Chunking-based Chinese Word Tokenization

GuoDong ZHOU
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore, 119613
zhougd@i2r.a-star.edu.sg

Abstract

This paper introduces a Chinese word tokenization system through HMM-based chunking. Experiments show that such a system can well deal with the unknown word problem in Chinese word tokenization.

1 Introduction

Word Tokenization is regarded as one of major bottlenecks in Chinese Language Processing. Normally, word tokenization is implemented through word segmentation in Chinese Language Processing literature. This is also affected in the title of this competition.

There exists two major problems in Chinese word segmentation: ambiguity and unknown word detection. While ngarm modeling and/or word co-ocurrence has been successfully applied to deal with ambiguity problem, unknown word detection has become major bottleneck in word tokenization.

This paper proposes a HMM-based chunking scheme to cope with unkown words in Chinese word tokenization. The unknown word detection is re-casted as chunking several words (single-character word or multi-character word) together to form a new word.

2 HMM-based Chunking

2.1 HMM

Given an input sequence $G_1^n = g_1 g_2 \cdots g_n$, the goal of Chunking is to find a stochastic optimal tag sequence $T_1^n = t_1 t_2 \cdots t_n$ that maximizes (Zhou and Su 2000) (2-1)

$$\log P(T_1^n \mid G_1^n) = \log P(T_1^n) + \log \frac{P(T_1^n, G_1^n)}{P(T_1^n) \cdot P(G_1^n)}$$

The second term in (2-1) is the mutual information between T_1^n and G_1^n. In order to simplify the computation of this term, we assume mutual information independence (2-2):

$$MI(T_1^n, G_1^n) = \sum_{i=1}^{n} MI(t_i, G_1^n) \text{ or}$$

$$\log \frac{P(T_1^n, G_1^n)}{P(T_1^n) \cdot P(G_1^n)} = \sum_{i=1}^{n} \log \frac{P(t_i, G_1^n)}{P(t_i) \cdot P(G_1^n)}$$

That is, an individual tag is only dependent on the token sequence G_1^n and independent on other tags in the tag sequence T_1^n. This assumption is reasonable because the dependence among the tags in the tag sequence T_1^n has already been captured by the first term in equation (2-1). Applying it to equation (2-1), we have (2-3):

$$\log P(T_1^n \mid G_1^n) = \log P(T_1^n) - \sum_{i=1}^{n} \log P(t_i) + \sum_{i=1}^{n} \log P(t_i \mid G_1^n)$$

From equation (2-3), we can see that:

- The first term can be computed by applying chain rules. In ngram modeling, each tag is assumed to be probabilistically dependent on the N-1 previous tags.
- The second term is the summation of log probabilities of all the individual tags.
- The third term corresponds to the "lexical" component (dictionary) of the tagger.

We will not discuss either the first or the second term further in this paper because ngram modeling has been well studied in the literature. We will focus on the third term $\sum_{i=1}^{n} \log P(t_i \mid G_1^n)$.

2.2 Chinese Word Tokenization

Given the previous HMM, for Chinese word tokenization, we have (Zhou and Su 2002):

- $g_i = < p_i, w_i >$; $W_1^n = w_1 w_2 \cdots w_n$ is the word sequence; $P_1^n = p_1 p_2 \cdots p_n$ is the word formation pattern sequence and p_i is the word formation pattern of w_i. Here p_i consists of:
  - The percentage of w_i occurring as a whole word (round to 10%)
  - The percentage of w_i occurring at the beginning of other words (round to 10%)
  - The percentage of w_i occurring at the end of other words (round to 10%)
  - The length of w_i
  - The occurring frequence feature, which is set to max(log(Frequence), 9).
- tag t_i : Here, a word is regarded as a chunk (called "Word-Chunk") and the tags are used to bracket and differentiate various types of Word-chunks. Chinese word tokenization can be regarded as a bracketing process while differentiation of different word types can help the bracketing process. For convenience, here the tag used in Chinese word tokenization is called “Word-chunk tag”. The Word-chunk tag t_i is structural and consists of three parts:
  - **Boundary category (B)**: it is a set of four values: 0,1,2,3, where 0 means that current word is a whole entity and 1/2/3 means that current word is at the beginning/in the middle/at the end of a word.
  - **Word category (W)**: used to denote the class of the word. In our system, word is classified into two types: pure Chinese word type and mixed word type (for example, including English characters/Chinese digits/Chinesenumbers).
  - **Word Formation Pattern(P)**: Because of the limited number of boundary and word categories, the word formation pattern is added into the structural chunk tag to represent more accurate models.

3 Context-dependent Lexicons

The major problem with Chunking-based Chinese word tokenization is how to effectively approximate $P(t_i / G_1^n)$. This can be done by adding lexical entries with more contextual information into the lexicon Φ. In the following, we will discuss five context-dependent lexicons which consider different contextual information.

3.1 Context of current word formation pattern and current word

Here, we assume:

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_i w_i) & p_i w_i \in \Phi \\ P(t_i / p_i) & p_i w_i \notin \Phi \end{cases}$$

where

$\Phi = \{p_i w_i, p_i w_i \exists C\} + \{p_i, p_i \exists C\}$ and $p_i w_i$ is a word formation pattern and word pair existing in the training data C .

3.2 Context of previous word formation pattern and current word formation pattern

Here, we assume :

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1} p_i) & p_{i-1} p_i \in \Phi \\ P(t_i / p_i) & p_{i-1} p_i \notin \Phi \end{cases}$$

where

$\Phi = \{p_{i-1} p_i, p_{i-1} p_i \exists C\} + \{p_i, p_i \exists C\}$ and $p_{i-1} p_i$ is a pair of previous word formation pattern and current word formation pattern existing in the training data C .

3.3 Context of previous word formation pattern, previous word and current word formation pattern

Here, we assume :

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1}w_{i-1}p_i) & p_{i-1}w_{i-1}p_i \in \Phi \\ P(t_i / p_i) & p_{i-1}w_{i-1}p_i \notin \Phi \end{cases}$$

where

$\Phi = \{p_{i-1}w_{i-1}p_i, p_{i-1}w_{i-1}p_i \exists C\} + \{p_i, p_i \exists C\}$, where $p_{i-1}w_{i-1}p_i$ is a triple pattern existing in the training corpus.

3.4 Context of previous word formation pattern, current word formation pattern and current word

Here, we assume :

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1}p_iw_i) & p_{i-1}p_iw_i \in \Phi \\ P(t_i / p_i) & p_{i-1}p_iw_i \notin \Phi \end{cases}$$

where

$\Phi = \{p_{i-1}p_iw_i, p_{i-1}p_iw_i \exists C\} + \{p_i, p_i \exists C\}$, where $p_{i-1}p_iw_i$ is a triple pattern.

3.5 Context of previous word formation pattern, previous word, current word formation pattern and current word

Here, the context of previous word formation pattern, previous word, current word formation pattern and current word is used as a lexical entry to determine the current structural chunk tag and $\Phi = \{p_{i-1}w_{i-1}p_iw_i, p_{i-1}w_{i-1}p_iw_i \exists C\} + \{p_i, p_i \exists C\}$, where $p_{i-1}w_{i-1}p_iw_i$ is a pattern existing in the training corpus. Due to memory limitation, only lexical entries which occurs at least 3 times are kept.

4 Error-Driven Learning

In order to reduce the size of lexicon effectively, an error-driven learning approach is adopted to examine the effectiveness of lexical entries and make it possible to further improve the chunking accuracy by merging all the above context-dependent lexicons in a single lexicon.

For a new lexical entry e_i, the effectiveness $F_\Phi(e_i)$ is measured by the reduction in error which results from adding the lexical entry to the lexicon : $F_\Phi(e_i) = F_\Phi^{Error}(e_i) - F_{\Phi+\Delta\Phi}^{Error}(e_i)$. Here, $F_\Phi^{Error}(e_i)$ is the chunking error number of the lexical entry e_i for the old lexicon Φ and $F_{\Phi+\Delta\Phi}^{Error}(e_i)$ is the chunking error number of the lexical entry e_i for the new lexicon $\Phi + \Delta\Phi$ where $e_i \in \Delta\Phi$ ($\Delta\Phi$ is the list of new lexical entries added to the old lexicon Φ). If $F_\Phi(e_i) > 0$, we define the lexical entry e_i as positive for lexicon Φ . Otherwise, the lexical entry e_i is negative for lexicon Φ .

5 Implementation

In training process, only the words occurs at least 5 times are kept in the training corpus and in the word table while those less-freqently occurred words are separated into short words (most of such short words are single-character words) to simulate the chunking. That is, those less-frequently words are regarded as chunked from several short words.

In word tokenization process, the Chunking-based Chinese word tokenization can be implemented as follows:

1) Given an input sentence, a lattice of word and word formation pattern pair is generated by skimming the sentence from left-to-right, looking up the word table to determine all the possible words, and determining the word formation pattern for each possible word.
2) Viterbi algorithm is applied to decode the lattice to find the most possible tag sequence.
3) In this way, the given sentence is chunked into words with word category information discarded.

6 Experimental Results

Table 1 shows the performance of our chunking-based Chinese word tokenization in the competition.

	PK (closed, official)	CTB (closed, unofficial)
Precision	94.5	90.7
Recall	93.6	89.6
F	94.0	90.1
OOV	6.9	18.1
Recall on OOV	76.3	75.2
Recall on In-Voc	94.9	92.7
Speed on P1.8G	420 KB/min	390 KB/min

The most important advantage of chunking-based Chinese word segmentation is the ability to cope with the unknown words. Table 1 shows that about 75% of the unknown words can be detected correctly using the chunking approach on the PK and CTB corpus.

7 Conclusion

This paper proposes a HMM-based chunking scheme to cope with the unkown words in Chinese word tokenization. In the meantime, error-driven learning is applied to effectively incorporate various context-dependent information. Experiments show that such a system can well deal with the unknown word problem in Chinese word tokenization.

References

Rabiner L. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *IEEE 77(*2), pages257-285.

Viterbi A.J. 1967. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory,* IT 13(2), 260-269.

Zhou GuoDong and Su Jian. 2000. Error-driven HMM-based Chunk Tagger with Context-dependent Lexicon. *Proceedings of the Joint Conference on Empirical Methods on Natural Language Processing and Very Large Corpus (EMNLP/ VLC'2000)*. Hong Kong, 7-8 Oct.