Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Word Segmentation for Chinese Novels

Likun Qiu and Yue Zhang
Singapore University of Technology and Design
20 Dover Drive, Singapore 138682
qiulikun@gmail.com, yue_zhang@sutd.edu.sg

Abstract

Word segmentation is a necessary first step for automatic syntactic analysis of Chinese text. Chinese segmentation is highly accurate on news data, but the accuracies drop significantly on other domains, such as science and literature. For scientific domains, a significant portion of out-of-vocabulary words are domain-specific terms, and therefore lexicons can be used to improve segmentation significantly. For the literature domain, however, there is not a fixed set of domain terms. For example, each novel can contain a specific set of person, organization and location names. We investigate a method for automatically mining common noun entities for each novel using information extraction techniques, and use the resulting entities to improve a state-of-the-art segmentation model for the novel. In particular, we design a novel double-propagation algorithm that mines noun entities together with common contextual patterns, and use them as plug-in features to a model trained on the source domain. An advantage of our method is that no retraining for the segmentation model is needed for each novel, and hence it can be applied efficiently given the huge number of novels on the web. Results on five different novels show significantly improved accuracies, in particular for OOV words.

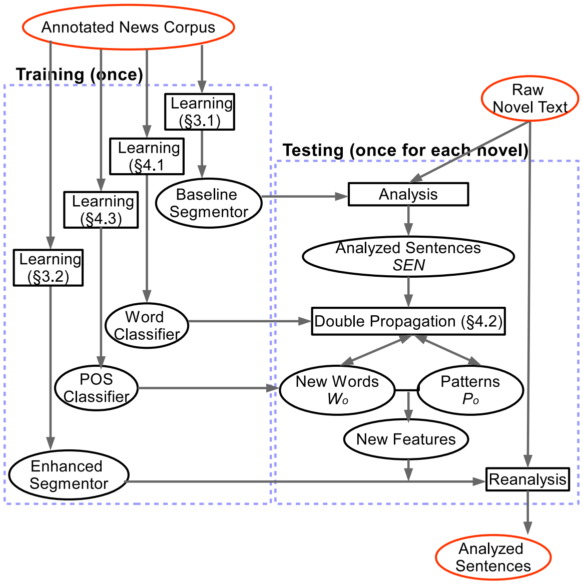

Figure 1: Flowchart of the proposed method.

1 Introduction

Word segmentation is a necessary first step for automatic syntactic analysis of Chinese text. Statistical Chinese word segmentation systems perform highly accurately on the news domain, thanks to large-scale manually-annotated training data. However, robust wide-coverage Chinese word segmentation is still an open problem, because the performance usually degrades significantly for other domains, such as science and literature. There has been a line of research on improving cross-domain word segmentation (Chang and Han 2010; Liu and Zhang 2012; Li and Xue 2014); for scientific texts such as patents, a domain dictionary can enhance the performance significantly.

The challenges to word segmentation for the literature domain, and novels in particular, are quite different from those for scientific texts. For scientific domains such as chemistry and computer science, OOV words mainly belong to domain terms, which form a relatively stable vocabulary. For novels, however, OOV words are usually named entities such as person, location and organization names, and other common noun entities that are specific to the setting of each individual novel. Apparently, novel-specific lexicons or annotated sentences can reduce a large proportion of segmentation errors for a specific novel (Zhang et al. 2014). However, the large number of novels on the web makes it unfeasibly expensive to annotate resources manually for each novel.

This paper addresses the domain adaptation problem for segmenting Chinese novels by automatically mining novel-specific noun entities using information extraction (IE) techniques. Our method does not use any target-domain annotation, such as domain dictionaries or small-scale annotated target-domain sentences. There has been work on semi-supervised word segmentation under the same setting, typically by incorporating target-domain statistics into source news-domain training (Suzuki and Isozaki 2008; Chang and Han 2010; Wang et al. 2011). However, such methods require the retraining of a statistical model for each

First type of context		**Second type of context**
Correctly segmented instances	**Simlar words from news corpus**	**Incorrectly segmented instances**
田灵儿 ... 跑 到 **田不易** 身旁 (Tianlinger ... went to Tianbuyi's side.)	他 ... 来 到 **王储** 身旁 (He ... came to the Crown Prince's side.)	眼看 着**田 不易** 晃悠悠 走 了 进来 (He saw thatTian Buyi came in wobbling.)
田不易 听 着 女儿 的 话 (Tianbuyi listened to his daughter's words.)	**高国珠** 听 了 很 生气 (Gaoguozhu was very angry after listening.)	**田 不易** 缓缓 点头 , (Tian Buyi nodded slowly.)
目光 离开 了 **田不易** , (His looks left Tianbuyi,)	越南队 输给 了 **泰国队** , (The Vietnamese team lost to Thailand.)	替 夫君 **田 不易** 教诲 这 帮 弟子 (She helped her husband Tian Buyi to teach these students.)

Table 1: Instances of the novel-specific word "田不易" occurring in different contexts.

target domain, using the news-domain annotated data. Given the large number of novels on the Internet, these methods can be unfeasibly costly in terms of training time. In contrast, the proposed model takes novel-specific nouns as plugin resources to a model trained on a large-scale annotated news corpus, and does not need retraining for each novel.

Our method is inspired by the observation that noun entities tend to occur frequently in common contexts. Formally, a context pattern *(l, x, r)* consists of a left context word *l*, a right context word *r* and a noun entity *x*, which we call the *target word*. For example, in the context pattern (*来到 (come to), x, 身旁 (side)*), *l*=来到 (come to), *r*=身旁 (side) and *x* is typically a person name; in the context pattern (*进入 (go), x, 内 (inside)*), *x* is typically a location or organization name. Context patterns can be classified into two types according to their occurrences in the source and target domains. In the first type of patterns, the context words *l* and *r* occur frequently in both the source-domain corpus and the target-domain corpus. In the second type of patterns, at least one of the context words is common in the target-domain but not in the source-domain.

It is likely for a segmentation algorithm to segment a new word correctly under contexts similar to the source-domain training data, but incorrectly under contexts rarely seen in the source domain. Several examples from our experimental data are given in Table 1. The left and right columns list instances of the target word "田不易 (Tianbuyi)" occurring under the two types of contexts, respectively. It is more frequently recognized correctly under the first type of context patterns, but incorrectly under the second type, being split into "田 (Tian; field)" and "不易 (Buyi; hard)".

For a human reader, the first type of context patterns can help recognize the meaning of a target word, which in turn helps the understanding of the second type of context patterns in a novel. Based on this observation, we develop a bootstrapping process that iteratively mines new words and context patterns. Given a novel, a double propagation process is used to detect new noun entities using the first type of context, which are in turn used to find the second type of context, before more noun entities and patterns are detected iteratively. We use the joint segmentation and part-of-speech (POS) tagging model of Zhang and Clark (2010) as the baseline segmentor, which gives the state-of-the-art accuracies on news data. The set of noun entities and patterns mined from each novel are incorporated into the baseline as new features to improve segmentation for the novel.

We perform experiments on segmenting five different novels. For each model, we manually annotate a set of sentences as the test data. Results show that the proposed method can significantly improve the accuracies of both word segmentation and POS tagging over a self-training baseline. In addition, unlike most previous semi-supervised segmentation methods, which improve recall on out-of-vocabulary (OOV) words at the expense of slightly decreased in-vocabulary (IV) recall, our method achieves an overall 30.2% error reduction on OOV recall, together with 8.7% error reduction on IV recall.

To facilitate future comparisons, we release our annotated datasets at http://people.sutd.edu.sg/\%7Eyue_zhang/publication.html.

2 Overview of the Proposed Method

The flowchart of the proposed method is shown in Figure 1. It consists of a training process and a testing process. Given an annotated news corpus, which we call *GenCorpus*, the training process is executed only once, resulting in a set of models that are used to analyze different novels without retraining. The testing process refers to the segmentation process of novels; it is carried out for each novel separately.

In the training process, a baseline segmentor (§3.1) and an enhanced segmentor (§3.2) that supports noun entity and pattern features are trained on *GenCorpus* separately. In addition, two log-linear classifiers (§4.1 and §4.3) are trained on *GenCorpus* for detecting new words and assigning POS tags to unknown words, respectively.

The segmentation process for a given novel consists of three steps. First, the novel is segmented and POS-tagged by the baseline segmentor (§3.1), resulting in a set of automatically segmented sentences *SEN*. Second, through a double propagation process, new word and context pattern mining is executed iteratively on *SEN* (§4.2), using the log-linear word (§4.1) and POS (§4.3) classifiers through a double propagation process. Finally, the newly-mined words together with their context patterns are converted into features, and given to the enhanced segmentor (§3.2) to produce the final segmentation for the novel.

3 Segmentation and POS-tagging

Joint segmentation and POS-tagging has received growing research attention due to improvements of lexical analysis over pipelined segmentors and POS taggers (Zhang and Clark 2008; Jiang et al. 2008; Kruengkrai et al. 2009; Zhang and Clark 2010). It makes better use of POS information and reduces segmentation error propagation, and is preferred when POS annotation is available.

3.1 The Baseline Segmentor

We apply the joint segmentor and POS-tagger of Zhang and Clark (2010)[1] as our baseline system. The segmentor processes a sentence from left to right, using a buffer to maintain partially-built outputs and a queue to hold the next incoming characters. In the *initial state*, the buffer is empty, and the queue contains the whole input sentence. Two transition actions are defined to consume input characters from the queue and construct output sentences on the buffer:

- APPEND, which removes the front character from the queue and appends it to the last word in the buffer;
- SEPARATE-*x*, which removes the front character from the queue and puts it as the start of a new word in the buffer, assigning the POS-tag *x* to the new word.

Given an input sentence, the system starts from the initial state, and repeatedly applies transition actions until all the characters on the queue are consumed and a full sentence is constructed on the buffer. Beam-search is applied to find a highest-scored sequence of transitions heuristically. The system scores search candidates using a linear model, which is trained using the averaged perceptron (Collins 2002) and early-update (Collins and Roark 2004).

The "Base" rows of Table 2 lists the the feature templates of our baseline segmentor, which are taken from Zhang and Clark (2010). *w*, *t* and *c* denote a word, a POS-tag and a character, respectively. The subscripts are based on the current character, which is the front character in the queue. w_{-1} represents the first word to the left of the current character, and t_{-2} represents the POS-tag on the second word to the left of the current character. *start(w)*, *end(w)* and *len(w)* indicate the first character, the last character and the length of word *w*, respectively. *cat(c)* represents the set of all possible POS-tags seen on the character *c*.

3.2 The Enhanced Segmentor

The enhanced segmentor is the baseline segmentor with additional feature templates that support information on new words (*W*) and patterns (*P*) in a target novel. The new features are shown in the "New" rows of Table 2. ISNOUN(*w*) indicates whether the word *w* is in *W*. ISPATTERN(w_{-2}, c_0) represents whether the word w_{-2} and the word starting with c_0 form a pattern in *P*, regardless whether w_{-1} is in *W* or not. ISTRIPLE(w_{-2}, w_{-1}, c_0) represents whether words w_{-2} and w_{-1}, and the word starting with c_0 form a pattern in *P*, with the noun w_{-1} being in *W*.

To train weights for *W* and *P* on the source-domain *GenCorpus*, the new feature templates are instantiated using a set of source-domain noun entities W_s and P_s. We construct P_s by extracting the set of source-domain patterns that occur at least 10 times in *GenCorpus*, and W_s by extracting the set of source-domain noun entities that occur at least under two context patterns in P_s. All the words in the general dictionary *the PKU Grammatical Dictionary* (Yu et al. 1998) are removed from W_s so that the remaining words simulate the distribution of new words in target novels.

Type	Feature
Base	$w_{-1}; w_{-1}w_{-2}; w_{-1},\ where\ len(w_{-1}) = 1;$
Base	$start(w_{-1})len(w_{-1}); end(w_{-1})len(w_{-1});$
Base	$c_{-1}c_0; begin(w_{-1})end(w_{-1}); end(w_{-2}w_{-1};$
Base	$start(w_{-1})c_0; end(w_{-2}end(w_{-1}); w_{-1}c_0;$
Base	$w_{-2}len(w_{-1}); len(w_{-2})w_{-1}; end(w_{-1})c_0);$
Base	$w_{-1}t_{-1}; t_{-1}t_0; t_{-2}t_{-1}t_0; w_{-1}t_0; t_{-2}w_{-1};$
Base	$w_{-1}t_{-1}end(w_{-2}); w_{-1}t_{-1}c_0; start(w_0)t_0;$
Base	$c_{-2}c_{-1}c_0t_{-1}\ (len(w_{-1}) = 1); t_{-1}start(w_{-1});$
Base	$ct_{-1}end(w_{-1})\ (c \in w_{-1}\ and\ c \neq end(w_{-1}));$
Base	$t_0c_0; c_0t_0start(w_0); c_0t_0c_{-1}t_{-1}; c_0t_0c_{-1};$
Base	$c_0t_0cat(start(w_0))\ (c \in w_{-1}\ and\ c \neq end(w_{-1}));$
New	ISNOUN$(w_{-2})t_{-2}len(w_{-2});$
New	ISNOUN$(w_{-1})t_{-1}len(w_{-1});$
New	ISNOUN$(c_0)t_0len(w_0);$
New	ISTRIPLE$(w_{-2}, w_{-1}, c)t_{-1}len(w_{-1});$
New	ISPATTERN$(w_{-2}, c)t_{-1}len(w_{-1});$

Table 2: Feature templates for the joint word segmentation and POS tagging system.

Type	Feature
Context	Both end and start with punctuations
Context	20⩽COUNT(*p*); 10⩽COUNT(*p*)<20
Context	2⩽COUNT(*p*)<10; COUNT(*p*)=1
Context	50⩽FREQ(*p*); 20⩽FREQ(*p*)<50
Context	5⩽FREQ(*p*)<20; FREQ(*p*)<5
Structure	PMI(C_1, $C_{2,n}$); PMI($C_{1,n-1}$, C_n)
Structure	PMI($C_{1,2}$,$C_{3,n}$); PMI($C_{1,n-2}$, $C_{n-1,n}$)

Table 3: Feature templates of the word classifier.

For the final segmentation of a novel, a set of novel-specific nouns (*W*) and the corresponding set of patterns (*P*) are extracted from the novel by using the double propagation algorithm in §4, and replace W_s and P_s for instantiating the new features in the enhanced segmentor.

4 Noun Entity and Pattern Mining

Our approach identifies new noun entities and context patterns using known and extracted noun entities and patterns iteratively, using noun entities to identify new patterns, and vice versa. Because of the two-way information passage between noun entities and patterns, this method is also called **double propagation** (Wu et al. 2009; Carlson et al. 2010; Qiu et al. 2011; Qiu and Zhang 2014).

4.1 Training a Word Classifier

For noun entity mining, a logistic regression classifier *WordClassifier* is used to score candidates, with features shown in Table 3. In the table, *p* denotes the context patterns that a target noun entity occurs in, and c_i denotes a substring of the target noun entity. For example, C_1 and $C_{(1,2)}$ denote the substrings consisting of the first character and the first

[1] from http://sourceforge.net/zpar version 0.6, using the *agenda* implementation of *chinese.postagger*.

two characters, respectively. COUNT(p) and FREQ(p) indicate the number of distinct patterns p and their total count, respectively, while PMI(C_1, $C_{(2,n)}$) is used to measure the point mutual information between the two substrings C_1 and $C_{(2,n)}$ in the target noun entity. The feature templates cover both context and word structure patterns.

We train *WordClassifier* using *GenCorpus*. All the nouns and their context patterns in the corpus can be taken as positive training examples for the word classifier. However, in order to simulate the test scenario when analyzing novels, we choose the top 30% most frequent person names, locations, organization names, and common nouns as the set of positive examples W_i. The frequencies of different types of nouns are counted separately. In addition, the context patterns of the chosen nouns are taken as the pattern set P_i.

For negative examples, we concatenate each noun in W_i and its right context word. However, if the resulting word also occurs in W_i, it is removed from the set of negative examples. This ensures that a word does not occur in both the positive and negative set.

4.2 Double Propagation

The pattern mining algorithm consists of three steps. Step 1 is an initialization process, where the tokens in the automatically-segmented novel *SEN* that have been tagged as nouns (including person names, locations, organization names, and common nouns) are extracted and put into the candidate new noun entity set W_c. The output noun entity set W_o is initialized to an empty set, and the output pattern set P_o is initialized to P_i. In Step 2, *WordClassifier* is used to classify the candidate new noun entities in W_c, taking words with probabilities above a threshold α as novel-specific nouns, and putting them into the set W_o. *WordClassifier* uses P_o for context features in the classification. In Step 3, the updated noun set W_o is used to expand P_o, with the context patterns of all words in W_o being added into P_o. If new patterns are detected, the process is repeated from Step 2. Otherwise, the algorithm finishes, returning W_o and P_o.

Algorithm 1 shows pseudocode of the double propagation algorithm. We take the novel 诛仙 *(Zhuxian)* as an example to illustrate the workflow of the algorithm. The original context pattern set P_i contains the context patterns in the left column of Table 1. Using the lines 2 to 4 of Algorithm 1, the target word "田不易 (Tianbuyi)" is put into the candidate new word set W_c. The target word "田不易 (Tianbuyi)" passes the confidence test in line 8 and is removed from W_c and put into W_o. As a result, the context patterns in the right column of Table 1 will be put into P_o. After repeating the execution of lines 7 to 15, more new words, including the person names "万剑一 (Wanjianyi)" and "申天斗 (Shentiandou)", and the locations "龙首峰 (Mount Longshou)" and "通天峰 (Mount Tongtian)", are extracted.

4.3 Tagging Unknown Words

Because our POS set differentiates common nouns (*n*), person names (*nr*), locations (*ns*) and organization names (*nt*), a new noun typically should have only one POS tag. However, some of the mined words are tagged with two or more POS under different contexts by the baseline segmentor, due to segmentation errors or inherent ambiguities.

```
Input  : auto-analyzed sentences $SEN$, context
         patterns $P_i$, annotated news corpus
         $GenCorpus$, noun entity classifier
         $WordClassifier$.
Output: New words $W_o$, context patterns $P_o$.
1  $W_o = \Phi$, $W_c = \Phi$; $P_o = P_i$;
2  for each $word \in SEN$ and $word \notin GenCorpus$ do
3      AddToSet($word$, $W_c$);
4  end
5  while True do
6      $count$ =0;
7      for each $word \in W_c$ do
8          if IsWord($word, P_o, WordClassifier$) then
9              RemoveFromSet($word, W_c$);
10             AddToSet($word, W_o$);
11             $pattern$ =GetContextPat($word$);
12             AddToSet($pattern, P_o$);
13             $count$ ++;
14         end
15     end
16     if $count$ =0 then
17         break;
18     end
19 end
```

Algorithm 1: The double propagation algorithm.

Type	Feature
Context	first context word on the left
Context	second context word on the left
Context	first context word on the right
Context	second context word on the right
Context	POS of first context word on the left
Context	POS of first context word on the right
Structure	the first character of the target word
Structure	the last character of the target word

Table 4: Feature templates of the POS classifier.

To disambiguate between the POS tags, we develop a log-linear regression classifier *POSClassifier* with the feature templates listed in Table 4, and try to attach a single POS tag to each new word. The POS classifier incorporates both global context features and internal structure features, and is trained on the source-domain news corpus. The training data is selected using the same way as selecting the positive examples for the *WordClassifier*.

POSClassifier is used to filter POS tags for the words in W_o. For a word with the most probable POS having a probability above a threshold β, only the most probable POS is given to the word. For the remaining words, all POS in W_o are kept. In the segmenting process, the word and POS information in W_o is used to instantiate features using the feature templates such as ISNOUN$(w_{-2})t_{-2}len(w_{-2})$ and ISNOUN$(c_0)t_0len(w_0)$ in Table 2.

Data Set	#Sents	#Words	OOV Rate
GenCorpus	42,132	1,052,119	—
诛仙*(ZX-dev)*	300	8,117	0.137
诛仙*(ZX-test)*	667	21,066	0.161
凡人修仙传*(FR)*	1,020	16,957	0.139
寻龙记*(XL)*	942	25,049	0.129
斗罗大陆*(DL)*	1,000	31,734	0.111
绝代双骄*(JD)*	632	17,759	0.109

Table 5: Corpus statistics.

5 Experiments

5.1 Experimental Setup

The People's Daily Corpus, which contains all the articles of People's Daily in January 1998, is used as the source-domain annotated corpus *GenCorpus*. The corpus consists of 42,000 sentences and 1.1M Chinese words. It contains both segmentation and POS annotation, and has been annotated according to the standard of Yu et al. (2003). Five contemporary novels, including 诛仙 *(Zhuxian, ZX)*, 凡人修仙传 *(Fanren Xiuxian Zhuan, FR)*, 寻龙记 *(XunLong Ji, XL)*, 斗罗大陆 *(Douluo Dalu, DL)* and 绝代双骄 *(Juedai Shuangjiao, JD)*, are chosen for the domain adaptation experiments. We select one chapter from the middle of each novel and annotate the sentences manually, so that the segmentation accuracies of the novels can be evaluated.

We take 300 annotated sentences from 诛仙 *(ZX)* as the development data (*ZX-dev*), which is used to determine the amount of sentences for a self-training baseline, and for the parameter tuning of the double propagation algorithm, the noun entity classifier, and the POS classifier. Detailed information of the training set (*GenCorpus*), the ZX development set and the five test sets is shown in Table 5.

We use the F1-score to evaluate both the segmentation accuracy and the overall segmentation and POS tagging accuracy. For the overall accuracy, a word is marked as correct only if both its segmentation and POS are correct. The recalls of IV and OOV words, which are defined as the percentages of IV and OOV words in the reference that are correctly segmented, respectively, are also measured.

5.2 Overall Results

Table 6 summarizes the best results on the development set, and the final results on the five test sets in terms of segmentation F1-score (the "Word" column), overall segmentation and POS-tagging F1-score (the "POS" column), and IV (the "IV" column) and OOV recalls (the "OOV" column) of segmentation. Improvements and error reduction of the enhanced segmentor over the baseline segmentor are listed in the "im" and "err" columns, respectively.

Compared with the baseline, the enhanced segmentor gives significant improvements on both the development set and the test sets. On average, it results in relative error reductions of 18.4% and 14.6% in segmentation and POS F1-scores, respectively.

The POS F1 improvement on the novel *JD* is relatively lower (+8.07% error reduction) than the other novels. Error analysis shows that the person names in this novel, including the names of the two main characters "小鱼儿 (Xiaoyuer; little fish)" and "花无缺 (Huawuque; flowers without flaw)", are highly similar to common nouns. In this novel, the POS-tagging errors by incorrectly tagging person names as common nouns occupy 51% of the OOV tagging errors. In contrast, the errors on other novels vary from 11% to 23%.

Previous work on semi-supervised domain-adaptation uses various sources of statistical information from raw target domain data, and improves the OOV recall with slightly decreased IV recall (Chang and Han 2010). This is mainly because target-domain statistical information such as character mutual information and χ^2 information helps the identification of target words, but can hurt the identification of source domain words if conflicts exist, which leads to the decrease of IV recall. In contrast, our proposed method can detect OOV words in high precision (§5.4) without affecting IV words. As a result, it achieves 30.4% error reduction in terms of OOV recall, together with 7.5% error reduction in terms of IV recall.

5.3 Comparison with Self-training

We compare the enhanced segmentor with the self-training method of Liu and Zhang (2012). Neither method uses target domain annotation, and self-training gives a simple baseline for unsupervised domain adaptation. We use the *ZX* development data to decide the best number of target-domain sentences from 4000 to 20000, and the best result is achieved using 8000 sentences. As a result, 8000 target-domain sentences are used for each novel in the final self-training tests.

Experimental results on the development and test sets are listed in Table 7. The results indicate that self-training can achieve a relative error reduction of about 6% on both segmentation and POS F1. This result is similar to the finding of Liu and Zhang (2012). In contrast, our proposed method shows significant advantage in terms of error reduction (Word 18.4% vs 5.8%, POS 14.6% vs 4.13%, IV 7.50% vs 4.75% and OOV 30.4% vs 6.6%). The self-training method gives a relatively better impact of IV, while our proposed method is relatively more effective in improving OOV. This is mainly because self-training tends to learn from the correctness of the baseline system, and hence improve what has been segmented correctly. In contrast, our double propagation method can mine words that have been rarely segmented correctly by the baseline, using context information. In addition, it takes six hours to segment each novel using self-training due to retraining of the segmentation model, but only about ten minutes using our method. The speed advantage results from the fact that our method does not need the retraining of segmentation models, and makes our method particularly useful compared with self-training and other semi-supervised methods that require retraining of a statistical model for a given novel.

5.4 The Effect of Noun Entity Mining

The main benefit of our segmentation method comes from noun entity mining. We evaluate the quality of mined noun entities in terms of precision and recall. The precision (*Noun P*) is evaluated manually on the 100 most frequent entities

	Word (%)				**POS (%)**				**IV (%)**				**OOV(%)**			
Nov	**base**	**en**	**im**	**err**	**base**	**en**	**im**	**err**	**base**	**en**	**im**	**err**	**base**	**en**	**im**	**err**
ZX-dev	89.5	92.1	+2.6	24.7	82.8	86.3	+3.5	20.3	92.6	93.8	+1.2	16.2	67.8	80.3	+12.5	38.8
ZX-test	88.3	90.2	+1.9	16.2	79.9	82.9	+3.0	14.9	91.3	92.0	+0.7	8.04	69.3	78.1	+8.8	28.6
FR	87.6	90.1	+2.5	20.1	81.3	84.5	+3.2	17.1	90.4	91.5	+1.1	11.4	68.3	79.4	+11.1	35.0
XL	87.5	89.5	+2.0	16.0	80.9	84.2	+3.3	17.2	90.2	91.1	+0.9	9.18	65.5	74.7	+9.2	26.6
JD	88.7	90.9	+2.2	19.4	83.9	85.2	+1.3	8.07	92.2	92.7	+0.5	6.41	59.4	71.1	+11.7	28.8
DL	92.6	94.1	+1.5	20.2	87.5	89.5	+2.0	16.0	95.9	96.0	+0.1	2.43	72.1	81.3	+9.2	32.9
avg-test			**+2.0**	**18.4**			**+2.6**	**14.6**			**+0.66**	**7.5**			**+10.0**	**30.4**

Table 6: Main experimental results using the model trained on *GenCorpus*. (base: the base segmentor. en: the enhanced segmentor. im: improvement of the enhanced segmentor over the baseline segmentor. err: error reduction.)

	Word (%)		**POS (%)**		**OOV (%)**	
Nov	**self**	**im**	**self**	**im**	**self**	**im**
ZX-Dev	90.2	+0.7	83.6	+0.8	71.4	+3.6
ZX-Test	89.0	+0.7	80.8	+0.9	71.9	+2.6
FR	89.0	+1.4	82.6	+1.3	72.8	+4.5
XL	87.7	+0.2	80.9	+0.0	66.1	+0.6
JD	89.1	+0.4	84.5	+0.6	56.4	+0.2
DL	93.1	+0.5	88.1	+0.7	74.1	+2.3
avg-test		**+0.64**		**+0.7**		**+2.04**

Table 7: Comparison with self-training. (self: self-training. im: improvement of self-training over the baseline.)

	Noun P (%)		**Noun R (%)**		**Seg R (%)**	
Nov	**type**	**token**	**type**	**token**	**base**	**en**
ZX	95	94	49.4	76.6	74.7	87.1
FR	94	92	60.8	82.6	76.9	91.4
XL	97	96	40.9	71.9	77.0	91.3
JD	96	95	24.4	53.2	64.2	96.7
DL	94	94	55.1	84.5	80.8	93.3
avg					74.7	**92.0**

Table 8: Evaluation of noun entity mining.

mined from each novel, while the recall (*Noun R*) is evaluated on the test sets automatically. The columns "Noun P" and "Noun R" of Table 8 show the results. Our method gives high precisions (about 95%), with recalls from 53% to 84.5% on the novels.

The effect of mined noun entities on the segmentation accuracies are shown in the "Seg R" column of Table 8. We measure the recalls in the segmentation of noun entities (*Seg R*), which occupy 60% to 76% of the total OOV words of the five novels. The enhanced segmentor can segment these mined noun entities with a relatively high recall (92%), giving an average error reduction about 68.2%, which contributes to the overall OOV recall improvement.

6 Related Work

Given manually annotated news data, there are three main approaches to improve segmentation accuracies on a target domain. The first is unsupervised domain adaptation, which uses no annotated sentences or lexicons on the target domain. The simplest method is self-training (Chang and Han 2010; Liu and Zhang 2012) and co-training (Zhang et al. 2013), while more complex methods include those based on feature augmentation (Wang et al. 2011) and training regularization (Suzuki and Isozaki 2008). In all the methods above, various sources of information from raw or automatically-segmented target domain data can be used, including mutual information (Sun, Shen, and Tsou 1998; Zhao and Kit 2008; Sun and Xu 2011), χ^2 information (Chang and Han 2010), branching entropy (Jin and Tanaka-Ishii 2006) and character clusters (Liang 2005). For each target domain, all the methods above require the retraining of a model using the source-domain corpus, which can be overly expensive if each novel is taken as its own domain. In contrast to the methods above, our approach does not require retraining of segmentation models, and hence is more suitable for the literature domain.

The second and third approaches are type- and token-supervised training, respectively (Li and Sun 2009; Jiang et al. 2013; Li and Xue 2014; Zhang et al. 2014). Both methods require manual annotation on the target domain, with the former requiring annotated target-domain lexicons and the latter requiring annotated target-domain sentences. Zhang et al. (2014) show that both types of annotations can lead to the same accuracy improvement given the same efforts, while a mixture of both types of annotations can lead to increased effect. Target domain annotation can be the most useful method for domains with a relatively stable vocabulary, such as scientific texts. However, for the literature domain, vocabularies are highly flexible, and it is unfeasibly expensive to annotate manually for the domain. We take an approach based on information extraction, automatically mining domain vocabularies for each novel. To our knowledge, this is the first to study segmentation by taking literature as a whole domain.

7 Conclusions and Future Work

We studied word segmentation for Chinese novels, presenting a double propagation method for mining noun entities and context patterns, and using them as features to enhance a cross-domain segmentation model. Our method is special in that no retraining of a segmentation model is required for each novel, and therefore it is practically more useful given the large number of novels on the Internet. Experimental results show that our approach achieves substantial improvement over a state-of-the-art baseline system.

By analyzing the contents of novels automatically, our work is a basic step in using AI algorithms to aid social science research. For example, based on the analysis of syntax, major events can be extracted from the novel, the relationship between characters can be automatically detected, and sentiment of the author can be analyzed. These provide empirical support for literary criticism. In addition, applications such as novel recommendation systems can be developed based on these results.

For future work, we plan to study the effects of context patterns beyond the immediate neighboring words, and external knowledge such as common surnames.

Acknowledgments

We gratefully acknowledge the invaluable assistance of Yijia Liu and Hui Zhao. We also thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Singapore Ministry of Education (MOE) AcRF Tier 2 grant T2MOE201301, the startup grant SRG ISTD 2012 038 from Singapore University of Technology and Design, the National Natural Science Foundation of China (No. 61103089), Scientific Research Foundation of Shandong Province Outstanding Young Scientist Award (No. BS2013DX020) and Humanities and Social Science Projects of Ludong University (No. WY2013003).

References

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka Jr, E. R.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Chang, B., and Han, D. 2010. Enhancing domain portability of Chinese segmentation model using chi-square statistics and bootstrapping. In *Proceedings of EMNLP*, 789–798.

Collins, M., and Roark, B. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*, 111.

Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of ACL*, 1–8.

Jiang, W.; Huang, L.; Liu, Q.; and Lü, Y. 2008. A cascaded linear model for joint Chinese word segmentation and part-of-speech tagging. In *In Proceedings of ACL*. Citeseer.

Jiang, W.; Sun, M.; Lü, Y.; Yang, Y.; and Liu, Q. 2013. Discriminative learning with natural annotations: Word segmentation as a case study.

Jin, Z., and Tanaka-Ishii, K. 2006. Unsupervised segmentation of Chinese text by use of branching entropy. In *Proceedings of COLING/ACL*, 428–435.

Kruengkrai, C.; Uchimoto, K.; Kazama, J.; Wang, Y.; Torisawa, K.; and Isahara, H. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and pos tagging. In *Proceedings of ACL and AFNLP*, 513–521.

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

Li, S., and Xue, N. 2014. Effective document-level features for Chinese patent word segmentation.

Liang, P. 2005. *Semi-supervised learning for natural language*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Liu, Y., and Zhang, Y. 2012. Unsupervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of COLING (Posters)*, 745–754. Citeseer.

Qiu, L., and Zhang, Y. 2014. Zore: A syntax-based system for chinese open relation extraction. In *Proceedings of EMNLP*, 1870–1880. Doha, Qatar: ACL.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics* 37(1):9–27.

Sun, W., and Xu, J. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of EMNLP*, 970–979.

Sun, M.; Shen, D.; and Tsou, B. K. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of ACL and COLING*, 1265–1271.

Suzuki, J., and Isozaki, H. 2008. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proceedings of ACL*, 665–673. Citeseer.

Wang, Y.; Jun'ichi Kazama, Y. T.; Tsuruoka, Y.; Chen, W.; Zhang, Y.; and Torisawa, K. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of IJCNLP*, 309–317.

Wu, D.; Lee, W. S.; Ye, N.; and Chieu, H. L. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of EMNLP*, 1523–1532.

Yu, S.; Zhu, X.; Wang, H.; and Zhang, Y. 1998. The grammatical knowledge-base of contemporary Chinese—a complete specification.

Yu, S.; Duan, H.; Zhu, X.; Swen, B.; and Chang, B. 2003. Specification for corpus processing at peking university: Word segmentation, POS tagging and phonetic notation. *Journal of Chinese Language and Computing* 13(2):121–158.

Zhang, Y., and Clark, S. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL-08: HLT*, 888–896. Columbus, Ohio: ACL.

Zhang, Y., and Clark, S. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In *Proceedings of EMNLP*, 843–852.

Zhang, L.; Wang, H.; Sun, X.; and Mansur, M. 2013. Exploring representations from unlabeled data with co-training for Chinese word segmentation. In *Proceedings of EMNLP*, 311–321.

Zhang, M.; Zhang, Y.; Che, W.; and Liu, T. 2014. Type-supervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of EACL*, 588–597.

Zhao, H., and Kit, C. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Proceedings of IJCNLP*, 106–111.