Abstract

This paper addresses two remaining challenges in Chinese word segmentation. The challenge in HLT is to find a robust segmentation method that requires no prior lexical knowledge and no extensive training to adapt to new types of data. The challenge in modelling human cognition and acquisition is to segment words efficiently without using knowledge of wordhood. We propose a radical method of word segmentation to meet both challenges. The most critical concept that we introduce is that Chinese word segmentation is the classification of a string of character-boundaries (CB’s) into either word-boundaries (WB’s) and non-word-boundaries. In Chinese, CB’s are delimited and distributed in between two characters. Hence we can use the distributional properties of CB among the background character strings to predict which CB’s are WB’s.

1 Introduction: modeling and theoretical challenges

The fact that word segmentation remains a main research topic in the field of Chinese language processing indicates that there maybe unresolved theoretical and processing issues. In terms of processing, the fact is that none of existing algorithms is robust enough to reliably segment unfamiliar types of texts before fine-tuning with massive training data. It is true that performance of participating teams have steadily improved since the first SigHAN Chinese segmentation bakeoff (Sprat and Emerson, 2004). Bakeoff 3 in 2006 produced best f-scores at 95% and higher. However, these can only be achieved after training with the pre-segmented training dataset. This is still very far away from real-world application where any varieties of Chinese texts must be successfully segmented without prior training for HLT applications.

In terms of modeling, all exiting algorithms suffer from the same dilemma. Word segmentation is supposed to identify word boundaries in a running text, and words defined by these boundaries are then compared with the mental/electronic lexicon for POS tagging and meaning assignments. All existing segmentation algorithms, however, presuppose and/or utilize a large lexical databases (e.g. (Chen and Liu, 1992) and many subsequent works), or uses the position of characters in a word as the basis for segmentation (Xue, 2003).

In terms of processing model, this is a contradiction since segmentation should be the pre-requisite of dictionary lookup and should not presuppose lexical information. In terms of cognitive modeling, such as for acquisition, the model must be able to account for how words can be successfully segmented and learned by a child/speaker without formal training or a priori knowledge of that word. All current models assume comprehensive lexical knowledge.

2 Previous work

Tokenization model. The classical model, described in (Chen and Liu, 1992) and still adopted in many recent works, considers text segmentation as a tokenization. Segmentation is typically divided into two stages: dictionary lookup and out of vocabulary (OOV) word identification. This approach requires comparing and matching tens of thousands of dictionary entries in addition to guessing thousands of OOV words. That is, this is a $10^4 \times 10^4$ scale mapping problem with unavoidable data sparseness.

More precisely the task consist in finding all sequences of characters $C_1, \ldots, C_n$ such that $[C_1, \ldots, C_n]$ either matches an entry in the lexicon or is guessed to be so by an unknown word resolution algorithm. One typical kind of the complexity this model faces is the overlap-
Any Chinese text is envisioned as a sequence of characters and character-boundaries $C_B_0, C_B_1, C_B_2, \ldots, C_B_{n-1}, C_B_n$. The segmentation task is reduced to finding all $C_B$s which are also wordbreaks $W_B$.

3.2 Modeling character-based information

Since $C_B$s are all the same and do not carry any information, we have to rely on their distribution among different characters to obtain useful information for modeling. In a segmented corpus, each $W_B$ can be differentiated from a non-$W_B$ $C_B$ by the character string before and after it. We can assume a reduced model where either one character immediately before and after a $C_B$ is considered or two characters (bigram). These options correspond to consider (i) only word-initial and word-final positions (hereafter the 2-CB-model or 2CBM) or (ii) to add second and penultimate positions (hereafter the 4-CB-model or 4CBM). All these positions are well-attested as morphologically significant.

3.3 The nature of segmentation

It is important to note that in this approaches, although characters are recognized, unlike (Xue, 2003) and Huang et al. (2006), characters simply are in the background. That is, they are the necessary delimiter, which allows us to look at the string of $C_B$s and obtaining distributional information of them.

4 Implementation and experiments

In this section we slightly change our notation to allow for more precise explanation. As noted before, Chinese text can be formalized as a sequence of characters and intervals as illustrated in we call this representation an interval form.

$c_1I_1c_2I_2\ldots c_{n-1}I_{n-1}c_n$.

In such a representation, each interval $I_k$ is either classified as a plain character boundary ($C_B$) or as a word boundary ($W_B$).

We represent the neighborhood of the character $c_i$ as $(c_{i-2}, I_{i-2}, c_{i-1}, I_{i-1}, c_i, I_i, c_{i+1}, I_{i+1})$, which we can be simplified as $(I_{i-2}, I_{i-1}, c_i, I_i, I_{i+1}, I_{i+2})$ by removing all the neighboring characters and retaining only the intervals.

4.1 Data collection models

This section makes use of the notation introduced above for presenting several models accounting for character-interval class co-occurrence.

Word based model. In this model, statistical data about word boundary frequencies for each character is...
retrieved word-wise. For example, in the case of a monosyllabic word only two word boundaries are considered: one before and one after the character that constitutes the monosyllabic word in question.

The method consists in mapping all the Chinese characters available in the training corpus to a vector of word boundary frequencies. These frequencies are normalized by the total frequency of the character in a corpus and thus represent probability of a word boundary occurring at a specified position with regard to the character.

Let us consider for example, a tri-syllabic word \( W = c_1c_2c_3 \), that can be rewritten as the following interval form as \( W^I = I^n_1c_1I^n_2c_2I^n_3c_3I^n_4 \).

In this interval form, each interval \( I_k \) is marked as word boundary \( B \) or \( N \) for intervals within words. When we consider a particular character \( c_1 \) in \( W \), there is a word boundary at index \(-1 \) and \( 3 \). We store this information in a mapping \( c_1 = \{-1 : 1, 3 : 1\} \). For each occurrence of this character in the corpus, we modify the character vector accordingly, each WB corresponding to an increment of the relevant position in the vector. Every character in every word of the corpus in processed in a similar way.

Obviously, each character yields only information about positions of word boundaries of a word this particular character belongs to. This means that the index \( I_{-1} \) and \( I_3 \) are not necessarily incremented every time (e.g. for monosyllabic and bi-syllabic words)

**Sliding window model.** This model does not operate on words, but within a window of a give size (span) sliding through the corpus. We have experimented this method with a window of size 4. Let us consider a string, \( s = "c_1c_2c_3c_4" \) which is not necessarily a word and is rewritten into an interval form as \( s^I = "c_1I_1c_2I_2c_3I_3c_4I_4" \). We store the co-occurrence character/word boundaries information in a fixed size (span) vector.

For example, we collect the information for character \( c_3 \) and thus arrive at a vector \( c_3 = (I_1, I_2, I_3, I_4) \), where 1 is incremented at the respective position if \( I_k = WB \), zero otherwise.

This model provides slightly different information that the previous one. For example, if a sequence of four characters is segmented as \( c_1I^n_1c_2I^n_2c_3I^n_3c_4I^n_4 \) (a sequence of one bi-syllabic and two monosyllabic words), for \( c_3 \) we would also get probability of \( I_4 \), i.e. an interval with index +2 . In other words, this model enables to learn \( WB \) probability across words.

### 4.2 Training corpus

In the next step, we convert our training corpus into a corpus of interval vectors of specified dimension. Let’s assume we are using dimension \( span = 4 \). Each value in such a vector represents the probability of this interval to be a word boundary. This probability is assigned by character for each position with regard to the interval. For example, we have segmented corpus \( C = c_1I_1c_2I_2\ldots c_{n-1}I_{n-1}c_n \), where each \( I_k \) is labeled as \( B \) for word boundary or \( N \) for non-boundary.

In the second step, we move our 4-sized window through the corpus and for each interval we query a character at the corresponding position from the interval to retrieve the word boundary occurrence probability. This procedure provides us with a vector of 4 probability values for each interval. Since we are creating this training corpus from an already segmented text, a class \((B \text{ or } N)\) is assigned to each interval.

An example of a part of the training corpus “台北市政府”, which is segmented as “台北市 政府”, can be seen in Figure 1.

The testing corpus (unsegmented) is encoded in a similar way, but does not contain the class labels \( B \) and \( N \).

Finally, we automatically assign probability of 0.5 for unseen events.

### 4.3 Predicting word boundary with a classifier

The Sinica corpus contains 6820 types of characters (including Chinese characters, numbers, punctuation, Latin alphabet, etc.). When the Sinica corpus is converted into our interval vector corpus, it provides 14.4 million labeled interval vectors. In this first study we have implement a baseline model, without any pre-processing of punctuation, numbers, names.

A decision tree classifier (Ruggieri, 2004) has been adopted to overcome the non-linearity issue. The classifier was trained on the whole Sinica corpus, i.e. on 14.4 million interval vectors. Due to space limit, actual bake-off experiment result will be reported in our poster presentation.

Our best results is based on the sliding window model, which provides better results. It has to be emphasized that the test corpora were not processed in any way, i.e. our method is sufficiently robust to account for a large number of ambiguities like numerals, foreign words.

### 5 Conclusion

In this paper, we presented a radical and robust model of Chinese segmentation which is supported by initial experiment results. The model does not pre-suppose any
lexical information and it treats character strings as context which provides information on the possible classification of character-breaks as word-breaks. We are confident that once a standard model of pre-segmentation, using textual encoding information to identify WB’s which involves non-Chinese characters, will enable us to achieve even better results. In addition, we are looking at other alternative formalisms and tools to implement this model to achieve the optimal results. Other possible extensions including experiments to simulate acquisition of wordhood knowledge to provide support of cognitive modeling, similar to the simulation work on categorization in Chinese by (Redington et al., 1995). Last, but not the least, we will explore the possibility of implementing a sharable tool for robust segmentation for all Chinese texts without training.

References


