Approaching Neural Chinese Word Segmentation as a Low-Resource Machine Translation Task

Pinzhen Chen **Kenneth Heafield**
School of Informatics, University of Edinburgh
{pinzhen.chen, kenneth.heafield}@ed.ac.uk

Abstract

Chinese word segmentation has entered the deep learning era which greatly reduces the hassle of feature engineering. Recently, some researchers attempted to treat it as character-level translation, which further simplified model designing, but there is a performance gap between the translation-based approach and other methods. This motivates our work, in which we apply the best practices from low-resource neural machine translation to supervised Chinese segmentation. We examine a series of techniques including regularization, data augmentation, objective weighting, transfer learning, and ensembling. Compared to previous works, our low-resource translation-based method maintains the effortless model design, yet achieves the same result as state of the art in the constrained evaluation without using additional data.

1 Introduction

Chinese text is written in characters as the smallest unit, and it has no explicit word boundary. Therefore, Chinese word segmentation (CWS) serves as upstream tokenization and disambiguation for Chinese language processing. The task is often viewed as sequence labelling, where each character receives a label indicating its relative position in a segmented sequence (e.g. whether the character is at the word boundary). While traditional machine learning methods have attained strong results, recent investigations focus on neural networks given their rise in the entire NLP field. Distinctive to sequence labelling, Shi et al. (2017) first treat CWS as neural machine translation (NMT). Nonetheless, Zhao et al. (2018) point out that without extra resources, all previous neural methods are not yet comparable with the non-neural state of the art from Zhao and Kit (2008); the NMT practice is even behind.

We note two advantages of treating the task as neural translation: 1) the entire input sentence is encoded before making any segmentation decision; 2) such a model jointly trains character embeddings with sequence modelling. Thus, we try to bridge the gap between the translation-based approach and state-of-the-art models, using low-resource techniques commonly seen in NMT. The translation-based method can be easy to adopt without the need for feature extraction and model modifictaion. Although NMT is known to be data hungry, our approach is able to achieve competitive results in the constrained evaluation scenario, where introducing extra data is forbidden. In specific, when benchmarked on the second CWS bakeoff (Emerson, 2005), our system reaches the top of the MSR leaderboard and achieves a strong result on the PKU dataset.

2 Related Work

Chinese segmentation is traditionally tackled as sequence labelling, which predicts whether each input character should be split from neighbouring characters (Xue, 2003). Earlier approaches relied on conditional random fields or maximum entropy Markov models (Peng et al., 2004; Ng and Low, 2004). Zhao and Kit (2008) leveraged unsupervised features to attain state-of-the-art results in the data-constrained track.

Recent research has shifted towards neural networks: feed-forward, recurrent and convolutional (Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015a,b; Wang and Xu, 2017). Without external data, these models did not surpass the best non-neural method, but instead, provided great ease of data engineering. Researchers also studied better representations for segments and characters, as well as the incorporation of external resources (Liu et al., 2016; Zhou et al., 2017; Yang et al., 2017). By carefully tuning model configurations, Ma et al. (2018) achieved strong results. The task can also be done through learning to score global word segmentation schemes given charac-

ters (Zhang and Clark, 2007, 2011; Cai and Zhao, 2016; Cai et al., 2017). On top of this, Wang et al. (2019) proved that it is beneficial to integrate unsupervised segmentation. A recent work used a modified Transformer for sequence tagging to attain the same results as the state of the art (Duan and Zhao, 2020).

The most relevant to our research is Shi et al. (2017)'s suggestion to formalize Chinese word segmentation as character-level neural machine translation. It differs from global segmentation scoring in that the NMT directly generates Chinese characters with delimiters. It can also be equipped with post-editing that adds back characters omitted by the model. Later, Wei et al. (2019) restrict the NMT decoding to follow all and only the input characters. This proposal, together with existing NMT toolkits, eases the model design and implementation for neural Chinese segmentation. However, even with external resources, the two systems are inferior to the previous works concerning performance. This encourages us to explore low-resource techniques to enhance the NMT-based approach.

3 Methodology

An NMT model is trained to minimize the sum of an objective function L over each target sentence $y_0^n = y_0, y_1, ..., y_n$ given a source sentence X. We list below per-character conditional cross-entropy as an example:

$$L = -\frac{1}{n}\sum_{i=1}^{n} \log P(y_i|y_0^{i-1}, X) \qquad (1)$$

Following Shi et al. (2017), we make use of character-level NMT, and add an extra delimiter token "$\langle D \rangle$" to the target vocabulary. The delimiter token on the target side implies that the previous and next words are separated. To visualize, given an unsegmented sentence in characters "我会游泳", the model should output character-by-character "我$\langle D \rangle$ 会$\langle D \rangle$ 游泳" (English: I can swim). This in reality resembles how a human would read an unsegmented Chinese sentence.

We argue that NMT can model word segmentation well because the decoder has access to the global information in both decoder and attention states. Moreover, the output segmented characters may display stronger probabilistic patterns than the position labels do, resulting in more explicit modelling of the word boundary "$\langle D \rangle$". This characteristic is also robust to out-of-vocabulary words because NMT can freely "insert" the boundary delimiter anywhere to form words. Finally, this method does not require any alteration to the model architecture.

However, it poses a challenge when CWS is approached as NMT, that NMT usually performs poorly under a low-resource condition (Koehn and Knowles, 2017), which is exactly the case of Chinese segmentation datasets. A CWS corpus provides fewer than 100k sentences, whereas a typical translation task provides data at least one order of magnitude larger. To address this issue, we apply low-resource NMT practices: regularization and data augmentation. Then, we examine several other broadly used techniques.

3.1 Hyperparameter tuning

Hyperparameter tuning is often the first step to build a machine learning model. In the field of neural translation, Sennrich and Zhang (2019) show that carefully tuning hyperparameters results in substantial improvement in low-resource scenarios. In our case, we concentrate on regularization: label smoothing, network dropout, and source token dropout (Szegedy et al., 2016; Srivastava et al., 2014; Sennrich et al., 2016a). Additionally, we switch between GRU and LSTM, and increase the model depth (Hochreiter and Schmidhuber, 1997; Cho et al., 2014).

3.2 Objective weighting

The generic NMT objective function considers the loss from each target sentence or token equally. By adjusting the objective function we can make it weigh some components more than others, in order to better learn the desired part of the training data. It can be applied at the token or sentence level, for various purposes including domain adaptation and grammatical error correction (Chen et al., 2017; Wang et al., 2017; Yan et al., 2018; Junczys-Dowmunt et al., 2018b).

We propose to put more emphasis on the delimiter token in target sentences because they correspond to word boundaries directly. We weight delimiters k times as many as other tokens in the objective function, where k can be empirically determined on a validation set. The new token-weighted objective function L_{token} is as Equation 2, where the weight coefficient $\boldsymbol{\lambda}_i = k$ if y_i is

a delimiter and $\boldsymbol{\lambda}_i = 1$ otherwise.

$$L_{token} = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\lambda_i} \log P(y_i|y_0^{i-1}, X) \quad (2)$$

3.3 Data augmentation

Data augmentation is widely adopted in NMT. The paradigm is to generate source side data from existing (monolingual) target side data (Sennrich et al., 2016b; Grundkiewicz et al., 2019), but this does not apply to CWS since there is no extra gold segmented data. Hence, we experiment with two methods that could suit CWS better: sentence splitting and unsupervised segmentation.

3.3.1 Sentence splitting

The surface texts of inputs and outputs are consistent in the NMT approach to CWS, with the only exception being the added delimiters. With a potential quality degrade, we assume that segmentation can be inferred locally, i.e. within a phrase instead of the whole sentence. It enables us to split a sentence into multiple shorter segments, with the gold segmentation unchanged. This can hugely expand the amount of training data, and reduce the input and output sequence lengths. In practice, we break full sentences down at comma and period symbols, since they are always separated from other characters.

3.3.2 Unsupervised segmentation

Both Zhao and Kit (2008)'s, and Wang et al. (2019)'s papers show that unsupervised segmentation helps supervised CWS. We use an external tool to segment our training data in an unsupervised way, to create augmented data (detailed later in Section 4.3). The data is utilized in two scenarios different from previous works: sentence-level weighting and transfer learning. The methods are depicted below.

Sentence weighting Weighting objective function at the sentence level can distinguish high- and low-quality training data. We designate our unsupervised segmentation result as low-quality augmented data, and the original training sentences as high-quality data. After combining them into a single training set, the high-quality data is weighted k times as much as the low-quality data. Equation 3 shows that sentence-weighted objective function $L_{sentence}$, where weight $\boldsymbol{\lambda} = k$ for gold sentences and 1 for augmented sentences. In contrast with Equation 2, the sentence weight is not token-dependant.

$$L_{sentence} = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\lambda} \log P(y_i|y_0^{i-1}, X) \quad (3)$$

Transfer learning It means to pre-train a model on high-resource data and then optimize it for a low-resource task. It often yields enhanced results over directly training on a small dataset (Zoph et al., 2016), as the knowledge learned from the high-resource task can be beneficial. Moreover, Aji et al. (2020) claim that starting a model from trained parameters is better than random initialization. We first train a model on the augmented data from an unsupervised segmenter, then further optimize it on the genuine training data.

3.4 Ensembling

An ensemble of diverse and independently trained and models enhances prediction. In our work, we combine models trained with different techniques and random seeds, and integrate a neural generative language model (LM) trained on the gold segmented training data. It works as follows: at each inference time step, all models' predictions are simply averaged to form the ensemble's prediction over the target vocabulary.

4 Experiments and Results

4.1 Task description

Evaluation takes place on the Microsoft Research (MSR) and Peking University (PKU) corpora in the second CWS bakeoff (Emerson, 2005).[1] The datasets are of sizes 87k and 19k, which are considered low-resource in machine translation tasks. Regarding preprocessing, our own training and validation sets are created randomly at a ratio of 99:1, from the supplied training data. We normalize characters, and convert continuous digits and Latin alphabets to "$\langle N \rangle$" and "$\langle L \rangle$" symbols without affecting segmentation.

There are both closed (constrained) and open tests in the CWS bakeoff. The former requires a system to only use the supplied data. Since we aim to strengthen the translation-based approach itself, we select the closed test condition and compare with other papers that report closed test results. The evaluation metric F1 (%) is calculated by the script from the bakeoff. We test different

[1] sighan.cs.uchicago.edu/bakeoff2005.

techniques on MSR and apply the best configurations to PKU without further tuning.

	$drop_{state}$	best loss	
$drop_{src}$ = 0	0	0.0333	
	0.1	0.0271	
	0.2	**0.0262**	✓
	0.3	0.0272	
	0.4	0.0303	

	$drop_{src}$	best loss	
$drop_{state}$ = 0.2	**0**	**0.0262**	✓
	0.15	0.2081	
	0.3	0.4496	

(a) Experiments on two dropout methods. $drop_{src}$ indicates entire source word dropout and $drop_{state}$ indicates dropout between RNN states.

	label smoothing	best loss	
$drop_{src}$= 0, $drop_{cell}$= 0.2	**0**	**0.0262**	✓
	0.1	0.1161	
	0.2	0.2220	

(b) Experiments on label smoothing.

cell	encoder depth	decoder depth	best loss	
GRU	1	1	0.0262	✓
	1	2	0.0251	
	2	1	0.0261	
	2	2	0.0264	
	3	3	0.0276	
	4	4	0.0268	
LSTM	1	1	0.0286	

(c) Experiments on model depth and the RNN type. No obvious winner is observed.

Table 1: Hyperparameter searches.

4.2 Baseline with regularization

We start with a 1-layer bi-directional GRU with attention (Bahdanau et al., 2015) containing 36M parameters. Adam (Kingma and Ba, 2015) is used to optimize for per-character (token) cross-entropy until the cost on the validation set stalls for 10 consecutive times. We set the learning rate to 10^{-4}, beam size to 6, and enable layer normalization (Ba et al., 2016). Since the model input and output share the same set of characters, we use a shared vocabulary and tied embeddings for source, target, and output layers (Press and Wolf, 2017). Training such a model on the MSR dataset takes 5 hours on a single GeForce GTX TITAN X GPU with the Marian toolkit (Junczys-Dowmunt et al., 2018a).[2]

Regarding hyperparameter selection, we always select the best settings based on the loss on the validation set. The tuning procedures are reported in Table 1. We see that a small dropout of 0.2 is helpful; source token dropout and label smoothing both cause adverse effects. Changing model depth and switching from GRU to LSTM make a negligible impact, so we stick to the single-layer GRU.

The first row in Table 4 shows that our carefully-tuned baseline achieves an F1 of 96.8% on the MSR test set. Next, we find that weighting delimiters twice as other tokens brings a 0.1% improvement. Delimiter weight tuning is presented in Table 2. These numbers already outperform previous translation-based works.

weight λ on delimiters	best loss	
1 (no weighting)	0.0262	
1.5	0.0197	
2	**0.0191**	✓
4	0.0204	
10	0.0210	
50	0.0253	

Table 2: Experiments on delimiter (word) weighting. λ is the weight on the delimiter, and other words are always given a weight of 1.

weight λ on original data	best loss	
1 (no weighting)	0.0462	
2	0.0346	
5	0.0309	
10	0.0268	
20	0.0227	
40	**0.0226**	✓
100	0.0230	
200	0.0245	
only genuine data	0.0268	

Table 3: Experiments on weighting augmented and original data. λ represents the weight on original sentences; augmented data always have a weight of 1.

4.3 Leveraging augmented data

Sentence splitting is done on both sides of the training and validation sets. Test sentences are

[2] https://github.com/marian-nmt/marian.

Techniques	F1 (%)
baseline w/ regularization (base)	96.8
base + delimiter weight	96.9
base + sentence splitting (split)	97.1
base + split + unsupervised + transfer	97.1
base + split + unsupervised + weight	**97.3**
2 × baseline	97.2
2 × transfer + 2 × weight + LM	**97.6**

Table 4: F1 of our techniques on MSR test set.

split, segmented by the model, and then concatenated, ensuring a consistent evaluation outcome. This leads to a better F1 of 97.1%, thanks to a 3-fold increase in data size to 257k for MSR.

We employ the segmental language model (Sun and Deng, 2018) for unsupervised segmentation.[3] We used the MSR model optimized on the training, validation, and test sets with a maximum word length of 4. Since the system is fully unsupervised, it is fair to include the test set; yet we only apply it to our training split to generate augmented data. In this way, no external resource is introduced. While transfer learning brings no gain, sentence-level weighting lifts the overall score to 97.3%, as shown in Table 4. We see that the cost on the validation set improves, and then degrades as sentence weight gets larger; the best sentence weight is determined to be 40 for MSR. The detailed weight selection process is described in Table 3.

4.4 Ensembling

During decoding, all models' predictions are averaged to produce an output token at each step. We first test an ensemble consisting of two baselines. Next, we combine two transfer learning models, two sentence-weighting models, and a character RNN LM. The LM has the same architecture as our NMT decoder. It is optimized for perplexity on the segmented side of the train set. Ensembling is done in one shot without tuning weights and it achieves the highest F1 of 97.6%.

5 Results and Analysis

In addition to MSR test, we keep the best hyperparameters determined on the MSR corpus unchanged, and run the same set of experiments on the PKU dataset.

[3]Their code and released models: github.com/edward-sun/slm.

System		MSR	PKU
non-neural	Zhao and Kit, 2008	**97.6**	95.4
	Zhang and Clark, 2011	97.3	94.4
neural	Pei et al., 2014	94.4	93.5
	Cai and Zhao, 2016	96.4	95.2
	Wang and Xu, 2017	96.7	94.7
	Cai et al., 2017	97.0	95.4
	Zhou et al., 2017	97.2	95.0
	Ma et al., 2018	**97.5**	95.4
	Wang et al., 2019	97.4	**95.7**
	Duan and Zhao, 2020	**97.6**	95.5
NMT-based	Shi et al., 2017	94.1	87.8
	+ external resources†	96.2	95.0
	Wei et al., 2019†	94.4	92.0
	Our best single model	97.3	95.0
	Our best ensemble‡	**97.6**	95.4

† The results are advantaged as extra resources are used.
‡ 97.61±0.16 on MSR and 95.43±0.38 on PKU, with $p < 0.05$ using bootstrapping (Ma et al., 2018), detailed in Appendix A.

Table 5: Previous and our systems' F1 (%) on MSR and PKU corpora under the constrained condition.

Table 5 compares our MSR and PKU results with previous papers. Our best single models are remarkably ahead of other NMT-based methods. With ensembling, our result on MSR ties with state of the art, showing that empirically neural methods can reach the top without external data. However, as data size significantly drops in the case of PKU, we observe a declined performance and larger variance on the PKU dataset. This is expected as NMT is known to be sensitive to a smaller data size.

Regarding regularization, we discover that low-resource NMT techniques are not always constructive for CWS. Dropping out source tokens is harmful because CWS is not a language generation task and the decoder output heavily relies on the input. A similar rationale explains why label smoothing causes rocketing cross-entropy: unlike language generation where a variety of outputs are accepted, for CWS there is always just one single correct scheme. Smoothing out the decoder probability distribution results in confusion.

Further, unsupervised data augmentation with weighting achieves the best single-model result. We suggest a possible reason: the augmented data has the same source side as the original data, but a noisier target side. When weighted appropriately, the noise might act as a smoothing tech-

nique for sequence modelling, especially in the low-resource condition (Xie et al., 2017). From the transfer learning aspect, pre-training on the augmented data does not lead to a higher number than starting from a randomly initialized state.

6 Conclusion

Our low-resource translation-based approach to Chinese word segmentation achieves strong performance and is easy to adopt. Data augmentation, objective weighting and ensembling are the most favourable. In future, it is worth extending this perspective to word segmentation of other languages, as well as re-basing it on Transformer models.

Acknowledgements

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The work reflects the view of the authors and not necessarily that of the funders.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *ACL*, pages 7701–7710.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *ACL*, pages 409–420.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for Chinese. In *ACL*, pages 608–615.

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015a. Gated recursive neural network for Chinese word segmentation. In *ACL-IJCNLP*, pages 1744–1753.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015b. Long short-term memory neural networks for Chinese word segmentation. In *EMNLP*, pages 1197–1206.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.

Sufeng Duan and Hai Zhao. 2020. Attention is all you need for Chinese word segmentation. In *EMNLP*, pages 3862–3872.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018a. Marian: Fast neural machine translation in C++. In *ACL*, pages 116–121.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018b. Approaching neural grammatical error correction as a low-resource machine translation task. In *NAACL*, pages 595–606.

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

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. 2016. Exploring segment representations for neural segmentation models. In *IJCAI*, pages 2880–2886.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art Chinese word segmentation with bi-LSTMs. In *EMNLP*, pages 4902–4908.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *EMNLP*, pages 277–284.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *ACL*, pages 293–303.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING*, pages 562–568.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *EACL*, pages 157–163.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *WMT*, pages 371–376.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *ACL*, pages 86–96.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *ACL*, pages 211–221.

Xuewen Shi, Heyan Huang, Ping Jian, Yuhang Guo, Xiaochi Wei, and Yi-Kun Tang. 2017. Neural chinese word segmentation as sequence to sequence translation. In *Chinese National Conference on Social Media Processing*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(56):1929–1958.

Zhiqing Sun and Zhi-Hong Deng. 2018. Unsupervised neural word segmentation for Chinese via segmental language modeling. In *EMNLP*, pages 4915–4920.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826.

Chunqi Wang and Bo Xu. 2017. Convolutional neural network with word embeddings for Chinese word segmentation. In *IJCNLP*, pages 163–172.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *EMNLP*, pages 1482–1488.

Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019. Unsupervised learning helps supervised neural word segmentation. In *AAAI*, pages 7200–7207.

Yuekun Wei, Binbin Qu, Nan Hu, and Liu Han. 2019. An improved method of applying a machine translation model to a chinese word segmentation task. In *ICANN*, pages 44–54.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *ICLR*.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*.

Shen Yan, Leonard Dahlmann, Pavel Petrushkov, Sanjika Hewavitharana, and Shahram Khadivi. 2018. Word-based domain adaptation for neural machine translation. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 31–38.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *ACL*, pages 839–849.

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *ACL*, pages 840–847.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.

Hai Zhao, Deng Cai, Changning Huang, and Chunyu Kit. 2018. Chinese word segmentation: Another decade review (2007-2017). In *Frontiers of Empirical and Corpus Linguistics*, pages 139–162.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *EMNLP*, pages 647–657.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2017. Word-context character embeddings for Chinese word segmentation. In *EMNLP*, pages 760–766.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *EMNLP*, pages 1568–1575.

A Results with a confidence interval

We report our final score with a confidence interval since the top results are very close. As there is only one test set, we create another 599 test sets of the same size as the original one, through resampling with replacement. Our best system obtains an F1 of 97.61 ± 0.16 on the MSR dataset and 95.43 ± 0.38 on the PKU dataset with 95% confidence (2 standard deviations).