A Resource-independent Word Segmentation Toolkit

Jingjing Xu
MOE Key Laboratory of Computational Linguistics, Peking University
School of Electronics Engineering and Computer Science, Peking University
{jingjingxu}@pku.edu.cn

1 Overview

This toolkit is a word segmentation framework on top of c# that aims to provide an efficient way of implementing deep neural networks. It can be applied to a variety of sequence labelling tasks, such as POS tagging, chunking and so on.

We release two versions of the code in this toolkit. If provided with high-resource datasets, you can directly run our simple version which is composed of the basic neural network. If provided with low-resource datasets, you can run our low-resource version which incorporates some transfer learning technologies.

Main features:

- This tooklit is developed with C#. Traditional neural networks are usually developed with Python. We supply the version of C# to help developers build word segmentation systems.
- This toolkit achieves competitive results on MSR, PKU and CTB (8.0) datasets.
- To speed up training, we explore mini-batch asynchronous parallel learning on neural word segmentation.

2 Installation

Need C# and java compilers.

3 Format of Data Files

The sample train/test files are given in our toolkit for illustrating the format of data files. As for how to create these train/test files, we also give the detailed codes. Here is the detailed introduction about data files.

4 How to Run

For training: Run ~\LSTM\bin\x64\Release\LSTM.exe and choose "training". ~ represents a root directory. The model reads the last saved model on default. If you want train the model from randomly initialized weights, set "isread=false" in global.cs. The model is saved every iteration. If restarted, the training would resume from the last saved model. For low-resource datasets, we use pre-training method to leverage high-resource corpora. For example, we choose MSRA as a high-resource dataset and PKU as a low-resource dataset. You can first run our code on MSRA dataset and then copy the last saved model to train the model on PKU dataset.

For testing: Run ~\LSTM\bin\x64\Release\LSTM.exe and choose "testing".

5 About Output Files

- log.txt records detailed training information of each iteration.
- answer.txt is generated by the testing mode. It records predicted tags for test data.

6 How to Evaluate on Test Data

We provide two methods for evaluating predicted results on test data. On one hand, we implement the evaluation method, F-score, in the toolkit. On the other hand, the standard F-score evaluation script is also provided. If you want to use the standard evaluation scripts, just run "post-processing.java" and make sure that you put it in the same directory with the input file, "answer.txt" which is produced by the testing mode. The code produces the standard predicted file which you can use it to run evaluation scripts.

7 Code Structure

- Global.cs contains several hyperparameter settings, such as hidden dimension, minibatch size and so on.
- Main.cs is the program entry file.
- Matrix.cs defines the basic matrix operations.
- Datasequence.cs transforms the input files to data structures.
- F-score.cs implements the evaluation method.
- GRNN.cs defines the gated recursive neural network layer.
- LSTM.cs defines the long short term memory layer.
- ForwardBackwardProp.cs defines several gradient operations.
- Losssoftmax.cs defines several kinds of loss functions.
- RunThread.cs is used to train the model in parallel.

8 Related Work

Chinese word segmentation (CWS) is an important step in Chinese natural language processing. The most widely used approaches treat CWS as a sequence labelling problem in which each character is assigned with a tag. Many exsiting techniques, such as conditional random fields, have been successfully applied to CWS (Lafferty et al., 2001; Xue and Shen, 2003; Peng et al., 2004; Tseng, 2005; Sun et al., 2009; Sun et al., 2012; Sun et al., 2014). However, these approaches incorporate many handcrafted features.

In recent years, neural networks have become increasingly popular in CWS, which focused more on the ability of automated feature extraction. Recent works (Collobert et al., 2011; Pei et al., 2014; Cai and Zhao, 2016; Xu and Sun, 2016; Sun, 2016) have been shown effective in using neural networks for Chinese word segmentation. Collobert et al. (2011) developed a general neural architecture for sequence labelling tasks. Pei et al. (2014) used convolutioanl neural networks to capture local features within a fixed size window. Chen et al. (2015) proposed gated recursive neural networks to model feature combinations. The gating mechanism was also used by Cai and Zhao (2016).

References

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for chinese. In *Meeting of the Association for Computational Linguistics*.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015. Gated recursive neural network for chinese word segmentation. In *ACL (1)*, pages 1744–1753. The Association for Computer Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, number 8 in ICML '01, pages 282–289.

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, June. 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*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A discriminative latent variable chinese segmenter with hybrid features. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64, Boulder, Colorado, June. Association for Computational Linguistics.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262, Jeju Island, Korea, July. Association for Computational Linguistics.

Xu Sun, Wenjie Li, Houfeng Wang, and Qin Lu. 2014. Feature-frequency-adaptive on-line training for fast and accurate natural language processing. *Computational Linguistics*, 40(3):563–586.

Xu Sun. 2016. Asynchronous parallel learning for neural networks and structured models with dense features. In *COLING*.

Huihsin Tseng. 2005. A conditional random field word segmenter. In *In Fourth SIGHAN Workshop on Chinese Language Processing*.

Jingjing Xu and Xu Sun. 2016. Dependency-based gated recursive neural network for chinese word segmentation. In *Meeting of the Association for Computational Linguistics*, pages 567–572.

N. Xue and L. Shen. 2003. Chinese Word Segmentation as LMR Tagging. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*.