Reduce Meaningless Words for Joint Chinese Word Segmentation and Part-of-speech Tagging

Kaixu Zhang
Xiamen University
karey-zhang@gmail.com

Maosong Sun
Tsinghua University
sunmaosong@gmail.com

Abstract

Conventional statistics-based methods for joint Chinese word segmentation and part-of-speech tagging (S&T) have generalization ability to recognize new words that do not appear in the training data. An undesirable side effect is that a number of meaningless words will be incorrectly created. We propose an effective and efficient framework for S&T that introduces features to significantly reduce meaningless words generation. A general lexicon, Wikipedia and a large-scale raw corpus of 200 billion characters are used to generate word-based features for the wordhood. The word-lattice based framework consists of a character-based model and a word-based model in order to employ our word-based features. Experiments on Penn Chinese treebank 5 show that this method has a 62.9% reduction of meaningless word generation in comparison with the baseline. As a result, the F1 measure for segmentation is increased to 0.984.

1 Introduction

Chinese word segmentation is to segment Chinese sentence into words. Part-of-speech (POS) tagging is to further assign each segmented word a POS tag.

Statistics-based methods are used for joint Chinese word segmentation and part-of-speech tagging (S&T). There are character-based methods (Ng and Low, 2004), word-based methods (Zhang and Clark, 2008; Zhang and Clark, 2010), and hybrid methods employing both character-based features and word-based features (Kruengkrai et al., 2009).

| gold standard | result |
|---------------|--------|
| 拉脱维亚 (Latvija) | 拉脱维亚 |
| 射电望远镜 (radio telescope) | 射电望远镜 |
| 水污染 (water pollution) | 水污染 |
| 纯碱 (sodium carbonate, literally, pure alkali) | 纯碱 |

Figure 1: There are errors that contain incorrectly created meaningless words such as “拉脱维亚” and “射电望远镜”.

Figure 2: There are some errors that do not contain meaningless words. In the first case, to treat “water pollution” as two words “water” and “pollution” in Chinese is also grammatical and meaningful. In the second case, “sodium carbonate” may also be treated as a single word.

Conventional statistics-based methods for S&T have the generalization ability to recognize new words that do not appear in the training data. However, as a side effect, they also erroneously create a number of meaningless words.

Figure 1 shows some segmentation errors that result in meaningless words; while Figure 2 shows segmentation errors without producing meaningless words. There are several specifications for S&T. Results in Figure 2 may not be wrong according to some of the specifications; while results in Figure 1 will always be wrong. Moreover, since errors with no meaningless words are still meaningful and grammatical, errors that contain meaningless words are more serious and may have more negative impact.
on the downstream tasks.

In this paper, we introduce word-based features as criteria for wordhood (quality of being word) in an efficient S&T framework to reduce meaningless words and to improve the S&T performance.

There are at least two challenges to reduce the meaningless words for S&T.

- Reliable criteria for wordhood are needed for accurate word identification. The errors that contain meaningless words are caused by the limited training data. If the word “拉脱维亚” do not appear in the training data, it is possible for the model to guess that this is two separated new words. Thus we need extra resources for the criteria for wordhood. In this paper, we use a general lexicon, the Chinese Wikipedia and a large-scale raw corpus to provide information for the criteria. We describe the criteria in detail in Section 2.

- A fast word-based S&T framework is needed for employing word-based features we introduce. Since the word-based S&T methods are reported much slower than the character-based ones (Zhang and Clark, 2010), it might become intractable to employ more features. Inspired by Jiang et al. (2008b) and Sun (2011), we propose an effective and efficient S&T framework based on word-lattice and stacked learning. We describe this framework in Section 3.

We conduct our experiments on Penn Chinese Treebank 5.0. We first show that our S&T framework outperforms the state-of-art methods in closed test with a speed of about 100 sentences per second. Then we show that after further employing features of criteria for wordhood, the number of meaningless words has a reduction of 65.7% comparing to the baseline. As a result, the F1 measure for segmentation is increased to 0.984.

2 Criteria for Wordhood

2.1 General Lexicon

Using a general lexicon is a straightforward way to determine the wordhood. If a substring \( w \) of a sentence can be found in a general lexicon, heuristically it is possible to be segment as a word in this sentence.

SogouW\(^1\) is an available large lexicon. There are 79,019 words with their word categories in this lexicon. We will use this lexicon to generate features in order to provide a criterion for wordhood.

\[
\text{SogouW}(w) = \{\tau_i\}
\]  
(1)

The function returns a set containing word categories for the string \( w \).

Note that the word categories of this lexicon do not need to be the same with the POS tags we use in our S&T model.

2.2 Wikipedia

Besides general words, words that could be incorrectly segmented are named entities and domain-specific words. We use Chinese Wikipedia\(^2\), an online encyclopedia, as a lexicon for these words. There are totally 1,310,114 identical entries in the Chinese Wikipedia.

Unlike the general lexicon, there is no word category information about the Wikipedia entries. We simply define a function \( \text{Wiki}(w) \) to indicate whether a string \( w \) is an entry in Wikipedia or not:

\[
\text{Wiki}(w) = \begin{cases} 
1 & \text{w is an entity in Wikipedia} \\
0 & \text{otherwise}
\end{cases}
\]  
(2)

2.3 Large-scale Raw Corpus

Using lexicons may still have limitation, since they can not include all the low-frequency words. We use a large-scale raw corpus of web pages to provide another criterion for wordhood.

In this paper, we use SogouT\(^3\) as the large-scale raw corpus. It consists of web pages with more than 200 billion Chinese characters. We use \( \text{freq}(w) \) to indicate the frequency of string \( w \) in this corpus, and use \( \text{freq}(l,w,r) \) to indicate the frequency of string \( w \) that occurs right before character \( l \) and after character \( r \). The symbol “#” is used as a wildcard for sentence boundaries such as punctuation marks.

Large-scale raw corpus with the advantages of wide coverage and large duplication can provide rich information for language processing, although it is not manually annotated. It plays an important role in

\(^1\)http://www.sogou.com/labs/dl/w.html
\(^2\)http://zh.wikipedia.org/
\(^3\)http://www.sogou.com/labs/dl/t.html
shows the RA V values (a) is an overview of our proposed frame-
(b) shows examples of each
define the RA V of a string as:

\[ \text{RAV}(w) = \sum_{\langle l, r \rangle \in P} \text{match}(w, \langle l, r \rangle) \]  (4)

The larger the RA V is, the more possible that \( w \) could be a word.

In this paper, \( \epsilon \) is set to 0.0001, and 30 restricted pairs are automatically selected as the restricted pair set. The last row of Table 1 shows the RA V values of some strings as examples.

### 3 Word Lattice Based S&T Framework

#### 3.1 Overview

The goal of this section is to build an effective and efficient S&T framework which can employ both character-based features and word-based features. Figure 3 (a) is an overview of our proposed framework inspired by Jiang et al. (2008b) and Sun (2011); while Figure 3 (b) shows examples of each step.

The input of S&T is a sequence of Chinese characters:

\[ c = (c_1, c_2, \ldots, c_{|c|}) \]  (5)

where \( c_i \) are Chinese characters and \( |c| \) denotes the number of elements in the sequence \( c \).

With the help of the character-based model (Section 3.2), a word lattice generation method (Section 3.3) is used to generate the word lattice which contains one or more possible outputs (Figure 3 (b)).

The word lattice is defined as a set of edges

\[ \mathcal{L} = \{ \langle p_i, w_i, t_i \rangle \} \]  (6)

where each edge is a tuple \( \langle p_i, w_i, t_i \rangle \). \( w_i \) is the corresponding word of the edge, \( t_i \) is the POS tag, and \( p_i \) denotes the position which is the number of characters before \( w_i \).

Then a pruned word-based S&T model (Section 3.4) is used to search the final output in the word lattice. Our features for the wordhood will be employed in this model.

The output of S&T is defined as a sequence of edges:

\[ y = (\langle p_{11}, w_{11}, t_{11} \rangle, \ldots, \langle p_{|y|}, w_{|y|}, t_{|y|} \rangle) \]  (7)
where $p_i = |w_1| + \cdots + |w_{i-1}|$.

Additionally, we use the stacked learning framework (Wolpert, 1992) to train the character-based S&T model and the pruned word-based S&T model. Similar to Sun (2011), a 10-fold cross-validation is used to train the pruned word-based model.

There is one problem that the word lattices for the training set may not always contain all the edges of the gold standard output $y^*$. Jiang et al. (2008b) provided a solution to use the “oracle diameter” as the gold standard in the training process of pruned word-based model. In our system, we directly add missing edges into the word lattice without defining the weights. The observation is that using our treatment is better at least in our framework.

### 3.2 Character-Based Model

The character-based models for Chinese word segmentation (Seg) and S&T are well studied (e.g., Xue and Shen (2003)).

The output of the character-based model is a sequence of character labels:

$$a = (\langle s_1,t_1 \rangle, \ldots, \langle s_{|a|}, t_{|a|} \rangle)$$

where $|a| = |c|$. Each label is a tuple $\langle s_i,t_i \rangle$. $s_i \in S = \{b,m,e,s\}$ is used to denote the position of $c_i$ in a word. The elements “b” / “m” / “e” indicate that the character is at the beginning / in the middle / at the end a multi-character word; the element “s” indicates that the corresponding character forms a single character word. And $t_i \in T$ is the POS tag of the word containing $c_i$.

The label $a_i = (s_i,t_i)$ indicates the S&T result of the corresponding character $c_i$. For example, the label $(b,NN)$ indicates that the corresponding character is at the beginning of a multi-character noun with the POS tag “NN”. These labels are used by Ng and Low (2004) and Jiang et al. (2008b).

An averaged perceptron model (Collins, 2002) is used for the learning and prediction. Following the convention, the feature templates of our model are showed in Table 2. The optimal label sequence $\hat{a}$ is calculated as:

$$\hat{a} = \arg \max_a f_c(c,a)$$

Table 2: The feature templates of the character-based model

| Unigram | $\langle c_{i-1}, a_i \rangle, \langle c_i, a_i \rangle, \langle c_{i+1}, a_i \rangle$ |
| Bigram  | $\langle c_{i-2}, c_{i-1}, a_i \rangle, \langle c_{i-1}, c_i, a_i \rangle, \langle c_i, c_{i+1}, a_i \rangle, \langle c_{i+1}, c_{i+2}, a_i \rangle$ |
| Transitional | $\langle a_{i-1}, a_i \rangle$ |

Roughly speaking, our model is similar to the pre-
vious proposed models with three differences. First, we do not use the feature template \( \langle c_{i-1}, c_{i+1}, a_i \rangle \). According to our empirical observation, this template will cause overfitting due to the data sparseness. Second, we define special characters for sentence boundaries \( c_{-1} = c_0 = \# = c_{|e|+1} = c_{|e|+2} \) to generate corresponding features. Third, we find that the search order of the dynamic programming algorithm of the decoding process is important. In our model, a label is calculated earlier if it appears earlier in the gold standard of the training data.

### 3.3 Word Lattice Generation

We will use the objective function \( f(c, a) \) of the character-based model to generate the word lattice.

The study by Jiang et al. (2008b) showed that using word lattice is better than using the \( n \)-best list for the reranking. However, in Jiang et al., the algorithm to generate the word lattice is still using a local \( n \)-best strategy. Here we present a more elaborate method to generate the word lattice.

In our system, the word lattice based on the input \( c \) and a threshold \( \delta \) is a set of edges, formally defined as:

\[
\mathcal{L}_{c, \delta} = \{ \langle p_i, w_i, t_i \rangle | \text{there exists an output } y \]
\[
\text{such that } \langle p_i, w_i, t_i \rangle \in y \}
\]

(10)

where \( a(y) \) denote the character label sequence corresponding to the output \( y \).

The motivation is that an edge is good enough to be in the word lattice if and only if there exists an output \( y \) containing this edge and has a high objective function value which is close to the best \( f(c, \hat{a}) \). Similar idea is also proposed by Huang (2008) and Mi et al. (2010).

We can weight an edge \( y_i = \langle p_i, w_i, t_i \rangle \) in the word lattice as the minimal margin between the best output \( \hat{a} \) and the output containing \( y_i \):

\[
m_i = f(c, \hat{a}) - \max_{y_i \in y} f(c, a(y))
\]

(11)

### 3.4 Pruned Word-based Model

Given a word lattice, our system needs to find a sequence of edges to form the output. This can be formalized as:

\[
\hat{y} = \arg \max_{y_i \in \mathcal{L}_{c, \delta}} f_w(y)
\]

(12)

|   | \( \langle w_i \rangle \) |
|---|------------------|
| 2 | \( \langle w_i, t_i \rangle \) |
| 3 | \( \langle |w_i| > 1 \rangle \) |
| 4 | \( \langle h(m_i) \rangle \) |
| 5 | \( \langle h(m_i), |w_i| > 1 \rangle \) |
| 6 | \( \langle t_i \rangle \) |
| 7 | \( \langle t_i, |w_i| > 1 \rangle \) |
| 8 | \( \langle t_{i-1}, t_i \rangle \) |
| 9 | \( \langle t_{i-1}, |w_{i-1}| > 1, t_i, |w_i| > 1 \rangle \) |
| 10 | \( \langle t_i, \tau \in \text{SogouW}(w_i) \rangle \) |
| 11 | \( \langle t_i, \text{Wiki}(w_i) \rangle \) |
| 12 | \( \langle t_i, \lceil \text{RAV}(w_i) / 2 \rceil \rangle \) |

Table 3: The feature templates of the pruned word-based S&T model. Templates 10 to 12 are only for the open test.

We call this pruned word-based model since we only consider words from \( \mathcal{L}_{c, \delta} \).

Also, an averaged perceptron model is used for the learning and prediction. Word-based features based on the criteria in Section 2 are used in this model.

The feature templates used for the pruned word-based model are listed in Table 3. Templates 1 to 9 are used for both closed and open test; while Templates 10 to 12 related to the wordhood are only used for the open test.

Template 1 represents the word corresponding to the edge. Since word length is useful, we use Template 3 to represent such information. Template 8 and Template 9 are used to represent the information of two contiguous words.

Stacked features (related to the previous character-based model) are also represented as Template 4 and Template 5. The weight of an edge \( m_i \) is defined in Eq. 11. \( h(m) \) is a genralized step function that maps a continuous value to a discrete value: \( h(m) = \lceil \log_2 |m| \rceil \) and we define \( h(0) = -\infty \).

Template 10 is based on the SogouW. If a word is assigned with more than one word categories in SogouW, each of the categories \( \tau \) will generate a feature together with \( t_i \). The machine learning algorithm will automatically learn the relation between the word categories and the POS tags. Template 11 is based on the Wikipedia. And Template 12 is based on the RAV calculated from the large-scale corpus.
shows the summary of these sets.

Sun, 2011 shows that the quality of the first step of a two-step S&T method is less than 1,000.

### 4 Experiments

We conduct our experiments on the Penn Chinese Treebank 5.0 (CTB 5.0). We use the same training, development and test sets as the previous work (Jiang et al., 2008b; Zhang and Clark, 2010; Sun, 2011). Table 4 shows the summary of these sets. The development set is used to determine some parameters of our system such as the feature templates and the threshold \( \delta \) for the word lattice generation.

An edge is correct if and only if it appears in both \( \hat{y} \) and the gold standard output sequence \( y^* \). We use the F1 measure to measure the performance of our S&T system. The F1 for S&T is defined as:

\[
F1 = \frac{2 \sum_{c} |\hat{y} \cap y^*|}{\sum_{c} |\hat{y}| + \sum_{c} |y^*|}
\]

where \( |\hat{y} \cap y^*| \) is the number of correct edges of sentence \( c \).

If we do not distinguish POS tags of edges in \( \hat{y} \) and \( y^* \) in the equation, we get the definition of the F1 for Chinese word segmentation (Seg). We use this F1 measure to evaluate the performance of Seg.

### 4.1 Influence of Lattice Scale

Word lattices are generated with the help of the character-based model and the word lattice generation algorithm described in Section 3.3. The parameter \( \delta \) is used to control the lattice scale.

We define the lattice scale which is the ratio of the size of the word lattice to the size of the gold standard: lattice scale = \( \frac{\sum_{c} |L_{c,\delta}|}{\sum_{c} |y^*|} \). And we define the oracle recall which is the ratio of the number of the edges in both the word lattice and the gold standard to the size of the gold standard to indicate in what degree the gold standard outputs are contained in the word lattices: oracle recall = \( \frac{\sum_{c} |L_{c,\delta} \cap y^*|}{\sum_{c} |y^*|} \). The oracle recall is similar to the oracle F1 used by related work to indicate the quality of the first step of a two-step S&T method (Jiang et al., 2008b; Sun, 2011).

In our system, we can easily manipulate the lattice scale and the oracle recall in a wide range by changing \( \delta \). The relation between these three variables are shown in Fig. 4.

As shown in (Jiang et al., 2008b), when the degree is set to 10, the lattice scale for the test set is larger than 10 and the oracle F1 is 0.9779. The oracle F1 reported by Sun (2011) is 0.9929, which relies on three models.

Then, we find a local optimal \( \delta \) to determine the lattice scale used for the pruned word-based model. Results on the development set are shown in Table 5. Feature Templates 1 to 9 are used. We set \( \delta \) to 15 in the rest of our experiments. Note that a large \( \delta \) will also cause a longer decoding time.

| Data set | CTB files | Sentences | Words |
|----------|-----------|-----------|-------|
| Training | 1-270     | 18,086    | 493,938 |
|          | 400-931   |           |       |
|          | 1001-1151 |           |       |
| Dev.     | 301-325   | 350       | 6,821  |
| Test     | 271-300   | 348       | 8,008  |

Table 4: Summary of the training, development and test sets of CTB 5.0

For the limitation of time and storage, we do not calculate and store the RAV for strings with frequency less than 1,000.

![Figure 4: The lattice scale and the oracle recall can be easily manipulated by changing the threshold \( \delta \)](image-url)
### Closed Test for Pruned Word-based Model

We conduct the experiment on the test set for closed test and compare our results with the previous work in Table 6.

Our system outperforms the previous systems listed in the table in closed test according to the Seg F1 and the S&T F1. Especially, our system has a high performance of Seg F1.

We perform the significant test using method for Chinese word segmentation in SIGHAN bakeoff (Emerson, 2005). The 95% confidence intervals of our full system are listed below the F1 measures. We see that the differences between those methods are not significant enough. This issue is for all the related work.

We measure the speed of our system for the closed test on the test set. The program uses one core of 2.53GHz (Intel Core 2 Duo E7200). Results are shown in Table 7. The total time of the test procedure is 3.229 seconds, including 1.041 seconds for lattice generation and 2.188 seconds for the pruned word-based model. And note that the feature extraction of the pruned word-based model (a pruned word-based S&T model) is implemented using Python, which is much slower than C/C++. The speed of that word-based S&T model using cross validation is reported by Zhang and Clark (2010). The speed of that word-based S&T model is 24.92 sentences per second (a single 2.66GHz Intel Core 2 CPU), while the speed of our system is 107.77 sentences per second.

### Features for Wordhood

Finally, we conduct the experiment on the test set for open test and compare our results with the previous work in Table 8.

We try different combinations of our three feature templates for wordhood. We find that in both development set and test set, features derived from the general lexicon and the raw corpus are helpful while features derived from the Wikipedia is not helpful.

The 95% confidence intervals are calculated for the best results of our model. The differences are also not significant.
5 Related Work

Zhang and Clark (2008), Mochihashi et al. (2009) and Krueckrai et al. (2009) proposed models for S&T or Seg with both character-based features and word-based features in a single model of which the time spending is considerable.

Sun (2011) proposed a stacked framework to connect a character-based model and a word-based model in parallel; while Jiang et al. (2008b) proposed a word lattice based reranking method in which the character-based model and word-based model are connected in series. The main difference between the method by Jiang et al. and ours are the lattice generation method and the feature templates. And we use the way proposed by Sun to train the character-based model and word-based model.

Several resources were used for the open test of S&T and Seg. Sun (2011) used a lexicon of idioms. Jiang et al. (2009) used an annotated corpus with different annotation standard. Wang et al. (2011) used auto-analyzed data. The data used in those methods are not comparable with the corpus of ours.

Li and Sun (2009) used nearly the large-scale corpus with ours in a semi-supervised way. The learned model had a better ability to recognize new word, while it did not outperform the baseline. In this paper, we employ the statistical information for that corpus as features in our model.

There are proposed criteria for Chinese word extraction (Chien, 1997; Chen and Ma, 2002). For example, Feng et al. (2004b; 2004a) proposed the criterion accessor variety for Chinese word extraction and used it for Chinese word segmentation. Jin et al. (2006) used branching entropy to perform unsupervised Chinese word segmentation. Since these methods will become intractable for a large scale corpus, we adapted the method by Zhang et al. (2011) in our experiment.

6 Conclusion

In this paper, we proposed an S&T method based on the word lattice adapted from Jiang et al. (2008b). A simple but more effective method is used for the word lattice generation based on the objective function of a character-based S&T model. The scale of lattice can be easily controlled by a simple parameter $\delta$. Then a pruned word-based S&T model is used to find an optimal path in the word lattice as the output. Our experiments on Penn Chinese treebank 5 show that our system is faster than the single word-based model and outperforms the previous work in the closed test.

We employ features of Chinese wordhood from lexicon and large-scale raw corpus of 200 billion characters. Our final system has a 62.9% reduction of meaningless word generation in comparison with the character-based model. As a result, the F1 measure for segmentation is increased to 0.984.

The CTB corpus mainly consists of news and magazines. And the performance is already high. When we want to do S&T on documents such as blogs, microblogs or domain-specified documents other than news, the meaningless words in the output will be an even greater challenge. We will test our method on such documents in the future.

References

M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2009. Open information extraction for the web. University of Washington.

T. Brants, A.C. Popat, P. Xu, F.J. Och, and J. Dean. 2007. Large language models in machine translation. In EMNLP. Citeseer.

Keh-Jiann Chen and Wei-Yun Ma. 2002. Unknown word extraction for chinese documents. In Proceedings of the 19th international conference on Computational linguistics-Volume I, pages 1–7. Association for Computational Linguistics.

Lee-Feng Chien. 1997. PAT-tree-based keyword extraction for chinese information retrieval. In ACM SIGIR Forum, volume 31, pages 50–58. ACM.
Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, page 1C8.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 123–133. Jeju Island, Korea.

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004a. Accessor variety criteria for chinese word extraction. Computational Linguistics, 30(1):75–93.

Haodi Feng, Kang Chen, Chunyu Kit, and Xiaotie Deng. 2004b. Unsupervised segmentation of chinese corpus using accessor variety. Natural Language Processing IJCNLP 2004, pages 694–703.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In Proceedings of ACL-08: HLT, pages 586–594, Columbus, Ohio, June. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan L. 2008a. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics.

Wenbin Jiang, Haitao Mi, and Qun Liu. 2008b. Word lattice reranking for chinese word segmentation and Part-of-Speech tagging. In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pages 385–392. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging c a case study. In Proceedings of the 47th ACL, page 522C530, Suntec, Singapore, August. Association for Computational Linguistics.

Zhihui Jin and Kuniko Tanaka-Ishii. 2006. Unsupervised segmentation of chinese text by use of branching entropy. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, page 428C435, Sydney, Australia, July. Association for Computational Linguistics.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun’ichi Kazama, Yiu Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An Error-Driven Word-Character hybrid model for joint chinese word segmentation and POS tagging. In Proc. of ACL-IJCNLP 2009, page 513C521, Suntec, Singapore. Association for Computational Linguistics.

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