Towards a Hybrid Model for Chinese Word Segmentation

Xiaofei Lu
Department of Linguistics
The Ohio State University
Columbus, OH 43210, USA
xflu@ling.osu.edu

Abstract

This paper describes a hybrid Chinese word segmenter that is being developed as part of a larger Chinese unknown word resolution system. The segmenter consists of two components: a tagging component that uses the transformation-based learning algorithm to tag each character with its position in a word, and a merging component that transforms a tagged character sequence into a word-segmented sentence. In addition to the position-of-character tags assigned to the characters, the merging component makes use of a number of heuristics to handle non-Chinese characters, numeric type compounds, and long words. The segmenter achieved a 92.8% F-score and a 72.8% recall for OOV words in the closed track of the Peking University Corpus in the Second International Chinese Word Segmentation Bakeoff.

1 Introduction

This paper describes a hybrid Chinese word segmenter that participated in the closed track of the Peking University Corpus in the Second International Chinese Word Segmentation Bakeoff. This segmenter is still in its early stage of development and is being developed as part of a larger Chinese unknown word resolution system that performs the identification, part of speech guessing, and sense guessing of Chinese unknown words (Lu, 2005).

The segmenter consists of two major components. First, a tagging component tags each individual character in a sentence with a position-of-character (POC) tag that indicates the position of the character in a word. This could be one of the following four possibilities, i.e., the character is either a monosyllabic word or is in a word-initial, middle, or final position. This component is based on the transformation-based learning (TBL) algorithm (Brill, 1995), where a simple first-order HMM tagger (Charniak et al., 1993) is used to produce an initial tagging of a character sequence. Second, a merging component transforms the output of the tagging component, i.e., a POC-tagged character sequence, into a word-segmented sentence. Whereas this process relies largely on the POC tags assigned to the individual characters, it also takes advantage of a number of heuristics generalized from the training data to handle non-Chinese characters, numeric type compounds, and long words.

The approach adopted here is reminiscent of the line of research that employs the idea of character-based tagging for Chinese word segmentation and/or unknown word identification (Goh et al., 2003; Xue, 2003; Zhang et al., 2002). The notion of character-based tagging allows us to model the tendency for individual characters to combine with other characters to form words in different contexts. This property gives the model a good potential for improving the performance of Chinese unknown word identification, a major concern of the Chinese unknown word resolution system that the segmenter is a part of.

The rest of the paper is organized as follows. Section two describes the system architecture. Section three reports the results of the system in the bakeoff. Section four concludes the paper.

2 System Description

The overall architecture of the segmenter is described in Figure 1. An input sentence is first segmented into a character sequence, with a space inserted after each character. The segmented character sequence is then processed by the tagging component, where it is initially tagged by an HMM tagger, and then by a TBL tagger. Finally, the tagged character sequence is transformed into a word-segmented sentence by the merging component.

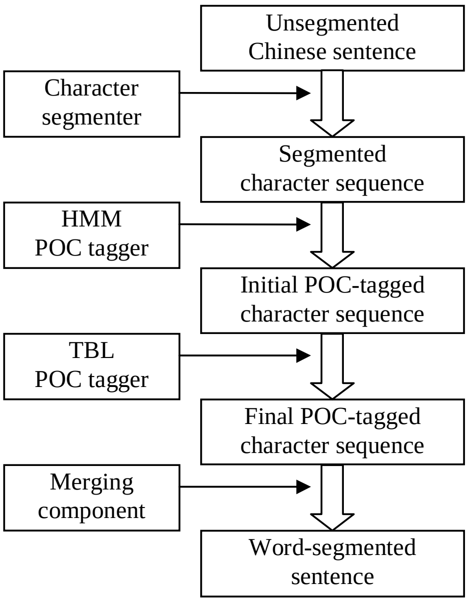
Unsegmented Chinese sentence
Character segmenter
Segmented character sequence
HMM POC tagger
Initial POC-tagged character sequence
TBL POC tagger
Final POC-tagged character sequence
Merging component
Word-segmented sentence

Figure 1: System Architecture.

2.1 The Tagging Component

The tagset used by the tagging component consists of the following four tags: *L*, *M*, *R*, and *W*, each of which indicates that the character is in a word-initial, word-middle, or word-final position or is a monosyllabic word respectively. The transformation-based error-driven learning algorithm is adopted as the backbone of the tagging component over other promising machine learning algorithms because, as Brill (1995) argued, it captures linguistic knowledge in a more explicit and direct fashion without compromising performance. This algorithm requires a gold standard, some initial tagging of the training corpus, and a set of rule templates. It then learns a set of rules that are ranked in terms of the number of tagging error reductions they can achieve.

A number of different initial tagging schemes can be used, e.g., tagging each character as a monosyllabic word or with its most probable POC tag. We used a simple first-order HMM tagger to produce an initial tagging. Specifically, we calculate

$$\arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} p(t_i|t_{i-1})p(t_i|w_i) \qquad (1)$$

where t_i denotes the *i*th tag in the tag sequence and w_i denotes the *i*th character in the character sequence. The transition probabilities and lexical probabilities are estimated from the training data. The lexical probability for an unknown character, i.e., a character that is not found in the training data, is by default uniformly distributed among the four POC tags defined in the tagset. The Viterbi algorithm (Rabiner, 1989) is used to tag new texts.

The transformation-based tagger was implemented using fnTBL (Florian and Ngai, 2001). The rule templates used are the same as the contextual rule templates Brill (1995) defined for the POS tagging task. These templates basically transform the current tag into some other tag based on the current character/tag and the character/tag one to three positions before/after the current character. An example rule template is given below:

(1) Change tag *a* to tag *b* if the preceding character is tagged *z*.

The training process is iterative. At each iteration, the algorithm picks the instantiation of a rule template that achieves the greatest number of tagging error reductions. This rule is applied to the text, and the learning process repeats until no more rules reduce errors beyond a predefined threshold. The learned rules can then be applied to new texts that are tagged by the initial HMM tagger.

2.2 The Merging Component

The merging component transforms a POC-tagged character sequence into a word-segmented sentence. In general, the characters in a sequence are concatenated, and a space is inserted after each character tagged *R* (word-final position) or *W* (monosyllabic word).

In addition, two sets of heuristics are used in this process. One set (H1) is used to handle non-Chinese characters and numeric type compounds, e.g., numbers, time expressions, etc. A few patterns of non-Chinese characters and numeric type compounds are generalized from the training data. If the merging algorithm detects such a pattern in the character sequence, it groups the characters that are part of the pattern accordingly.

The second set of heuristics (H2) is used to handle words that three or more characters long. Our hypothesis is that long words tend to have less fluidity than shorter words and their behavior is more predictable (Lu, 2005). We extracted a wordlist from the training data. Based on our hypothesis, if the merging algorithm detects that a group of characters form a long word found in the wordlist, it groups these characters into one word.

3 Results

The segmenter was evaluated on the closed track of the Peking University Corpus in the bakeoff. In the development stage, we partitioned the official training data into two portions: the training set consists of 90% of the data, and the development set consists of the other 10%. The POC tagging accuracy on the development set is summarized in Table 1. The results indicate that the TBL tagger significantly improves the initial tagging produced by the HMM tagger.

	Accuracy
HMM tagger	0.814
TBL tagger	0.936

Table 1: Tagging Results on the Development Set.

The performance of the merging algorithm on the development set is summarized in Table 2. To understand whether and how much the heuristics contribute to improving segmentation, we evaluated four versions of the merging algorithm. The set of heuristics used to handle non-Chinese characters and numeric type compounds did not seem to improve segmentation results on the development set, suggesting that these characters are handled well by the tagging component. However, the second set of heuristics improved segmentation accuracy significantly. This seems to confirm our hypothesis that longer words tend to behave more stably.

Resources used	R	P	F
POC Tags only	0.928	0.926	0.927
+ H1	0.929	0.925	0.927
+ H2	0.938	0.959	0.948
+ H1 & H2	0.940	0.960	0.950

Table 2: Segmentation Results on the Development Set. H1 stands for the set of heuristics used to handle non-Chinese characters and numeric type compounds. H2 stands for the set of heuristics used to handle long words.

Corpus	R	P	F	R_{OOV}	R_{IV}
PKU	0.922	0.934	0.928	0.728	0.934

Table 3: Official Results in the Closed-Track of the Peking University Corpus.

The official results of the segmenter in the closed-track of the Peking University Corpus are summarized in Table 3. It is somewhat unexpected that the results on the official test data dropped over 2% compared with the results obtained on the development set. Compared with the other systems, the segmenter performed relatively well on OOV words.

Our preliminary error analysis indicates that this discrepancy in performance is partially attributable to two kinds of inconsistencies between the training and test datasets. One is that there are many ASCII numbers in the test set, but none in the training set. These numbers became unknown characters to the tagger and affected tagging accuracy. It is possible that this inconsistency affected our system more than other systems. Second, there are also a number of segmentation inconsistencies between the training and test sets, but these should have affected all systems more or less equally. The error analysis also indicates that the current segmenter performed poorly on transliterations of foreign names.

4 Conclusions

We described a hybrid Chinese word segmenter that combines the transformation-based learning algorithm for character-based tagging and linguistic heuristics for transforming tagged character sequences into word-segmented sentences.

As the segmenter is in its first stage of development and is far from mature, the bakeoff provided an especially valuable opportunity for evaluating its performance. The results suggest that:

1. Despite the lack of a separate mechanism for unknown word recognition, the segmenter performed relatively well on OOV words. This confirms our hypothesis that character-based tagging has a good potential for improving Chinese unknown word identification.
2. Using linguistic heuristics at the merging stage can help improve segmentation results.
3. There is much room for improvement for both the tagging algorithm and the merging algorithm. This is being undertaken.

References

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543-565.

Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. 1993. Equations for part-of-speech tagging. In *Proceedings of AAAI-1993*, pp. 784-789.

Chooi Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2003. Chinese unknown word identification using character-based tagging and chunking. In *Proceedings of ACL-2003 Interactive Posters and Demonstrations*, pp. 197-200.

Xiaofei Lu. 2005. Hybrid methods for POS guessing of Chinese unknown words. In *Proceedings of ACL-2005 Student Research Workshop*, pp. 1-6.

Grace Ngai and Radu Florian. 2001. Transformation-based learning in the fast lane. In *Proceedings of NAACL-2001*, pp. 40-47.

Lawrence R. Rabiner. 1989. A tutorial of hidden Markov models and selected applications in speech recognition. In *Proceedings of IEEE-1989*, pp. 257-286.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1):29-48.

Kevin Zhang, Qin Liu, Hao Zhang, and Xue-Qi Cheng. 2002. Automatic recognition of Chinese unknown words based on roles tagging. In *Proceedings of the 1st SIGHAN Workshop on Chinese Language Processing*, pp. 71-78.