Enhancing Chinese Word Segmentation Using Unlabeled Data

Weiwei Sun[†‡] and **Jia Xu**[‡]
[†]Department of Computational Linguistics, Saarland University
[‡]German Research Center for Artificial Intelligence (DFKI)
D-66123, Saarbrücken, Germany
wsun@coli.uni-saarland.de, Jia.Xu@dfki.de

Abstract

This paper investigates improving supervised word segmentation accuracy with unlabeled data. Both large-scale in-domain data and small-scale document text are considered. We present a unified solution to include features derived from unlabeled data to a discriminative learning model. For the large-scale data, we derive string statistics from Gigaword to assist a character-based segmenter. In addition, we introduce the idea about transductive, document-level segmentation, which is designed to improve the system recall for out-of-vocabulary (OOV) words which appear more than once inside a document. Novel features[1] result in relative error reductions of 13.8% and 15.4% in terms of F-score and the recall of OOV words respectively.

1 Introduction

Chinese sentences are written in continuous sequence of characters without explicit delimiters such as space characters. To find the basic language units, i.e. words, segmentation is a necessary initial step for Chinese language processing. Previous research shows that word segmentation models trained on labeled data are reasonably accurate. In this paper, we investigate improving supervised word segmentation with unlabeled data.

We distinguish three types of unlabeled data, namely large-scale in-domain data, out-of-domain data and small-scale document text. Both large-scale in-domain and out-of-domain data is popular for enhancing NLP tasks. Learning from these two types of unlabeled data normally involves semi-supervised learning. The difference between them is that out-of-domain data is usually used for domain adaptation. For a number of NLP tasks, there are relatively large amounts of labeled training data. In this situation, supervised learning can provide competitive results, and it is difficult to improve them any further by using extra unlabeled data. Chinese word segmentation is one of this kind of tasks, since several large-scale manually annotated corpora are publicly available. In this work, we first exploit unlabeled in-domain data to improve strong supervised models. We leave domain adaptation for our future work.

We introduce the third type of unlabeled data with a *transductive learning*, *document-level* view. Many applications of word segmentation involve processing a whole document, such as information retrieval. In this situation, the text of the current document can provide additional useful information to segment a sentence. Take the word "氨纶丝/elastane" for example[2]. As a translated terminology word, it lacks compositionality. Moreover, this word appears rarely in general texts. As a result, if it does not appear in the training data, it is very hard for statistical models to recognize this word. Nevertheless, when we deal with an article discussing an elastane company, this word may appear more than once in this article, and the document information can help recognize this word. This idea is closely related to transductive learning in the sense that the segmentation model knows something about the problem it

[1]You can download our derived features at http://www.coli.uni-saarland.de/~wsun/semi-cws-feats-emnlp11.tgz.

[2]This example is from an article indexed as chtb_0041 in the Penn Chinese Treebank corpus.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 970–979,
Edinburgh, Scotland, UK, July 27–31, 2011.

is going to resolve. In this work, we are also concerned with enhancing word segmentation with the document information.

We present a unified "feature engineering" approach for learning segmentation models from both labeled and unlabeled data. Our method is a simple two-stage process. First, we use unannotated corpus to extract string and document information, and then we use these information to construct new statistics-based and document-based feature mapping for a discriminative word segmenter. We are relying on the ability of discriminative learning method to identify and explore informative features, which play central role to boost the segmentation performance. This simple solution has been shown effective for named entity recognition (Miller et al., 2004) and dependency parsing (Koo et al., 2008). In their implementations, word clusters derived from unlabeled data are imported as features to discriminative learning approaches.

To demonstrate the effectiveness of our approach, we conduct experiments on the Penn Chinese Treebank (CTB) data. CTB is a collection of documents which are separately annotated. This annotation style allows us to evaluate our transductive segmentation method. Our experiments show that both statistics-based and document-based features are effective in the word segmentation application. In general, the use of unlabeled data can be motivated by two concerns: First, given a fixed amount of labeled data, we might wish to leverage unlabeled data to improve the performance of a supervised model. Second, given a fixed target performance level, we might wish to use unlabeled data to reduce the amount of annotated data necessary to reach this target. We show that our approach yields improvements for fixed data sets, even when large-scale labeled data is available. The new features result in relative error reductions of 13.8% and 15.4% in terms of the balanced F-score and the recall of out-of-vocabulary (OOV) words respectively. By conducting experiments on data sets of varying sizes, we demonstrate that for fixed levels of performance, the new features derived from unlabeled data can significantly reduce the need of labeled data.

The remaining part of the paper is organized as follows. Section 2 describes the details of our system, especially the design of the derived features.

B	Current character is the start of a word consisting of more than one character.
E	Current character is the end of a word consisting of more than one character.
I	Current character is a middle of a word consisting of more than two characters.
S	Current character is a word consisting of only one character.

Table 1: The start/end representation.

Section 3 presents experimental results and empirical analysis. Section 4 reviews the related work. Section 5 concludes the paper.

2 Method

2.1 Discriminative Character-based Word Segmentation

The Character-based approach is a dominant word segmentation solution for Chinese text processing. This approach treats word segmentation as a sequence tagging problem, assigning labels to the characters indicating whether a character locates at the beginning of, inside or at the end of a word. This character-by-character method was first proposed by (Xue, 2003), and a number of discriminative sequential learning algorithms have been exploited, including structured perceptron (Jiang et al., 2009), the Passive-Aggressive algorithm (Sun, 2010), conditional random fields (CRFs) (Tseng et al., 2005), and latent variable CRFs (Sun et al., 2009). In this work, we use the *Start/End* representation to express the position information of every character. Table 2.1 shows the meaning of each character label. For example, the target label representation of the book title "赵紫阳总理的秘密日记/The Secret Journal of Premier Zhao Ziyang" is as follows.

赵	紫	阳	总	理	的	秘	密	日	记
B	I	E	B	E	S	B	E	B	E

Key to our approach is to allow informative features derived from unlabeled data to assist the segmenter. In our experiments, we employed three different feature sets: a baseline feature set which draws upon "normal" information from labeled training data, a statistics-based feature set that uses statistical information derived from a large-scale in-domain corpus, and a document-based feature set

that uses information encoded in the surrounding text.

2.2 Baseline Features

In this work, to train a good traditional supervised segmenter, our baseline feature templates includes the ones described in (Sun et al., 2009; Sun, 2010). These features are divided into two types: character features and word type features. Note that the word type features are indicator functions that fire when the local character sequence matches a word uni-gram or bi-gram. Dictionary containing word uni-grams and bi-grams is collected from the training data. To conveniently illustrate, we denote a candidate character token c_i with a context $...c_{i-1}c_ic_{i+1}...$. We use $c_{[s:e]}$ to express a string that starts at the position s and ends at the position e. For example, $c_{[i:i+1]}$ expresses a character bi-gram c_ic_{i+1}. The character features are listed below.

- *Character uni-grams*: c_s $(i-3<s<i+3)$
- *Character bi-grams*: c_sc_{s+1} $(i-3<s<i+3)$
- Whether c_s and c_{s+1} are *identical*, for $i-2<s<i+2$.
- Whether c_s and c_{s+2} are *identical*, for $i-4<s<i+2$.

The word type features are listed as follows.

- The *identity* of the string $c_{[s:i]}$ $(i-6<s<i)$, if it matches a word from the list of uni-gram words;
- The *identity* of the string $c_{[i:e]}$ $(i<e<i+6)$, if it matches a word; multiple features could be generated.
- The *identity* of the bi-gram $c_{[s:i-1]}c_{[i:e]}$ $(i-6<s,e<i+6)$, if it matches a word bi-gram from the list of uni-gram words.
- The *identity* of the bi-gram $c_{[s:i]}c_{[i+1:e]}$ $(i-6<s,e<i+6)$, if it matches a word bi-gram; multiple features could be generated.

Idiom In linguistics, idioms are usually presumed to be figures of speech contradicting the principle of compositionality. As a result, it is very hard to recognize out-of-vocabulary idioms for word segmentation. Nonetheless, the lexicon of idioms can be taken as a close set, which helps resolve the problem well. In our previous work (Sun, 2011), we collect 12992 idioms from several free online Chinese dictionaries. This linguistic resource is publicly available[3]. In this paper, we use this idiom dictionary to derive the following feature.

- Does c_i locate at the beginning of, inside or at the end of an idiom? If the string $c_{[s:i]}$ $(s<i)$ matches an item from the idiom lexicon, the feature template receives a string value "E-IDIOM". Similarly, we can define when this feature ought to be set to "B-IDIOM" or "I-IDIOM". Note that all idioms are larger than one character, so there is no "S-IDIOM" feature here.

2.3 Statistics-based Features

In order to distill information from unlabeled data, we borrow ideas from some previous research on unsupervised word segmentation. The statistical information acquired from a relatively large amount of unlabeled data are designed as features correlated with the position where a character locates in a word token. These features are based on three widely used criteria.

2.3.1 Mutual Information

Empirical mutual information is widely used in NLP. Informally, mutual information compares the probability of observing x and y together with the probabilities of observing x and y independently. If there is a genuine association between x and y, the $I(x,y)=\log\frac{p(x,y)}{p(x)p(y)}$ should be greater than 0.

Some previous work claimed that the larger the mutual information between two consecutive strings, the higher the possibility of the two strings being combined together. We adopt this idea in our character-based segmentation model. The empirical mutual information between two character bi-grams is computed by counting how often they appear in the large-scale unlabeled corpus. Given a

[3] http://www.coli.uni-saarland.de/~wsun/idiom.txt.

Chinese character string $c_{[i-2:i+1]}$, the mutual information between substrings $c_{[i-2:i-1]}$ and $c_{[i:i+1]}$ is computed as:

$$MI(c_{[i-2:i-1]}, c_{[i:i+1]}) = \log \frac{p(c_{[i-2:i+1]})}{p(c_{[i-2:i-1]})p(c_{[i:i+1]})}$$

For each character c_i, we incorporate the MI of the character bi-grams into our model. They include,

- $MI(c_{[i-2:i-1]}, c_{[i:i+1]})$,
- $MI(c_{[i-1:i]}, c_{[i+1:i+2]})$.

2.3.2 Accessor Variety

When a string appears under different linguistic environments, it may carry a meaning. This principle is introduced as the *accessor variety* criterion for identifying meaningful Chinese words in (Feng et al., 2004). This criterion evaluates how independently a string is used, and thus how likely it is that the string can be a word. Given a string s, which consists of l ($l \geq 2$) characters, we define the *left accessor variety* of $L^l_{av}(s)$ as the number of distinct characters that precede s in a corpus. Similarly, the *right accessor variety* $R^l_{av}(s)$ is defined as the number of distinct characters that succeed s.

We first extract all strings whose length are between 2 and 4 from the unlabeled data, and calculate their accessor variety values. For each character c_i, we then incorporate the following information into our model,

- Accessor variety of strings with length 4: $L^4_{av}(c_{[i:i+3]})$, $L^4_{av}(c_{[i+1:i+4]})$, $R^4_{av}(c_{[i-3:i]})$, $R^4_{av}(c_{[i-4:i-1]})$;
- Accessor variety of strings with length 3: $L^3_{av}(c_{[i:i+2]})$, $L^3_{av}(c_{[i+1:i+3]})$, $R^3_{av}(c_{[i-2:i]})$, $R^3_{av}(c_{[i-3:i-1]})$;
- Accessor variety of strings with length 2: $L^2_{av}(c_{[i:i+1]})$, $L^2_{av}(c_{[i+1:i+2]})$, $R^2_{av}(c_{[i-1:i]})$, $R^2_{av}(c_{[i-2:i-1]})$.

2.3.3 Punctuation as Anchor Words

Punctuation marks are symbols that indicate the structure and organization of written language, as well as intonation and pauses to be observed when reading aloud. Punctuation marks can be taken as perfect word delimiters, and can be used as anchor words to harvest lexical knowledge. The preceding and succeeding strings of punctuations carry additional wordbreak information, since punctuations should be segmented as a word. Note that such information is biased because not all words can appear before or after punctuations. For example, punctuations can not be followed by particles, such as "了", "着" and "过" which are indicators of aspects. Nevertheless, our experiments will show this kind of information is still useful for word segmentation.

When a string appears many times preceding or succeeding punctuations, there tends to be wordbreaks succeeding or preceding that string. To utilize the wordbreak information provided by punctuations, we extract all strings with length $l (2 \leq l \leq 4)$ which precede or succeed punctuations in the unlabeled data. We define the *left punctuation variety* of $L^l_{pv}(s)$ as the number of times a punctuation precedes s in a corpus. Similarly, the *right punctuation variety* $R^l_{pv}(s)$ is defined as the number of how many times a punctuation succeeds s. These two variables evaluate how likely a string can be separated at its start or end positions.

We first gather all strings surrounding punctuations in the unlabeled data, and calculate their punctuation variety values. The length of each string is also restricted between 2 and 4. For each character c_i, we import the following information into our model,

- Punctuation variety of strings with length 4: $L^4_{pv}(c_{[i:i+3]})$, $R^4_{pv}(c_{[i-3:i]})$;
- Punctuation variety of strings with length 3: $L^3_{pv}(c_{[i:i+2]})$, $R^3_{pv}(c_{[i-2:i]})$;
- Punctuation variety of strings with length 2: $L^2_{pv}(c_{[i:i+1]})$, $R^2_{pv}(c_{[i-1:i]})$.

Punctuations can be viewed as *mark-up's* of Chinese text. Our motivation to use the punctuation information to assist a word segmenter is similiar to (Spitkovsky et al., 2010) in a way to explore "artificial" word (or phrase) break symbols. In their work, four common HTML tags are successfully used as raw phrase bracketings to improve unsupervised dependency parsing.

2.3.4 Binary or Numeric Features

The derived information introduced above is all expressed as real values. The natural way to incorporate these statistics into a discriminative learning model is to directly use them as numeric features. However, our experiments show that this simple choice does not work well. The reason is that these statistics actually behave non-linearly to predict character labels. For each type of statistics, one weight alone cannot capture the relation between its value and the possibility that a string forms a word. Instead, we represent these statistics as discrete features.

For the mutual information, this is done by rounding down decimal number. The integer part of each MI value is used as a string feature. For the accessor variety and punctuation variety information, since their values are integer, we can directly use them as string features. The accessor variety and punctuation variety could be very large, so we set thresholds to cut off large values to deal with the data sparse problem. Specially, if an accessor variety value is greater than 50, it is incorporated as a feature "> 50"; if the value is greater than 30 but not greater than 50, it is incorporated as a feature "$30 - 50$"; else the value is individually incorporated as a string feature. For example, if the left accessory variety of a character bi-gram $c_{[i:i+1]}$ is 29, the binary feature "$L^2_{av}(c_{[i:i+1]})$=29" will be set to 1, while other related binary features such as "$L^2_{av}(c_{[i:i+1]}) = 15$" or "$L^2_{av}(c_{[i:i+1]}) > 50$" will be set to 0. Similarly, we can discretize the punctuation variety features. However, we only set one threshold, 30, for this value. These thresholds can be tuned by using held-out data.

2.4 Document-based Features

It is meaningless to derive statistics of a document and use it for word segmentation, since most documents are relatively short, and values are statistically unreliable. Our experiments confirm this idea. Instead, we propose the following binary features which are based on the *string count* in the given document that is simply the number of times a given string appears in that document. For each character c_i, our document-based features include,

- Whether the string count of $c_{[s:i]}$ is equal to that of $c_{[s:i+1]}$ $(i - 3 \leq s \leq i)$. Multiple features are generated for different string length.

- Whether the string count of $c_{[i:e]}$ is equal to that of $c_{[i-1:e]}$ $(i \leq e \leq i + 3)$. Multiple features are generated for different string length.

The intuition is as follows. The string counts of $c_{[s:i]}$ and $c_{[s:i+1]}$ being equal means that when $c_{[s:i]}$ appears, it appears inside $c_{[s:i+1]}$. In this case, $c_{[s:i]}$ is not independently used in this document, and this feature suggests the segmenter not assign a "S" or "E" label to the character c_i. Similarly, the string counts of $c_{[i:e]}$ and $c_{[i-1:e]}$ being equal means $c_{[i:e]}$ is not independently used in this document, and this feature suggests segmenter not assign a "S" or "B" label to c_i. We do not directly use the string counts to prevent a bias towards longer documents.

3 Experiments

3.1 Setting

The SIGHAN Bakeoffs provide several large-scale labeled data for the research on Chinese word segmentation. Although these data sets are labeled on continuous run texts, they do not contain the document boundary information. CTB is a segmented, part-of-speech tagged, and fully bracketed corpus in the constituency formalism. It is also an popular data set to evaluate word segmentation methods, such as (Jiang et al., 2009; Sun, 2011). CTB is a collection of documents which are separately annotated. This annotation style allows us to calculate the so-called document-based features and to further evaluate our approach. In this paper, we use CTB 6.0 as our main corpus and define the training, development and test sets according to the Chinese sub-task of the CoNLL 2009 shared task[4]. Table 2 shows the statistics of our experimental settings.

Data set	# of sent.	# of words	# of char.
Training	22277	609060	1004266
Devel.	1763	49646	83710
Test	2557	73152	121008

Table 2: Training, development and test data on CTB 6.0

[4]We would like to thank Prof. Nianwen Xue for the help with the division of the data

Chinese Gigaword is a comprehensive archive of newswire text data that has been acquired over several years by the Linguistic Data Consortium (LDC). The large-scale unlabeled data we use in our experiments comes from the Chinese Gigaword (LDC2005T14). We choose the Mandarin news text, i.e. Xinhua newswire. This data covers all news published by Xinhua News Agency (the largest news agency in China) from 1991 to 2004, which contains over 473 million characters.

F-score is used as the accuracy measure. Define precision p as the percentage of words in the decoder output that are segmented correctly, and recall r as the percentage of gold standard output words that are correctly segmented by the decoder. The (balanced) F-score is $2pr/(p+r)$. We also report the recall of OOV words. Note that, all idioms in our extra idiom lexicon are added into the in-vocabulary word list.

CRFsuite (Okazaki, 2007) is an implementation of Conditional Random Fields (CRFs) (Lafferty et al., 2001) for labeling sequential data. It is a speed-oriented implementation, which is written in pure C. In our experiments, we use this toolkit to learn global linear models for segmentation. We use the stochastic gradient descent algorithm to resolve the optimization problem, and set default values for other learning parameters.

3.2 Main Results

Table 3 summarizes the segmentation results on the development data with different configurations, representing a few choices between baseline, statistics-based and document-based feature sets. In this table, the symbol "+" means features of current configuration contains both the baseline features and new features for semi-supervised or transductive learning. From this table, we can clearly see the impact of features derived from the large-scale unlabeled data and the current document. Comparison between the performance of the baseline and "+MI" shows that the widely used mutual information is not helpful. Both good segmentation techniques and valuable labeled corpora have been developed, and pure supervised systems can provide strong performance. It is not a trial to design new features to enhance supervised models.

There are significant increases when accessor variety features and punctuation variety features are separately added. Extending the length of neighboring string helps a little from 2 to 3. Although the OOV recall increases when the length is extended from 3 to 4, there is no improvement of the overall balanced F-score. The line "+MI+AV(2,3,4)+PU(2,3,4)" shows the performance when all statistics-based features are added. The combination of the "AV" and "PU" features gives further helps. This system can be seen as a pure semi-supervised system. The line "+DOC" is the result when document-based features are added. In spite of its simplicity, the document-based features can help the task. However, when we combine statistics-based features with document-based features, we cannot get further improvement in terms of F-score.

Devel.	P	R	$F_{\beta=1}$	R_{oov}
Baseline	95.41	95.52	95.46	77.68
+MI	95.50	95.48	95.49	77.98
+AV(2)	95.85	96.04	95.94	79.31
+AV(2,3)	95.95	96.19	96.07	80.61
+AV(2,3,4)	96.14	95.99	96.07	81.83
+PU(2)	95.86	96.07	95.97	79.70
+PU(2,3)	95.98	96.25	96.11	80.42
+PU(2,3,4)	96.00	96.19	96.10	80.53
+MI+AV(2,3,4)+PU(2,3,4)				
	96.17	96.22	96.19	80.42
+DOC	95.69	95.64	95.66	79.89
+MI+AV(2,3,4)+PU(2,3,4)+DOC				
	96.21	96.23	96.22	81.75

Table 3: Segmentation performance with different feature sets on the development data. Abbreviations: MI=mutual information; AV=accessor variety; PU=punctuation variety; DOC=document features. The numbers in each bracket pair are the lengths of strings. For example, PU(2,3) means punctuation variety features of character bi-grams and tri-grams are added.

Table 4 shows the segmentation performance on the test data set. The final results of our system are achieved with the "+MI+AV(2,3,4)+PU(2,3,4)+DOC" feature configuration. The new features result in relative error reductions of 13.8% and 15.4% in terms of the balanced F-score and the recall of OOV words respectively.

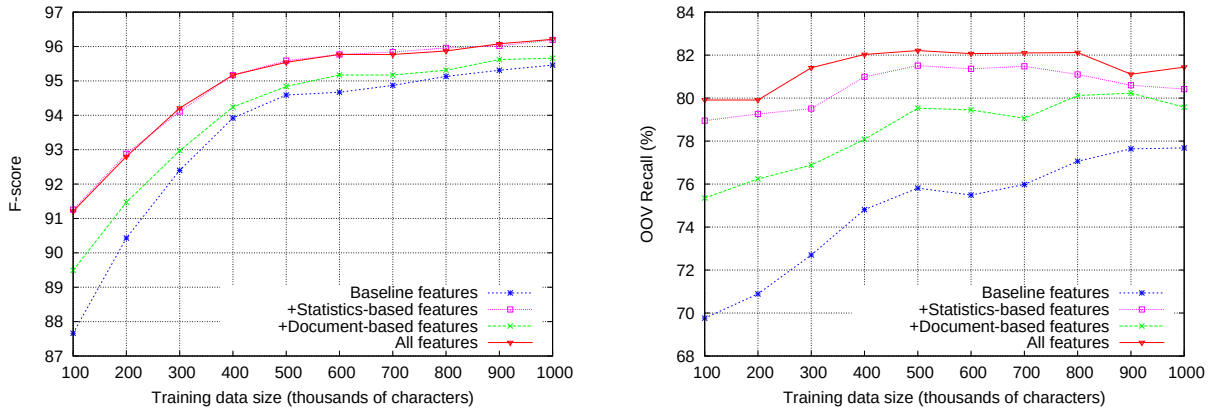

Figure 1: The learning curves of different models.

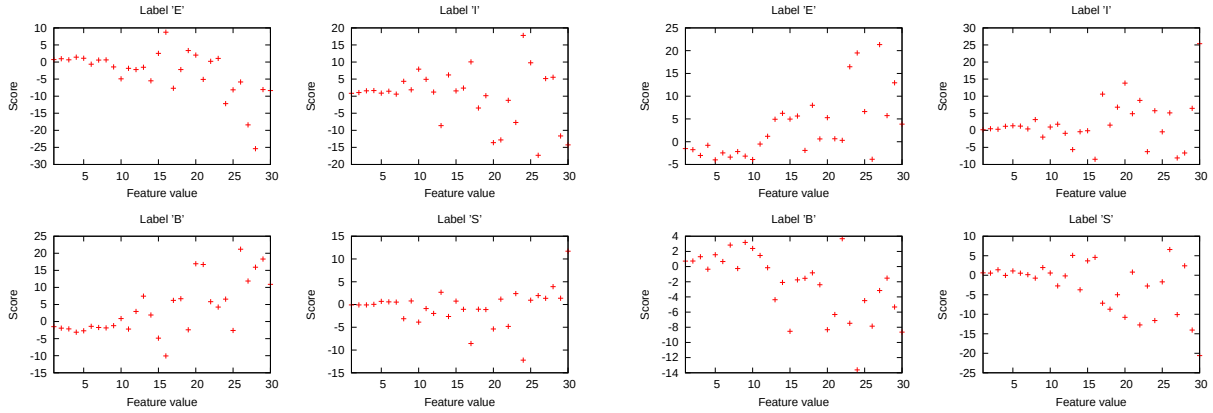

Figure 2: Scatter plot of feature score against feature value. The left side shows is $L^2_{pv}(c_{[i:i+1]}$ feature while the right side is the $R^2_{pv}(c_{[i:i+1]}$ feature.

Test	P	R	$F_{\beta=1}$	R_{oov}
Baseline	95.21	94.90	95.06	75.52
Final	95.86	95.62	95.74	79.28

Table 4: Segmentation performance on the test data.

3.3 Learning Curves

We performed additional experiments to evaluate the effect of the derived features as the amount of training data is varied. Figure 1 displays the F-score and the OOV recall of systems with different feature sets when trained on smaller portions of the labeled data. Note that there is no change in the configuration of the unlabeled data. We can clearly see that the derived features obtain consistent gains regardless of the size of the training set. The improvement is more significant when little labeled data is applied. Both statistics-based features and document-based features can help improve the overall performance. Especially, they can help to recognize more unknown words, which is important for many applications. The F-score of semi-supervised models, i.e. models trained with statistics-based features, does not achieve further improvement when document-based features are added. Nonetheless, the OOV recall obtains slightly improvements.

It is interesting to consider the amount by which derived features reduce the need for supervised data, given a desired level of accuracy. The change of the F-score in Figure 1 suggests that derived features reduce the need for supervised data by roughly a factor of 2. For example, the performance of the model with extra features trained on 500k characters

is slightly higher than the performance of the model with only baseline features trained on the whole labeled data.

3.4 Feature Analysis

We discussed the choice of using binary or numeric features in Section 2.3.4. In our experiment, when the accessor variety and punctuation variety information are integrated as numeric features, they do not contribute. To show the non-linear way that these features contribute to the prediction problem, we present the scatter plots of the score of each feature (i.e. the weight multiply the feature value) against the value of the feature. Figure 2 shows the relation between the score and the value of the punctuation variety features. For example, the weight of the binary feature "$L^2_{pu}(c_{[i:i+1]}) = 26$ combined with the label "B" learned by the final model is 0.815141, so the score of this combination is $0.815141 \times 26 = 21.193666$ and a point $(26, 21.193666)$ is drawn. These plots indicate the punctuation variety features contribute to the final model in a very complicated way. It is impossible to use one weight to capture it. The accessor variety features affect the model in the same way, so we do not give detailed discussions. We only show the same scatter plot of the $L^2_{av}(c_{[i:i+1]})$ feature template in Figure 3.

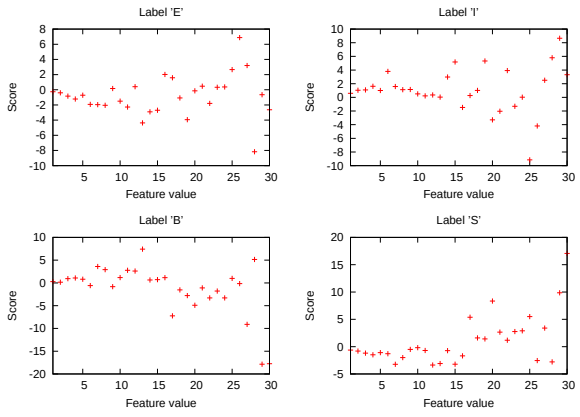

Figure 3: Scatter plot of feature score against feature value for $L^2_{av}(c_{[i:i+1]})$.

4 Related Work

Xu et al. (2008) presented a Bayesian semi-supervised approach to derive task-oriented word segmentation for machine translation (MT). This method learns new word types and word distributions on unlabeled data by considering segmentation as a hidden variable in MT. Different from their concern, our focus is general word segmentation.

The "feature-engineering" semi-supervised approach has been successfully applied to named entity recognition (Miller et al., 2004) and dependency parsing (Koo et al., 2008). These two papers demonstrated the effectiveness of using word clusters as features in discriminative learning. Moreover, Turian et al. (2010) compared different word clustering algorithms and evaluated their effect on both named entity recognition and text chunking.

As mentioned earlier, the feature design is inspired by some previous research on word segmentation. The accessor variety criterion is proposed to extract word types, i.e. the list of possible words, in (Feng et al., 2004). Different from their work, our method resolves the segmentation problem of running texts, in which this criterion is used to define features correlated with the character position labels. Li and Sun (2009) observed that punctuations are perfect delimiters which provide useful information for segmentation. Their method can be viewed as a self-training procedure, in which extra punctuation information is incorporated to filter out automatically predicted samples. We use the punctuation information in a different way. In our method, the counts of the preceding and succeeding strings of punctuations are incorporated directly as features into a supervised model.

In machine learning, transductive learning is a learning framework that typically makes use of unlabeled data. The goal of transductive learning is to only infer labels for the unlabeled data points in the test set rather than to learn a general classification function that can be applied to any future data sets. This means that the test data is known as a priori knowledge and can be used to construct better hypotheses. Although the idea to explore the document-level information in our work is similar to transductive learning, we do not use state-of-the-art transductive learning algorithms which involve learning when they meet the test data. For real-world applications, our approach is efficient by avoiding re-training.

5 Conclusion and Future Work

In this paper, we have presented a simple yet effective approach to explore unlabeled data for Chinese word segmentation. We are concerned with large-scale in-domain data and the document text. Experiments show that our approach achieves substantial improvement over a competitive baseline. Especially, the informative features derived from unlabeled data lead to significant improvement of the recall of unknown words. Our immediate concern for future work is to exploit the out-of-domain data to improve the robustness of current word segmentation systems. The idea would be to extract domain information from unlabeled data and define them as features in our unified approach. The word-based approach is an alternative for word segmentation. This kind of segmenters sequentially predicts whether the local sequence of characters make up a word. A natural avenue for future work is the extension of our method to the word-based approach. The word segmentation task is similar to constituency parsing, in the sense of finding boundaries of language units. Another interesting question is whether our method can be adapted to resolve constituency parsing.

Acknowledgments

The work is supported by the project TAKE (Technologies for Advanced Knowledge Extraction), funded under contract 01IW08003 by the German Federal Ministry of Education and Research. The author is also funded by German Academic Exchange Service (DAAD).

References

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Comput. Linguist.*, 30:75–93.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging – a case study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 522–530. Association for Computational Linguistics, Suntec, Singapore.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603. Association for Computational Linguistics, Columbus, Ohio.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for Chinese word segmentation. *Comput. Linguist.*, 35:505–512.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 337–342. Association for Computational Linguistics, Boston, Massachusetts, USA.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Valentin I. Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. 2010. Profiting from mark-up: Hypertext annotations for guided parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1278–1287. Association for Computational Linguistics, Uppsala, Sweden.

Weiwei Sun. 2010. Word-based and character-based word segmentation models: Comparison and combination. In *Coling 2010: Posters*, pages 1211–1219. Coling 2010 Organizing Committee, Beijing, China.

Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the ACL 2011 Conference*. Association for Computational Linguistics, Portland, Oregon, United States.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A

discriminative latent variable Chinese segmenter with hybrid word/character information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64. Association for Computational Linguistics, Boulder, Colorado.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *In Fourth SIGHAN Workshop on Chinese Language Processing*.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, Uppsala, Sweden.

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. Coling 2008 Organizing Committee, Manchester, UK.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. In *International Journal of Computational Linguistics and Chinese Language Processing*.