Word Segmentation on Chinese Mirco-Blog Data with a Linear-Time Incremental Model

Kaixu Zhang † **Maosong Sun** ‡ **Changle Zhou** †
†Xiamen University, Fujian Province 361005, China
‡State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China
kareyzhang@gmail.com sunmaosong@gmail.com dozero@xmu.edu.cn

Abstract

This paper describes the model we designed for the word segmentation bake-off on Chinese micro-blog data in the 2nd CIPS-SIGHAN joint conference on Chinese language processing. We presented a linear-time incremental model for word segmentation where rich features including character-based features, word-based features as well as other possible features can be easily employed. We report the performances of our model on four datasets in the SIGHAN bake-off 2005. After adding more features designed for the micro-blog data, the performance of our model is further improved. The F-score of our model for this bake-off is 0.9478 and 44.88% of the sentences are segmented correctly.

1 Introduction

Chinese word segmentation is an important and fundamental task for Chinese language processing. General-purpose word segmentation is widely studied. Micro-blog-related topic emergences and becomes a new research topic in recent years. Therefore researchers pay more and more attention to the word segmentation model for Chinese micro-blog data.

Motivated by the linear-time incremental parser proposed by Huang and Sagae (2010) and the word-based word segmentation model proposed by Zhang and Clark (2011), first we presented a linear-time incremental word segmentation model. Various features including character-based features and word-based features can be employed while exponentially many segmented results can be tested in linear-time. We report the performances of our model on four datasets in the SIGHAN bake-off 2005.

One of the difficulties of training word segmentation model on micro-blog data is the lack of annotated micro-blog data (only 500 sentences of micro-blog data are provided and used by us). Following the annotation adaptation method proposed by Jiang et al. (2009), we train a general-purpose joint word segmentation and part-of-speech tagging model using People's Daily corpus. Then, the decoding results of such a model are used as features in the final word segmentation model for micro-blog data.

Moreover, various lexicon features such as dictionaries and word list of idioms are employed to segment micro-blog data. Preprocessing is also conducted to deal with URLs and special characters.

Finally, The F-score of our model for the bake-off is 0.9478 and 44.88% of the sentences are segmented correctly. The performance of our method is still far from perfect. The lack of segmented micro-blog data is one of the bottlenecks of our model. If more training data is provided, our model can reach better performance.

2 The Linear-Time Incremental Word Segmentation Model

2.1 Word Segmentation Definition

First, we give a formal general definition of word segmentation.

A raw sentence X is a Chinese sentence where no spaces are presented to separate words, while a segmented sentence Y is a sentence in which words are separated by spaces. For example, "材料利用率高" is a raw sentence, and "材料 利用率 高" is one of the possible segmented sentences corresponding to the raw sentence.

Given a raw sentence X, a word segmentation model needs to find a segmented sentence $\hat{Y}$ among all possible segmented sentences GEN(X) corresponding to the raw sentence. This can be

Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 41–46,
Tianjin, China, 20-21 DEC. 2012

seen as an optimization problem:

$$\hat{Y} = \underset{Y \in \text{GEN}(x)}{\arg\max}\, f(Y, \Lambda) \quad (1)$$

where the objective function $f(Y, \Lambda)$ is used to evaluate segmented sentences and Λ is the parameter.

In the following subsections, we will describe the detail of this function and how to learn the parameter.

2.2 Word Segmentation as Action Sequence Generation

In this paper, word segmentation is treated as action sequence generation. Each action is corresponding to a character interval of the input sentence. For an input sentence of $|X|$ characters, the corresponding action sequence $A = (a_0, \ldots, a_{|X|})$ has a length of $|X| + 1$ (including the "intervals" at very beginning and very end of the sentence). There are two kinds of actions ($a_i \in \{\mathbf{s}, \mathbf{c}\}$), namely separate (denoted as **s**) and combine (denoted as **c**). The action $a_i = \mathbf{s}$ means that the i-th character and the $i+1$-th character in the input sentence are belong to two separated words; while $a_i = \mathbf{c}$ means that they are belong to the same word.

Given A, the corresponding segmented sentence Y is determined and denoted as Y_A. For example, for the input sentence "材料利用率高", the action sequence $(\mathbf{s}, \mathbf{c}, \mathbf{s}, \mathbf{c}, \mathbf{s}, \mathbf{s}, \mathbf{s})$ could generate a segmented sentence Y_A as "材料 利用 率高".

The problem of finding a best segmented sentence is now equivalent to the problem of generating a best action sequence.

We introduce $S = (s_0, \ldots, s_{|X|})$ determined by A as a sequence of statuses to generate feature vectors for the action sequence and then evaluate any segmented sentence Y_A as

$$f(Y_A, \Lambda) = \sum_{i=0}^{|X|+1} \Phi(s_i, X) \cdot \Lambda_{a_i}^T \quad (2)$$

where $\Phi(s_i, X)$ is a feature vector generated by the input and status s_i corresponding to action a_i. And $\Lambda_{\mathbf{s}}$ and $\Lambda_{\mathbf{c}}$ are two weight vectors for two kinds of actions.

The status sequence S can be defined in different ways. In this paper, we define it as follows.

A status s_i in S is defined as a tuple $\langle i, u_i, v_i \rangle$, where u_i is the index of the last **s** action, and v_i is the index of the second last **s** action. Thus given A, s_i can be formally recursively calculated as

$$s_i = \begin{cases} \langle i, -1, -1 \rangle & \text{if } i = 0 \\ \langle i, i-1, u_{i-1} \rangle & \text{if } a_{i-1} = \mathbf{s} \\ \langle i, u_{i-1}, v_{i-1} \rangle & \text{if } a_{i-1} = \mathbf{c} \end{cases} \quad (3)$$

Following Huang and Sagae (2010), the generation of the action sequence can also be formalized as a deductive system described in Figure 2.2.

The next subsection will describe the feature vector $\Phi(s_i, X)$ in detail.

$$\text{input} \quad X$$

$$\text{axiom} \quad \langle 0, -1, -1 \rangle : 0$$

$$\mathbf{s} \quad \frac{\langle i, u, \_ \rangle : c}{\langle i+1, i, u \rangle : c + \sigma}$$

$$\mathbf{c} \quad \frac{\langle i, u, v \rangle : c}{\langle i+1, u, v \rangle : c + \gamma}$$

$$\text{goal} \quad \langle |X| + 1, \_, \_ \rangle : c$$

Figure 1: The deductive system used to describe our model. In this system, i is the step, c is the cost, $\sigma = \Phi(s_i, X) \cdot \Lambda_{\mathbf{s}}^T$ is the **s** cost and $\gamma = \Phi(s_i, X) \cdot \Lambda_{\mathbf{c}}^T$ is the **c** cost. The best derivation is the derivation of the goal with the highest cost.

Atom features	Description
x_j	characters in X
a_{i-1}, a_{i-2}	last two actions
$\mathbf{w}_0$	current (partial) word
$\mathbf{w}_{-1}$	last determined word

Table 2: Atom features for the i-th action a_i

2.3 Feature Templates

We define feature vectors by using feature templates. First, atom features are generated based on s_i and X. All the feature templates can then be generated by using atom features.

Atom features are shown in Table 2. The last two actions a_{i-1} and a_{i-2} can be determined by the status s_i. The (partial) word $\mathbf{w}_0$ is the string between the index of last **s** action u_i and the current position i.

Feature templates are defined as tuples and shown in Table 1. $|\mathbf{w}|$ is the length of the word $\mathbf{w}$. $\mathbf{w}[0]$ and $\mathbf{w}[-1]$ are the first and last character

action-based	$\langle$**a-1**, $a_{i-2}, a_{i-1}\rangle$
character-based	$\langle$**c-1**, $x_{i-2}, a_{i-1}\rangle$, $\langle$**c-2**, $x_{i-1}, a_{i-1}\rangle$, $\langle$**c-3**, $x_i, a_{i-1}\rangle$
	$\langle$**c-4**, $x_{i-3}, x_{i-2}, a_{i-1}\rangle$, $\langle$**c-5**, $x_{i-2}, x_{i-1}, a_{i-1}\rangle$,
	$\langle$**c-6**, $x_{i-1}, x_i, a_{i-1}\rangle$, $\langle$**c-7**, $x_i, x_{i+1}, a_{i-1}\rangle$
word-based	$\langle$**w-1**, $\mathbf{w}_0\rangle$, $\langle$**w-2**, $\|\mathbf{w}_0\|\rangle$
	$\langle$**w-3**, $\|\mathbf{w}_0\|, \mathbf{w}_0[0]\rangle$, $\langle$**w-4**, $\|\mathbf{w}_0\|, \mathbf{w}_0[-1]\rangle$, $\langle$**w-5**, $\mathbf{w}_0[0], \mathbf{w}_0[-1]\rangle$
	$\langle$**w-6**, $\mathbf{w}_{-1}[-1], \mathbf{w}_0[-1]\rangle$, $\langle$**w-7**, $\|\mathbf{w}_{-1}\|, \mathbf{w}_0\rangle$, $\langle$**w-8**, $\mathbf{w}_{-1}, \|\mathbf{w}_0\|\rangle$
	$\langle$**w-9**, $\mathbf{w}_0[0], x_i\rangle$, $\langle$**w-10**, $\mathbf{w}_0[-1], x_i\rangle$

Table 1: Feature templates

of word **w**, respectively. Each tuple is corresponding to one dimension of the feature vector and the value of that dimension will be set to 1 if this corresponding feature was generated.

There are action-based, character-based and word-based templates. Note that when only action-based and character-based templates are used, these feature templates are equivalent to the templates used by conventional word segmentation models based on character tagging (Zhang et al., 2011). And the word-based features are mainly based on the work by Zhang and Clark (2011).

2.4 Decoding and Learning Algorithms

We apply the decoding algorithm used by Huang and Sagae (2010).

Beam search is used in the decoding algorithm, while different hypotheses with the same status at a certain step will be merged in a dynamic programming manner. This decoding algorithm can efficiently search exponentially many hypotheses in linear-time ($O(nb)$ where b is the width of the beam). Comparatively, the time complexity of the decoding algorithm using fully dynamic programming is $O(n^3)$ (or $O(nL^2)$ if the max length of words L is specified).

The parameter Λ is trained using an average perceptron algorithm (Collins, 2002). We also tried early update (Collins and Roark, 2004) in the learning algorithm. Although it is reported that early update helps the learning of parsers, we do not observe that early update helps the learning of word segmentation models. So we do not implement early update in our experiments.

3 Word Segmentation for Micro-Blog Data

In order to segment the micro-blog data better, we modified the word segmentation model described in the last section by adding a preprocessing and more features.

We just perform feature engineering manually for the development to decide which feature is useful for segmenting micro-blog data [1].

3.1 Preprocessing

A rule-based preprocessing is conducted before the statistical model. This preprocessing is mainly used to reduce the search space of the statistical model by assigning the action a_i of certain position before the decoding algorithm. Thus the decoding algorithm will only search either hypotheses that $a_i = \mathbf{s}$ or hypotheses that $a_i = \mathbf{c}$.

URLs and other micro-blog-specified characters (such as "@" means "at somebody" and "#" means to annotate a tag) are first recognized. The boundaries of these components are assigned to **s**, while the inner character intervals of the URLs are assigned to **c**.

Likewise, the punctuations (such as Chinese full stop "。" and comma "，") are recognized and the boundaries of these are assigned to **s**. The intervals between two Arabic numbers or two Latin letters are assigned to **c**.

White spaces can also be found in the raw micro-blog data between two English words or at the end of a micro-blog user's name after the '@' character. The preprocessing will remove these white spaces and assigned **s** for the left character intervals.

3.2 Character-Type-Based Features

Since there are more non-standard uses of non-Chinese characters in micro-blog data than in news data and adding character-type-based features can improve the performance of general-

[1] Word-based feature templates in Table 1 are also modified slightly for the word segmentation model for micro-blog data.

Method	Dataset			
	AS	CityU	MSR	PKU
Best05	0.952	0.943	0.964	0.950
	(Asahara et al., 2005)	(Tseng et al., 2005)		(Chen et al., 2005)
(Wang et al., 2010)	0.956	0.956	0.972	0.957
(Zhang and Clark, 2011)	0.954	0.951	0.973	0.944
(Sun et al., 2012)	NA	0.948	0.974	0.954
Our model	0.953	0.948	0.973	0.952

Table 3: F-scores of our model and models in related work on SIGHAN 05 bake-off data

purpose word segmentation model (Zhao et al., 2006), we employ character-type-based features.

We define a function type(x_i) that returns the type of the characters

$$\text{type}(x_i) = \begin{cases} \mathbf{C} & \text{if } x_i \text{ is a Chinese character} \\ \mathbf{L} & \text{if } x_i \text{ is a Latin letter} \\ \mathbf{A} & \text{if } x_i \text{ is a Arabic numeric character} \\ x_i & \text{otherwise} \end{cases} \tag{4}$$

The additional feature templates that we use are $\langle$**ct-1**, type$(x_i)\rangle$, $\langle$**ct-2**, type$(x_{i-1})\rangle$, $\langle$**ct-3**, type$(x_{i+1})\rangle$, $\langle$**ct-4**, type(x_{i-1}), type$(x_i)\rangle$ and $\langle$**ct-5**, type(x_i), type$(x_{i+1})\rangle$.

3.3 Lexical Features

Lexical features are used as additional word-based features for word segmentation for micro-blog data. Each lexical feature template $\langle$**lex-**$k, \text{lex}_k(\mathbf{w}_0)\rangle$ is based on a function whose variable is a word.

Since we have various lexical resources, we can define several functions lex_k to create different lexical feature templates. If the lexical resource is just a word list, the $\text{lex}_k(\mathbf{w}_0)$ could just return a binary value to indicate whether this word $\mathbf{w}_0$ is in the word list or not. If the lexical resource is about the frequencies of words, $\text{lex}_k(\mathbf{w}_0)$ could return $\log_2(\text{freq}(\mathbf{w}_0)+1)$ where $\text{freq}(\mathbf{w}_0)$ is the frequency of word $\mathbf{w}_0$.

We use several word lists to add lexical feature templates, including a word list of idioms from Sun (2011), word lists based on People's Daily corpus, Yuwei Corpus and Tsinghua Treebank. We also use words with frequencies counted from the three mentioned segmented corpora.

Additionally, we add another lexical feature template based on whether these four characters x_{u_i}, x_{u_i+1}, x_{u_i+2} and x_{u_i+3} form an idiom.

3.4 Tagger-Based Features

The annotated micro-blog data contains only 500 micro-blogs. So more annotated data are required. We train a character-based joint word segmentation and part-of-speech tagging model using the People's Daily corpus (Zhang, 2012)[2], and then use the decoding results of this model as features for the word segmentation model for the micro-blog data.

Three templates $\langle$**tb-1**, $a'_i\rangle$, $\langle$**tb-2**, a'_i, POS$_{i-1}\rangle$ and $\langle$**tb-3**, a'_i, POS$_i\rangle$ are added. a'_i is the action based on the results of the tagger, and POS$_i$ is the part-of-speech tag of the word that x_i belongs to.

4 Experiments

We report the performances of our model on four SIGHAN05 datasets (Emerson, 2005). Then we report the performance our model on the micro-blog data. We use 5-fold cross validation for the development and use the whole dataset to train the final model for the test.

The F-score is used to evaluate the performance, which is the harmonic mean of precision (percentage of words that are correctly segmented in the results) and recall (percentage of words that are correctly segmented in the gold standard).

The results of our model and related work on the SIGHAN05 datasets are listed in Table 3.

The results of the micro-blog data are listed in Table 4. The first row is the final performance on the test data, while the following rows show the performances with different feature sets for the cross validation using 500 micro-blog sentences. We can see that the additional features of the micro-blog data improve the performance.

[2]The code we use is a part of the tool THULAC (Tsinghua University - Lexical Analyzer for Chinese) `http://nlp.csai.tsinghua.edu.cn/thulac/`.

	F-score
All features for test	0.9478
All features for cross validation	0.9413
w/o character-type-based features	0.9383
w/o lexical features	0.9201
w/o tagger-based features	0.9310

Table 4: Experiment results of our model on the micro-blog data

For the annotated micro-blog data for the training is quite limited, the lexical features and tagger-based features are important for the performance. Note that the F-score for the test is better than the F-score for the cross validation. This may caused by that the training set for the former model is one-quarter larger. It may imply that the performance of our model is limited by the size of the training data and the performance of our model will be improved when larger training data was provided.

5 Discussion and Conclusion

In this paper, we describe the model we designed for the word segmentation bake-off on Chinese micro-blog data in the 2nd CIPS-SIGHAN joint conference on Chinese language processing. We presented a linear-time incremental word segmentation model in which various features can be easily employed. After employing more features of the micro-blog data, the performance of our model is further improved. The final F-score of our model on the test set is 0.9478 and 44.88% of the micro-blogs are segmented correctly.

The performance of our model is still far from perfect. Word segmentation for micro-blog data is not as good as word segmentation for news data (see Table 3 and Table 4). More manually annotated data or employing semi-supervised method can be used to improve the performance. We also notice that outputting inconsistence words is a problem for statistical word segmentation models. Therefore a post-processing could be used to adjust the output for better performance. We spend much time comparing the performances of combinations of different feature templates. A more sophisticated method is needed for the selection of feature templates.

Acknowledgments

The authors want to thank ZHANG Junsong from the cognitive lab and SHI Xiaodong, CHEN Yidong and SU Jinsong from the NLP lab of Xiamen University for the support of experiments.

The authors are supported by NSFC under Grant No. 61133012 and 61273338.

References

M. Asahara, K. Fukuoka, A. Azuma, C. L. Goh, Y. Watanabe, Y. Matsumoto, and T. Tsuzuki. 2005. Combination of machine learning methods for optimum chinese word segmentation. In *Proc. Fourth SIGHAN Workshop on Chinese Language Processing*, pages 134–137.

A. Chen, Y. Zhou, A. Zhang, and G. Sun. 2005. Unigram language model for chinese word segmentation. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 138–141.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 111–118, Barcelona, Spain, July.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. pages 1–8.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 123–133. Jeju Island, Korea.

Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086, Uppsala, Sweden, July. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging - a case study. In *Proceedings of the 47th ACL*, pages 522–530, Suntec, Singapore, August. Association for Computational Linguistics.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262, Jeju Island, Korea, July. Association for Computational Linguistics.

Weiwei Sun. 2011. A stacked sub-word model for joint chinese word segmentation and part-of-speech

tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1385–1394, Portland, Oregon, USA, June. Association for Computational Linguistics.

H. Tseng, 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*, pages 168–171. Jeju Island, Korea.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A character-based joint model for chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1173–1181, Beijing, China, August. Coling 2010 Organizing Committee.

Y. Zhang and S. Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, (Early Access):1–47.

Kaixu Zhang, Ruining Wang, Ping Xue, and Maosong Sun. 2011. Extract chinese unknown words from a large-scale corpus using morphological and distributional evidences. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 837–845, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Kaixu Zhang. 2012. *Study on Chinese Word Segmentation and Part-of-Speech Tagging with Compact Representations*. Ph.D. thesis, Tsinghua University.

H. Zhao, C. N. Huang, and M. Li. 2006. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165. Sydney: July.