Exploring Representations from Unlabeled Data with Co-training for Chinese Word Segmentation

Longkai Zhang Houfeng Wang* Xu Sun Mairgup Mansur
Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China
zhlongk@qq.com, wanghf@pku.edu.cn, xusun@pku.edu.cn, mairgup@gmail.com,

Abstract

Nowadays supervised sequence labeling models can reach competitive performance on the task of Chinese word segmentation. However, the ability of these models is restricted by the availability of annotated data and the design of features. We propose a scalable semi-supervised feature engineering approach. In contrast to previous works using pre-defined task-specific features with fixed values, we dynamically extract representations of label distributions from both an in-domain corpus and an out-of-domain corpus. We update the representation values with a semi-supervised approach. Experiments on the benchmark datasets show that our approach achieve good results and reach an f-score of 0.961. The feature engineering approach proposed here is a general iterative semi-supervised method and not limited to the word segmentation task.

1 Introduction

Chinese is a language without natural word delimiters. Therefore, Chinese Word Segmentation (CWS) is an essential task required by further language processing. Previous research shows that sequence labeling models trained on labeled data can reach competitive accuracy on the CWS task, and supervised models are more accurate than unsupervised models (Xue, 2003; Low et al., 2005). However, the resource of manually labeled training corpora is limited. Therefore, semi-supervised learning has become one of the most natural forms of training for CWS. Traditional semi-supervised methods focus on adding new unlabeled instances to the training set by a given criterion. The possible mislabeled instances, which are introduced from the automatically labeled raw data, can hurt the performance and not easy to exclude by setting a sound selecting criterion.

In this paper, we propose a simple and scalable semi-supervised strategy that works by providing semi-supervision at the level of representation. Previous works mainly assume that context features are helpful to decide the potential label of a character. However, when some of the context features do not appear in the training corpus, this assumption may fail. An example is shown in table 1. Although the context of "水" and "篮" is totally different, they share a homogeneous structure as "verb-noun". Therefore. A much better way is to map the context information to a kind of representation. More precisely, the mapping should let the similar contexts map to similar representations, while let the distinct contexts map to distinct representations.

	吃水果	打篮球
Label	B	B
Character	吃 水 果	打 篮 球
Context Features	C-1= 吃 C0= 水 C1= 果	C-1= 打 C0= 篮 C1= 球

Table 1: Example of the context of "水" in "吃水果 (Eat fruits)" and the context of "篮" in "打篮球 (Play basketball)"

We use the label distribution information that

*Corresponding author

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 311–321,
Seattle, Washington, USA, 18-21 October 2013.

is extracted from the unlabeled corpus as this representation to enhance the supervised model. We add "pseudo-labels" by tagging the unlabeled data with the trained model on the training corpus. These "pseudo-labels" are not accurate enough. Therefore, we use the label distribution, which is much more accurate.

To accurately calculate the precise label distribution, we use a framework similar to the co-training algorithm to adjust the feature values iteratively. Generally speaking, unlabeled data can be classified as in-domain data and out-of-domain data. In previous works these two kinds of unlabeled data are used separately for different purposes. In-domain data is mainly used to solve the problem of data sparseness (Sun and Xu, 2011). On the other hand, out-of domain data is used for domain adaptation (Chang and Han, 2010). In our work, we use in-domain and out-of-domain data together to adjust the labels of the unlabeled corpus.

We evaluate the performance of CWS on the benchmark dataset of Peking University in the second International Chinese Word Segmentation Bakeoff. Experiment results show that our approach yields improvements compared with the state-of-art systems. Even when the labeled data is insufficient, our methods can still work better than traditional methods. Compared to the baseline CWS model, which has already achieved an f-score above 0.95, we further reduce the error rate by 15%.

Our method is not limited to word segmentation. It is also applicable to other problems which can be solved by sequence labeling models. We also applied our method to the Chinese Named Entity Recognition task, and also achieved better results compared to traditional methods.

The main contributions of our work are as follows:

- We proposed a general method to utilize the label distribution given text contexts as representations in a semi-supervised framework. We let the co-training process adjust the representation values from label distribution instead of using manually predefined feature templates.
- Compared with previous work, our method achieved a new state-of-art accuracy on the CWS task as well as on the NER task.

The remaining part of this paper is organized as follows. Section 2 describes the details of the problem and our algorithm. Section 3 describes the experiment and presents the results. Section 4 reviews the related work. Section 5 concludes this paper.

2 System Architecture

2.1 Sequence Labeling

Nowadays the character-based sequence labeling approach is widely used for the Chinese word segmentation problem. It was first proposed in Xue (2003), which assigns each character a label to indicate its position in the word. The most prevalent tag set is the BMES tag set, which uses 4 tags to carry word boundary information. This tag set uses B, M, E and S to represent the Beginning, the Middle, the End of a word and a Single character forming a word respectively. We use this tag set in our method. An example of the "BMES" representation is shown in table 2.

Character:	我	爱	北	京	天	安	门
Tag:	S	S	B	E	B	M	E

Table 2: An example for the "BMES" representation. The sentence is "我爱北京天安门" (I love Beijing Tian-an-men square), which consists of 4 Chinese words: "我" (I), "爱" (love), "北京" (Beijing), and "天安门" (Tian-an-men square).

2.2 Unlabeled Data

Unlabeled data can be divided into in-domain data and out-of-domain data. In previous works, these two kinds of unlabeled data are used separately for different purposes. In-domain data only solves the problem of data sparseness (Sun and Xu, 2011). Out-of domain data is used only for domain adaptation (Chang and Han, 2010). These two functionalities are not contradictory but complementary. Our study shows

that by correctly designing features and algorithms, both in-domain unlabeled data and out-of-domain unlabeled data can work together to help enhancing the segmentation model. In our algorithm, the dynamic features learned from one corpus can be adjusted incrementally with the dynamic features learned from the other corpus.

As for the out-of-domain data, it will be even better if the corpus is not limited to a specific domain. We choose a Chinese encyclopedia corpus which meets exactly this requirement. We use the corpus to learn a large set of informative features. In our experiment, two different views of features on unlabeled data are considered:

Static Statistical Features (SSFs): These features capture statistical information of characters and character n-grams from the unlabeled corpus. The values of these features are fixed during the training process once the unlabeled corpus is given.

Dynamic Statistical Features (DSFs): These features capture label distribution information from the unlabeled corpus given fixed text contexts. As the training process proceeds, the value of these features will change, since the trained tagger at each training iteration may assign different labels to the unlabeled data.

2.3 Framework

Suppose we have labeled data L, two unlabeled corpora U_a and U_b (one is an in-domain corpus and the other is an out-of-domain corpus). Our algorithm is shown in Table 3.

During each iteration, we tag the unlabeled corpus U_a using T_b to get pseudo-labels. Then we extract features from the pseudo-labels. We use the label distribution information as dynamic features. We add these features to the training data to train a new tagger T_a. To adjust the feature values, we extract features from one corpus and then apply the statistics to the other corpus. This is similar to the principle of co-training (Yarowsky, 1995; Blum and Mitchell, 1998; Dasgupta et al., 2002). The difference is that there are not different views of features, but different kinds of unlabeled data. Detailed description of features is given in the next section.

Algorithm
Init:
Using baseline features only:
Train an initial tagger T_0 based on L ()
Label U_a and U_b individually using T_0
BEGIN LOOP:
Generate DSFs from tagged U_a
Augment L with DSFs to get L_a
Generate DSFs from tagged U_b
Augment L with DSFs to get L_b
Using baseline features, SSFs and DSFs:
Train new tagger T_a using L_a
Train new tagger T_b using L_b
Label U_a using T_b
Label U_b using T_a
LOOP until performance does not improve
RETURN the tagger which is trained with in-domain features.

Table 3: Algorithm description

2.4 Features

2.4.1 Baseline Features

Our baseline feature templates include the features described in previous works (Sun and Xu, 2011; Sun et al., 2012). These features are widely used in the CWS task. To be convenient, for a character c_i with context $\ldots c_{i-1}c_ic_{i+1}\ldots$, its baseline features are listed below:

- Character uni-grams: c_k $(i-3<k<i+3)$
- Character bi-grams: c_kc_{k+1} $(i-3<k<i+2)$
- Whether c_k and c_{k+1} are identical $(i-2<k<i+2)$
- Whether c_k and c_{k+2} are identical $(i-4<k<i+2)$

The last two feature templates are designed to detect character reduplication, which is a morphological phenomenon in Chinese language. An example is "十全十美" (Perfect), which is a Chinese idiom with structure "ABAC".

2.4.2 Static statistical features

Statistical features are statistics that distilled from the large unlabeled corpus. They are proved useful in the Chinese word segmentation task. We define Static Statistical Features (SSFs) as features whose value do not change during the training process. The SSFs in our approach includes Mutual information, Punctuation information and Accessor variety. Previous works have already explored the functions of the three static statistics in the Chinese word segmentation task, e.g. Feng et al. (2004); Sun and Xu (2011). We mainly follow their definitions while considering more details and giving some modification.

Mutual information

Mutual information (MI) is a quantity that measures the mutual dependence of two random variables. Previous works showed that larger MI of two strings claims higher probability that the two strings should be combined. Therefore, MI can show the tendency of two strings forming one word. However, previous works mainly focused on the balanced case, i.e., the MI of strings with the same length. In our study we find that, in Chinese, there remains large amount of imbalanced cases, like a string with length 1 followed by a string with length 2, and vice versa. We further considered the MI of these string pairs to capture more information.

Punctuation information

Punctuations can provide implicit labels for the characters before and after them. The character after punctuations must be the first character of a word. The character before punctuations must be the last character of a word. When a string appears frequently after punctuations, it tends to be the beginning of a word. The situation is similar when a string appears frequently preceding punctuations. Besides, the probability of a string appears in the corpus also affects this tendency. Considering all these factors, we propose "punctuation rate" (PR) to capture this information. For a string with length len and probability p in the corpus, we define the left punctuation rate LPR_{len} as the number of times the string appears after punctuations, divided by p. Similarly, the right punctuation rate RPR_{len} is defines as the number of times it appears preceding punctuations divided by its probability p. The length of string we consider is from 1 to 4.

Accessor variety

Accessor variety (AV) is also known as letter successor variety (LSV) (Harris, 1955; Hafer and Weiss, 1974). If a string appears after or preceding many different characters, this may provide some information of the string itself. Previous work of Feng et al. (2004), Sun and Xu (2011) used AV to represent this statistic. Similar to punctuation rate, we also consider both left AV and right AV. For a string s with length l, we define the left accessor variety (LAV) as the types of distinct characters preceding s in the corpus, and the right accessor variety (RAV) as the types of distinct characters after s in the corpus. The length of string we consider is also from 1 to 4.

2.4.3 Dynamic statistical features

The unlabeled corpus lacks precise labels. We can use the trained tagger to give the unlabeled data "pseudo-labels". These labels cannot guarantee an acceptable precision. However, the label distribution will not be largely affected by small mistakes. Using the label distribution information is more accurate than using the pseudo-labels directly.

Based on this assumption, we propose "dynamic statistical features" (DSFs). The DSFs are intended to capture label distribution information given a text context. The word "Dynamic" is in accordance with the fact that these feature values will change during the training process.

We give a formal description of DSFs. Suppose there are K labels in our task. For example, $K = 4$ if we take BMES labeling method. We define the whole character sequence with length n as $X = (x_1, x_2 \cdots x_j \cdots x_n)$. Given a text context C_i, where i is current character position, the DSFs can be represented as a list,

$$DSF(C_i) = (DSF(C_i)_1, \cdots, DSF(C_i)_K)$$

Each element in the list represents the probability of the corresponding label in the distribution.

For convenience, we further define function 'count(condition)' as the total number of times a 'condition' is true in the unlabeled corpus. For example, count (current='a') represents the times the current character equals 'a', which is exactly the number of times character 'a' appears in the unlabeled corpus.

According to different types of text context C_i, we can divide DSFs into 3 types:

1.Basic DSF

For Basic DSF of C_i, we define $D(C_i)$:

$$D(C_i) = (D(C_i)_1, \ldots, D(C_i)_K)$$

We define Basic DSF with current character position i, text context C_i and label l (the lth dimension in the list) as:

$$\begin{aligned} D(C_i)_l &= P(y = l | C_i = x_i) \\ &= \frac{count(C_i = x_i \wedge y = l)}{count(C_i = x_i)} \end{aligned}$$

In this equation, the numerator counts the number of times current character is x_i with label l. The denominator counts the number of times current character is x_i.

We use the term "Basic" because this kind of DSFs only considers the character of position i as its context. The text context refers to the current character itself. This feature captures the label distribution information given the character itself.

2.BigramDSF

Basic DSF is simple and very easy to implement. The weakness is that it is less powerful to describe word-building features. Although characters convey context information, characters themselves in Chinese is sometimes meaningless. Character bi-grams can carry more context information than uni-grams. We modify Basic DSFs to bi-gram level and propose Bigram DSFs.

For Bigram DSF of C_i, we define $B(C_i)$:

$$B(C_i) = (B(C_i)_1, \ldots, B(C_i)_K)$$

We define Bigram DSF with current character position i, text context C_i and label l (the lth dimension in the list) as:

$$\begin{aligned} B(C_i)_l &= P(y = l | C_i = x_{i-j}x_{i-j+1}) \\ &= \frac{count(C_i = x_{i-j}x_{i-j+1} \wedge y = l)}{count(C_i = x_{i-j}x_{i-j+1})} \end{aligned}$$

j can take value 0 and 1.

In this equation, the numerator counts the number of times current context is $x_{i-j}x_{i-j+1}$ with label l. The denominator counts the number of times current context is $x_{i-j}x_{i-j+1}$.

3.WindowDSF

Considering Basic DSF and Bigram DSF only might cause the over-fitting problem, therefore we introduce another kind of DSF. We call it Window DSF, which considers the surrounding context of a character and omits the character itself.

For Window DSF, we define $W(C_i)$:

$$W(C_i) = (W(C_i)_1, \ldots, W(C_i)_K)$$

We define Window DSF with current character position i, text context C_i and label l (the lth dimension in the list) as:

$$\begin{aligned} W(C_i)_l &= P(y = l | C_i = x_{i-1}x_{i+1}) \\ &= \frac{count(C_i = x_{i-1}x_{i+1} \wedge y = l)}{count(C_i = x_{i-1}x_{i+1})} \end{aligned}$$

In this equation, the numerator counts the number of times current context is $x_{i-1}x_{i+1}$ with label l. The denominator counts the number of times current context is $x_{i-1}x_{i+1}$.

2.4.4 Discrete features VS. Continuous features

The statistical features may be expressed as real values. A more natural way is to use discrete values to incorporate them into the sequence labeling models . Previous works like Sun and Xu (2011) solve this problem by setting thresholds and converting the real value into boolean values. We use a different method to solve this, which does not need to consider tuning thresholds. In our method, we process static and dynamic statistical features using different strategies.

For static statistical value:

For mutual information, we round the real value to their nearest integer. For punctuation rate and accessor variety, as the values tend to be large, we first get the log value of the feature and then use the nearest integer as the corresponding discrete value.

For dynamic statistical value:

Dynamic statistical features are distributions of a label. The values of DSFs are all percentage values. We can solve this by multiply the probability by an integer N and then take the integer part as the final feature value. We set the value of N by cross-validation..

2.5 Conditional Random Fields

Our algorithm is not necessarily limited to a specific baseline tagger. For simplicity and reliability, we use a simple Conditional Random Field (CRF) tagger, although other sequence labeling models like Semi-Markov CRF Gao et al. (2007) and Latent-variable CRF Sun et al. (2009) may provide better results than a single CRF. Detailed definition of CRF can be found in Lafferty et al. (2001); McCallum (2002); Pinto et al. (2003).

3 Experiment

3.1 Data and metrics

We used the benchmark datasets provided by the second International Chinese Word Segmentation Bakeoff[1] to test our approach. We chose the Peking University (PKU) data in our experiment. Although the benchmark provides another three data sets, two of them are data of traditional Chinese, which is quite different from simplified Chinese. Another is the data from Microsoft Research (MSR). We experimented on this data and got 97.45% in f-score compared to the state-of-art 97.4% reported in Sun et al. (2012). However, this corpus is much larger than the PKU corpus. Using the labeled data alone can get a relatively good tagger and the unlabeled data contributes little to the performance. For simplicity and efficiency, our further experiments are all conducted on the PKU data. Details of the PKU data are listed in table 4.

We also used two un-segmented corpora as unlabeled data. The first one is Chinese Gigaword[2] corpus. It is a comprehensive archive of newswire data. The second one is articles from Baike[3] of baidu.com. It is a Chinese encyclopedia similar to Wikipedia but contains more Chinese items and their descriptions. In the experiment we used about 5 million characters from each corpus for efficiency. Details of unlabeled data can be found in table 5.

In our experiment, we did not use any extra resources such as common surnames, part-of-speech or other dictionaries.

F-score is used as the accuracy measure. We define precision P as the percentage of words in the output that are segmented correctly. We define recall R as the percentage of the words in reference that are correctly segmented. Then F-score is as follows:

$$F = \frac{2 \times P \times R}{P + R}$$

The recall of out-of-vocabulary is also taken into consideration, which measures the ability of the model to correctly segment out of vocabulary words.

3.2 Main Results

Table 6 summarizes the segmentation results on test data with different feature combinations. We performed incremental evaluation. In this table, we first present the results of the tagger only using baseline features. Then we show the results of adding SSF and DSF individually. In the end we compare the results of combining SSF and DSF with baseline features.

Because the baseline features is strong to reach a relative good result, it is not easy to largely enhance the performance. Nevertheless, there are significant increases in f-score and OOV-Recall when adding these features. From table 6 we can see that by adding SSF and DSF individually, the F-score is improved by +1.1%

[1] http://www.sighan.org/bakeoff2005/

[2] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2003T09

[3] http://baike.baidu.com/

Identical words	Total word	Identical Character	Total character
5.5×10^4	1.1×10^6	5×10^3	1.8×10^6

Table 4: Details of the PKU data

Corpus	**Character used**
Gigaword	5000193
Baike	5000147

Table 5: Details of the unlabeled data.

	P	R	F	OOV
Baseline	0.950	0.943	0.946	0.676
+SSF	0.961	0.953	0.957	0.728
+DSF	0.958	0.953	0.955	0.678
+SSF+DSF	0.965	0.958	0.961	0.731

Table 6: Segmentation results on test data with different feature combinations. The symbol "+" means this feature configuration contains features set containing the baseline features and all features after '+' . The size of unlabeled data is fixed as 5 million characters.

and +0.9%. The OOV-Recall is also improved, especially after adding SSFs. When considering SSF and DSF together, the f-score is improved by +1.5% while the OOV-Recall is improved by +5.5%.

To compare the contribution of unlabeled data, we conduct experiments of using different sizes of unlabeled data. Note that the SSFs are still calculated using all the unlabeled data. However, each iteration in the algorithm uses unlabeled data with different sizes.

Table 7 shows the results when changing the size of unlabeled data. We experimented on three different sizes: 0.5 million, 1 million and 5 million characters.

	P	R	F	OOV
DSF(0.5M)	0.962	0.954	0.958	0.727
DSF(1M)	0.963	0.955	0.959	0.728
DSF(5M)	0.965	0.958	0.961	0.731

Table 7: Comparison of results when changing the size of unlabeled data. (0.5 million, 1 million and 5 million characters).

We further experimented on unlabeled corpus with larger size (up to 100 million characters). However the performance did not change significantly. Besides, because the number of features in our method is very large, using too large unlabeled corpus is intractable in real applications due to the limitation of memory.

Our method can keep working well even when the labeled data are insufficient. Table 8 shows the comparison of f-scores when changing the size of labeled data. We compared the results of using all labeled data with 3 different situations: using 1/10, 1/2 and 1/4 of all the labeled data. In fact, the best system on the Second International Chinese Word Segmentation bakeoff reached 0.95 in f-score by using all labeled data. From table 8 we can see that our algorithm only needs 1/4 of all labeled data to achieve the same f-score.

	Baseline	+SSF+DSF	Improve
1/10	0.934	0.943	+0.96%
1/4	0.946	0.951	+0.53%
1/2	0.952	0.956	+0.42%
All	0.957	0.961	+0.42%

Table 8: Comparison of f-scores when changing the size of labeled data. (1/10, 1/4, 1/2 and all labeled data. The size of unlabeled data is fixed as 5 million characters.)

We also explored how the performance changes as iteration increases. Figure 1 shows the change of F-score during the first 10 iterations. From figure 1 we find that f-score has a fast improvement in the first few iterations, and then stables at a fixed point. Besides, as the size of labeled data increases, it converges faster.

Using an in-domain corpus and an out-of-domain corpus is better than use one corpus alone. We compared our approach with the method which uses only one unlabeled corpus. To use only one corpus, we modify our algorithm to extract DSFs from the Chinese Giga word corpus and apply the learned features to itself.

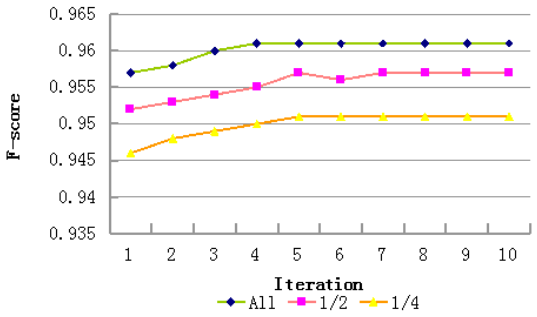

Figure 1: Learning curve of using different size of labeled data

Table 9 shows the result. We can see that our method outperforms by +0.2% in f-score and +0.7% in OOV-Recall.

Finally, we compared our method with the state-of-art systems reported in the previous papers. Table 10 listed the results. Best05 represents the best system reported on the Second International Chinese Word Segmentation Bakeoff. CRF + Rule system represents a combination of CRF model and rule based model presented in Zhang et al. (2006). Other three systems all represent the methods using their corresponding model in the corresponding papers. Note that these state-of-art systems are either using complicated models with semi-Markov relaxations or latent variables, or modifying models to fit special conditions. Our system uses a single CRF model. As we can see in table 10, our method achieved higher F-scores than the previous best systems.

3.3 Results on NER task

Our method is not limited to the CWS problem. It is applicable to all sequence labeling problems. We applied our method on the Chinese NER task. We used the MSR corpus of the sixth SIGHAN Workshop on Chinese Language Processing. It is the only NER corpus using simplified Chinese in that workshop. We compared our method with the pure sequence labeling approach in He and Wang (2008). We reimplemented their method to eliminate the difference of various CRFs implementations. Experiment results are shown in table 11. We can see that our methods works better, especially when handling the out-of-vocabulary named entities;

4 Related work

Recent studies show that character sequence labeling is an effective method of Chinese word segmentation for machine learning (Xue, 2003; Low et al., 2005; Zhao et al., 2006a,b). These supervised methods show good results. Unsupervised word segmentation (Maosong et al., 1998; Peng and Schuurmans, 2001; Feng et al., 2004; Goldwater et al., 2006; Jin and Tanaka-Ishii, 2006) takes advantage of the huge amount of raw text to solve Chinese word segmentation problems. These methods need no annotated corpus, and most of them use statistics to help model the problem. However, they usually are less accurate than supervised ones.

Currently "feature-engineering" methods have been successfully applied into NLP applications. Miller et al. (2004) applied this method to named entity recognition. Koo et al. (2008) applied this method to dependency parsing. Turian et al. (2010) applied this method to both named entity recognition and text chunking. These papers shared the same concept of word clustering. However, we cannot simply equal Chinese character to English word because characters in Chinese carry much less information than words in English and the clustering results is less meaningful.

Features extracted from large unlabeled corpus in previous works mainly focus on statistical information of characters. Feng et al. (2004) used the accessor variety criterion to extract word types. Li and Sun (2009) used punctuation information in Chinese word segmentation by introducing extra labels 'L' and 'R'. Chang and Han (2010), Sun and Xu (2011) used rich statistical information as discrete features in a sequence labeling framework. All these approaches can be viewed as using static statistics features in a supervised approach. Our method is different from theirs. For the static statistics features in our approach, we not only consider richer string pairs with the different lengths, but also consider term frequency when processing

	P	R	F	OOV
Using one corpus	0.963	0.955	0.959	0.724
Our method	0.965	0.958	0.961	0.731

Table 9: Comparison of our approach with using only the Gigaword corpus

Method	P	R	F-score
Best05 (Chen et al. (2005))	0.953	0.946	0.950
CRF + rule-system (Zhang et al. (2006))	0.947	0.955	0.951
Semi-perceptron (Zhang and Clark (2007))	N/A	N/A	0.945
Latent-variable CRF (Sun et al. (2009))	0.956	0.948	0.952
ADF-CRF (Sun et al. (2012))	0.958	0.949	0.954
Our method	0.965	0.958	0.961

Table 10: Comparison of our approach with the state-of-art systems

	P	R	F	OOV
Traditional	0.925	0.872	0.898	0.712
Our method	0.916	0.887	0.902	0.737

Table 11: Comparison of our approach with traditional NER systems

punctuation features.

There are previous works using features extracted from label distribution of unlabeled corpus in NLP tasks. Schapire et al. (2002) use a set of features annotated with majority labels to boost a logistic regression model. We are different from their approach because there is no pseudo-example labeling process in our approach. Qi et al. (2009) investigated on large set of distribution features and used these features in a self-training way. They applied the method on three tasks: named entity recognition, POS tagging and gene name recognition and got relatively good results. Our approach is different from theirs. Although we all consider label distribution, the way we use features are different. Besides, our approach uses two unlabeled corpora which can mutually enhancing to get better result.

5 Conclusion and Perspectives

In this paper, we presented a semi-supervised method for Chinese word segmentation. Two kinds of new features are used for the iterative modeling: static statistical features and dynamic statistical features. The dynamic statistical features use label distribution information for text contexts, and can be adjusted automatically during the co-training process. Experimental results show that the new features can improve the performance on the Chinese word segmentation task. We further conducted experiments to show that the performance is largely improved, especially when the labeled data is insufficient.

The proposed iterative semi-supervised method is not limited to the Chinese word segmentation task. It can be easily extended to any sequence labeling task. For example, it works well on the NER task as well. As our future work, we plan to apply our method to other natural language processing tasks, such as text chunking.

Acknowledgments

This research was partly supported by Major National Social Science Fund of China(No. 12&ZD227),National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101) and National Natural Science Foundation of China (No.91024009). We also thank Xu Sun and Qiuye Zhao for proofreading the paper.

References

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Chang, B. and Han, D. (2010). Enhancing domain portability of chinese segmentation model using chi-square statistics and bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 789–798. Association for Computational Linguistics.

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

Dasgupta, S., Littman, M. L., and McAllester, D. (2002). Pac generalization bounds for co-training. *Advances in neural information processing systems*, 1:375–382.

Feng, H., Chen, K., Deng, X., and Zheng, W. (2004). Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Gao, J., Andrew, G., Johnson, M., and Toutanova, K. (2007). A comparative study of parameter estimation methods for statistical natural language processing. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 824.

Goldwater, S., Griffiths, T., and Johnson, M. (2006). Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 673–680. Association for Computational Linguistics.

Hafer, M. A. and Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information storage and retrieval*, 10(11):371–385.

Harris, Z. S. (1955). From phoneme to morpheme. *Language*, 31(2):190–222.

He, J. and Wang, H. (2008). Chinese named entity recognition and word segmentation based on character. In *Sixth SIGHAN Workshop on Chinese Language Processing*, page 128.

Jin, Z. and Tanaka-Ishii, K. (2006). Unsupervised segmentation of chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 428–435. Association for Computational Linguistics.

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Li, Z. and Sun, M. (2009). Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512.

Low, J., Ng, H., and Guo, W. (2005). A maximum entropy approach to chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 164. Jeju Island, Korea.

Maosong, S., Dayang, S., and Tsou, B. (1998). Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 1265–1271. Association for Computational Linguistics.

McCallum, A. (2002). Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 403–410. Morgan Kaufmann Publishers Inc.

Miller, S., Guinness, J., and Zamanian, A. (2004). Name tagging with word clusters and discriminative training. In *Proceedings of HLT-NAACL*, volume 4.

Peng, F. and Schuurmans, D. (2001). Self-supervised chinese word segmentation. *Ad-*

vances in Intelligent Data Analysis, pages 238–247.

Pinto, D., McCallum, A., Wei, X., and Croft, W. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM.

Qi, Y., Kuksa, P., Collobert, R., Sadamasa, K., Kavukcuoglu, K., and Weston, J. (2009). Semi-supervised sequence labeling with self-learned features. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 428–437. IEEE.

Schapire, R., Rochery, M., Rahim, M., and Gupta, N. (2002). Incorporating prior knowledge into boosting. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 538–545.

Sun, W. and Xu, J. (2011). Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 970–979. Association for Computational Linguistics.

Sun, X., Wang, H., and Li, W. (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. Association for Computational Linguistics.

Sun, X., Zhang, Y., Matsuzaki, T., Tsuruoka, Y., and Tsujii, J. (2009). A discriminative latent variable chinese segmenter with hybrid word/character information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64. Association for Computational Linguistics.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Xue, N. (2003). Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Zhang, R., Kikui, G., and Sumita, E. (2006). Subword-based tagging by conditional random fields for chinese word segmentation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 193–196. Association for Computational Linguistics.

Zhang, Y. and Clark, S. (2007). Chinese segmentation with a word-based perceptron algorithm. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 840.

Zhao, H., Huang, C., and Li, M. (2006a). An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, volume 117. Sydney: July.

Zhao, H., Huang, C., Li, M., and Lu, B. (2006b). Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of PACLIC*, volume 20, pages 87–94.