Improving Chinese Word Segmentation and POS Tagging with Semi-supervised Methods Using Large Auto-Analyzed Data

Yiou Wang[†], Jun'ichi Kazama[†], Yoshimasa Tsuruoka[‡†], Wenliang Chen[§†], Yujie Zhang[*†], Kentaro Torisawa[†]
[†]National Institute of Information and Communications Technology (NICT), Japan
[‡]School of Information Science, JAIST, Japan; [*]Beijing Jiaotong University, China
[§]Human Language Technology, Institute for Infocomm Research, Singapore
{wangyiou, kazama, torisawa}@nict.go.jp; tsuruoka@jaist.ac.jp
wechen@i2r.a-star.edu.sg;yjzhang@bjtu.edu.cn

Abstract

This paper presents a simple yet effective semi-supervised method to improve Chinese word segmentation and POS tagging. We introduce novel features derived from large auto-analyzed data to enhance a simple pipelined system. The auto-analyzed data are generated from unlabeled data by using a baseline system. We evaluate the usefulness of our approach in a series of experiments on Penn Chinese Treebanks and show that the new features provide substantial performance gains in all experiments. Furthermore, the results of our proposed method are superior to the best reported results in the literature.

1 Introduction

In Chinese, most language processing starts from word segmentation and part-of-speech (POS) tagging. These two steps tokenize a sequence of characters without delimiters into words and predict a syntactic label (POS tag) for each segmented word. They are considered indispensable steps for higher-level NLP tasks such as parsing and information extraction. Although the performance of Chinese word segmentation and POS tagging has been greatly improved over the past years, the task is still challenging.

To improve the accuracy of NLP systems, one of the current trends is semi-supervised learning, which utilizes large unlabeled data in supervised learning. Several studies have demonstrated that the use of unlabeled data can improve the performance of NLP tasks, such as text chunking (Ando and Zhang, 2005), POS tagging and named entity recognition (Suzuki and Isozaki, 2008), and parsing (Suzuki et al., 2009; Chen et al., 2009; Koo et al., 2008). Therefore, it is attractive to consider adopting semi-supervised learning in Chinese word segmentation and POS tagging tasks.

In this paper, we present an approach to improve the performance of both segmentation and POS tagging by incorporating large unlabeled data. We first preprocess unlabeled data with our baseline models. We then extract various items of dictionary information from the auto-analyzed data. Finally, we generate new features that incorporate the extracted information for both word segmentation and POS tagging. We also perform word clustering on the auto-segmented data and use word clusters as features in POS tagging. In addition, we introduce lexicon features by using a cross-validation technique.

The use of sub-structures from the auto-annotated data has been presented previously (Noord, 2007; Chen et al., 2008; Chen et al., 2009). Chen et al. (2009) extracted different types of subtrees from the auto-parsed data and used them as new features in standard learning methods. They showed this simple method greatly improves the accuracy of dependency parsing. The idea of combining word clusters with discriminative learning has been previously reported in the context of named entity recognition (Miller et al., 2004; Kazama and Torisawa, 2008) and dependency parsing (Koo et al., 2008). We adapt and extend these techniques to Chinese word segmentation and POS tagging, and demonstrate their effectiveness in this task.

One of our criteria in this study was to achieve high accuracy with simple and easy-to-implement techniques. To meet this, the whole system is a pipeline with a character-based CRF for word segmentation and a word-based CRF for POS tagging. The information of unlabeled data is incorporated as additional new features without changing the learning algorithm.

To demonstrate the effectiveness of our approach, we conduct segmentation and POS tagging experiments on three versions of Penn Chinese Treebank, including the newly released CTB

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 309–317,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

Word Length	1	2	3	4	5	6	7 or more
Tags	S	BE	BB_2E	BB_2B_3E	BB_2B_3ME	BB_2B_3MME	$BB_2B_3M...ME$

Table 1: Word representation with a 6-tag tagset : S, B, B_2, B_3, M, E

Type	Feature	Description
Character Unigram	c_{-1}, c_0, c_1	Previous, current and next character
Nearing Character Bigram	(c_{-1} c_0), (c_0 c_1)	Previous (next) character and current character
Jump Character Bigram	c_{-1} c_1	Previous character and next character
Punctuation	IsPu(c_0)	Current character is punctuation
Character Type	$K(c_{-2})K(c_{-1})K(c_0)K(c_1)K(c_2)$	Types of character: date, numeral, alphabet, Chinese

Table 2: Feature templates for word segmentation

version 7.0. We show that our semi-supervised approach yields improvements for all the test collections and achieves better results than the best reported results in the literature.

2 Segmentation and POS tagging Models

We implement our approach using sequential tagging models. Following the previous work (Zhao et al., 2006; Zhao et al., 2010), we employ the linear chain CRFs (Lafferty et al., 2001) as our learning model. Specifically, we use its CRF++ (version 0.54) implementation by Taku Kudo. [1]

2.1 Baseline Segmentation Model

We employ character-based sequence labeling for word segmentation. In addition to its simplicity, the advantage of a character-based model is its robustness to the unknown word problem (Xue, 2003). In a character-based Chinese word segmentation task, a character in a given sequence is labeled by a tag that stands for its position in the word that the character belongs to. Zhao et al. (2006) reported that a 6-tag tagset shown in Table 1 is the best choice among the tagsets tested for Chinese word segmentation under the CRF framework. Therefore we also use this 6-tag tagset.

The basic types of features of our word segmentation model are listed in Table 2. These basic feature templates are based on Zhao et al. (2006; 2010) and Low et al. (2005).

2.2 Baseline POS Tagging Model

Since we employ a pipelined method, the POS tagging can be performed as a word labeling task, where the input is a sequence of segmented words. We use a CRF here as well. The feature set of our baseline POS tagger, is listed in Table 3. These are adopted from Wu et al. (2008).

[1] Available from http://crfpp.sourceforge.net/

3 Our New Features

In this section, we describe our approach of effectively integrating useful information from unlabeled (and labeled) data into the above baseline models through features. We preprocess unlabeled data with our baseline models and obtain word-segmented sentences with POS tags, and generate new features from the auto-analyzed data. Although the focus of the paper is semi-supervised learning, we also extract a lexicon from the training corpus and use it to generate features. Figure 1 shows an overview of our approach. The rest of this section describes our features in detail.

3.1 New features for Word Segmentation

3.1.1 Semi-supervised *n*-gram features

In this section, we describe our approach of extracting character-level *n*-gram lists and generating *n*-gram features from unlabeled data. We followed the method of Chen et al. (2009), and modified the method for word segmentation and POS tagging. First, we preprocess unlabeled data using the baseline segmenter and obtain auto-segmented data. We then extract character *n*-gram lists from auto-segmented sentences. Finally, we generate *n*-gram features for word segmentation.

By using the baseline segmenter, each character c_i in the unlabeled data is labeled with a tag t_i. In other words, the output of auto-segmentation is a sequence $\{(c_i, t_i)\}_{i=1}^{L}$. Let g be a character *n*-gram (e.g., uni-gram c_i, bi-gram c_ic_{i+1}, tri-gram $c_{i-1}c_ic_{i+1}$ and so on)[2], and seg be a segmentation profile for n-gram g observed at each position. The segmentation profile can be tag t_i or the combination of tags. Take a bi-gram for example, seg may be t_i or t_it_{i+1}. Then,

[2] Note that there are several alternative ways for extracting n-grams at position i, for example $c_{i-1}c_i$ for a bi-gram. In this paper, we used the way as explained here.

Feature Type	Context Position	Description
Word Unigram	w_{-2},w_{-1},w_0,w_1,w_2	Word unigram
Nearing Word Bigram	($w_{-2}w_{-1}$),($w_{-1}w_0$),(w_1w_0),(w_1w_2)	Word bigram
Jump Word Bigram	(w_{-1},w_1)	Previous word and next word
First Character	$Fc(w_0)$	First character of current word
Last Character	$Lc(w_0)$	Last character of current word
Length	$Len(w_0)$	Length of current word

Table 3: Feature templates for POS tagging

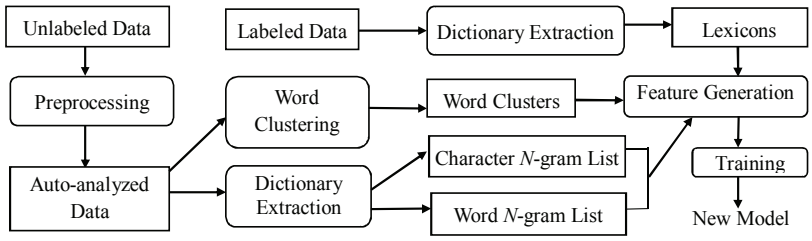

Figure 1: Overview of the proposed approach

we can extract a list of $\{(g, seg, f(g, seg))\}$ from the auto-segmented data. Here, $f(g, seg)$ is the frequency of the cases where n-gram g is segmented with the segmentation profile seg. Then, following Chen et al. (2009), we group entries in this list into three sets: high-frequency (HF), middle-frequency (MF), and low-frequency (LF). The sets are defined as follows: if (g, seg) is one of the top 5% most frequent entries, it is labeled as HF; if it is between top 5% and 20%, it is labeled as MF, otherwise it is labeled as LF. Finally the list can be transformed as a *n*-gram list $L_{ng} = \{(g, seg, FL(g, seg))\}$, with $FL(g, seg)$ being the frequency label determined as above.

We attempted to encode the information of the above *n*-gram list into a new type of features, called *n*-gram features. We tried several feature representations and generation methods and found that the feature derived from the bi-gram list with $seg = t_i$ was most effective.

We generate the feature for the current character c_0 as follows. We retrieve a set of entries, whose g part matches the bi-gram c_0c_1, from L_{ng}. Let this set be L_m. From an entry in L_m, we generate a feature string represented by

(a) $seg - FL(g, seg)$

Then, we concatenate the feature strings of all the entries in L_m as one n-gram feature. If there is no entry in L_m, the feature representation is "*ND*".

For example, consider that L_m is { (幸*(Xing)*/福*(Fu)*, *B*, *HF*), (幸*(Xing)*/福*(Fu)*, *B2*, *MF*), (幸*(Xing)*/福*(Fu)*, *E*, *LF*)} and we are processing c_kc_{k+1} = "幸*(Xing)*/福*(Fu)*"; consequently, the *n*-gram feature of c_k is represented as "*B-HF|B2-MF|E-LF*". Note that the concatenation is in lexicographic order of the feature strings for standardization.

3.1.2 Lexicon features

Although a character-based model is simple and robust to unknown words, a limitation is its inability to consider word-level information. If a sequence of characters matches a word in an existing dictionary, it may be a clue that the character sequence should be segmented as one word. Several studies showed that using a dictionary brings improvement for Chinese word segmentation (Low et al., 2005; Zhao et al., 2010). For a corpus-based word segmenter, a manually annotated corpus is essential. Thus we can easily compile a lexicon from a training corpus. We refer to the features related to this lexicon as lexicon features.

In this study, we extract a lexicon in the following way. We collect words and all possible POS tags of the words from the training corpus. For instance, for the word "交流(*JiaoLiu*)", the collected entry may be (交流(*JiaoLiu*), *NN-VV*). Here, "*NN-VV*" is a concatenation of all the observed POS tags. POS tags are in lexicographical order, as in "*NN-VV*". However, we were concerned that a lexicon compiled in this way could cause an overfitting problem and that meaningful weights for the lexicon features may not be learned. This concern was indeed confirmed by the preliminary experiments using the development set. To solve this problem, we used the following method to build and use lexicons. The method is based on the idea

of cross-validation.

- Divide the training corpus into ten equal-sized sets, as in the data preparation for 10-fold cross-validation.
- For each set, we compile a lexicon using the remaining nine sets and use this lexicon to generate features for this set.
- For the development and test sets, we collect a lexicon using the entire training corpus and use it for feature generation.

Because the lexicon is extracted from other sets, the weights for this feature will not be overestimated by the learning algorithm. This kind of cross-validation-like techniques are used in studies such as Collins (2002) and Martins et al. (2008) to avoid over-fitting to the training data. Our method can be considered as its application to lexicon extraction.

Using the extracted lexicon, we generate lexicon features as follows. If a character sequence starting with character c_0 matches some words in the lexicon, we greedily choose the longest such matching word w. Letting $LEN(w)$ be the length (the number of characters) of w, we add the following feature for each character c_k in $c_0, c_1, ..., c_{LEN(w)}$:

(b) $P(c_k)/LEN(w)\text{-}POSs(w)$

Here, $P(c_k)$ is the position number (i.e., k) of the character c_k in w and $POSs(w)$ represents the POS tags of w in the lexicon. After generating these features, we increment the current position by $LEN(w)$. If there is no matching word, we proceed to the next character. That is, the forward maximum matching is used.

For example, consider that the current character sequence c_0c_1= "幸*(Xing)*/福*(Fu)*" was matched with a lexicon entry (幸福*(XingFu), JJ-NN-VA*), the feature for c_0 "幸*(Xing)*" is "*1/2-JJ-NN-VA*" and the feature for c_1 "福*(Fu)*" is "*2/2-JJ-NN-VA*".

Several feature representations have been attempted: (i) using only position information, (ii) representing the position information in a 6-tag or 4-tag tagset, or (iii) representing only one POS tag with the highest frequency. Development experiments showed that the current encoding is more effective than others in word segmentation tasks.

Note that our lexicon feature uses POS tag information for word segmentation. The fact that this feature is very effective as reported in Section 4.3 is interesting, since this can be considered as "loose" information feedback from the later process. Although we need a POS tagged corpus even for segmentation, this will not be a big problem since we usually perform POS tagging as well in many applications.

3.2 New Features for POS Tagging

We generate *n*-gram and lexicon features for POS tagging as well. In addition, the features that incorporate word clusters derived from a large auto-analyzed corpus (referred to as cluster features) are introduced.

3.2.1 Semi-supervised *n*-gram features

We preprocess auto-segmented data using the baseline POS tagger and can generate word-level *n*-gram lists $L_{wg} = \{w, pos, FL(w, pos)\}$. Here, w is a word *n*-gram and pos is the POS tagging profile of the word *n*-gram. Different from segmentation, features generated from the word unigram list yielded the best results.

A feature of this type for the current word w_0 is generated as follows. We retrieve a set of entries, whose w part matches the uni-gram w_0, from L_{wg}. Let this set be L_m. In the error analysis, we found that some words were associated with several odd POS tags in the uni-gram list. For instance, in addition to (研究*(YanJiu), NN, HF*) and (研究*(YanJiu), VV, HF*), (研究*(YanJiu), VA, LF*) and (研究*(YanJiu), CD, LF*) may appear as entries in the word unigram list, due to mis-tagging by the baseline POS-tagger. Therefore we further impose a restriction based on the frequency as follows: if the number of entries with a HF label $\geq$ threshold, only the entries with HF will be selected, and if the sum of entries with a HF or MF label $\geq$ threshold, the entries with either HF or MF will be selected, otherwise, all of the entries in L_m will be selected. Here the threshold is set to 2 based on the development experiments. Let these selected entries be L_s. From an entry in L_s, we generate a feature string represented by

(c) $pos - FL(w, pos)$.

Then, we concatenate the feature strings of all entries in L_s as one n-gram feature. As for the previous instance, the feature for "研究*(YanJiu)*" is encoded as "$NN\text{-}HF|VV\text{-}HF$".

3.2.2 Semi-supervised cluster features

Following the work of Koo et al. (2008), we produced the clusters of various levels of granularity,

Data set	CTB chapter IDs
Dev	41-80,203-233,301-325,400-409,591,613-617,643-673,1022-1035,1120-1129,2110-2159,2270-2294,2510-2569,2760-2799,3040-3109,4040-4059,4084-4085,4090,4096,4106-4108,4113-4115,4121,4128,4132,4135,4158-4162,4169,4189,4196,4236-4261,4322,4335-4336,4407-4411
Test	1-40,144-174,271-300,410-428,592,900-931,1009-1020,1036,1044,1060-1061,1072,1118-1119,1132,1141-1142,1148,2000-2010,2160-2220,2295-2330,2570-2640,2800-2845,3110-3145,4030-4039,4060-4070,4086-4087,4091,4097,4109-4112,4118-4120,4127,4133-4134,4136-4139,4163-4168,4188,4197-4235,4321,4334,4337,4400-4406

Table 4: Dev-set and test-set of CTB7 data split

	Total	Dev-$_{LDC}$	Test-$_{LDC}$	Dev	Test
NS	107,14	561	981	2,084	2,028
NM	8,420	682	917	1,618	1,646
BN	10,079	836	898	2,067	2,038
BC	12,049	0	0	2,367	2,382
NW	10,181	0	0	2,000	2,086

Table 5: Statistics of each genre of CTB7 split (Dev(Test)-$_{LDC}$ are sets of LDC split)

by using the prefixes of the Brown cluster hierarchy at various lengths [3]. After experimenting with many different feature configurations, we eventually settled on the following features:

(d) full string prefixes for w_{-1}, w_0, w_1

6-bit string prefixes for w_{-1}, w_0, w_1

The clusters are best exploited when "anchored" to words or parts of speech (Koo et al., 2008). We found it useful to make the above features in Bigram template, in CRF++ with the first character "B". With this template, a combination of the current output tag and the previous output tag (bigram) is automatically generated. In this case, the combination of the current POS tag and the previous POS tag output is automatically generated.

3.2.3 Lexicon features

We use the same lexicon extracted for word segmentation for POS tagging. We add the following feature for the current word w_0:

(e) $POSs(w_0)$

Here, $POSs(w_0)$ are all possible POS tags of the current word w_0 in the lexicon. We also tried to use different lexicons, as well as representing the feature with only one POS tag with the highest frequency. However, the experimental results were not better than those by using the above simple method.

[3]We used the word clustering tool, available from http://www.cs.berkeley.edu/p̃liang/software/brown-cluster-1.2.zip, to produce word clusters.

4 Experiments

We conducted word segmentation and POS tagging experiments on Penn Chinese Treebanks incorporating up to 200-million-word unlabeled data.

4.1 Data Set

To compare with previous studies, we selected the widely used CTB5 (LDC2005T01), and defined the training, development and test sets according to Kruengkrai (2009a). In order to increase the reliability of our findings, we also used CTB6 (LDC2007T36) and CTB7 (LDC2010T07), which are larger than CTB5. For CTB6, we used the same data split as recommended in the CTB6 document [4]. Because CTB7 includes data from various sources and various genres, we made a new data split according to the following criteria:

• Put the test set and the development set data described in CTB7 documents [5] into each data set.

• Put the test set and the development set data of CTB5 into each set.

• Put all double checked files into the test-set. [6]

• Keep the data of different genres and sources in balance.

• Increase the size of the development and test sets to make the evaluation more reliable. [7]

The test set and development set of the CTB7 data split used in this paper are detailed in Table 4, and we used the rest as the training set. Table 5 provides the detailed statistics of each genre: NS (Newswire), NM (News magazine), BN (Broadcast news), BC (Broadcast conversation), NW (Newsgroups weblogs). Table 6 provides the statistics of our experimental settings on the treebanks. The out-of-vocabulary (OOV) is defined as

[4]list-of-files.pdf

[5]This is the same as the CTB6 data split.

[6]In CTB7, sentences checked twice are marked, and they are expected to have higher annotation quality.

[7]CTB5 and CTB6 data splits include small development and test sets.

	# of sent. training	# of sent. dev	OOV rate (word) dev	OOV rate(word & tag) dev	# of sent. test	OOV rate (word) test	OOV rate(word & tag) test
CTB5	18,089	350	0.0811	0.0877	348	0.0347	0.0420
CTB6	23,420	2,079	0.0545	0.0635	2,796	0.0557	0.0636
CTB7	31,131	10,136	0.0549	0.0634	10,180	0.0521	0.0608

Table 6: Statistics of CTB5, CTB6 and CTB7 data splits

	CTB5			CTB6			CTB7		
method	R	P	F_1	R	P	F_1	R	P	F_1
Baseline	0.9791	0.9715	0.9753	0.9504	0.9521	0.9513	0.9503	0.9492	0.9498
+(a) *n*-gram	0.9830	0.9766	0.9798	0.9567	0.9568	0.9567	0.9562	0.9546	0.9554
+(b) lexicon	0.9809	0.9743	0.9776	0.9545	0.9555	0.9550	0.9548	0.9535	0.9542
+(a)+(b)	0.9845	0.9777	0.9811	0.9575	0.9583	0.9579	0.9576	0.9554	0.9565

Table 7: Results of word segmentation

POS tag method	CTB5	CTB6	CTB7
Baseline	0.9318	0.8999	0.8937
+(c) *n*-gram	0.9333	0.9014	0.8958
+(d) cluster	0.9350	0.9026	0.8959
+(e) lexicon	0.9346	0.9015	0.8959
+(c)+(d)+(e)	0.9359	0.9048	0.8985

Table 8: F_1 results of segmentation and POS tagging (baseline model for word segmentation)

POS tag method	CTB5	CTB6	CTB7
Baseline	0.9362	0.9061	0.8996
+(c) *n*-gram	0.9382	0.9078	0.9017
+(d) cluster	0.9403	0.9089	0.9020
+(e) lexicon	0.9399	0.9081	0.9019
+(c)+(d)+(e)	0.9418	0.9112	0.9046

Table 9: F_1 results of segmentation and POS tagging (our best model for word segmentation)

the words in the test set that are not in the training set (Sproat and Emerson, 2003). The development sets were used to obtain the optimal values of tunable parameters and feature configurations.

The unlabeled data for our experiments were taken from the XIN_CMN portion of Chinese Gigaword Version 2.0 (LDC2009T14), which has approximately 311 million words. Some of CTB data and Chinese Gigaword data are from the same source: Xinhua newswire between 1994 and 1998. In order to avoid overlap between the CTB data and the unlabeled data, we used only the articles published in 1991- 1993 and 1999-2004 as unlabeled data, with 204 million words.[8] Note that we only used one million words from this data for word clustering, because the clustering process is time-consuming and the amount is enough to show the impact of cluster feature.

4.2 Parameter Tuning

CRF++ has four major tunable parameters to control the training condition: *a*, the regularization algorithm; *c*, the balance between over-fitting and under-fitting; *f*, the cut-off threshold for the feature frequencies; and *p*, the number of threads. We used $a = CRF\text{-}L2$ (Gaussian regularization) and $f = 1$. We set p to 12 for all experiments to speed up the training. For the baseline segmentation model, we varied c in the range of [1.0, 5.0] and found that setting $c = 4.0$ yielded the best performance on the development set of CTB7. For our approach, we varied c in the range of [0.3, 5.0] and found that setting $c = 1.0$ yielded the best performance. For the POS tagging model, c was set to 4.0 in all of the methods . For the clustering tool, c (the number of clusters) was set to 1000.

4.3 Experimental Results

We evaluated both word segmentation (Seg) and joint word segmentation and POS tagging (Seg &Tag). We used recall (R), precision (P) and F_1 as evaluation metrics.

The experimental results of word segmentation on CTB5, CTB6 and CTB7 test sets are shown in Table 7, where (a) refers to the *n*-gram feature generated from the unlabeled data and (b) refers to the lexicon feature. The results show that the *n*-gram feature was very effective in all experiments and that the combination of (a) and (b) can provide further improvement.

The experimental results of segmentation and POS tagging on CTB5, CTB6 and CTB7 test sets are shown in Table 8 and Table 9. Table 8 shows the results when we used the baseline segmenta-

[8]This may be a too strict setting, but we wanted to test our approach in the fairest way.

	CTB5		CTB6		CTB7	
Method	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag
Ours	0.9628	**0.9316**	**0.9619**	**0.9138**	0.9536	**0.9027**
Baseline	0.9493	0.8934	0.9564	0.9052	0.9493	0.8934
K 09b	0.9628	0.9291	0.9577	0.9063	**0.9547**	0.8989
K 09a	**0.9642**	0.9288	0.9574	0.9061	0.9533	0.8984

Table 12: F_1 Results comparison on development set

Method	Seg	Seg&Tag
Ours	**0.9811**	**0.9418**
Baseline	0.9753	0.9318
Z&C 10	0.9778	0.9367
K 09a	0.9787	0.9367
K 09b	0.9798	0.9400
Jiang 08a	0.9785	0.9341
Jiang 08b	0.9774	0.9337
N&U 07	0.9796	0.9338

Table 10: Comparison with previous studies on CTB5

	CTB6		CTB7	
Methods	Seg	Seg&Tag	Seg	Seg&Tag
Ours	**0.9579**	**0.9112**	**0.9565**	**0.9046**
Baseline	0.9513	0.8999	0.9498	0.8937
K 09a	0.9550	0.9050	0.9540	0.8986
K 09b	0.9551	0.9053	0.9546	0.8990

Table 11: Comparison with previous studies on CTB6 and CTB7

tion model and Table 9 shows the results when we used our best segmentation model (i.e., (a)+(b)). The results show that the cluster features were the most effective ones and that a combination of three types of features achieves the best performance. This suggests that these features are relatively independent in feature characteristics.

The results of our best system are compared with the previous methods in the next section.

4.4 Comparative Results

In this section, we compare our approach with the best previous approaches reported in the literature. The performance scores of previous studies are directly taken from their papers, except for N&U 07 (Nakagawa and Uchimoto, 2007), which is taken from Kruengkrai et al. (2009b). Z&C 10 refers to Zhang and Clark (2010). Two methods in Kruengkrai et al. (2009a; 2009b) are referred to as K 09a and K 09b. Jiang 08a and Jiang 08b refer to Jiang et al. (2008a; 2008b). Table 10 compares F_1 results on CTB5.0. The best score in each column is in boldface. The results of our approach are superior to those of previous studies for both Seg and Seg&Tag.

	p-value		
Models	CTB5	CTB6	CTB7
Ours vs. K 09b(Seg)	0.8054	5.0e-08	≈ 0.0
Ours vs. K 09b(Seg&Tag)	0.7060	1.6e-14	≈ 0.0
Ours vs. Base(Seg)	4.0e-06	1.8e-11	≈ 0.0
Ours vs. Base(Seg&Tag)	2.1e-06	≈ 0.0	≈ 0.0

Table 13: Results of McNemar's test.

We also conducted experiments using the system implemented by Kruengkrai for comparison on CTB6 and CTB7 with two methods (K 09a and K 09b) and the F_1 results are shown in Table 11.

For reference, the results of the development set are also shown in Table 12. Although the Seg performances of CTB5 and CTB7 are lower than K 09a and K 09b , Seg&Tag achieves the best performance on all development sets.

4.5 Statistical Significance Tests

We evaluated statistical significance using McNemar's test [9]. With McNemar' test, we compared the correctness of the labeling decisions of the two models. The null hypothesis is that the disagreements (correct vs. incorrect) are due to chance. For Seg, a word in the system output is considered correct if the word boundary is correctly identified. For Seg &Tag, a word is considered correct only when both the word boundary and its POS tag are correctly identified. Table 13 summarizes the results on test sets. These tests suggest that although the difference from K 09b for CTB5 data is not statistically significant, all other differences are clearly statistically significant ($p < 10^{-5}$).

4.6 Comparison with Self-Training

An alternative method of incorporating unlabeled data is self-training, so we also compared our results to the self-training method. Because no existing research was found concerning the self-training method on word segmentation and POS

[9] We used the version with Yates' correction, using correction factor 0.5

Sentences added	Segmentation F_1
0(Baseline)	0.9498
5k	0.9493
10k	0.9492
30k	0.9482
150k	0.9469
300k	0.9469
600k	0.9468

Table 14: Comparison with self-training (Seg)

sentences added	POS tagging F_1
0(Baseline)	0.8937
5k	0.8926
10k	0.8922
30k	0.8911
50k	0.8908

Table 15: Comparison with self-training (POS)

tagging for Chinese, we tested the simplest self-training here. We analyzed the unlabeled data with the baseline models, added the newly auto-labeled data to the training corpus, and trained a new model. Since the manually labeled data should be considered more important than the unlabeled data (McClosky et al., 2006), we also adjusted the weight of the labeled data to the integer in the range of [1,5] in experiments. The results of all the experiments were not positive – we were not able to obtain any improvement over the baseline models in either word segmentation or POS tagging. Due to space limitation, we only include the results with the labeled data weight = 1. Other weights did not change the conclusion here. Table 14 shows the F_1 results on segmentation with different sizes of the additional data on the CTB7 test set. Table 15 shows the F_1 results on segmentation and POS tagging. The segmentation by the baseline model was used for all of the POS tagging experiments here.

5 Related Work

Our approach is to incorporate large unlabeled data in Chinese word segmentation and POS tagging.

For research using large unlabeled data, Suzuki and Isozaki (2008) and Suzuki et al. (2009) proposed semi-supervised learning algorithms on giga-word-scale unlabeled data and showed performance improvement in POS tagging, syntactic chunking, and named entity recognition. Instead of using specialized semi-supervised learning algorithms, Chen et al. (2009) used features based on sub-structures in auto-parsed data and demonstrated the effectiveness of these features. Koo et al. (2008) presented the use of cluster features. The advantage of the methods by Chen et al. (2009) and Koo et al. (2008) is their simplicity and flexibility. Our research applied these techniques to word segmentation and POS tagging rather than dependency parsing.

Yu et al. (2007) proposed a character-based joint method for word segmentation and POS tagging, in which they introduced an unsupervised method for unknown word learning. However, they only learned the unknown words from the test set. Zhao and Kit (2007; 2008) proposed an approach using unsupervised segmentation criteria as features for Chinese word segmentation. However, their features were only accumulated from the training and test data. Our approach differs in that we used features generated from large unlabeled data and provided richer information, which may be unseen from the training corpus.

Kruengkrai et al. (2009a; 2009b) presented a discriminative word-character hybrid model for joint Chinese word segmentation and POS tagging and achieved the state-of-the-art accuracy for the CTB test sets. Instead of using the hybrid model, we used conceptually simpler pipelined models built with standard CRF tools. Compared with their method, our approach achieved higher performance with the help of unlabeled data.

6 Conclusion

In this paper, we presented a simple yet effective semi-supervised approach to pipelined Chinese segmentation and POS tagging. Through a series of experiments, we demonstrated that our approach provides substantial improvement over the best previously reported methods as well as the baseline methods.

References

Andr F. T.Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing 2008. *Stacking Dependency Parsers*. In Proceedings of EMNLP-2008, pages 513-521.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara 2009. *An Error-Driven Word-Character Hybird Model for Joint Chinese Word Segmentation and POS Tagging*. In Proceedings of ACL-IJCNLP-2009, pages 513-521.

Canasai Kruengkrai Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara 2009. *Joint Chinese Word Segmentation and POS Tagging Using an Error-Driven Word-Character Hybrid Model*. IEICE transactions on information and systems 92(12), pages 2298-2305.

David McClosky, Eugene Charniak, and Mark Johnson 2006. *Effective self-training for parsing*. In Proceedings of the Human Language Technology Conference of the NAACL-2006, pages 152-159.

Gertjan van Noord 2007. *Using Self-trained Ailexical Preferences to Improve Disambiguation Accuracy*. In Proceedings of IWPT-07, pages 1-10

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu 2006. *Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling*. In Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation.pages 87-94.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu 2010. *A Unified Character-Based Tagging Framework for Chinese Word Segmentation*. ACM Transactions on Asian Language Information Processing, 9(2), Article 5.

Hai Zhao and Chunyu Kit 2008. *Exploiting Unlabeled Text with Different Unsupervised Segmentation Criteria for Chinese Word Segmentation*. Research in Computing Science, Vol. 33, pages 93-104.

Hai Zhao and Chunyu Kit 2007. *Incorporating Global Information into Supervised Learning for Chinese Word Segmentation*. In Proceedings of PACLING-2007, pages 66-74.

Jin Kiat Low, Hwee Tou Ng and Wenyuan Guo 2005. *A Maximum Entropy Approach to Chinese Word Segmentation*. In Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing (SIGHAN05), pages 161-164.

John Lafferty, Andrew McCallum, and JFernando Pereira 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In Proceedings of ICML01, pages 282-289.

Jun'ichi Kazama and Kentaro Torisawa 2008. *Inducing Gazetteers for Named Entity Recognition by Large-scale Clustering of Dependency Relations*. In Proceedings of ACL-2008, pages 665-673.

Jun Suzuki and Hideki Isozaki 2008. *Semi-Supervised Sequential Labeling and Segmentation using Giga-word Scale Unlabeled Data*. In Proceedings of ACL-08: HLT, pages 407-415.

Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins 2009. *An Empirical Study of Semi-supervised Structured Conditional Models for Dependency Parsing*. In Proceedings of EMNLP-2009, pages 551-560.

Kun Yu, Sadao Kurohashi, Hao Liu 2007. *Character-based Chinese Word Segmentation and Pos-tagging with Unsupervised Unknown Word Learning*. In Proceedings of NLP-2007, pages 823-826.

Michael Collins 2002. *Ranking Algorithms for Named-entity Extraction: Boosting and the Voted Perceptron*. In Proceedings of ACL-2002, pages 489-496

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

Richard Sproat and Thomas Emerson 2003. *The First International Chinese Word Segmentation Bakeoff*. In Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing, pages, 133-143.

Rie Kubota Ando and Tong Zhang 2005. *A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data*. Journal of Machine Learning Research, 6, pages 1817-1853

Scott Miller, Jethran Guinness, and Alex Zamanian 2004. *Name Tagging with Word Clusters and Discriminative Training*. In Proceedings of HLT-2004, pages 337-342

Terry Koo, Xavier Carreras and Michael Collins 2008. *Simple Semi-supervised Dependency Parsing*. In Proceedings of ACL-2008, pages 595-603

Tetsuji Nakagawa and Kiyotaka Uchimoto 2007. *Hybrid Approach to Word Segmentation and Pos Tagging*. In Proceedings of ACL Demo and Poster Sessions, pages 217-220

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lu. 2008. *A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging*. In Proceedings of ACL-2008, pages 897-904

Wenbin Jiang, Haitao Mi and Qun Liu 2008. *Word Lattice Reranking for Chinese Word Segmentation and Part-of-Speech Tagging*. In Proceedings of COLING-2008, pages 385-392

Wenliang Chen, Daisuke Kawahara, Kiyotaka Uchimoto, Yujie Zhang, and Hitoshi Isahara 2008. *Dependency Parsing with Short Dependency Relations in Unlabeled Data*. In Proceedings of IJCNLP-2008

Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa 2009. *Improving Dependency Parsing with Subtrees from auto-Parsed Data*. In Proceedings of EMNLP-2009, pages 570–579,

Yu-Chieh Wu Jie-Chi Yang and Yue-Shi Lee 2008. *Description of the NCU Chinese Word Segmentation and Part-of-Speech Tagging for SIGHAN Bakeoff 2008*. In Proceedings of the SIGHAN Workshop on Chinese Language Processing, pages 161-166.

Yue Zhang and Stephen Clark 2010. *A Fast Decoder for Joint Word Segmentation and POS-Tagging Using a Single Discriminative Model*. In Proceedings of EMNLP-2010, pages 843-852