Unigram Language Model for Chinese Word Segmentation

Aitao Chen
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94089
aitao@yahoo-inc.com

Yiping Zhou
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94089
zhouy@yahoo-inc.com

Anne Zhang
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94089
annezhangya@yahoo.com

Gordon Sun
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94089
gzsun@yahoo-inc.com

Abstract

This paper describes a Chinese word segmentation system based on unigram language model for resolving segmentation ambiguities. The system is augmented with a set of pre-processors and post-processors to extract new words in the input texts.

1 Introduction

The Yahoo team participated in all four closed tasks and all four open tasks at the second international Chinese word segmentation bakeoff.

2 System Description

The underlying algorithm in our word segmentation system is the unigram language model in which words in a sentence are assumed to occur independently. For an input sentence, we examine all possible ways to segment the new sentence with respect to the segmentation dictionary, and choose the segmentation of the highest probability, which is estimated based on the unigram model.

Our system also has a few preprocessors and postprocessors. The main preprocessors include recognizers for extracting names of people, places and organizations, and recognizer for numeric expressions. The proper name recognizers are built based on the maximum entropy model, and the numeric expression recognizer is built as a finite state automaton. The conditional maximum entropy model in our implementation is based on the one described in Section 2.5 in (Ratnaparkhi, 1998), and features are the same as those described in (Xue and Shen, 2003).

One of the post-processing steps is to combine single characters in the initial segmentation if each character in a sequence of characters occurs in a word much more frequently than as a word on its own. The other post-processing procedure checks the segmentation of a text fragment in the input text against the segmentation in the training data. If the segmentation produced by our system is different from the one in the training data, we will use the segmentation in the training data as the final segmentation. More details on the segmentation algorithm and the preprocessors and postprocessors can be found in (Chen, 2003).

Our system processes a sentence independently. For an input sentence, the preprocessors are applied to the input sentence to extract numeric expressions and proper names. The extracted numeric expressions and proper names are added to the segmentation dictionary, if they are not already in the dictionary. Then the input sentence is segmented into words. Finally the post-processing procedures are applied to the initial segmentation to produce the final segmentation. Our system processes texts encoded in UTF-8; and it is used in all 8 tasks.

3 Results

Table 1 presents the results of the 10 official runs we submitted in all 8 tasks.

Run id	R	P	F	R-oov	R-in
as-closed	0.955	0.934	0.947	0.468	0.978
as-open	0.958	0.938	0.948	0.506	0.978

cityu-closed	0.949	0.931	0.940	0.561	0.980
cityu-open	0.952	0.937	0.945	0.608	0.980
pku-closed	0.953	0.946	0.950	0.636	0.972
pku-open-a	0.964	0.966	0.965	0.841	0.971
msr-closed-a	0.969	0.952	0.960	0.379	0.985
msr-closed-b	0.968	0.953	0.960	0.381	0.984
msr-open-a	0.970	0.957	0.963	0.466	0.984
msr-open-b	0.971	0.961	0.966	0.512	0.983

Table 1: Summary of Yahoo official results.

The first element in the run id is the corpus name, *as* referring to the Academia Sinica corpus, *cityu* the City University of Hong Kong corpus, *pk*u the Peking University Corpus, and *msr* the Microsoft Research corpus. The second element in the run id is the type of task, *closed* or *open*. The second column shows the recall, the third column the precision, and the fourth column F-score. The last two columns present the recall of the out-of-vocabulary words and the recall of the words in the training data, respectively.

3.1 Closed Tasks

For the AS closed task run *as-closed*, we manually identified about 15 thousands person names and about 4 thousands place names from the AS training corpus. We then built a person name recognizer and a place name recognizer from the AS training data. All the name recognizers we built are based on the maximum entropy model. We also built a rule-based numeric expression recognizer implemented as a finite state automaton.

The segmentation dictionary consists of the words in the training data with occurrence frequency compiled from the training data. For each character, the probability that a character occurs in a word is also computed from the training data only.

Each line of texts in the testing data set is processed independently. From an input line, first the person name recognizer and place name recognizer are used to extract person and place names; the numeric expression recognizer is used to extract numeric expressions. The extracted new proper names and new numeric expressions are added to the segmentation dictionary with a constant occurrence frequency of 0.5 before the input text is segmented. After the segmentation, a sequence of single characters is combined into a single unit if each of the characters in the sequence occurs much more frequently in a word than as a word on its own. The threshold of a character occurring in a word is set to 0.80. Also the quad-grams down to unigrams in the segmentation are checked against the training data. When a text fragment is segmented in a different way by our system than in the training data, we use the segmentation of the text fragment in the training data as the final output.

The runs *cityu-closed* and *pku-closed* are produced in the same way. We first manually identified the person names and place names in the training data, and then built name recognizers from the training data. The name recognizers and numeric expression recognizer are used first to extract proper names and numeric expressions before segmentation. The post-processing is also the same.

Two runs, named *msr-closed-a* and *msr-closed-b*, respectively, are submitted using the Microsoft Research corpus for the closed task. Unlike in the other three corpora, the numeric expressions are much more versatile, and therefore, more difficult to write regular expressions to identify them. We manually identified the numeric expressions, person names, place names, and organization names in the training data, and then built maximum entropy model-based recognizers for extracting numeric expressions and names of people, place, and organizations. Also the organization names in this corpus are not segmented into words like in the other three corpora. The organization name recognizer is word-based while the other three recognizers are character-based. The only difference between these two runs is that the run *msr-closed-b* includes an organization name recognizer while the other run *msr-closed-a* does not.

3.2 Open Tasks

For the AS open task, we used a user dictionary and a person name recognizer and a place name recognizer, both trained on the combined AS corpus and the CITYU corpus. However, the base dictionary and word frequency counts are compiled from only the AS corpus. For the open run, we used the annotated AS corpus we acquired from Academia Sinica. Also the phrase segmentation table is built from the AS training data only. The AS open run *as-open* was produced with the new person and place name recognizers and with the user dictionary. The performance of the open run is almost the same as that of the close run.

The training data used in the CITYU open task is the same as in the closed task. We built a person name recognizer and a place name recognizer from the combined AS and CITYU corpora. In training a recognizer, we only kept the sentences that contain at least one person or place name. The run *cityu-open* was produced with new person name and place name recognizers trained on the combined corpora but *without* user dictionary. The base dictionary and frequency counts are from the CITYU training data. We prepared a user dictionary for the CITYU open run but forgot to turn on this feature in the configuration file. We repeated the CITYU open run *cityu-open* with user dictionary. The recall is 0.959; precision is 0.953; and F-score is 0.956.

For the PKU open task run *pku-open-a*, we trained our segmenter from the word-segmented People's Daily corpus covering the period of January 1 through June 30, 1998. Our base dictionary with word frequency counts, character counts, and phrase segmentation table are built from this larger training corpus of about 7 million words. The words in this corpus are annotated with part-of-speech categories. Both the names of people and the names of places are uniquely tagged in this corpus. We created a training set for person name recognizer by combining the sentences in the People's Daily corpus that contain at least one person name with the sentences in the MSR training corpus that contain at least one person name. The person names in the MSR corpus were manually identified. From the combined training data for person names, we built a person name recognizer based on the maximum entropy model. The place name recognizer was built in the same way. The PKU open run *pku-open-a* was produced using the segmenter trained on the 6-month People's Daily corpus with the new person and place name recognizer trained on the People's Daily corpus and the MSR corpus. A user dictionary of about 100 thousand entries, most being proper names, was used in the PKU open run.

The training data used for the MSR open runs is the same MSR training corpus. Our base dictionary, together with word frequency counts, and phrase segmentation table are built from the MSR training data only. The numeric expression recognizer is the same as the one used in the closed task. The person name recognizer and place name recognizer are the same as those used in the PKU open task. We built an organization name recognizer from the People's Daily corpus where organization names are marked. For example, the text fragment "[上海/ns 社科院/j]nt" is marked by a pair of brackets and tagged with "nt" in the annotated People's Daily corpus. We extracted all the sentences containing at least one organization name and built a word-based recognizer. The feature templates are the same as in person name or place name recognizer. We submitted two MSR open task runs, named *msr-open-a* and *msr-open-b*, respectively. The only difference between these two runs is that the first run *msr-open-a* did not include an organization name recognizer, while the run *msr-open-b* used the organization name recognizer built on the annotated People's Daily corpus. Both runs were produced with a user dictionary, the new person name recognizer and new place name recognizer. The increase of F-score from 0.963 to 0.966 is due to the organization name recognizer. While the organization name recognizer correctly identified many organization names, it also generated many false positives. So the positive impact was offset by the false positives.

At about 12 hours before the due time, we learned that multiple submissions for the same task are acceptable. A colleague of ours submitted one PKU open run with the run id 'b' and one MSR open run with the run id 'c' in the bakeoff official results using a different word segmentation system without being tuning for the bakeoff. These two open runs are not discussed in this paper.

4 Discussions

The differences between our closed task runs and open task runs are rather small for both the AS corpus and the CITYU corpus. Our CITYU open run would be substantially better had we used our user dictionary. The open task run using the PKU corpus is much better than the closed task run. We performed a number of additional evaluations in both the PKU closed task and the PKU open task. Table 2 below presents the evaluation results with different features activated in our system. The PKU training corpus was used in all the experiments presented in Table 2.

Run	Features	R	P	F
1	base-dict	0.9386	0.9095	0.9238
2	1+num-expr	0.9411	0.9161	0.9285
3	2+person+place	0.9440	0.9249	0.9343
4	3+single-char	0.9404	0.9420	0.9412
5	4+consistency-checking	0.9529	0.9464	0.9496

Table 2: Results with different features applied in PKU closed task.

Table 3 presents the results with different features applied in the PKU open task. The 6-month annotated People's Daily corpus was used in all the experiments shown in Table 3.

Run	Features	R	P	F
1	base-dict	0.9523	0.9503	0.9513
2	1+user-dict	0.9534	0.9565	0.9549
3	2+num-expr	0.9547	0.9605	0.9576
4	3+person+place	0.9562	0.9647	0.9604
5	4+single-char	0.9487	0.9650	0.9568
6	5+consistency-checking	0.9637	0.9664	0.9650

Table 3: Results with different features applied in PKU open task.

In the features column, *base-dict* refers to the base dictionary built from the training data only; *user-dict* the additional user dictionary*; num-expr* the numeric expression recognizer implemented as a finite state automaton; *person* the person name recognizer; *place* the place name recognizer; *single-char* combining a sequence of single characters when each one of them occurs in words much more frequently than as a word on its own; and lastly *consistency-checking* checking segmentations against the training texts and choosing the segmentation in the training texts if the segmentation of a text fragment produced by our system is different from the one in the training data. The tables show the results with more and more features included. Each run in the both tables includes one or two new features over the previous run. The last run numbered 5 in Table 2 is our official PKU closed run labeled *pku-closed* in Table 1; and the last run numbered 6 in Table 3 is our official PKU open run labeled *pku-open-a* in Table 1.

The F-score for our closed PKU task run is 0.950 with all available features, while using the larger People's Daily corpus as training data and its dictionary alone, the F-score is 0.9513. So a larger training data contributed significantly to the increase in performance in our PKU open task run. The user dictionary, the numeric expression recognizer, the person name recognizer, and the place name recognizer all contributed to the better performance of our PKU closed run and open run. Selectively combining sequence of single characters appreciably improved the precision while marginally decreased the recall in the PKU closed run. However, in the open task run, combining single characters did not result in better performance, probably because the new words recovered by combining single characters are already in our user dictionary for the open run. Finally consistency checking substantially improved the performance for both the closed run and the open run.

5 Conclusion

We presented a word segmentation system that uses unigram language model to select the most probable segmentation among all possible candidates for an input text. The system is augmented with proper name recognizers, numeric expression recognizers, and post-processing modules to extract new words. Overall the recognizers and the post-processing modules substantially improved the baseline performance. The larger training data set used in the PKU open task also significantly increased the performance of our PKU open run. The additional user dictionary is another major contributor to our better performance in the open tasks over the closed tasks.

References

Aitao Chen. 2003. *Chinese Word Segmentation Using Minimal Linguistic Knowledge.* In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing.

Nianwen Xue and Libin Shen. 2003. *Chinese Word Segmentation as LMR Tagging.* In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* Dissertation in Computer and Information Science, University of Pennsylvania.