An Empirical Study of Automatic Chinese Word Segmentation for Spoken Language Understanding and Named Entity Recognition

Wencan Luo[*]
University of Pittsburgh
Pittsburgh, PA 15260
wencan@cs.pitt.edu

Fan Yang
Nuance Communications, Inc.
Seattle, WA 98104
fan.yang@nuance.com

Abstract

Word segmentation is usually recognized as the first step for many Chinese natural language processing tasks, yet its impact on these subsequent tasks is relatively under-studied. For example, how to solve the *mismatch* problem when applying an existing word segmenter to new data? Does a better word segmenter yield a better subsequent NLP task performance? In this work, we conduct an initial attempt to answer these questions on two related subsequent tasks: semantic slot filling in spoken language understanding and named entity recognition. We propose three techniques to solve the mismatch problem: using word segmentation outputs as additional features, adaptation with partial-learning and taking advantage of n-best word segmentation list. Experimental results demonstrate the effectiveness of these techniques for both tasks and we achieve an error reduction of about 11% for spoken language understanding and 24% for named entity recognition over the baseline systems.

1 Introduction

Unlike English text in which sentences are sequences of words separated by white spaces, in Chinese text (as are some other languages including Arabic, Japanese, etc.), sentences are represented as strings of characters without similar natural delimiters. Therefore, it is generally claimed that the first step in a Chinese language processing task is to identify the sequence of words in a sentence and mark boundaries in appropriate places, which is refereed to as the task of Chinese Word Segmentation (CWS).

(1) Input: 能穿多少穿多少
CWS 1: 能|穿|**多少**|穿|多少[1]
(Put on as **much** clothes as possible.)
CWS 2: 能|穿|多|**少**|穿|多|少
(Put on as **little** clothes as possible.)

Word segmentations in Chinese text do reduce ambiguities. In the example (1), the same span of text (the input) can convey entirely opposite meanings (the English sentences in parentheses) depending on how word boundaries (CWS 1 and CWS 2) are labeled. Therefore, it is generally believed that more accurate word segmentations should benefit more the subsequent Chinese language processing tasks, such as part-of-speech tagging, named entity recognition, etc. There has been quite a number of research in the field of CWS to improve segmentation accuracy, yet its impact on the subsequent processing is relatively under-studied. Chang et al. (2008) explore how word segmentation improves machine translation; and Ni and Leung (2014) explore how word segmentation impacts automatic speech recognition yet do not have conclusive findings. In this research, we aim to better understand how CWS benefits the subsequent NLP tasks, using semantic slot filling in spoken language understanding (SLU) and named entity recognition (NER) as two case studies.

In particular, we investigate the impact of Chinese word segmentation in three different situations.

[*] Work done at Nuance during an internship.

[1] We use '|' to indicate a word boundary. Example is borrowed and revised from (Chen et al., 2015).

First, assuming domain data (the data for a particular subsequent task, e.g. SLU or NER) having no word boundary annotation (§4), we can apply word segmenters trained with publicly-available data to the domain data to get the word boundary. However, existing word segmenters may have a domain mismatch problem due to the fact that they may have different genre from the subsequent task and are usually segmented with different standards (Huang and Zhao, 2007). Therefore, we propose three techniques to solve this problem. Note, these techniques can be used together.

1) We use word segmentation outputs as additional features in subsequent tasks (§3.2), which is more robust against error propagation than using segmented word units.

2) We adapt existing word segmenters with partially-labeled data derived from the subsequent task training data (§3.3), further improving the end-to-end performance.

3) We take advantage of the n-best list of word segmentation outputs (§3.4), making the subsequent task less sensitive to word segmentation errors.

Second, assuming domain training data (e.g., NER) is already segmented with word boundary (§5), we are able to train a domain word segmenter with the data itself and apply it to the testing data. This allows us to see the differences between a word segmenter trained with in-domain data and one trained with publicly-available data.

Last, assuming both domain training and testing data have word boundary information (§5), it allows to explore the upper bound performance of the subsequent task with a perfect word segmenter.

Experimental results show that the proposed techniques do improve the end-to-end performance and we achieve an error rate reduction of 11% for SLU and 24% for NER over their corresponding baseline systems. In addition, we found that even a word segmenter that is only moderately reliable is still able to improve the end-to-end performance, and a word segmenter trained with in-domain data is not necessarily better compared to a word segmenter trained with out-domain data in terms of the end-to-end performance.

2 Related Work

Word segmentation has received steady attention over the past two decades. People have shown that models trained with limited text can have a reasonable accuracy (Li and Sun, 2009; Zhang et al., 2013a; Li et al., 2013; Cheng et al., 2015). However, the fact is that none of existing algorithms is robust enough to reliably segment unfamiliar types of texts without fine-tuning (Huang et al., 2007). Several approaches have proposed to eliminate this issue, for example the use of unlabeled data (Sun and Xu, 2011; Wang et al., 2011; Zhang et al., 2013b) and partially-labeled data (Yang and Vozila, 2014; Takahasi and Mori, 2015). In our work, we encounter the same issue when applying word segmentation to the subsequent tasks and thus we propose three approaches to solve this problem.

Word segmentation has been applied in several subsequent tasks, e.g. NER (Zhai et al., 2004), information retrieval (Peng et al., 2002), automatic speech recognition (Ni and Leung, 2014), machine translation (Xu et al., 2008; Chang et al., 2008; Zhang et al., 2008; Zeng et al., 2014), etc. In general, there are two types of approaches to utilize word segmentation in subsequent tasks: *pipeline* and *joint-learning*. The pipeline approach creates word segmentation first and then feeds the segmented words into subsequent task(s). It is straightforward, but suffers from error propagation since an incorrect word segmentation would cause an error in the subsequent task. The joint-learning approach trains a model to learn both word segmentation and the subsequent task(s) at the same time. A number of subsequent tasks have been unified into joint models, including disambiguation (Wang et al., 2012), part-of-speech tagging (Jiang et al., 2008a; Jiang et al., 2008b; Zhang and Clark, 2010; Sun, 2011), NER (Gao et al., 2005; Xu et al., 2014; Peng and Dredze, 2015), and parsing (Hatori et al., 2012; Qian and Liu, 2012). However, the joint-learning process generally assumes the availability of manual word segmentations for the training data, which limits the use of this approach. Thus in this work, we focus on the pipeline approach, but instead of feeding the segmented words, we use word segmentation results as additional features in the subsequent tasks, which is more robust against error propagation.

3 Applying CWS to Subsequent Tasks

In this section, we describe how to integrate word segmentation information when domain data having no word boundary information, using SLU and NER as two case studies.

We first introduce the baseline system, and then describe the techniques that we propose to solve the domain mismatch problem when applying automatic CWS to the subsequent NLP tasks.

3.1 Baseline system

Both of the SLU and NER can be formulated as sequence labeling tasks, and can be solved using machine learning techniques such as Conditional Random Field (CRF), Recurrent Neural Network, or their combinations (Wan et al., 2011; Mesnil et al., 2015; Rondeau and Su, 2015). We adopt the tool wapiti (Lavergne et al., 2010), which is an implementation of CRF. In the baseline system, each Chinese character is treated as a labeling unit. Here is an example of our training sentences for SLU:'三|division 元|division 里|division 莫|street 干|street 山|street 路|street 周|locref 围|locref 的|unk 餐|query 厅|query' (Find the restaurants near Sanyuanli Mogan Mountain road). The input features for training the baseline CRF model are character ngrams in the K-window and label bigrams. For computational efficiency, we use trigram within 5-character window. Given the current character c_0, we extract the following character ngram features: c_{-2}, c_{-1}, c_0, c_1, c_2, $c_{-2}c_{-1}$, $c_{-1}c_0$, c_0c_1, c_1c_2, $c_{-2}c_{-1}c_0$, $c_{-1}c_0c_1$, and $c_0c_1c_2$.

3.2 Using CWS as features

When word segmentation information is not available within the domain data, we can use publicly-available corpora such as the Chinese Tree Bank (Levy and Manning, 2003), to train an automatic word segmenter.

A dominant approach for supervised CWS is to formulate it as a character sequence labeling problem, and label each character with its location in a word (Xue, 2003). A popular labeling scheme is 'BIES': 'B' for the beginning character of a word, 'I' for the internal characters, 'E' for the ending character, and 'S' for single-character word. Following (Yang and Vozila, 2014), we train our automatic word segmenter with CRF using the input features of character unigrams and bigrams, consecutive character equivalence, separated character equivalence, punctuation, character sequence pattern, anchor of word unigram and bigram. This word segmenter achieves state-of-the-art or comparable performance.

A straightforward way to integrate word segmentation is the traditional pipeline approach. It uses word segmentation first and feeds the segmented words to subsequent task(s), named as **Word Unit**. However, this method suffers from the error propagation problem since an incorrect word segmentation would cause an error in the subsequent task. Therefore, we proposed to use word segmentation outputs as additional features (**As Features**) in the subsequent tasks, as introduced below. We hypothesize the **As Features** is less sensitive to word segmentation errors since the CRF model can still rely on the character features when a word segmentation is not perfect.

Word Unit We can use segmented words instead of characters as labeling units for the CRF learning. During training we can run *forced-decoding* (Lavergne et al., 2010) on word segmentation so that word boundaries are consistent with semantic slot or named entity boundaries. During testing we simply apply the word segmenter to the sentences.

As Features We can still keep using characters as the labeling units, but add the word segmentation information as additional features. Given the current character c_0 and word segmentation output represented as 'BIES' tag t_0, we extract the character ngram features together with the following word segmentation tag ngram features: t_{-2}, t_{-1}, t_0, t_1, t_2, $t_{-2}t_{-1}$, $t_{-1}t_0$, t_0t_1, t_1t_2, $t_{-2}t_{-1}t_0$, $t_{-1}t_0t_1$, and $t_0t_1t_2$. The tag ngram features provide word segmentation information indirectly. For example, t_0t_1='BE' indicates c_0 initiates a two-character word, while $t_0t_1t_2$='BII' means that c_0 is probably a beginning of a long word.

3.3 Adaptation with Partial-learning

The publicly-available corpora for word segmentation, however, may create a domain-mismatch prob-

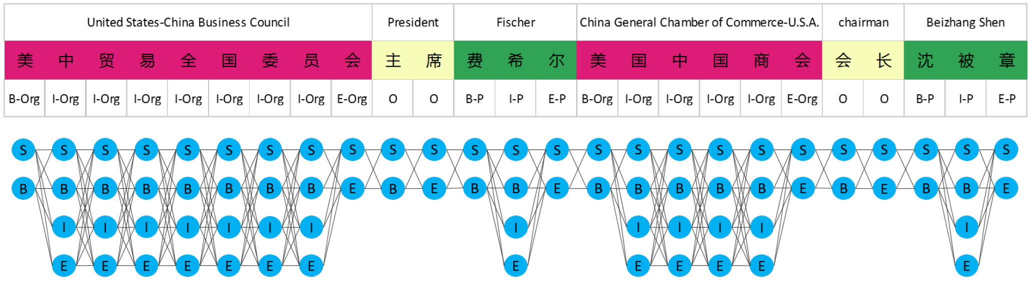
United States-China Business Council
President
Fischer
China General Chamber of Commerce-U.S.A.
chairman
Beizhang Shen
美 中 贸 易 全 国 委 员 会 主 席 费 希 尔 美 国 中 国 商 会 会 长 沈 被 章
B-Org I-Org I-Org I-Org I-Org I-Org I-Org I-Org E-Org O O B-P I-P E-P B-Org I-Org I-Org I-Org I-Org E-Org O O B-P I-P E-P
S
B
I
E

Figure 1: Partially-labeled word segmentations derived from named entity labels. The first character in a name ('美' in the organization name '美中贸易全国委员会') can only be labeled as 'S' or 'B', while the last one ('会') can only be labeled as 'S' or 'E'; similarly, a character after a name ('主') can only be labeled as 'S' or 'B', while a character before a slot ('席') can only be labeled as 'S' or 'E'.

lem (especially for the SLU data). First, these corpora tend to be news articles and thus have different genre in content. Second, these corpora are usually segmented with different standards (Huang and Zhao, 2007) and it is unclear which one would serve the purpose of the subsequent task.

Even if the NER/SLU task training data is not word segmented, the semantic slot and named entity labels actually provide valuable information on word boundaries. As illustrated in Fig. 1, the first character in an organization/person/location name can only be labeled as 'S' or 'B', while the last one can only be labeled as 'S' or 'E'; similarly, a character after a name can only be labeled as 'S' or 'B', while a character before a name can only be labeled as 'S' or 'E'. We can thus create partially-labeled CWS data from SLU and NER labels. These partially-labeled data can then be used to adapt the out-of-domain word segmenter trained from publicly-available corpus.

Täckström et al. (2013) propose the approach *partial-label learning* to learn from partially-labeled data, and Yang and Vozila (2014) apply it to Chinese word segmentation. In partial-label training, each item in the sequence receives multiple labels, and each sequence has a lattice constraint, as shown in Fig. 1. The basic idea is to marginalize the probability mass of the constrained lattice in a cost function. The marginal probability of the lattice is defined as Equation 1, where C denotes the input character sequence, L denotes the label sequence, and $\hat{Y}(C, \tilde{L})$ denotes the constrained lattice (with regard to the input sequence C and the partial-labels $\tilde{L}$).

$$p_\theta(\hat{Y}(C, \tilde{L})|C) = \sum_{L \in \hat{Y}(C, \tilde{L})} p_\theta(L|C) \qquad (1)$$

The optimization objective function is to maximize the log-likelihood of the training set, in which likelihood is calculated via the probability mass of the constrained lattice, as shown in Equation 2. Here n denotes the number of sentences in the training set.

$$L(\theta) = \sum_{i=1}^{n} \log p_\theta(\hat{Y}(C_i, \tilde{L_i})|C_i) \qquad (2)$$

With CRF[2], a gradient-based approach such as L-BFGS can be used to optimize Equation 2. We expect that this adaptation process should help to provide better word segmentation information that further improves the subsequent task performance.

3.4 N-best CWS

Only using the best word segmentation output as features for the subsequent tasks might not be sufficient (as we will show in our experiments). Indeed we can make use of the n-best word segmentation outputs. The task of SLU or NER is to find the best label sequence L, given the character sequence C, represented as $\arg\max_L P(L|C)$. By including the word segmentation information, we can rewrite it by marginalizing over all possible word segmentations.

$$\begin{aligned} \arg\max_L P(L|C) &= \arg\max_L \sum_j P(L, W_j|C) \\ &= \arg\max_L \sum_j P(L|W_j, C) \cdot P(W_j|C) \end{aligned} \qquad (3)$$

[2] We modified wapiti to implement partial learning.

Where, W_j is each possible word segmentation. This formula can be understood as two components: $P(W_j|C)$ is the word segmentation model and $P(L|W_j, C)$ is the SLU/NER model. In practice, we can use the n-best outputs associated posterior probabilities from the wapiti, for both $P(W_j|C)$ and $P(L|W_j, C)$.[3]

4 CWS for SLU

In this section, we investigate the impact of CWS to the task of spoken language understanding (SLU) by making use of existing word segmenters trained with publicly-available data (1^{st} situation in §1). This is motivated by the fact that our SLU training and testing data are not pre-segmented by semantic word units.

We choose semantic slot filling in SLU because it is becoming popular as it is a critical component to support conversational virtual assistants, such as Apple Siri, Samsung S Voice, Microsoft Cortana, Nuance Nina, just to name a few. The task of SLU is to convert a user utterance into a machine-readable semantic representation, which typically includes two sub-tasks: intent recognition and semantic slot filling (Tur et al., 2013). Intent recognition is to determine the intention of the user utterance. For example, for the input utterance 'book a ticket from Boston to Seattle', SLU will determine that its intent is *ticket-booking* as opposed to *music-playing*. Semantic slot filling is to extract the designated slot values for the recognized intent from the input utterance. For example, SLU will extract 'depart:Boston' and 'arrive:Seattle' from the above user utterance. In this paper, we assume the availability and correctness of intent recognition, and focus only on semantic slot filling.

4.1 SLU experiments setting

As described above, intent recognition is the first step in SLU, and the availability of which is assumed in this research work. We organize our training and testing data for semantic slot filling according to their intents. A single model for semantic slot filling is trained for each individual intent because different intents have different designated slots. For example, for the intent *ticket-booking*, the designated slots are the arrival and departure city/airport, airline, date, etc.; While the *local-search* intent is more interested in the city, address, street name, type of point of interest, etc. For evaluation, each model is applied to the corresponding intent's testing data. At the end, we gather the automatic semantic labels of all intents in a pool and calculate F-measure.

Our SLU data consists of about 2 million sentences for training and 260 thousand sentences for testing, distributing into 170 intents.

	CTB6	**PKU**
number of sentences	23,458	19,058
number of unique character	4,223	4,685
number of unique word	42,127	55,302
average sentence length	45.0	95.8
average word length	1.7	1.7

Table 1: Statistics of two publicly-available corpus for CWS training.

4.2 Results and discussion

We build two word segmenters from two public corpora, the Chinese Tree Bank 6 (CTB6) and the PKU corpus from the SIGHAN Bakeoff 2005, respectively. The data statistics of the two corpora are shown in Table 1.

The SLU performances are summarized in Table 2. **Baseline** using only character ngram features gives an F-measure of 93.92%. When switching to using automatic segmented words as the labeling units (**Word Unit**), the performance is a lot worse in both cases (87.10% for CTB6 and 88.68% for PKU). This indeed is not too surprising because errors in CWS propagate into SLU semantic slot filling. If an error results in a word crossing the boundary of semantic slots, it will definitely lead to an error in SLU semantic slot filling.

On the other hand, when supplying the automatic 'BIES' ngrams from CWS to SLU semantic slot filling (**As Features**), we observe a nice gain in both cases, 94.41% for CTB6 and 94.13% for PKU. Using the ngram 'BIES' as input features provides useful information of word segmentation to SLU semantic slot filling, while it is less sensitive to word segmentation errors.

[3] During training we build the SLU/NER model with 1-best word segmentation; during evaluation, we use n-best word segmentation and n-best SLU/NER.

	CTB6			PKU		
	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)
Baseline	94.10	93.73	93.92	94.10	93.73	93.92
Word Unit	89.42	84.90	87.10	90.29	87.12	88.68
As Features	94.13	94.70	94.41*	94.10	94.16	94.13*
Partial Learning	94.18	94.76	94.47*	94.19	94.77	94.48*
N-best	**94.36**	**94.84**	**94.60***	**94.37**	**94.85**	**94.61***

Table 2: SLU Results in Recall (R), Precision (P), and F-measure (F). * means it is statistically significant better than **Baseline** using a Z-test with a confidence level of 99%.

(2) Input: 查找[**湖南财政**经济学院][附近]的[餐厅][4]
CWS: 查找|**湖南**|**财政**|经济|学院|附近|的|餐厅
(Find the restaurants near Hunan College of Finance and Economics)

Example (2) illustrates that how CWS helps SLU semantic slot filling. For the sentence, the baseline system extracts '湖南财' as a location name. However, the word segmentation separates the words '湖南' (Hunan) and '财政' (Finance), which reduces the probability score of '湖南财' being a slot value because it crosses word boundaries. With CWS information, the system is able to extract '湖南财政经济学院' (Hunan College of Finance and Economics) as a slot value.

(3) Input: 转发[**淘宝网**的链接]
CWS 1: 转发|**淘**|**宝网**|的|链接
(Forward the link of bao.com)
CWS 2: 转发|**淘宝网**|的|链接
(Forward the link of taobao.com)

(4) Input: 亲[**四季酒店**]在哪里
CWS 1: **亲四季酒店**|在|哪里
(Where is the Kiss Four Seasons Hotel)
CWS 2: 亲|**四季酒店**|在|哪里
(Dear, where is the Four Seasons Hotel)

Adapting the word segmentation with SLU partially-labeled data gives further gain to semantic slot filling. In the case of CTB6 it reaches an F-measure of 94.47%, and 94.48% in PKU, using the ngram of 'BIES' labels from the adapted segmenters. Here are two examples showing how the adaptation process further improves SLU. In the example (3), we have the incorrect word segmentation (CWS 1) before adaptation. It splits a word '淘宝网' (taobao.com) and thus labels '宝网的链接' as a semantic slot. From the adaptation the system learns that '淘宝网' is a word, and it generates the correct word segmentation (CWS 2) and thus is able to create the correct semantic slot value '淘宝网的链接' (the link of taobao.com). Similarly, in the example (4), the sentence is initially under-segmented (CWS 1) and it creates the incorrect semantic slot value '亲四季酒店'. From the adaptation the system learns to put a word boundary between '亲' and '四' and then the correct slot value '四季酒店' (Four Seasons Hotel) is extracted.

Finally, we take 10-best outputs from the adapted word segmenter, for each word segmentation generate 10-best SLU outputs, sum up the probabilities, and search for the best semantic label sequence following Equation 3. We further push the performance to an F-measure of 94.60% for CTB6 and 94.61% for PKU. Compared with the baseline system that uses character ngrams as input features, the information of CWS helps us achieve an error reduction of about 11%.

5 CWS for NER

In our experiments on SLU, we showed how CWS helps the subsequent task when no in-domain word segmentation data is available (1^{st} situation in §1). In this section, we investigate the impact of CWS to another important subsequent task: named entity recognition (NER). For the NER data we use, both the domain training and testing data have word boundary information, which allows us to explore the differences between word segmenters trained with in-domain data and publicly-available data (2^{nd} situation). It also allows us to see the performance of the subsequent task using manual word segmen-

[4] Semantic slots in the input sentence are marked by '[]'.

tation (3^{rd} situation). Moreover, it allows us to see the relationship between the performance of word segmentation and the end-to-end subsequent task.

5.1 NER experiments setting

For NER experiments, we use the benchmark NER data from the third SIGHAN Chinese language processing Bakeoff (SIGHAN-3) (Levow, 2006). It consists of 46,364 sentences in the training set and 4,365 sentences in the testing set. These data are annotated with both word boundaries and NER information.

5.2 Results and discussion

Baseline system which only uses character ngram features (same configuration as the SLU task) gives the performance of 85.81% in F-measure, as shown in Table 3.[5]

Oracle system uses character ngram features together with manual in-domain word boundary information during both training and testing, showing that *perfect* word segmentation information does help NER a lot. Again this suggests that *good* word segmentation does reduce ambiguities for the subsequent NLP tasks, as we argue in the introduction. Of course, since manual word segmentation is not generally available (esp. on testing), this raises the motivation of our research work: what is the impact of automatic CWS on NER and how to make the best out of it.

To understand the impact of automatic CWS on NER, we discard the manual word segmentations in the NER data, and build two word segmenters from two public corpora, CTB6 and PKU respectively, same as we did for the SLU experiments. We also adapt them to NER with partial-label learning, and finally apply n-best CWS to NER decoding. Here we only report the results for **As Features**, as summarized in Table 3. Similar to SLU, when supplying the automatic 'BIES' ngrams from CWS to NER (**As Features**), we observe a nice gain in both cases of **CTB6** and **PKU**. The NER F-measure improves to 86.40% and 87.05% respectively. In addition, adapting the word segmentation with NER partially-labeled data gives a further gain for both CTB6 and PKU, with an F-measure of 86.96% and 87.64% respectively. Note that, the adaptation process does improve the CWS performances for both CTB6 and PKU.

In-domain CWS

NER system uses the NER training data to build a word segmenter and then apply it to the NER training and testing data to extract the word segmentation features. A naive thought is that it will result in a better NER performance than **CTB6** and **PKU** since a word segmenter trained with the in-domain data should be better than one trained with publicly-available data due to the domain mismatch issue. As shown in Table 3, it is true that the word segmentation F-measures of **NER** are much better than **CTB6** and **PKU**. However, to our surprise, the NER F-measure is only 83.45%, which is even worse than **Baseline**.

We hypothesize that this is due to the mismatch of the training CWS and testing CWS (as shown in Table 3, CWS F (train) and F (test)). When CWS accuracy is high on the training data, the NER model trained with such data puts more weight on word segmentation features rather than character features. However, during testing, the performance of CWS drops, resulting in more word segmentation errors, with a high chance to propagate to NER errors; even worse, a lot of these CWS errors are around NERs since a lot of NERs are OOVs and thus are challenging to segment correctly. To test this hypothesis, we use 3-fold cross-validation to get the word boundary information during the CWS training, and thus the model is named as **NER 3-fold**. Note, although the performance of CWS decreases in the training, it has a more balanced CWS performance between training and testing, which gives a better NER performance (improving 83.45% from **NER** to 86.80%).

N-best CWS

The model **N-best** takes N-best outputs from the adapted word segmenter, for each word segmentation generate K-best NER outputs, sums up the probabilities and searches for the best named-entity

[5] We also train a model to learn both word segmentation and NER at the same time (**Joint-learning**) using char ngram features, and then during decoding we marginalize all possible CWS sequences to search for the best NER labels. The performance, however, is only 85.39% in F-measure, suggesting it is non-trivial to leverage the gain from joint-training and the comparison between joint-training and our approaches is out of the scope of this paper.

		CWS		NER		
		F (Train) (%)	F (Test) (%)	R (%)	P (%)	F (%)
Baseline		-	-	81.63	90.44	85.81
Oracle		100	100	92.01	96.39	94.15*
CTB6	As Features	84.16	84.71	82.91	90.20	86.40
	Partial Learning	85.21	85.21	83.78	90.39	86.96*
	N-best	-	-	86.88	90.36	88.59*
PKU	As Features	86.53	87.37	84.04	90.29	87.05*
	Partial Learning	87.56	87.57	84.81	90.66	87.64*
	N-best	-	-	87.44	90.59	88.99*
NER	As Features	99.64	95.70	80.88	86.19	83.45
	N-best	-	-	84.55	87.47	85.98
NER 3-fold	As Features	94.69	95.70	83.61	90.25	86.80*
	N-best	-	-	87.22	91.30	89.21*
SIGHAN-3 Best System		-	-	84.20	88.94	86.51

Table 3: CWS and NER Results in F-measure. **CWS F (Train)** and **CWS F (Test)** are the word segmentation F-measure in the training and testing data respectively. **NER F** is the named-entity testing F-measure. '-' means that the metric does not apply. For example, **Baseline** has no word segmentation model and F-measure cannot be calculated for N-best models. For **N-best**, we set N=10 and K=2. '*' means it is statistically significant better than **Baseline** using a Z-test with a confidence level of 99%.

label sequence following Equation 3. We can see a big jump in **N-best** performances for all the models in Table 3. This verifies our hypothesis that 1-best CWS is not sufficient.

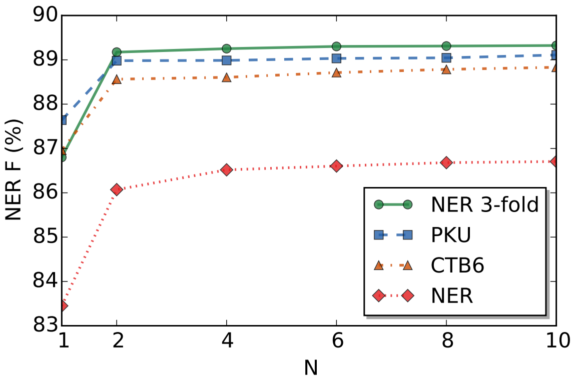

Figure 2: N-best results when varying N ($K = 1$)

To better understand how **N-best** helps NER, we vary the parameter N and the performance of NER (K=1) is shown in Fig. 2. The N-best performance improves dramatically when N jumps from 1 to 2. After that the performance seems to quickly saturate. We also found that the performance does not change much when changing K. These results show that in practice we can set N=2 and K=1, which is cost-efficient.

SIGHAN-3 evaluation

In the closed track evaluation of SIGHAN-3 (Levow, 2006), participants could only use the information found in the provided training data. Our best model (**NER 3-fold**) belongs to this track since it uses only the word segmentation annotation in the training data set. Our model outperforms all the submissions as shown in Table 3. Furthermore, even if manual word segmentation does not exist in the data, the model **CTB6 N-best** and **PKU N-best** which using existing word segmenters trained from publicly-available data can still outperform all the submissions in SIGHAN-3. Note that, these models use only character and word segmentation features without requiring additional name lists, part-of-speech taggers, etc.

6 Conclusion and future work

Chinese word segmentation is an important research topic and usually is the first step in Chinese natural language processing, yet its impact on the subsequent processing is relatively under-studied. To our knowledge, this research work is the first attempt to understand in depth how automatic CWS impacts the two related subsequent tasks: SLU semantic slot filling and named entity recognition.

In this work, we proposed three techniques to solve the domain mismatch problem when applying CWS to other tasks: using word segmentation outputs as additional features, adaptation with partial-learning and taking advantage of n-best list. All three techniques work for both tasks.

We also examined the impact of CWS in three different situations: First, when domain data has no word boundary information, we showed that a word segmenter built from public out-of-domain data is able to improve the end-to-end performance. In addition, adapting it with the partially-labeled data derived from human annotation can further improve the performance. Moreover, marginalizing n-best word segmentations leads to further improvement. Second, when domain word segmentation is available, the word segmenter trained with the domain data itself has a better CWS performance but it does not necessarily have a better end-to-end task performance. A word segmenter with more balanced performance on the training and testing data may obtain a better end-to-end performance. Third, when testing data is manually segmented, word segmentation does help the task a lot. This is not a typical use case in reality, but it does suggest that word segmentation does reduce ambiguities for the subsequent NLP tasks.

In the future, we can try to sequentially stack two CRFs (one for word segmentation and one of subsequent task). We also would like to explore more subsequent tasks beyond sequence labeling problems.

Acknowledgments

The authors would like to thank Paul Vozila, Xin Lu, Xi Chen, and Xiang Li for helpful discussions, and the anonymous reviewers for insightful comments.

References

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 224–232, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2015. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1197–1206, Lisbon, Portugal, September. Association for Computational Linguistics.

Fei Cheng, Kevin Duh, and Yuji Matsumoto. 2015. Synthetic word parsing improves chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 262–267, Beijing, China, July. Association for Computational Linguistics.

Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574, December.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 1045–1053, Stroudsburg, PA, USA. Association for Computational Linguistics.

Changning Huang and Hai Zhao. 2007. Chinese word segmentation: A decade review. *Journal of Chinese Information Processing*, 21(3):8–19, May.

Chu-Ren Huang, Petr Šimon, Shu-Kai Hsieh, and Laurent Prévot. 2007. Rethinking chinese word segmentation: Tokenization, character classification, or wordbreak identification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 69–72, Prague, Czech Republic, June. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008a. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL-08: HLT*, pages 897–904, Columbus, Ohio, June. Association for Computational Linguistics.

Wenbin Jiang, Haitao Mi, and Qun Liu. 2008b. Word lattice reranking for chinese word segmentation and part-of-speech tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 385–392, Manchester, UK, August. Coling 2008 Organizing Committee.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

Gina-Anne Levow. 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*, pages 108–117, Sydney, Australia, July. Association for Computational Linguistics.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 439–446, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Xiaoqing Li, Chengqing Zong, and Keh-Yih Su. 2013. A study of the effectiveness of suffixes for chinese word segmentation. *Sponsors: National Science Council, Executive Yuan, ROC Institute of Linguistics, Academia Sinica NCCU Office of Research and Development*, page 118.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *Trans. Audio, Speech and Lang. Proc.*, 23(3):530–539, March.

Chongjia Ni and Cheung-Chi Leung. 2014. Investigation of using different Chinese word segmentation standards and algorithms for automatic speech recognition. In *International Symposium on Chinese Spoken Language Processing*, pages 44–48.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 548–554, Lisbon, Portugal, September. Association for Computational Linguistics.

Fuchun Peng, Xiangji Huang, Dale Schuurmans, and Nick Cercone. 2002. Investigating the relationship between word segmentation performance and retrieval performance in chinese ir. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Xian Qian and Yang Liu. 2012. Joint chinese word segmentation, pos tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 501–511, Jeju Island, Korea, July. Association for Computational Linguistics.

Marc-Antoine Rondeau and Yi Su. 2015. Full-rank linear-chain neurocrf for sequence labeling. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015*, pages 5281–5285.

Weiwei Sun and Jia Xu. 2011. Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 970–979, Edinburgh, Scotland, UK., 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 - Volume 1*, HLT '11, pages 1385–1394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Fumihiko Takahasi and Shinsuke Mori. 2015. Keyboard logs as natural annotations for word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1186–1196, Lisbon, Portugal, September. Association for Computational Linguistics.

Gokhan Tur, Anoop Deoras, and Dilek Hakkani-Tur. 2013. Semantic parsing using word confusion networks with conditional random fields. In *Annual Conference of the International Speech Communication Association (Interspeech)*, September.

Xiaojun Wan, Liang Zong, Xiaojiang Huang, Tengfei Ma, Houping Jia, Yuqian Wu, and Jianguo Xiao. 2011. Named entity recognition in Chinese news comments on the web. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 856–864, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 309–317, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Longyue Wang, Shuo Li, Derek F. Wong, and Lidia S. Chao. 2012. A joint chinese named entity recognition and disambiguation system. In *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 146–151, Tianjin, China, December. Association for Computational Linguistics.

Jia Xu, Jianfeng Gao, Kristina Toutanova, and Hermann Ney. 2008. Bayesian semi-supervised chinese word segmentation for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1017–1024, Manchester, UK, August. Coling 2008 Organizing Committee.

Yan Xu, Yining Wang, Tianren Liu, Jiahua Liu, Yubo Fan, Yi Qian, Junichi Tsujii, and Eric I Chang. 2014. Joint segmentation and named entity recognition using dual decomposition in Chinese discharge summaries. *Journal of the American Medical Informatics Association*, 21(e1):84–92, February.

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

Fan Yang and Paul Vozila. 2014. Semi-supervised Chinese word segmentation using partial-label learning with conditional random fields. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–98, Doha, Qatar, October. Association for Computational Linguistics.

Xiaodong Zeng, Lidia S. Chao, Derek F. Wong, Isabel Trancoso, and Liang Tian. 2014. Toward better chinese word segmentation for smt via bilingual constraints. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1360–1369, Baltimore, Maryland, June. Association for Computational Linguistics.

Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu. 2004. Using n-best lists for named entity recognition from Chinese speech. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 37–40, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 843–852, Cambridge, MA, October. Association for Computational Linguistics.

Ruiqiang Zhang, Keiji Yasuda, and Eiichiro Sumita. 2008. Improved statistical machine translation by multiple Chinese word segmentation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 216–223, Columbus, Ohio, June. Association for Computational Linguistics.

Longkai Zhang, Li Li, Zhengyan He, Houfeng Wang, and Ni Sun. 2013a. Improving chinese word segmentation on micro-blog using rich punctuations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 177–182, Sofia, Bulgaria, August. Association for Computational Linguistics.

Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013b. Exploring representations from unlabeled data with co-training for Chinese word segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 311–321, Seattle, Washington, USA, October. Association for Computational Linguistics.