include the translations produced by the character-level *char2char* systems, which is the main focus of this paper. Full examples from the additional systems are available in the supplementary material.

In the first example, which is a short sentence resembling the headline of a document, both the *wubi2en* and cn2en models produced correct translations. When translating from English to Chinese, however, the *en2wubi* produced the word '与' (highlighted in red) which more correctly matches the ground truth text. In contrast, the *en2cn* model produced the synonym '和'. In the second example, the *en2wubi* output completely matches the ground truth and is superior to the *en2cn* output. The latter failed to correctly translate 'the' to '这次' (marked in green).

The *wubi2en* translation in the third example accurately translated the word 'believe' (marked in blue) and the full form of the abbreviation 'ldcs' – 'the least developed countries' (highlighted in green), whereas the *cn2en* chooses 'are convinced' and ignores 'ldcs' in its output sentence. Interestingly, although the ground truth text maps the word 'essential' (marked in red) to three Chinese words '至␣为␣重 要', both *en2wubi* and *en2cn* use only a single word to interpret it. Arguably, *en2wubi*'s translation '至关重要' is closer to the ground truth than *en2cn*'s translation '必不可少'.

The fourth example is more challenging. There, the English ground truth 'requested' (highlighted in blue) maps to two different parts of the Chinese ground truth '提出' (in blue) and '要求' (in green). This one-to-many mapping confuses both translation models. The *wubi2en* tries to match the Chinese text by translating '提出' into 'pro-

Table 5: Four examples from our test dataset, along with system-generated translations produced by the *char2char* models. We converted the Wubi translations to raw Chinese. Translations of words with a similar meaning are marked with the same color.

Translation Type		Example 1
English	ground truth	social and human rights questions
Chinese	ground truth	社会 与 人权 问题
Wubi	ground truth	py|wf gn w|sc ukd0|jghm1|
	wubi2en	social and human rights questions
	cn2en	social and human rights questions
	en2wubi	社会 与 人权 问题
	en2cn	社会 和 人权 问题
		Example 2
English	ground truth	the informal consultations is open to all member states .
Chinese	ground truth	所有 会员国 均 可 参加 这次 非正式 协商 。
Wubi	ground truth	rn|e wf|km|l fqu sk cd|lk p|uqw djd|ghd0|aa fl|um .
	wubi2en	this informal consultation may be open to all member states .
	cn2en	the informal consultations will be open to all member states .
	en2wubi	所有 会员国 均 可 参加 这次 非正式 协商 。
	en2cn	所有 会员国 均 可 进行 非正式 协商 。
		Example 3
English	ground truth	we believe that increased trade is essential for the growth and development of ldcs .
Chinese	ground truth	我们 相信 ， 增加 贸易 对 最 不 发达国家 的 增长 和 发展 至 为 重要 。
Wubi	ground truth	q|wu sh|wy , fu|lk qyv|jqr cf jb i v|dp|l|pe r fu|ta t v|nae gcf o tgj|s .
	wubi2en	we believe that increased trade is essential for the growth and development of the least developed countries .
	cn2en	we are convinced that increased trade growth and development is essential .
	en2wubi	我们 认为 ， 增加 贸易 对 最 不 发达国家 的 增长 和 发展 至关重要 。
	en2cn	我们 认为 ， 增加 贸易 对于 最 不 发达国家 的 增长 和 发展 来说 是 必不可少 的 。
		Example 4
English	ground truth	in some cases , additional posts were requested without explanation .
Chinese	ground truth	在 某些 情况 中 ， 提出 增加 员额 要求 时 ， 并未 作出 说明 。
Wubi	ground truth	d afs|hxf nge|ukq k , rj|bm fu|lk km|ptkm0 s|fiy jf , ua|fii wt|bm yu|je .
	wubi2en	in some cases , no indication was made when additional staffing requirements were proposed .
	cn2en	in some cases , there was no indication of the request for additional posts .
	en2wubi	在 有些 情况 下 ， 要求 增加 员额 。
	en2cn	在 有些 情况 下 还 要求 增设 员额 ， 但 没有 作出 任何 解释 。

posed' and '要求' into 'requirements': this model may have been misled by the word '时' (can be translated to 'when'); the output contains an adverbial clause. While the *wubi2en* output is closer to the ground truth, the two have little overlap. For the English-to-Chinese task, the *en2cn* translation is better than the one produced by *en2wubi*: while *en2cn* successfully translated 'without explanation' (in red), the *en2wubi* model ignored this part of the sentence.

The Wubi-based models tend to produce slightly shorter translations for both directions (see Table 6). In overall, the Wubi-based outputs appear to be visibly better than the raw Chinese-based outputs, in both directions.

5 Conclusion

We demonstrated that an intermediate encoding step to ASCII characters is suitable for the character-level Chinese-English translation task, and can even lead to performance improvements. All of our models trained using the Wubi encoding achieve comparable or better performance to the baselines trained directly on raw Chinese. On the character-level, using Wubi yields BLEU improvements when translating both to and from English, despite the increased length of the input or output sequences, and the smaller number of embedding parameters used. Furthermore, there are also improvements on the subword-level, when translating from English.

Table 6: Word counts of the outputs of the *char2char* models (mean and standard deviation).

Model	Word Count
wb2en	25.01 ± 10.95
cn2en	25.80 ± 11.72
en2wb	21.61 ± 9.68
en2cn	22.19 ± 10.11

Future work will focus on making use of the semantic structure of the Wubi encoding scheme, to develop architectures tailored to utilize it. Another exciting future direction is multilingual many-to-one character-level translation from Chinese and several Latin languages simultaneously, which becomes possible using encodings such as Wubi. This has previously been successfully realized for Latin and Cyrillic languages (Lee et al., 2017).

Acknowledgments

We acknowledge support from the Swiss National Science Foundation (grant 31003A_156976) and from the National Centre of Competence in Research (NCCR) Robotics.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations 2015*.

Qiming Chen and Ren Wu. 2017. CNN is all you need. *CoRR*, abs/1712.09662.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Jinhua Du and Andy Way. 2017. Pinyin as subword unit for chinese-sourced neural machine translation. In *Irish Conference on Artificial Intelligence and Cognitive Science 2017*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. In *Proceedings of the International Workshop on Spoken Language Translation 2016*, volume 1.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *CoRR*, abs/1610.10099.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

K. Lunde. 2009. *CJKV Information Processing*. O'Reilly Series. O'Reilly Media.

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)*, pages 1715–1725. Association for Computational Linguistics.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2377–2385. Curran Associates, Inc.

Rui Wang, Hai Zhao, and Bao-Liang Lu. 2015. English to Chinese translation: How chinese character matters. In *PACLIC*.

Yining Wang, Long Zhou, Jiajun Zhang, and Chengqing Zong. 2017. Word, subword or character? an empirical study of granularity in chinese-english nmt. In *Machine Translation*, pages 30–42, Singapore. Springer Singapore.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*.