Word Boundary Decision with CRF for Chinese Word Segmentation

Shoushan Li and Chu-Ren Huang

Department of Chinese and Bilingual Studies
The Hong Kong Polytechnic University
{shoushan.li, churenhuang}@gmail.com

Abstract. Chinese word segmentation systems necessarily perform both accurately and quickly for real applications. In this paper, we study on word boundary decision (WBD) approach for Chinese word segmentation and implement it as a 2-tag character tagging with conditional random filed (CRF). With a help of tag transition features, WBD with CRF segmentation approach can achieve comparative performances compared to 4-tag character tagging approach (represents the state-of-the-art segmentation approach). But it requires only about half training time and memory space as much as 4-tag character tagging approach. These results encourage that WBD segmentation approach is a good choice for real Chinese word segmentation systems.

Keywords: Chinese word segmentation, conditional random field, word boundary decision.

1 Introduction

Chinese word segmentation (CWS) is the task of segmenting text of character string into word list as original Chinese text contains no explicit boundaries between every two words. This task is an indispensible preprocessing requirement for many applications in Chinese language technology. A realistic CWS system necessarily performs well on both segmentation accuracy and speed.

Segmentation accuracy is essential for many applications. For instance, in machine translation for Chinese to English (Chang *et al.*, 2008), segmentation errors would cause translation mistakes directly. Translation systems without a wonderful CWS model are impossible to offer good results. State-of-the-arts approach called character tagging (Xue, 2003) has shown to be excellent in segmentation accuracy. This approach mainly aims to detect the character position in a certain word, e.g., beginning, middle or end of a word. It achieves much better performances than traditional word-based (or dictionary) approach, e.g., n-gram word maximum probability (Sun *et al.*, 2006), because of its apparent advantages on detecting out-of-vocabulary (OOV) words.

On the other side, the segmentation speed is also very important in some applications, such as information retrieval and online machine translation systems. Since CWS system is almost always used as a preprocessing step in the applications, long segmentation time would make the applications' whole running time unacceptable by users. Therefore, it is meaningful to simplify the complexity of CWS approaches so as to reducing segmentation training and testing time. In terms of this view, character tagging approach (often using 4-tags (Xue, 2003; Ng and Low, 2004) or even 6-tags (Zhao *et al.*, 2006)) is not so satisfactory in its training and testing time. Especially, given a very huge training data, this approach might not get a training model due to its large time and memory space demand.

Recently, Huang *et al.* (2007) propose an interesting approach, named word boundary decision (WBD), which turns from words towards word boundaries. WBD tries to detect the

nature of boundary between two characters, which can be either a word boundary or not, i.e. a boundary between two words or a mere character boundary. This approach performs better than traditional word-based (or dictionary) approach but still worse than character tagging approach (Huang *et al.*, 2008). However, this approach takes a big advantage over character tagging approach in its training and testing time.

In this paper, we deeply analyze the relationship between character tagging approach and WBD approach and propose a new implementation of WBD approach with conditional random field (CRF) learning approach. This implementation will make WBD approach achieve competitive performance compared to character tagging approach with 4-tags which represents the state-of-the-art approach in CWS studies but need much less training time and memory space.

In the remaining part of the paper, we review WBD approach and study the relationship between this approach and character tagging approach in Section 2. Then, we propose our implementation approach of WBD with CRF in Section 3. Experimental results are given and discussed in Section 4. Finally, we conclude our contribution on Chinese word segmentation in Section 5.

2 Word Boundary Decision

2.1 Approach Reviewing

Huang *et al.* (2007) propose an interesting approach called WBD which aims at classifying boundaries directly rather than classifying characters. As a result, word segmentation becomes a binary classification problem, which makes the segmentation task easier and faster.

Chinese text can be formalized as a sequence of characters and intervals

$$c_1 I_1 c_2 I_2, ..., c_{n-1} I_{n-1} c_n$$

where c_i means a character and I_i means a interval between two characters. There is no indication of word boundaries in Chinese text and each interval might be a word boundary ($I_i = 1$) or not ($I_i = 0$). The classification problem in WBD is to classify the intervals into word boundaries or non-boundaries.

WBD consists of two main steps: generating a set of character n-gram probabilities and classifier training and testing using probability vectors coined from n-gram set.

In the first step, different kinds of character n-gram probabilities are estimated from training data. Five different unigram and bi-gram probabilities are usually used in WBD. They are unigram probabilities of P_{CB}, P_{BC} and bigram probabilities of P_{CCB}, P_{CBC}, P_{BCC}. The definition of P_{CB} is given as

$$P_{CB}(I_i = 1 \mid c_i) = \frac{C(c_i, I_i = 1)}{C(c_i)}$$

where $C(c_i, I_i = 1)$ is the number of c_i which appears before a word boundary. $C(c_i)$ is the total number of c_i that appears in the training data. Similarly, definition of P_{CCB} is given as

$$P_{CCB}(I_i = 1 \mid c_{i-1}, c_i) = \frac{C(c_{i-1}, c_i, I_i = 1)}{C(c_{i-1}, c_i)}$$

where $C(c_{i-1}, c_i, I_i = 1)$ is the number of bigrams of characters c_{i-1}, c_i which appear together in front of a word boundary. $C(c_{i-1}, c_i)$ represents the total number of the bi-gram c_{i-1}, c_i.

After the estimating process on the training data, all unigrams and bi-grams will get their boundary probability information. The probabilities are then applied to generate the vectors in the second step. Once the frequency and probability information of all character n-grams is obtained, it can be easily preserved in a database (n-gram database).

In the second step, each boundary I_i would be represented as a vector

$$< P_{CCB}(I_i),\ P_{CB}(I_i),\ P_{CBC}(I_i),\ P_{BC}(I_i),\ P_{BCC}(I_i) >$$

Both training and testing process need to generate the vectors for each boundary. Interestingly, Huang *et al.* (2008) show that 1,000 vectors are enough to optimize a good classifier.

Table 1: Example of encoding and labeling of interval vectors

P_{CCB}	P_{CB}	P_{CBC}	P_{BCC}	P_{BC}	I_i	Inter.
0.5	0.60	0.00	0.17	0.02	0	時間
0.98	0.96	1.00	0.99	1.00	1	間：
1.00	1.00	1.00	0.71	0.99	1	：三
0.30	0.54	0.01	0.32	0.05	0	三月
0.96	0.85	1.00	0.43	0.47	1	月十
0.00	0.25	0.07	0.49	0.01	0	十日

Using the example from Huang *et al.* (2008), to segment the following Chinese sentence:

時 I_1 間 I_2 ： I_3 三 I_4 月 I_5 十 I_6 日

The corresponding vectors are generated and shown in Table 1.

Note that if an n-gram does not appear in the n-gram database, the probability is assigned automatically 0.5, which means that it offers no detection information for word boundary.

2.2 Relationship to Character Tagging Approach

Character tagging approach models Chinese word segmentation as a character-tag classification problem. Each character in an untagged text is labeled with a tag that represents the position in a word (Xue, 2003). The tag sets usually contains four labels: '*B*' for a character that begins a word; '*M*' for a character that occurs in the middle of a word; '*E*' for a character that ends a word; '*S*' for character that occurs as a single-character word. Therefore, Chinese text with word segmentation information is formulized as follows

$$c_1 T_1 c_2 T_2, ..., c_{n-1} T_{n-1} c_n T_n \qquad T_i \in \{B, M, E, S\}$$

With respect of classification vectors, each character is directly represented by the characters or character n-grams in its surrounding, e.g., whether one character appears in its left position. As a result, the dimension of the vector is extremely high which make this tagging approach takes a very long training time.

Compared to above WBD approach, there seems to be two differences between word boundary decision and character tagging approach: One is category definition (two categories vs. four categories) and the other is feature representation for statistical classification (meta-probabilities vs. character presence).

Actually, the first difference can be discarded if we use only two tags to represent the character positions. There are two corresponding implementations. One is using '*B*' and '*M*' tags, where '*B*' means the character is a beginning of a word, otherwise '*M*'. The other is using '*E*' and '*M*', where '*E*' means the character is an end of a word, otherwise '*M*'.

For example, when we define that a character is assigned 1 when a word boundary is existing after it, the sentence of "共同 创造 美好 的 新 世纪" can be represented as following in the WBD approach.

共 0 同 1 创 0 造 1 美 0 好 1 的 1 新 1 世 0 纪 1

Accordingly, the same representation can be given by using character tags of '*M*' and '*E*'.

共 *M* 同 *E* 创 *M* 造 *E* 美 *M* 好 *E* 的 *E* 新 *E* 世 *M* 纪 *E*

Meanwhile, when we define that a character is assigned 1 when a word boundary is existing before it, the sentence can be represented as following in the WBD approach.

共 1 同 0 创 1 造 0 美 1 好 0 的 1 新 1 世 1 纪 0

Accordingly, the same representation can be given by using character tags of '*M*' and '*B*'.

共 *B* 同 *M* 创 *B* 造 *M* 美 *B* 好 *M* 的 *B* 新 *B* 世 *B* 纪 *M*

Therefore, WBD can certainly be implemented through character tagging approach. But there are two different implementations. The difference mainly due to one special case when the character is a single character word, such as '的' and '新' in the example sentence.

Fortunately, we can use a special type of features to avoid do both two implementations. The special features are tag transition features which are supposed to incorporate the single character word information. That is to say, we consider not only the current character but also its previous tag to do the classification. For example, when classifying the character '新', we use the character features and also use the previous tag (the tag of the character '的') in the classification features.

3 WBD Implementation with Character Tagging using CRF

The segmentation task is to classify each character with a tag of '1' or '0', which represents a word boundary appears after this character or not. There are several classification algorithms which can be applied to do the segmentation, such as maximum entropy (Xue, 2003), conditional random field (CRF) (Tseng *et al.*, 2005) and perceptron algorithm (Jiang *et al.*, 2008). We use CRF learning method as it gives state-of-the-arts performance for word segmentation and can also easily incorporate different types of features (Tseng *et al.*, 2005).

CRF is a statistical sequence modeling framework which aims to compute the following probability of a label sequence for a particular of character string:

$$p_\lambda(Y|W) = \frac{1}{Z(W)} \exp\left(\sum_{t \in T} \sum_k \lambda_k f_k(y_{t-1}, W, t)\right)$$

where $Y = \{y_t\}$ is the label sequence for a character string. Here, $y_t \in \{1, 0\}$ which represents that whether there is a word boundary after the current character or not. W is the sequence of unsegmented characters. $Z(W)$ is a normalization term. f_k is a feature function and t is the index of one character in the string.

Specifically, we use a public tool for CRF implementation: CRF++[1] by Taku Kudo. The feature template is given in Table 2. The unigram and bi-gram features follows the character features which are used in WBD approach by (Huang *et al.*, 2007), i.e., CB, BC, CCB, CBC, and CCB. Third type of transition features is incorporating the segmentation information from single character words. This new type of features has not been carefully studied in previous work (e.g., in the implementation of 2-tag segmentation approach by Zhao *et al.* (2006)). We believe that using this type of features would make the performance of two tags similar to four tags (i.e., '*B*', '*M*', '*E*', and '*S*').

Table 2: Feature template

Type	Features	Function
Character Unigram	C_0, C_1	The single character features
Character Bi-gram	$C_{-1}C_0$, C_0C_1, C_1C_2	The character bi-gram features
Transition	$T_{-1}C_0T_0$, $C_{-1}T_{-1}C_0T_0$, $T_{-1}C_0T_0C_1$	The character adding tag transition features

[1] This tool is available at: *http://crfpp.sourceforge.net/*

4 Experimental Studies

In this section, we would empirically compare the two implementations: WBD with meta-probability classification (Huang *et al.*, 2007) and WBD with character tagging with CRF. Furthermore, we would compare the WBD with character tagging implement with traditional 4-tag character tagging approach.

We use SIGHAN Bakeoff 2 data (Levow, 2006) for experimental studies. The data consists of four different sources: PKU, MSR, CityU, and AS. Their detailed information is given in Table 3. In all experiments, we mainly use F-measure (*F1*) as the performance measurement. *F1* is defined as $F1 = 2PR/(P+R)$ where P is precision and R is recall. Another evaluation measurement is out-of-vocabulary (OOV) recall, which is used to evaluate the ability of OOV word recognition.

Table 3: Corpus Information

Corpus	Abbrv.	Training Size (Words/Types)	Test Size (Words/Types)
Beijing University	PKU	1.1M/55K	104K/13K
Microsoft Research	MSR	2.37M/88K	107K/13K
City University of Hong Kong	CityU	1.46M/69K	41K/9K
Academia Sinica	AS	5.45M/141K	122K/19K

First of all, WBD approach with different implements are tested on the four data sets and the results are shown in Table 4. Specifically, CRF without transition features means using the first and second types of features in Table 2 while CRF adding transition features means using all the three types of features in Table 2. From Table 4, we can see that WBD with meta-probability (Huang *et al.*, 2008) apparently performs worse than WBD with character features. Compared the tagging approach with and without transition features, we can find that transition features are very effective and able to make a improvement of more than 1% on *F1* score in each data set.

Table 4: WBD segmentation results with different implementations (*F1* score)

	Huang *et al.* (2008)	CRF without transition features	CRF adding transition features
PKU	0.895	0.920	0.937
MSR	0.932	0.951	0.961
CityU	0.908	0.932	0.946
AS	0.922	0.942	0.951

For further comparing WBD segmentation approach to the state-of-the-arts approaches, we implement the 4-tag (i.e., '*B*', '*M*', '*E*', and '*S*') character tagging approach with CRF using the same features shown in Table 2. Furthermore, we give some results from most related work Tseng (2005) along with the best performance in Sighan 2005 contest in each data set. All these results are shown in Table 5 where WBD with CRF means WBD approach with CRF adding transition features. Compared to 4-tag approach, WBD approach has shown comparative performances (merely a little worse in MSR and CityU data sets). This result is quite different from those reported by previous work, e.g., Zhao *et al.* (2006) which states that 2-tag segmentation performs much worse than 4-tag segmentation. We think this is mainly because we use the transition features which imply the segmentation information of single character

word. Their implementation of 2-tag approach is similar to our WBD implementation with CRF without transition features. Compared to other state-of-the-arts results from Tseng and Sighan Best, WBD approach with CRF provides comparative performances except in the PKU data set. We think the worse performance in PKU is because the digital character (e.g., 1, 2, 3) encoding are different in training data and testing data (halfwidth vs. fullwidth forms). Tseng and some Sighan systems consider the differences while we do not. We strictly follow the close-test instructions. Note that there are some other related work which perhaps presents better results, e.g., Jiang *et al.* (2008) and Zhao *et al.* (2006). However, they often use much more features or some digital and punctuation features. Therefore, the performance comparison to them becomes quite unfair.

Table 5: Comparison between the performance of WBD with CRF and state-of-the-arts results (*F1* score)

	WBD with CRF	4-tag character tagging	Tseng (2005)	Sighan Best
PKU	0.937	0.938	0.950	0.950
MSR	0.961	0.966	0.964	0.964
CityU	0.946	0.951	0.943	0.943
AS	0.951	0.952	0.947	0.952

Table 6 shows the OOV recall results of different approaches. Apparently, WBD using character tagging with CRF performs much better than WBD by Huang *et al.* (2008). But it performs a little worse than 4-tag character tagging approach in three data sets.

Table 6: OOV recall results of different approaches

	WBD by Huang *et al.* (2008)	WBD with CRF	4-tag character tagging
PKU	0.382	0.628	0.596
MSR	0.467	0.615	0.684
CityU	0.500	0.692	0.728
AS	0.504	0.652	0.669

Finally, let's see the time and space requirement of WBD approach and 4-tag character tagging approach. The training time and peer memory space is tested in each data set and the results are given in Table 7. From this table, we can see that WBD with CRF need only half time and memory space compared to 4-tag character tagging. In our work, we implement WBD with CRF is actually using 2-tag character tagging and thus the computational cost of WBD with CRF might be half as much as the cost of 4-tag character tagging.

Table 7: The time and space requirement of WBD approach and 4-tag character tagging approach

	WBD with CRF		4-tag character tagging	
	Training Time	Memory	Training Time	Memory
PKU	14min	0.9G	37min	1.8G
MSR	40min	1.5G	108min	3.1G
CityU	25min	1.2G	52min	2.4G
AS	150min	2.6G	350min	5.7G

5 Conclusion and Future Work

In this work, we analyze the relationship between WBD (Huang *et al.*, 2007) and 4-tag character tagging approach for Chinese word segmentation. There are two main differences between them: One is category definition (two categories vs. four categories) and the other is feature representation for statistical classification (meta-probabilities vs. character presence). Experimental results show that character presence is definitely more effective than meta-probabilities. Therefore, we implement WBD using character tagging approach (character presence features) with CRF and find that our implement can achieve comparative performance compared to 4-tag character tagging approach. This conclusion is quite different from most previous work. We think this is mainly due to our usage of the transition features which can imply the segmentation information of single character word. Moreover, our WBD implement can save about half training time and memory space, which makes it more practical for real applications.

References

Chang, P., M. Galley and C. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. In P*roceedings of the 3rd Workshop on Statistical Machine Translation (SMT'08)*.

Huang, C., P. Šimon, S. Hsieh and L. Prevot. 2007. Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak identification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-07)*.

Huang, C., T. Yo, P. Šimon and S. Hsieh. 2008. A Realistic and Robust Model for Chinese Word Segmentation. In *Proceedings of the Conference of Computational Linguistics and Speech Processing (ROCLING-08)*.

Jiang, W., L. Huang, Q. Liu and Y. Lu. 2008. A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-08)*.

Levow, G. 2006. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-06)*.

Ng, H. and J. Low. 2004. Chinese Part-of-speech Tagging: One-at-a-time or All-at-once? Word-Based or Character-based. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*.

Sun, M., D. Xu, B. Tsou and H. Lu. 2006. An Integrated Approach to Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of The International Conference on the Computer Processing of Oriental Languages (ICCPOL-06)*.

Tseng, H., P. Chang, G. Andrew, D. Jurafsky and C. Manning. 2005. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing (SIGHAN-05)*.

Xue, N. 2003. Chinese Word Segmentation as Character Tagging. *Computational Linguistics and Chinese Language Processing*, 8 (1). pages 29-48.

Zhao, H., C. Huang, M. Li and B. Lu. 2006. Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC-06)*