Chinese Word Segmentation with World Knowledge

Tom ma
8tom147@gmail.com

Abstract

Recent years deep learning has achieved great success for chinese word segmentation(CWS). And we human beings can do chinese word segmentation(CWS) well because we already have an language model(LM) and dictionary with word frequency in mind. In this paper we prpopose to use world knowledge(language model and dictionary with word frequency) in deep leaning framwork for CWS. Our method achieved new state-of-the-art result of SIGHAN 2005 bakeof.[1]

1 Introduction

Unlike english, chinese do not have explict word boundaries, so CWS is an essential task for chinese NLP tasks such as NER. Generally speaking we can divide CWS methods into unsupervised and supervised. The unsupervised methods include Mutual Information (Chang and Lin, 2003), normalized Variation of Branching Entropy(Magistry and Sagot, 2012), Minimum Description Length(Magistry and Sagot, 2013), Hidden Markov Model(Chen et al., 2014) and Nested Pitman-Yor Process(Mochihashi et al., 2009). And recently A generative unsupervised language model method (Sun and Deng, 2018) achieved new state-of-the-art.

For supervised methods there are two mainlines. The first is word level CWS which assign score to each segmentation of the sentence. (Zhang et al., 2016) proposed an transition-based model which incorporate word and character feature in sliddding window. (Cai and Zhao, 2016) proposed an LSTM scoring model which can catch the sengmentation history information. They further improve their model in (Cai et al., 2017) by greedy search and an new word representation network. Another mainline is char level CWS which treat the task as an sequence label problem. These models like Maximum Entropy (Low et al., 2005) and Conditional Random Fields(CRF)((Peng et al., 2004) (Zhao et al., 2006))rely on heavy hand-crafted features. In (Zheng et al., 2013) a deep learning framework was proposed to release heavy feature engineering. And (Chen et al., 2015) extended LSTM to explicitly model previously important information in memory cells to perform the task.

There are also models which try to incoporate external knowledge to improve the performance. (Qian and Liu, 2012), (Liu et al., 2014) and (Zhang et al., 2018) investigated how to incoporate dictionary in their models. Although these models has proved the effectiveness of dictionary knowledge they did not incoporate the word frequency knowledge. In (Yang et al., 2017) they exploited richer sources of external information by pretraining character and word embeddings to improve performance.

In this paper we use CRF based sequence label model to incoporate world knowledge(an language model and dictioary with word frequency). Our model is most similary with (Zhang et al., 2018).

2 World Knowledge

We human beings can do the CWS well because we have world knowledge such as language model and dictionary. For dictinary knowledge we follow (Zhang et al., 2018), for a given sentence $\mathbf{x} = (x_1, x_2, ..., x_n)$ we construct a feature vector $\mathbf{v_i}$ for each x_i based on the dictionary D and the n-gram context. If $n = 3$ the 3-grams for x_i is $(x_{i-2}x_{i-1}x_i, x_{i-1}x_i, x_ix_{i+1}, x_ix_{i+1}x_{i+2})$. Then for each n-gram we can get its frequency from the dictionary D, if the n-gram is not in the dictinary the frequency is 0. Then we use the logarithm of

[1] Our implementation can be found at `https://github.com/amxineohp/cwswk`

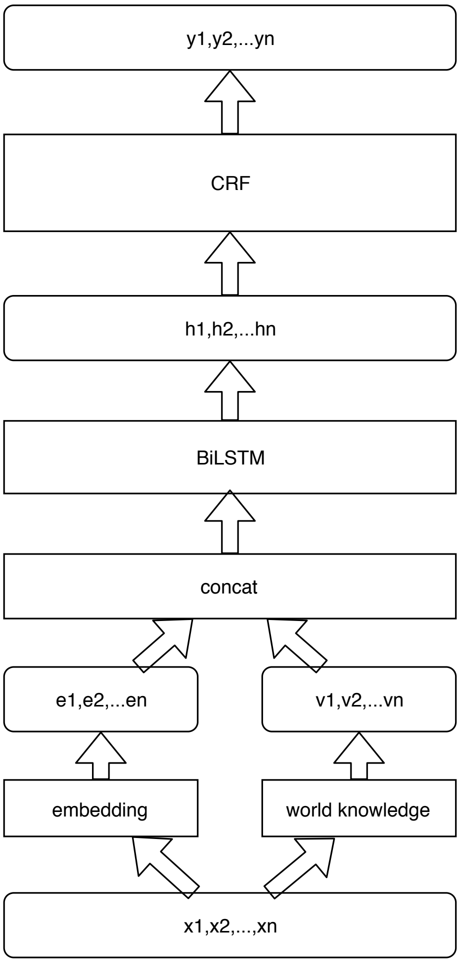

Figure 1: our proposed BiLSTM CRF model incoporated world knowledge

the frequency vectory $\mathbf{f_i}$ as the final feature vector $\mathbf{v_i} = log(1 + \mathbf{f_i})$.

Recently pretrainning an language model from exteranl dataset has been proved to be very usefully for many NLP tasks, see (Peters et al., 2018), BERT(Devlin et al., 2018) for details. In this paper we use BERT to extract the feature of each character in a sentence.

3 BiLSTM CRF Model

We treat CWS as an sequence label task. For a given sentence x, we need to label each character x_i as one of the tags B, M, E, S indicating beginning,middle,end of a word or a single word. In this paper we use BiLSTM and CRF as the backbone of our model.

Let $\mathbf{v} = [v_1, v_2, ..., v_n]$ be the corresponding feature inputs of an sequence $\mathbf{x} = (x_1, x_2, ..., x_n)$, an LSTM neural net will canculate the input, forget, output gate and cell memory for time step t as below:

$$
\begin{aligned}
\mathbf{i_t} &= sigmoid(\mathbf{W_i h_{t-1}} + \mathbf{U_i v_t} + \mathbf{b_i}) \\
\mathbf{f_t} &= sigmoid(\mathbf{W_f h_{t-1}} + \mathbf{U_f v_t} + \mathbf{b_f}) \\
\mathbf{o_t} &= sigmoid(\mathbf{W_o h_{t-1}} + \mathbf{U_o v_t} + \mathbf{b_o}) \\
\mathbf{\hat{c}_t} &= tanh(\mathbf{W_c h_{t-1}} + \mathbf{U_c v_t} + \mathbf{b_c}) \\
\mathbf{c_t} &= \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{c_{t-1}} \\
\mathbf{h_t} &= \mathbf{o_t} \odot tanh(\mathbf{c_t})
\end{aligned}
$$

$\odot$ is the elementwise multiplication. $\mathbf{h_t}$ only have past information so in order to catch information from future we use bi-directionaal LSTM. The hidden state $\mathbf{h_t}$ is:

$$\mathbf{h_t} = \overrightarrow{h_t} \oplus \overleftarrow{h_t}$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are hidden states of forward and backword LSTM at time t. $\oplus$ represent the concat operation. Although we can use $\mathbf{h_t}$ to do the classification for tags B,M,E,S of each character, it will ignore the fact that a M label can not be followed by a B label. Therefor it comes to the condition random field(CRF). For a sequence $\mathbf{x} = (x_1, x_2, ..., x_n)$ and the corresponding tag sequence $\mathbf{y} = (y_1, y_2, ..., y_n)$, CRF will calcuate an score as:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} (\mathbf{T}_{y_{t-1} y_t} + \mathbf{P}_{y_t}),$$

where $\mathbf{T}$ is the trainable transition score matrix and $\mathbf{T}_{ij}$ is the score from tag i to j. $\mathbf{P}_{y_t}$ is the score of y_t tag of x_t which is calculated as:

$$\mathbf{P}_{y_t} = \mathbf{W}_s \mathbf{h}_t + \mathbf{b}_t,$$

where $\mathbf{W_s} \in \mathbb{R}^{|T| \times d_h}$, and $b_t \in \mathbb{R}^{|T|}$ are trainable variables. Then CRF layer will calculate the probability of the tag sequence as:

$$p(\hat{\mathbf{y}}|\mathbf{x}) = \frac{e^{s(\mathbf{x},\mathbf{y})}}{\sum_{\hat{\mathbf{y}} \in \mathbf{Y}} e^{s(\mathbf{x},\hat{\mathbf{y}})}},$$

where $\mathbf{Y}$ is the set of all posssible tag sequence of $\mathbf{x}$. During training, we use the maximum likelihood estimation to maximize the log-probabilty:

$$Loss = -\sum_{i=1}^{N} log(p(\mathbf{y_i}|\mathbf{x_i})),$$

where $\mathbf{x_i}, \mathbf{y_i}$ is the ith training example and tag sequence. When predicting, the highest scoring tag sequence will be picked:

$$y = \operatorname*{argmax}_{\hat{\mathbf{y}} \in \mathbf{Y}} s(\mathbf{x}, \hat{\mathbf{y}})$$

Details of our model is shown in Figure 1. The world knowledge layer will extract the bert feature and dictionary feature for each character and concatenate them.

4 Experiments

4.1 Datasets

We use the SIGHAN2005 dataset(PKU, MSR, AS, CITYU) to evaluate our model. The first 90% lines of the oringal training dataset was used for training and the left was used for validation. Among them AS and CITYU datasets are transfered from traditional chinese to simplify chinese. Following with previous research we substutite continus english in a word to a single chracter X and digits to 0. In (Zhang et al., 2018) they use pretrained charcter embedding and an idiom dict to replace idioms in the dataset to character I, in order to compare with them we do the same.

For dictionary knowledge we use the dictionary sourced from jieba [2] which is one of the most popular outsourced CWS tool. For the language model knowledge, we use the pretrained bert-base model [3] released by google which use the chinese wiki dump as the training dataset.

4.2 Training and results

We trained three models CWSD which incoporate only LM knowledge, CWSB which use only dictinoary knowledge and CWSBD which use the both. Table 1 compares our results with previous models. We can see even with only dictinoary knowledge, our method is comparable with (Zhang et al., 2018) which used more complex model architecture.

During training we do early stop based on the loss on validation dataset. All the hyper-parameters used in the three models are the same except for AS which use a different learning rate 1e-4 [4]. We linearly project the bert features to a lower dimension. Hyper-parameter details are listed in table 2.

Model	PKU	MSR	AS	CITYU
(Zhang et al., 2016)	95.7	97.7	-	-
(Cai et al., 2017)	95.8	97.1	-	-
(Chen et al., 2015)	96.0	96.6	-	-
(Yang et al., 2017)	96.3	97.5	95.7	96.9
(Zhang et al., 2018)	96.5	97.8	95.9	96.3
CWSD	*96.5*	*97.8*	95.7	96.4
CWSB	*96.6*	97.5	*96.6*	*97.3*
CWSBD	**97.2**	**97.9**	**96.6**	**97.5**

Table 1: F1 score results on SIGHAN 2005 bake-off datasets with previous models. Bold mean the best, italic means equal or better then previous

embedding dim	100
LSTM hidden dim	64
learning rate	1e-2
bert projecting dim	64
l2 decay	1e-4
dropout	0.2
batch size	128
gradient clipping	5

Table 2: Hyper-parameters

5 Related work

(Chang et al., 2008) incorporated external dictioanry for CWS to improve machine translation performance. (Liu et al., 2014) use dictioanry to get partial annotation for their CWS model. (Chen et al., 2015) use idom dictionary to substitute chinese idioms in dataset to a special character. (Yang et al., 2017) pretrain word embeddings based on external information. (Chen et al., 2017) use adversarial learning to incoporate knowldge from different segmentation criteria. (Zhang et al., 2018) extract ngrams of the surrounding text of a character and use 0, 1 to indicate if the ngram is in a dictionary. In our study we incoporated the word frequency and LM knowledge.

6 Conclusion

In this paper we investigated the method to incoporate world knowledge(LM and dictionary) to CWS task. Even with only dictinoary knowledge our model is comparable with the state-of-the-art models. If incoporate both knowledge our model is the best on all SIGHAN2005 dataset.

Acknowledgments

I need say thanks to the opening of the deep leaning community, thanks for all the people who published their papers and outsource their code.

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

[3] https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

[4] we did not do much of hyper-parameter tuning because we do not want the improvement was caused by hyper-parameter tuning

References

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for chinese. *arXiv preprint arXiv:1606.04300*.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for chinese. *arXiv preprint arXiv:1704.07047*.

Jason S Chang and Tracy Lin. 2003. Unsupervised word segmentation without dictionary. *ROCLING 2003 Poster Papers*, pages 355–359.

Pi-Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*, pages 224–232. Association for Computational Linguistics.

Miaohong Chen, Baobao Chang, and Wenzhe Pei. 2014. A joint model for unsupervised chinese word segmentation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 854–863.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. 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.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. *arXiv preprint arXiv:1704.07556*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for crf-based chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 864–874.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Pierre Magistry and Benoît Sagot. 2012. Unsupervized word segmentation: the case for mandarin chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 383–387. Association for Computational Linguistics.

Pierre Magistry and Benoît Sagot. 2013. Can mdl improve unsupervised chinese word segmentation? In *Sixth International Joint Conference on Natural Language Processing: Sighan workshop*, page 2.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 100–108. Association for Computational Linguistics.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, page 562. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Xian Qian and Yang Liu. 2012. Joint chinese word segmentation, pos tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 501–511. Association for Computational Linguistics.

Zhiqing Sun and Zhi-Hong Deng. 2018. Unsupervised neural word segmentation for chinese via segmental language modeling. *arXiv preprint arXiv:1810.03167*.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. *arXiv preprint arXiv:1704.08960*.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 421–431.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *AAAI*.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165.

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.