

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

Pinzhen Chen Kenneth Heafield

School of Informatics, University of Edinburgh

10 Crichton Street, Edinburgh EH8 9AB

{pinzhen.chen, kenneth.heafield}@ed.ac.uk

Abstract

Supervised Chinese word segmentation has entered the deep learning era which reduces the hassle of feature engineering. Recently, some researchers attempted to treat it as character-level translation which further simplified model designing and building, but there is still a performance gap between the translation-based approach and other methods. In this work, we apply the best practices from low-resource neural machine translation to Chinese word segmentation. We build encoder-decoder models with attention, and examine a series of techniques including regularization, data augmentation, objective weighting, transfer learning and ensembling. Our method is generic for word segmentation, without the need for feature engineering or model implementation. In the closed test with constrained data, our method ties with the state of the art on the MSR dataset and is comparable to other methods on the PKU dataset.

1 Introduction

Written Chinese has no explicit word boundary, so Chinese word segmentation (CWS) serves as an upstream disambiguation step for Chinese language processing. The task is often viewed as sequence labeling, where each character receives a label indicating its relative position in a segmented sentence. While traditional machine learning methods have attained strong results, recent research focuses on neural networks. 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 (SOTA) from Zhao and Kit (2008), and the NMT method is even behind.

We note two advantages of NMT: the entire sentence is encoded before making any decision, and the model jointly trains character embeddings with sequence modeling. Thus, we try to bridge the

gap between the NMT approach and SOTA, using low-resource techniques such as regularization and data augmentation. Then, we explore more techniques commonly seen in NMT. The translation-based method is easy to adopt without the hassle of feature engineering and model design. In the constrained test condition, our method reaches the top on the MSR dataset and achieves a strong result on the PKU dataset without tuning. As a generic approach it can also be used for other languages.

2 Related Work

CWS is often tackled as sequence labeling, where each input character is assigned a label showing how it is positioned relative to neighboring words (Xue, 2003). Most traditional approaches rely on conditional random fields or maximum entropy Markov models (Peng et al., 2004; Ng and Low, 2004). Zhao and Kit (2008) leverage unsupervised features to attain state-of-the-art results in the closed track, which we tie.

Recent research shifted towards neural networks: feed-forward, recursive and convolutional (Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015a,b; Wang and Xu, 2017). Without external data, these do not surpass the best non-neural method, but they greatly reduce the hassle of feature engineering. Better representations for segments and characters, and incorporation of external data are studied too (Liu et al., 2016; Zhou et al., 2017; Yang et al., 2017). By tuning model configurations, Ma et al. (2018) achieve strong results. CWS can also be done through learning to score global word segmentation given characters (Zhang and Clark, 2007, 2011; Cai and Zhao, 2016; Cai et al., 2017). On top of this, Wang et al. (2019) prove that it is beneficial to integrate unsupervised segmentation. A concurrent work, which uses a modified Transformer for sequence tagging, achieves comparable results to SOTA too, but the model design is more complicated than ours (Duan and Zhao, 2020).

The most relevant to our work is Shi et al. (2017)’s proposal to formalize CWS as character-level NMT. It differs from global segmentation scoring in that the NMT directly generates characters segmented by delimiters. Afterward, Wei et al. (2019) constrain the NMT decoding to follow all and only the input characters. This approach, together with several existing NMT toolkits, ease the model design and implementation for neural CWS. However, even with external resources, the two methods are behind the previous work concerning performance. This motivates 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 $y_0^n = y_0, y_1, \dots, y_n$ given a source sentence X . We use per-character conditional cross-entropy here:

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

Following Shi et al. (2017), we use character-level NMT and add an extra delimiter token “ $\langle D \rangle$ ” to the target vocabulary. The delimiter token in a target sentence implies that its previous and next words are separated. For instance, given an unsegmented sentence “我会游泳”, the model will output “我 $\langle D \rangle$ 会 $\langle D \rangle$ 游泳” (English: I can swim).

We argue that NMT can model word segmentation well because the decoder has access to the global information in decoder and attention states. Moreover, the output segmented characters may display stronger probabilistic patterns than position labels do, resulting in more explicit modeling 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 model design or implementation.

However, CWS poses a challenge when approached as NMT: insufficient data (Koehn and Knowles, 2017). A CWS corpus provides fewer than 100k sentences, whereas a translation task provides more than a million. To address this issue, we apply low-resource NMT techniques: 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 model. Sennrich and Zhang (2019) show that carefully tuning hyperparameters results in substantial improvement for low-resource NMT. In our case, we focus on tuning regularization techniques: label smoothing, network dropout and source token dropout (Szegedy et al., 2016; Srivastava et al., 2014; Sennrich et al., 2016a). Additionally, we try both GRU and LSTM, and increase model depth (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Miceli Barone et al., 2017).

3.2 Objective weighting

The generic NMT objective function considers the cost from each target token equally. By modifying the objective function we can make it weight some components more, in order to better learn the desired part of the training data. It can be applied at 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 try 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 token 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 1, where weight coefficient $\lambda_i = k$ if y_i is a delimiter and $\lambda_i = 1$ otherwise.

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

3.3 Data augmentation

Data augmentation is widely adopted in NMT. The paradigm is to generate source data from existing target 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 try two methods that 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. We assume that segmentation can be inferred locally, i.e. within a phrase instead of the whole sentence. It allows us to split a sentence into multiple shorter segments,

with the gold segmentation unchanged. This can increase training data size and reduce the sequence length to be modeled. In practice, we break down sentences at comma and period symbols since they are always separated from other words.

3.3.2 Unsupervised segmentation

Zhao and Kit (2008) and Wang et al. (2019) show that unsupervised segmentation helps supervised learning. 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 work: sentence-level weighting and transfer learning. The methods are depicted below.

Sentence weighting: weighting objective function at 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 as a single training set, the high-quality data is weighted k times as many as the low-quality data. Equation 2 shows that sentence-weighted objective function $L_{sentence}$, where weight $\lambda = k$ for gold sentences and 1 for augmented sentences.

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

Transfer learning: it means to train a model on high-resource data then optimize it for a low-resource task. It yields enhanced results over directly training on a small dataset (Zoph et al., 2016). Aji et al. (2020) claim that starting 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 original training data.

3.4 Ensembling

An ensemble of independently trained and diverse models improves 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 training data. It works as follows: at each time step, all models’ predictions are simply averaged to form the ensemble’s prediction.

4 Experiments and Results

4.1 Task, data and evaluation metric

We evaluate 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. Regarding preprocessing, train and valid sets are created randomly at a 99:1 ratio. We normalize characters, and convert continuous digits and Latin alphabets to “⟨N⟩” and “⟨L⟩” without affecting segmentation.

There are both closed 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 focus on the closed test and compare with other methods that report closed test results. The evaluation metric F1 (%) is calculated by the script from the bakeoff. We test different techniques on MSR and apply the best configurations to PKU without further tuning.

4.2 Baseline with regularization

We start with a 1-layer bi-directional GRU with attention (Bahdanau et al., 2015), comprising 36M parameters. Adam (Kingma and Ba, 2015) is used to optimise for per-character (token) cross-entropy until the cost on valid set stalls for 10 consecutive times. We set the learning rate to 10^{-4} , beam size to 6, and enable layer normalisation (Ba et al., 2016). Since model input and output share the same set of characters, we use a shared vocabulary and tied-embeddings for source, target and output (Press and Wolf, 2017). With mini-batch size 6000 tokens, training on MSR corpus takes 5 hours on a GeForce GTX TITAN X GPU using Marian toolkit (Junczys-Dowmunt et al., 2018a).²

Regarding hyperparameter tuning, we always select the best settings based on per-character cross-entropy on the valid set. The tuning procedures are reported in Table 3, 4 and 5 in Appendix A. We see that a small dropout of 0.2 is helpful. However, randomly dropping out source tokens and label smoothing both cause an adverse effect. From the experiments on model depths and cell types, changing model depth and switching from GRU to LSTM make a negligible impact, so we stick to the single-layer GRU architecture.

Table 1 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

¹sighan.cs.uchicago.edu/bakeoff2005.

²It supports all proposed methods: [marian-nmt.github.io](https://github.com/marian-nmt).

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 1: F1 of our techniques on MSR test set.

tokens brings a 0.1% increase. Delimiter weight tuning is presented in Table 6 in Appendix B.

4.3 Leveraging augmented data

Sentence splitting is done on both sides of train and valid sets. Test sentences are split, segmented by the model and then concatenated, ensuring a genuine 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 adopt the segmental language model (Sun and Deng, 2018) for unsupervised segmentation.³ We download their MSR model optimised on the task’s train, valid and test sets with a maximum word length 4, but only apply it on our train set to get augmented data. In this way, no external data is introduced. While transfer learning brings no gain, sentence-level weighting lifts the overall score to 97.3%, as shown in Table 1. The best sentence weight is 40 for MSR, with details reported in Table 7 in Appendix B.

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 Analysis

After the best configurations are determined on MSR, we test on the PKU dataset. Table 2 compares our results with previous work. Our best

³Code and released models: github.com/edward-sun/slm.

single models are markedly ahead of other NMT-based methods. With ensembling, our MSR result is tied with SOTA, showing that empirically neural methods can reach the top without external data. An advantage of our approach is that neither feature engineering nor model design is required. However, as data quantity drops, we observe a worse performance and larger variance on the PKU dataset. This is expected as NMT is known to need a huge amount of data.

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: there is always just one single correct output, so smoothing out the decoder probability distribution is undesired.

Methods		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 C.

Table 2: F1 of previous and our work on MSR and PKU corpora from CWS bakeoff 2005 under closed test.

Further, unsupervised data augmentation with weighting achieves the best single-model result. We suggest a reason: the augmented data has the same source side as the original data, but a noisier target side. When weighted appropriately, the noise acts as a smoothing technique for sequence modeling (Xie et al., 2017). From another aspect, transfer learning from augmented data does not lead to a better result than training from a randomly initial-

ized state, which contradicts with Aji et al. (2020)'s findings.

6 Conclusion

Our low-resource translation approach to Chinese segmentation achieves strong performance and is easy to adopt. Data augmentation, objective weighting and ensembling are the most beneficial. In future, it is worth extending this to other languages.

References

- Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. [In neural machine translation, what does transfer learning transfer?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. [Layer normalization](#). In *NIPS 2016 Deep Learning Symposium*, Barcelona, Spain.
- 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*.
- Deng Cai and Hai Zhao. 2016. [Neural word segmentation learning for Chinese](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.
- Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. [Fast and accurate neural word segmentation for Chinese](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 608–615, Vancouver, Canada. Association for Computational Linguistics.
- 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, Vancouver. Association for Computational Linguistics.
- Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015a. [Gated recursive neural network for Chinese word segmentation](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1744–1753, Beijing, China. Association for Computational Linguistics.
- Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015b. [Long short-term memory neural networks for Chinese word segmentation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal. Association for Computational Linguistics.
- 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 *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Sufeng Duan and Hai Zhao. 2020. [Attention is all you need for Chinese word segmentation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3862–3872, Online. Association for Computational Linguistics.
- 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, Florence, Italy. Association for Computational Linguistics.
- 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 *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018b. [Approaching neural grammatical error correction as a low-resource machine translation task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Lei Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations*,

- ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*
- 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, Vancouver. Association for Computational Linguistics.
- Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. 2016. [Exploring segment representations for neural segmentation models](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, page 2880–2886. AAAI Press.
- Ji Ma, Kuzman Ganchev, and David Weiss. 2018. [State-of-the-art Chinese word segmentation with bi-LSTMs](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908, Brussels, Belgium. Association for Computational Linguistics.
- Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. [Deep architectures for neural machine translation](#). In *Proceedings of the Second Conference on Machine Translation*, pages 99–107, Copenhagen, Denmark. Association for Computational Linguistics.
- Howe 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 *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 277–284, Barcelona, Spain. Association for Computational Linguistics.
- Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. [Max-margin tensor neural network for chinese word segmentation](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland. Association for Computational Linguistics.
- Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. [Chinese segmentation and new word detection using conditional random fields](#). In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 562–568, Geneva, Switzerland. COLING.
- Ofir Press and Lior Wolf. 2017. [Using the output embedding to improve language models](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. [Edinburgh neural machine translation systems for WMT 16](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. [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 and Biao Zhang. 2019. [Revisiting low-resource neural machine translation: A case study](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.
- Xuwen Shi, Heyan Huang, Ping Jian, Yuhang Guo, Xiaochi Wei, and Yi-Kun Tang. 2017. [Neural chinese word segmentation as sequence to sequence translation](#). In *Social Media Processing*, pages 91–103. Springer Singapore.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research*, 15(56):1929–1958.
- Zhiqing Sun and Zhi-Hong Deng. 2018. [Unsupervised neural word segmentation for Chinese via segmental language modeling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4915–4920, Brussels, Belgium. Association for Computational Linguistics.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. [Rethinking the inception architecture for computer vision](#). *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chunqi Wang and Bo Xu. 2017. [Convolutional neural network with word embeddings for Chinese word segmentation](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 163–172, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. [Instance weighting for neural machine translation domain adaptation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark. Association for Computational Linguistics.
- Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019. [Unsupervised learning helps supervised neural word segmentation](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 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

- Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, pages 44–54. Springer International Publishing.
- 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 *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.
- Nianwen Xue. 2003. [Chinese word segmentation as character tagging](#). In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, pages 29–48.
- 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 Workshop on Spoken Language Translation*, pages 31–38, Bruges, Belgium.
- Jie Yang, Yue Zhang, and Fei Dong. 2017. [Neural word segmentation with rich pretraining](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, Vancouver, Canada. Association for Computational Linguistics.
- Yue Zhang and Stephen Clark. 2007. [Chinese segmentation with a word-based perceptron algorithm](#). In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 840–847, Prague, Czech Republic. Association for Computational Linguistics.
- 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, Beijing. China Social Sciences Press. Language: Chinese.
- 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 *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.
- Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2017. [Word-context character embeddings for Chinese word segmentation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 760–766, Copenhagen, Denmark. Association for Computational Linguistics.
- Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. [Transfer learning for low-resource neural machine translation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Hyperparameter Tuning Details

	D_{state}	best cost	
$D_{\text{src}} = 0$	0	0.0333	✓
	0.1	0.0271	
	0.2	0.0262	
	0.3	0.0272	
	0.4	0.0303	

	D_{src}	best cost	
$D_{\text{state}} = 0.2$	0	0.0262	✓
	0.15	0.2081	
	0.3	0.4496	

Table 3: Experiments on two dropout methods. D_{src} indicates entire source word dropout and D_{state} indicates dropout between RNN states.

	label smoothing	best cost	
$D_{\text{src}} = 0,$ $D_{\text{cell}} = 0.2$	0	0.0262	✓
	0.1	0.1161	
	0.2	0.2220	

Table 4: Experiments on label smoothing.

cell type	encoder depth	decoder depth	best cost	
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	

Table 5: Experiments on model depth and RNN cell.

B Weight Tuning Details

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

Table 6: 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 cost	
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 original data	0.0268	

Table 7: Experiments on sentence weighting of augmented and original data. λ is the weight on original sentences, and augmented sentences are always given a weight of 1.

C 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 additional 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).