

Word Segmentation on Micro-blog Texts with External Lexicon and Heterogeneous Data

Qingrong Xia, Zhenghua Li*, Jiayuan Chao, Min Zhang

Soochow University, Suzhou, China
 {kirosummer.nlp, chaojiayuan.china}@gmail.com,
 {zhli13, minzhang}@suda.edu.cn

Abstract. This paper describes our system designed for the NLPCC 2016 shared task on word segmentation on micro-blog texts (i.e., Weibo). We treat word segmentation as a character-wise sequence labeling problem, and explore **two directions** to enhance our CRF-based baseline. First, we employ **a large-scale external lexicon** for constructing extra lexicon features in the model, which is proven to be extremely useful. Second, we exploit **two heterogeneous datasets**, i.e., Penn Chinese Treebank 7 (*CTB7*) and People Daily (*PD*) to help word segmentation on Weibo. We adopt **two mainstream approaches**, i.e., **the guide-feature based approach** and the recently proposed **coupled sequence labeling approach**. We combine the above techniques in different ways and obtain four well-performing models. Finally, we merge the outputs of the four models and obtain the final results via Viterbi-based re-decoding. On the test data of Weibo, our proposed approach outperforms the baseline by $95.63 - 94.24 = 1.39\%$ in terms of F1 score. Our final system rank the first place among five participants in the open track in terms of F1 score, and is also the best among all 28 submissions. All codes, experiment configurations, and the external lexicon are released at <http://hlt.suda.edu.cn/~zhli>.

1 Introduction

Chinese word segmentation (WS) is the most fundamental task in Chinese language processing. In the past decade, supervised approaches have gained extensive progress on canonical texts, especially on texts from domains or genres similar to existing manually labeled data¹. However, the upsurge of web data imposes great challenges on existing techniques. The performance of the state-of-the-art systems degrades dramatically on informal web texts, such as micro-blogs, product comments, and so on. Driven by this challenge, NLPCC 2016 organizes a shared task with an aim of promoting WS on Weibo (WB, Chinese pinyin of micro-blogs) text [8].

* Corresponding Author.

¹ Please refer to <http://zhangkaixu.github.io/bibpage/cws.html> for a long list of related papers.

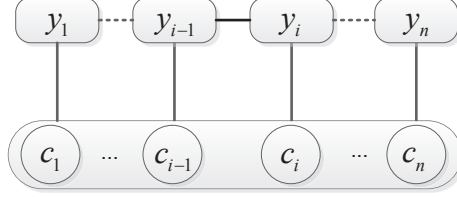


Fig. 1. Graphical structure of the baseline CRF model.

This paper describes our system designed for the shared task in detail. We treat WS as a character-wise sequence labeling problem, and build our model based on the standard conditional random field (CRF) [4] with bigram features. Our major contributions are three-fold. First, we employ a large-scale external lexicon for constructing extra lexicon features in the model, which is proven to be extremely useful.

Second, we exploit two mainstream approaches to exploit heterogeneous data, i.e., the guide-feature based approach and the recently proposed coupled sequence labeling approach. The third-party heterogeneous resources used in the work are **Penn Chinese Treebank 7.0** (CTB7, 50K) and **People’s Daily** (PD, 100K). Since CTB7 and PD have different annotation standards in word segmentation and part-of-speech (POS) tagging, PD has been automatically converted into the style of CTB7.

Third, we propose a **merge-then-re-decode** ensemble approach to combine the outputs of different base models.

On the test data of Weibo, our proposed approach outperforms the baseline by $95.63 - 94.24 = 1.39\%$ in terms of F1 score. Our final system rank the first place among five participants in the open track in terms of F1 score, and is also the best among all 28 submissions.

This paper is organized as follows. Section 2 introduces the baseline CRF-based word segmentation model. Section 3 describes how to employ external lexicon features into baseline CRF model. Section 4 briefly illustrates the guide-feature based approach while Section 5 briefly presents the coupled sequence labeling approach. Section 6 introduces the merge-then-re-decode ensemble approach. Section 7 presents the experimental results. We discuss closely related works in Section 8 and conclude this paper in Section 9.

2 The Baseline CRF-based WSTagger

We treat WS as a sequence labeling problem and employ the standard CRF with bigram features. We adopt the $\{B, I, E, S\}$ tag set, indicating the beginning of a word, the inside of a word, the end of a word and a single-character word [13].

Figure 1 shows the graphical structure of the CRF model. Given an input sentence, which is a sequence of n characters, denoted by $\mathbf{x} = c_1 \dots c_n$, WS aims

Table 1. Feature templates for $\mathbf{f}_{bs}(\mathbf{x}, i, y_{i-1}, y_i)$ used in the baseline CRF model. $T(c_i)$ returns the type of the character c_i (time, number, punctuation, special symbols, else). $I(c_i, c_j)$ judges whether the two characters c_i and c_j are the same.

Unigram: $\mathbf{f}_{bs_uni}(\mathbf{x}, i, y_i)$	Bigram: $\mathbf{f}_{bs_bi}(\mathbf{x}, i, y_{i-1}, y_i)$
01: $y_i \circ c_k$ $i - 2 \leq k \leq i + 2$	09: $y_{i-1} \circ y_i$
02: $y_i \circ c_{k-1} \circ c_k$ $i - 1 \leq k \leq i + 2$	10: $y_{i-1} \circ y_i \circ c_i$
03: $y_i \circ c_{k-1} \circ c_k \circ c_{k+1}$ $i - 1 \leq k \leq i + 1$	11: $y_{i-1} \circ y_i \circ c_{i-1} \circ c_i$
04: $y_i \circ T(c_k)$ $i - 1 \leq k \leq i + 1$	
05: $y_i \circ T(c_{k-1}) \circ T(c_k)$ $i \leq k \leq i + 1$	
06: $y_i \circ T(c_{i-1}) \circ T(c_i) \circ T(c_{i+1})$	
07: $y_i \circ I(c_i, c_k)$ $i - 2 \leq k \leq i + 2, k \neq i$	
08: $y_i \circ I(c_{i-1}, c_{i+1})$	

to determine the best tag sequence $\mathbf{y} = y_1 \dots y_n$, where $y_i \in \{B, I, E, S\}$. As a log-linear model, CRF defines the probability of a tag sequence as:

$$P(\mathbf{y}|\mathbf{x}; \theta) = \frac{e^{Score(\mathbf{x}, \mathbf{y}; \theta)}}{\sum_{\mathbf{y}'} e^{Score(\mathbf{x}, \mathbf{y}'; \theta)}} \quad (1)$$

$$Score(\mathbf{x}, \mathbf{y}; \theta) = \sum_{1 \leq i \leq n+1} \theta \cdot \mathbf{f}_{bs}(\mathbf{x}, i, y_{i-1}, y_i)$$

where $Score(\mathbf{x}, \mathbf{y}; \theta)$ is a scoring function; $\mathbf{f}_{bs}(\mathbf{x}, i, y_{i-1}, y_i)$ is the feature vector at the i^{th} character and θ is the feature weight vector. Please note that c_0 and c_{n+1} are two pseudo characters marking the beginning and end of the sentence. We use the features described in zhang et al. (2014) [16], as shown in Table 1.

3 Exploring External Lexicon Features

Inspired by the work of Yu et al., (2015) [14] who have participated last year’s shared task, we try to enhance the baseline CRF by using a large-scale word dictionary [15]. The dictionary we use is composed of two parts. The first part contains about 210K words, and is directly borrowed from Yu et al., (2015) [14].² The second part contains 217K words, and is collected by ourselves from the lexicon sharing website of Sogou (<http://pinyin.sogou.com/dict/>). In total, the external lexicon consists of 428,101 words, and is denoted as \mathcal{D} in this work.

² We are very grateful for their kind sharing. Their dictionary is composed of several word lists, the SogouW word dictionary (<http://www.sogou.com/labs/resource/w.php>), and a few lists on different domains (finance, sports, and entertainment) from the lexicon sharing website of Sogou (<http://pinyin.sogou.com/dict/>).

Table 2. Lexicon Feature templates $\mathbf{f}_{lex}(\mathbf{x}, i, y_i, \mathcal{D})$.

01: $F_B(\mathbf{x}, i-1, y_i, \mathcal{D})$	04: $F_B(\mathbf{x}, i, y_i, \mathcal{D})$	07: $F_B(\mathbf{x}, i+1, y_i, \mathcal{D})$
02: $F_I(\mathbf{x}, i-1, y_i, \mathcal{D})$	05: $F_I(\mathbf{x}, i, y_i, \mathcal{D})$	08: $F_I(\mathbf{x}, i+1, y_i, \mathcal{D})$
03: $F_E(\mathbf{x}, i-1, y_i, \mathcal{D})$	06: $F_E(\mathbf{x}, i, y_i, \mathcal{D})$	09: $F_E(\mathbf{x}, i+1, y_i, \mathcal{D})$

Apart from the features used in Table 1, denoted as $\mathbf{f}_{bs}(\mathbf{x}, i, y_{i-1}, y_i)$, the enhanced model adds extra lexicon features to the feature vector, denoted as $\mathbf{f}_{lex}(\mathbf{x}, i, y_i, \mathcal{D})$. Thus, the scoring function becomes:

$$Score(\mathbf{x}, \mathbf{y}; \theta) = \sum_{1 \leq i \leq n+1} \theta \cdot \begin{bmatrix} \mathbf{f}_{bs}(\mathbf{x}, i, y_{i-1}, y_i) \\ \mathbf{f}_{lex}(\mathbf{x}, i, y_i, \mathcal{D}) \end{bmatrix} \quad (2)$$

where the first term of the extended feature vector is the same as the baseline feature vector and the second term is the lexicon feature vector.

Table 2 lists the lexicon feature templates, which are mostly borrowed from Zhang et al. (2012) [15]. $F_B(\mathbf{x}, i, y_i, \mathcal{D})$ considers words beginning with c_i , and returns the maximum length m , so that the span $c_i c_{i+1} \dots c_{i+m-1}$ in \mathbf{x} is a word in \mathcal{D} . “Maximum” means that there is no $r > m$ so that $c_i c_{i+1} \dots c_{i+r-1}$ in \mathbf{x} is a word in \mathcal{D} . In contrast, $F_E(\mathbf{x}, i, y_i, \mathcal{D})$ considers words ending with c_i , and returns the maximum length m , so that the span $c_{i-m+1} \dots c_{i-1} c_i$ in \mathbf{x} is a word in \mathcal{D} . Analogously, $F_I(\mathbf{x}, i, y_i, \mathcal{D})$ considers words containing c_i (absolutely inside), and returns the maximum length m , so that the span $c_{i-(m-j-1)} \dots c_i \dots c_{i+j}$ (where $m > 2$ and $0 < j < m-1$) in \mathbf{x} is a word in \mathcal{D} .

4 The Guide-feature Based Approach for Exploiting CTB7 and PD

To use the heterogeneous data, we re-implement the guide feature baseline method [3]. The basic idea is to use one resource to generate extra guide features on another resource, as illustrated in Fig. 2. *PD* is converted into the style of *CTB*, as discussed in Section 7.1. First, we use *CTB7* and *PD* as the source data to train a source model $Tagger_{CTB7+PD}$. Then, $Tagger_{CTB7+PD}$ generates automatic tags on the target data *WB*, called *source annotations*. Finally, a target model $Tagger_{WB \leftarrow (CTB7+PD)}$ is trained on *WB*, using source annotations as extra guide features.

Table 3 lists the guide feature templates used in this work. Adding the guide features into the model feature vector, the scoring function becomes:

$$Score(\mathbf{x}, \mathbf{y}; \theta) = \sum_{1 \leq i \leq n+1} \theta \cdot \begin{bmatrix} \mathbf{f}_{bs}(\mathbf{x}, i, y_{i-1}, y_i) \\ \mathbf{f}_{guide}(\mathbf{x}, \mathbf{y}^S, i, y_i) \end{bmatrix} \quad (3)$$

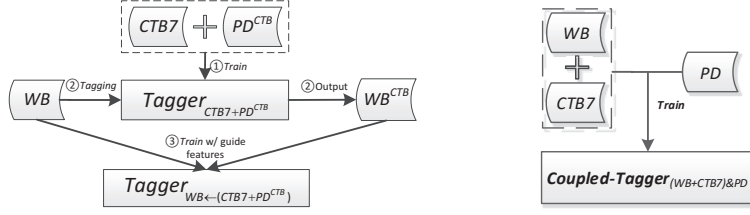


Fig. 2. Our model using guide feature **Fig. 3.** Graphical structure of the coupled CRF

Table 3. Guide feature templates for $\mathbf{f}_{guide}(\mathbf{x}, \mathbf{y}^S, i, y_i)$. where $\mathbf{f}_{bs_uni}(\mathbf{x}, i, y_i) \circ y_i^S$ means that each feature template in $\mathbf{f}_{bs_uni}(\mathbf{x}, i, y_i)$ is concatenated with y_i^S to produce a new feature template.

Guide Features: $\mathbf{f}_{guide}(\mathbf{x}, \mathbf{y}^S, i, y_i)$	
01: $\mathbf{f}_{bs_uni}(\mathbf{x}, i, y_i) \circ y_i^S$	05: $y_i \circ y_{i-1}^S \circ y_i^S$
02: $y_i \circ y_i^S$	06: $y_i \circ y_i^S \circ y_{i+1}^S$
03: $y_i \circ y_{i+1}^S$	07: $y_i \circ y_{i-1}^S \circ y_{i+1}^S$
04: $y_i \circ y_{i-1}^S$	08: $y_i \circ y_{i-1}^S \circ y_i^S \circ y_{i+1}^S$

5 The Coupled Approach for Exploring *CTB7* and *PD*

The coupled sequence labeling approach is proposed in our earlier work Li et al. (2015) [6], and aims to learn and predict two heterogeneous annotations simultaneously. The key idea is to bundle two sets of tags together, and build a conditional random field (CRF) based tagging model in the enlarged space of bundled tags with the help of *ambiguous labeling*. To train our model on two non-overlapping datasets that each has only one-side tags, we transform a one-side tag into a set of bundled tags by concatenating the tag with every possible tag at the missing side according to a predefined context-free tag-to-tag mapping function, thus producing ambiguous labeling as weak supervision. The bundled tag space contains $4 \times 4 = 16$ tags in our task of WS. Please refer to Chao et al. (2015) [2] for the detailed description of the coupled WS tagging model.

6 The Merge-then-re-decode Ensemble Approach

In this section, we propose a merge-then-re-decode ensemble approach to combine the outputs of different base models, which is inspired by the work of Sagae and Lavie (2006) [10]. First, given a sentence $\mathbf{x} = c_1 \dots c_n$, the outputs of several base models are treated as votes of character-wise tags with equal weights. For example, if three models assign *B* to the character c_i , and only one model assigns *S* to it, then the scores of tagging c_i as $\{B, I, E, S\}$ are $\{3, 0, 0, 1\}$ respectively. In such way, we can get all scores for all characters in \mathbf{x} . Then, we find the highest-scoring tag sequence using the Viterbi algorithm.

Table 4. Data statistics

Dataset	Partition	Sentences	Words	Characters
WB	train	20,135	421,166	688,734
	dev	2,052	43,697	73,244
	test	8,592	—	315,857
CTB7	train	46,572	1,039,774	1,682,485
	dev	2,079	59,955	100,316
	test	2,796	81,578	134,149
PD	train	106,157	1,752,502	2,911,489

To avoid that the re-decode procedure outputs a tag sequence containing illegal transitions ($B \rightarrow S$, $B \rightarrow B$, $I \rightarrow B$, $I \rightarrow S$, $E \rightarrow I$, $E \rightarrow E$, $S \rightarrow I$, $S \rightarrow E$), we make a slight modification to the standard Viterbi algorithm. The basic idea is to throw away illegal transitions from c_{i-1} to c_i when searching the best partial tag sequences for $c_1 \dots c_i$. Concretely, if we are searching the best tag sequences for $c_1 \dots c_i$ with c_i tagged as B , we only considers the results that tag c_{i-1} as E or S (but neither B nor I).

7 Experiments

7.1 Datasets

Table 4 shows the datasets used in this work. “WB”, short for Weibo, refers to the labeled data provided by the NLPCC 2016 shared task organizer. Actually, the organizer also provides a large set of unlabeled *WB* text, which is not considered in this work.

We adopt *CTB7* as a third-party resource and follow the suggestion in the data description guideline for data split.

We also use *PD* as another labeled resource. Since *PD* and *CTB7* have different word segmentation and POS tagging standards, we used a converted version of *PD* following the style of *CTB* for the sake of simplicity in this work.

Annotation Conversion: PD^{CTB} . We directly use the coupled WS&POS tagging model trained on *CTB5* and *PD* in Li et al. (2016) [5] for data conversion. As pointed in Li et al (2015) [6], the coupled model can be naturally used for annotation conversion via constrained decoding with the *PD*-side tags being fixed. After conversion, if a sentence in *PD* contains a character with a very low marginal probability (< 0.8), we throw away the sentence to guarantee the data quality. Finally, we get the *100KPD* dataset in the same style of *CTB7*, denoted as PD^{CTB} .

For **evaluation metrics**, we adopt character-level accuracy, and the standard Precision (P), Recall (R), and F1 score.

Training with multiple training datasets: For some models (such as $WSTagger_{CTB7+PD}$ and $CoupledWSTagger_{WB\&CTB7+PD}$), we use two or three training datasets simultaneously. To balance the contribution of different datasets, we adopt the simple corpus-weighting strategy proposed in Li et al. (2015) [6].

Table 5. WS accuracy: an investigation of the heterogeneity of *WB* and *CTB7*.

	on <i>CTB7</i>		on <i>WB</i>
	dev	test	dev
$WSTagger_{CTB7}$	96.37	95.81	91.77
$WSTagger_{WB}$	90.86	90.82	94.66

Table 6. Results on *CTB7*-dev/test.

	on Dev				on Test			
	Acc	P	R	F	Acc	P	R	F
$WSTagger_{CTB7}$	96.37	95.84	95.37	95.60	95.81	95.40	94.58	94.98
$WSTagger_{CTB7+PD}$	96.82	96.29	96.14	96.21	96.37	95.94	95.44	95.69
$WS\&POSTagger_{CTB7}$	96.70	96.21	95.78	96.00	96.25	95.92	95.13	95.52
$WS\&POSTagger_{CTB7+PD}$	97.04	96.62	96.34	96.48	96.61	96.30	95.66	95.98
$CoupledWSTagger_{WB\&CTB7}$	96.54	96.03	95.55	95.79	96.02	95.59	94.86	95.22
$CoupledWSTagger_{WB\&CTB7+PD}$	96.96	96.43	96.21	96.32	96.45	95.96	95.48	95.72
$CoupledWSTagger_{WB\&CTB7}$ w/ lexicon	96.82	96.29	95.96	96.12	96.42	95.95	95.39	95.67
$CoupledWSTagger_{WB\&CTB7+PD}$ w/ lexicon	97.25	96.79	96.51	96.65	96.83	96.45	95.88	96.16

Before each iteration, we randomly select 5000 sentences from each training datasets. Then, we merge and shuffle the selected sentences, and use them for one-iteration training.

7.2 Heterogeneity of *WB* and *CTB7*

To investigate the heterogeneity of *WB* and *CTB7*, we use the baseline model trained on *WB*-train, denoted as $WSTagger_{WB}$, to process *CTB7*-dev/test, and also use the baseline model trained on *CTB7*-train, denoted as $WSTagger_{CTB7}$, to process *WB*-dev. Table 5 shows the results. It is obvious that *CTB7* and *WB* differs a lot in the definition of word boundaries. In contrast, in the shared task of NLPCC 2015, we find that *CTB7* and the provided *WB* are very similar in the word boundary standard [2].

Based on this observation, we employ the guide-feature based approach and the coupled approach to exploit *CTB7*, instead of directly adding *CTB7* as extra training data.

7.3 Results on *CTB7*-dev/test

To investigate the performance on canonical texts of the models trained on *CTB7* (and *PD*), we evaluate the models on *CTB7*-dev/test. Table 6 shows the results on the task of WS. We can get several reasonable yet interesting findings. First, comparing the results in all four major rows, we can see that using *PD* as extra labeled data consistently improves the F1 score by about 0.5%. Second, comparing the results in the first two major rows, it is clear that jointly modeling WS&POS outperforms the pure WS tagging model by about 0.3 – 0.5%. Third,

Table 7. Performance of joint WS&POS tagging on *CTB7*-dev/test.

	on Dev			on Test		
	P	R	F	P	R	F
<i>WS&POSTagger_{CTB7}</i>	91.28	90.86	91.04	90.91	90.16	90.54
<i>WS&POSTagger_{CTB7+PD}</i>	92.19	91.92	92.06	91.80	91.19	91.49

comparing the results in the bottom two major rows, we can see that lexicon features are useful and improves F1 score by about 0.5%. Fourth, comparing the results in the first and third major rows, we can see that using WB as extra labeled data leads with the coupled approach to slight improvement in F1 score (0.03 – 0.24%).

Table 7 shows the results on the joint task of WS&POS. We can see that using *PD* as extra labeled data dramatically improves the word-wise F1 score by about 1%.

7.4 Results on WB-dev

In this part, we conduct extensive experiments to investigate the effectiveness of different methods for WS on *WB*-dev. Table 8 shows the results. From the results, we can obtain the following findings.

First, **lexicon features** are very useful. Comparing the first two major rows, we can see that using lexicon features leads to a large improvement of $94.88 - 93.65 = 1.23\%$ on F1 score over the baseline model. Comparing the third and fourth major rows, lexicon features boost F1 score by $95.15 - 94.16 = 0.99\%$ over the models with guide features. Comparing the fifth and sixth major rows, lexicon features boost F1 score by $95.30 - 94.64 = 0.66\%$ over the coupled models.

Second, the **coupled approach** is much more effective than the guide-feature based approach in exploiting multiple heterogeneous data. Comparing the third and fifth major rows, the coupled approach outperforms the guide-feature based approach by $94.64 - 94.16 = 0.48\%$ on F1 score. Comparing the fourth and sixth major rows, with the lexicon features, the coupled approach achieves higher F1 score by $95.30 - 95.15 = 0.15\%$ over its counterpart.

Third, looking into the third major row, we also get a few interesting findings: 1) using a joint **WS&POS tagger** to produce guide tags is better than using a WS tagger, indicating that jointly modeling WS&POS leads to better guide information, which is consistent with the results in Table 6; 2) **PD** is helpful by producing better guide tags, leading to higher F1 score on *WB*-dev by about 0.2%; 3) using both **WS&POS tags** for guide achieves nearly the same performance as using only WS tags.

Finally, the proposed merge-then-re-decode ensemble approach improves F1 score by $95.47 - 95.30 = 0.17\%$ over the best single model. However, we find that the performance drops when we use all model during ensemble, which may be caused by the very bad performance of some models.

Table 8. Results on *WB-dev*

Approaches		Acc	P	R	F
Baseline	1. <i>WSTagger</i> _{WB}	94.66	93.30	93.99	93.65
w/ lexicon features	2. <i>WSTagger</i> _{WB}	95.74	94.45	95.31	94.88
w/ guide features	3. WS-tag from <i>WSTagger</i> _{CTB7}	94.52	93.21	93.93	93.58
	4. WS-tag from <i>WSTagger</i> _{CTB7+PD}	94.80	93.41	94.40	93.90
	5. WS-tag from <i>WS&POSTagger</i> _{CTB7}	94.86	93.64	94.27	93.95
	6. WS-tag from <i>WS&POSTagger</i> _{CTB7+PD}	95.05	93.76	94.57	94.16
	7. WS&POS-tag from <i>WS&POSTagger</i> _{CTB7}	94.88	94.33	93.64	93.98
	8. WS&POS-tag from <i>WS&POSTagger</i> _{CTB7+PD}	95.03	93.83	94.50	94.16
w/ lexicon & guide	9. WS&POS-tag from <i>WS&POSTagger</i> _{CTB7+PD}	95.97	94.77	95.53	95.15
Coupled	10. <i>CoupledWSTagger</i> _{WB&CTB7}	95.38	94.12	94.91	94.51
	11. <i>CoupledWSTagger</i> _{WB&CTB7+PD}	95.50	94.25	95.03	94.64
Coupled w/ lexicon	12. <i>CoupledWSTagger</i> _{WB&CTB7}	96.01	94.74	95.61	95.17
	13. <i>CoupledWSTagger</i> _{WB&CTB7+PD} (submitted)	95.98	94.78	95.56	95.17
	14. <i>CoupledWSTagger</i> _{WB&CTB7+PD}	96.11	94.80	95.82	95.30
Merge-then-re-decode	On four models (2,9,12,13) (submitted)	96.14	95.03	95.72	95.37
	On four models (2,9,12,14)	96.22	95.10	95.84	95.47
	On all models (w/o 13)	95.88	94.76	95.48	95.12

Table 9. Results on *WB-test*

	P	R	F
Baseline	93.53	94.14	93.83
Merge-then-re-decode (2,9,12,13)	95.05 (+ 1.52)	95.70 (+ 1.56)	95.37 (+ 1.54)

7.5 Reported results on WB-test

Since we do not have the gold-standard labels for the test data, Table 9 shows the results provided by the shared task organizers. Our effort leads to an improvement on WS F1 score by $95.37 - 93.83 = 1.54\%$. And our results on test data rank the first place among five participants, and is also the best among all 28 submissions.

8 Related Work

Using external lexicon is first described in Pi-Chuan Chang et al. (2008) [1]. Zhang et al. (2012) [15] find the lexicon features are also very helpful for domain adaptation of WS models,

Jiang et al. (2009) [3] first propose the simple yet effective guild-feature based method, which is further extended in [11,7,12].

Qiu et al. (2013) [9] propose a model that performs heterogeneous Chinese word segmentation and POS tagging and produces two sets of results following *CTB* and *PD* styles respectively. Their model is based on **linear perceptron**, and uses approximate inference.

Li et al. (2015) [6] first propose the coupled sequence labeling approach. Chao et al., (2015) [2] make extensive use of the coupled approach in participating the

NLPCC 2015 shared task of WS&POS for Weibo texts. Li et al., (2016) [5] further improves the coupled approach in terms of efficiency via context-aware pruning, and first apply the coupled approach to the joint WS&POS task. In this work, we directly use the coupled model built in Li et al. for converting the WS&POS annotations in *PD* into the style of *CTB*.

9 Conclusion

We have participated in the NLPCC 2016 shared task on Chinese WS for Weibo Text. Our main focus is to make full use of an external lexicon and two heterogeneous labeled data (i.e., *CTB7* and *PD*). Moreover, we apply an merge-then-decode ensemble approach to combine the outputs of different base models. Extensive experiments are conducted in this work to fully investigate the effectiveness of methods in study. Particularly, this work leads to several interesting findings. First, lexicon features are very useful in improving performance on both canonical texts and WB texts. Second, the coupled approach is consistently more effective than the guide-feature based approach in exploiting multiple heterogeneous data. Third, using the same training data, a joint WS&POS model produces better WS results than a pure WS model, indicating that the POS tags are helpful for determining word boundaries. Our submitted results rank the first place among five participants in the open track in terms of F1 score, and is also the best among all 28 submissions.

For future work, we plan to work on word segmentation with **different granularity levels**. During this work, we carefully compared the outputs of different base models, and found that in many error cases, the results of the statistical models are actually correct from the human point view. Many results are considered as wrong answers simply because they are of different word granularity from the gold-standard references. Therefore, we are very interested in build statistical models that can output WS results with different granularities. And perhaps, we have to first construct some WS data with multiple-granularity annotations.

Acknowledgments

The authors would like to thank the anonymous reviewers for the helpful comments. This work was supported by National Natural Science Foundation of China (Grant No. 61502325, 61432013) and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 15KJB520031).

References

1. Chang, P.C., Galley, M., Manning., C.D.: Optimizing chinese word segmentation for machine translation performance. In: ACL 2008 Third Workshop on Statistical Machine Translation (2008)

2. Chao, J., Li, Z., Chen, W., Zhang, M.: Exploiting heterogeneous annotations for weibo word segmentation and pos tagging. In: Proceedings of the 4th CCF Conference on Natural Language Processing & Chinese Computing. pp. 495–506 (2015)
3. Jiang, W., Huang, L., Liu, Q.: Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging – a case study. In: Proceedings of ACL. pp. 522–530 (2009)
4. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
5. Li, Z., Chao, J., Zhang, M.: Fast coupled sequence labeling on heterogeneous annotations via context-aware pruning. In: Proceedings of EMNLP (2016)
6. Li, Z., Chao, J., Zhang, M., Chen, W.: Coupled sequence labeling on heterogeneous annotations: POS tagging as a case study. In: Proceedings of ACL (2015)
7. Li, Z., Che, W., Liu, T.: Exploiting multiple treebanks for parsing with quasisynchronous grammar. In: ACL. pp. 675–684 (2012)
8. Qiu, X., Qian, P., Shi, Z.: Overview of the NLPCC-ICCPOL 2016 shared task: Chinese word segmentation for micro-blog texts. In: Proceedings of The Fifth Conference on Natural Language Processing and Chinese Computing & The Twenty Fourth International Conference on Computer Processing of Oriental Languages (2016)
9. Qiu, X., Zhao, J., Huang, X.: Joint Chinese word segmentation and POS tagging on heterogeneous annotated corpora with multiple task learning. In: Proceedings of EMNLP. pp. 658–668 (2013)
10. Sagae, K., Lavie, A.: Parser combination by reparsing. In: Proceedings of NAACL. pp. 129–132 (2006)
11. Sun, W.: A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In: Proceedings of ACL. pp. 1385–1394 (2011)
12. Sun, W., Wan, X.: Reducing approximation and estimation errors for Chinese lexical processing with heterogeneous annotations. In: Proceedings of ACL. pp. 232–241 (2012)
13. Xue, N., et al.: Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing* 8(1), 29–48 (2003)
14. Yu, Z., Dai, X., Huang, S., Chen, J.: Word segmentation of micro blogs with bagging. pp. 573–580 (2015)
15. Zhang, M., Deng, Z., Che, W., Liu, T.: Combining statistical model and dictionary for domain adaption of chinese word segmentation. In: *Journal of Chinese Information Processing*. pp. 8–12 (2012)
16. Zhang, M., Zhang, Y., Che, W., Liu, T.: Character-level Chinese dependency parsing. In: Proceedings of ACL. pp. 1326–1336 (2014)