NLP Experiment One: Walk into Chinese Tokenization

Zhijun Wang
Computer Science and Technology, Harbin Institute of Technology, Harbin, China
1190303311@stu.hit.edu.cn

Abstract

Tokenization has always been an important part in Natural Language Processing(NLP). The results of the tokenization are close related to the performance of the NLP systems in either understanding or generation tasks. For Chinese, the tokenization process is often named Chinese Word Segmentation(CWS) as Chinese texts has no explicit delimiter for words. Sentences are continuous sequences of Chinese characters. So CWS aims to segment Chinese sentences into sequences of words, which contains at least one Chinese character. In this paper, we try to grep a view of the Chinese word segmentation techniques used before, which take simple ideas to get pretty good results. We perform how segmentation techniques like Forward Maximum Matching(FMM), Backward Maximum Mathcing(BMM) and Max Probability Tokenization using Directed Cyclic Graphs(DAG) work and try to decrease the time used to tokenize. We conduct experiments on People's Daily Corpus and analyese the performance of different techniques. Code and scripts are freely available at https://github.com/1190303311/CWS.git.

1 Introduction

Chinese Word Segmentation has become the mainstream paradigm in Chinese natural language processing tasks recently. In Chinese NLP tasks, Chinese sentences are segmented into sequences of meaning words and then algorithms are conducted to learn embeddings of them. For example, CWS could segment the Chinese sentence "我想吃苹果" to "我想吃苹果", which words are split by blanks. One simple idea to segment sentences is take each Chinese character as one word. This idea brings smaller vocabulary but suffers from the poor performance. Besides, tokenization of whatever languages all suffer from the multiple segmentation problems, which means that one sentence could have more than one possible segmentation. As for CWS, its also suffers from the open vocabulary problems. As language develops, new words are created rapidly, which makes the vocabulary size continuously grows. In the next several sections, we conduct several research on CWS and try to walk into some simple CWS techniques.

SRC	学生会组织义演活动
SegmentationA	学生会/ 组织/ 义演/ 活动/
SegmentationB	学生/ 会/ 组织/ 义演/ 活动/

Table 1: Multiple segmentation of Chinese sentence.

2 Related Work

Chinese word segmentation (CWS) is the mainstream paradigm in character level representation which cuts text into words consisting of at least one character. Existing research pays much attention to CWS tasks with neural network architecture. Ma et al. (2018) use Bi-LSTMs to conduct CWS, leveraging both previous and future information in a sentence while Gan and Zhang (2020) show that self-attention network gives more competitive results. These techniques perform well on Chinese NLP tasks. Si et al. (2021) propose sub-character tokenization to encode a Chinese character into a sequence of its glyph or pronunciation and learn a new vocabulary for Chinese language model pretraining, which is different from CWS. Recently, subword learning is widely used to address the limited vocabulary problem in NLP tasks and has been proved powerful (Sennrich et al., 2016). Several researches leverage different segmentation as augmented data or a noisy term during training. Kudo (2018) propose subword regularization by integrating different segmentation of words to NMT models by probability. Provilkov et al. (2020) propose the BPE-dropout technique to stochastically corrupt the segmentation procedure of BPE. Wang et al. (2021) propose multi-view subword regularization to make full use of different kinds of seg-

Model	Precision	Recall	F1
FMM	97.71%	97.04%	97.38%
BMM	97.89%	97.25%	97.57%
LM	98.85%	97.94%	98.40%

Table 2: Results of FMM and BMM on the first month data of People's Daily. Experiments are conducted in non-closed style.

Model	Precision	Recall	F1
FMM	70.73%	92.48%	80.16%
BMM	70.94%	92.74%	80.39%
LM	72.24%	94.67%	81.95%

Table 3: Results of FMM and BMM on the test data of People's Daily. Experiments are conducted in closed style.

mentation. Manghat et al. (2022) propose a hybrid subword segmentation algorithm to deal with out-of-vocabulary words. Tay et al. (2021) propose a soft gradient-based subword tokenization algorithm to learn subword representation in data-driven fashion. Ács et al. (2021) investigate how different strategies of subword pooling affect the downstream performance.

3 Chinese Word Segmentation System

3.1 Vocabulary Construction

We construct our vocabulary by simply extract all the words in the training corpus. The training data are raw texts which are segmented already. We split the sentences to get Chinese words and count their frequency of occurrence in data. We take all these words as the vocabulary. Each line of the vocabulary file is one word and its frequency of occurrence with a blank between them. The corpus contains 69K sentences which are segmented already. We randomly take 55K(80%) of them as the training data and the rest as the test data.

3.2 The implementation of FMM and BMM

We follow the code frameworks given by the experiment files. The ideas of FMM and BMM are simple. Given a string we try to match the longest substring in the vocabulary. The key part of this algorithm is the mathcing process. If we just use scan search, it would cost too much time. More about how to optimize this process would be discussed in section 3.3.

SRC	外交工作取得了重要成果
FMM	外交/ 工作/ 取得/ 了/ 重要/ 成果
BMM	外交/ 工作/ 取/ 得了/ 重要/ 成果
SRC	改革开放历史新时期
FMM	改革/ 开放/ 历史/ 新时/ 期
BMM	改革/ 开放/ 历史/ 新/ 时期

Table 4: Different segmentation of FMM and BMM. Red segmentation are corresponding to the references.

3.3 Optimize

We take two methods to optimize the speed of the segmentation process. The first is the Trie Tree, we follow the code framework released by the experiment. Experiments are conducted on the first month of People's Daily corpus. We take FMM as the algorithm. Baselines are implemented using list and cost 23Ks to segment the 23K sentences. Using Trie Tree costs only 1.2Ks. The second method is Hash algorithm. We use the unicode of each char as its hash value. For words consisting of more than one char, their hash values are calculated by the sum of each char. After encoding each word in the vocabulary, we construct a list contains 100K positions and surplus the encoding of each words as their index. Words with the same encoding are stored in a list. We leave the conflicts unsolved and it costs only 49s. We grep a view about the conflicts and find that most of them come from the date. Lots of date words have the same encoding and are stored in one list with the same index. For further optimization, the conflict problem is the key bottleneck. One possible idea is to design better encoding algorithm and try to reduce the conflicts. We leave this for further research.

3.4 Statistical Language Model

Here we implement one-gram language model and follow the code framework released by the experiment. As our vocabulary contains the word frequency, here we just use it to construct the prefix dictionary.

```
> n = len(line)
> route = [None]*(n+1)
> route[n] = (0, 0)
> for idx in range(n-1, -1, -1):
>     route[idx] = max((math.log(
>     dic.get(line[idx:x+1]) or 1) -
```

```
>     logtotal + route[x+1][0], x)
>     for x in DAG[idx])
```

The key code for this algorithm is the search process. Here we have the DAG of the text and each items of DAG contains possible end indexes of words while taking the DAG index as the word begin index. Here we use dynamic programming and try to scan the DAG from tail to head. Each step we scan every possible word from current index and calculate the log probability. We can see that when process index i, the segmentation of index over i have already been solved. So we can get a set of possible segmentation of current index and find the segmentation with the biggest probability and stored its index and probability. Results in non-closed experiments are showed in table 2 and results in closed experiments are showed in table 3. One-gram language model performs better than FMM and BMM.

4 Experimental Setup

Data We conduct two experiments in non-closed style and closed style. The non-closed style experiments use the only the first month data of People's Daily corpus as both the train and test data. The closed experiments use all the corpus and split them to train and test data by 8:2. We use the training data to construct our vocabulary and test the performance of different algorithms on the test data. The vocabulary sizes of these two experiments are 55K and 137K.

Model We explore three kinds of tokenization algorithms: Forward Maximum Matching, Backward Maximum Matching and One-gram Language Modeling.

4.1 Main Results

Results of the non-closed and closed style experiments are showed in table 2 and 3. In non-closed experiments, FMM, BMM and LM all get pretty good results. LM gains explicit improvements over FMM and BMM(at least +0.89%). In closed style experiment, they all suffer a performance reduction but LM still outperforms the other two algorithms.

Non-closed Experiments We first take the first month of People's Daily corpus to do non-closed test. The vocabulary construction keeps the same with section 3.1 and w e get a 76K vocabulary. We use it to segment the text and calculate the precision, recall and F1 scores. Table 2 shows the results. BMM performs slightly better than FMM. Table 4 shows some different segmentation made by FMM and BMM. It seems that there is no explicit reason why BMM would get better results than FMM. These two examples in table 4 both exist in the corpus. The difference is that the number of them. We speculate that in our corpus there are more sentences suitable for BMM to segment.

Closed Experiments We take all the data of People's Daily and split them into train and test data by 8:2. The training data contains 55K sentences and the test data contains 14K sentences. The vocabulary we get from the training data has been introduced in section 3.1. Table 3 shows the results. We can see that the performance decrease significantly for both FMM and BMM, especially the precision score. It makes sense that the recall score decreases to about 92% in closed test. But the precision score decreases to about 70%. After analysis on the segmented data we find that the difference comes from the date words. In the construction of vocabulary, we simply add them in our vocabulary. However, it cannot see the date in the test data. So the date words are split into sequence of single numbers, making the False Negative(FP) counts too large. The precision decrease explicitly as it is calculated by (TP)/(TP+FP).

5 Analysis

The vocabulary we get from the training data contains 137K words. The most common words appear to be some punctuation and daily used words. The key reason for too large vocabulary could be the date words. We just take them into our vocabulary and they account for a large proportion. We make statistics on the vocabulary and find that the long tail phenomenon exists in Chinese. We take words with frequency less than 100 as rare words. They contain 135K kinds of tokens in the vocabulary and account for 21% in the training data, which means that 3K words account for 79%. However, long tail phenomenon has a bad impact in neural-based systems that rare words cannot get enough attention. For techniques in this experiment, we can ignore temporarily.

6 Conclusion

In this paper we walk through several kinds of tokenization algorithms. We conduct no-closed and closed style experiments to verify their performance, and take some methods to optimize the

speed of the segmentation. Recently, the tokenization of texts is still a popular field for research. Algorithms like Byte-Pair Encoding(BPE) and other greatly promote the development of NLP tasks. Although nowadays these algorithms implemented in this experiment face many problems such as the open-vocabulary problem, they appear simple and powerful in some scenes. It is worth to learn these techniques.

Limitations

The speed of segmentation can be further optimized by designing better encoding and hash method.

Acknowledgments

We would like to thank the teachers and TAs for their efforts.

References

Judit Ács, Ákos Kádár, and Andras Kornai. 2021. Subword pooling makes a difference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2284–2295, Online. Association for Computational Linguistics.

Leilei Gan and Yue Zhang. 2020. Investigating self-attention network for chinese word segmentation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2933–2941.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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.

Sreeja Manghat, Sreeram Manghat, and Tanja Schultz. 2022. Hybrid sub-word segmentation for handling long tail in morphologically rich low resource languages. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6122–6126. IEEE.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

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, Berlin, Germany. Association for Computational Linguistics.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2021. SHUOWEN-JIEZI: linguistically informed tokenizers for chinese language model pretraining. *CoRR*, abs/2106.00400.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Prakash Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *ArXiv preprint*, abs/2106.12672.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. Multi-view subword regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.