Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information

Xiao LUO; Maosong SUN
National Lab. of Intelligent Tech. and Systems
Tsinghua University, Beijing, China, 100084
lkc-dcs@mail.tsinghua.edu.cn

Benjamin K TSOU
Language Information Sciences Research Centre
City University of Hong Kong, Hong Kong
rlbtsou@uxmail.cityu.edu.hk

Abstract

Covering ambiguity is one of the two basic types of ambiguities in Chinese word segmentation. We regard its resolution as equivalent to word sense disambiguation, and make use of the classical vector space model in information retrieval to formulate the contexts of ambiguous words. A variation form of TFIDF weighting is proposed and a Chinese thesaurus is additionally utilized to cope with data sparseness problem. We select 90 frequent cases of covering ambiguities as the target. The training set includes 77654 sentences, and the test set includes 19242 sentences. The experimental results showed that our model has achieved 96.58% accuracy, outperforming the original form of TFIDF weighting as well as another baseline model, the hidden Markov model.

1 Introduction

In Chinese texts, there are no separators, such as spaces in English, to explicitly indicate boundaries between words. So word segmentation is regarded as the first step in Chinese information processing systems. Ambiguity is one of the two main obstacles in Chinese word segmentation (another is unknown word). There are two basic types of ambiguities in Chinese: overlapping ambiguity and covering ambiguity. The focus of this paper is on covering ambiguity.

Covering ambiguity is defined as follows: Given a word $w \in$ W, W is a Chinese lexicon, if w is a concatenation of multiple words $w_1 \ldots w_n$ (n $\geq$ 2), $w_i \in$ W (i=1…n), and in addition, both the sequence w and the sequence $w_1 \ldots w_n$ can be realized in some sentences, then we term w a "covering ambiguity" (We also refer to w an "ambiguous word" throughout the paper), meanwhile, term the sequence w and $w_1 \ldots w_n$ the combined form and the separated form of the covering ambiguity w respectively.

Observe "中将", a two-character string with covering ambiguity. Firstly, three words, i.e. "中" (middle), "将"(will) and "中将"(lieutenant general), are involved in it. Secondly, both sequences can stand in real sentences. For examples:

(1a) 1955年他被授予中将军衔。
(1b) 信息在社会发展中将起到关键作用。

Correct segmentations are:

(1a) 1955年 | 他 | 被 | 授予 |
1955 year he by award
中将 | 军衔 | 。
lieutenant general military rank .
(*In 1955, he was awarded the military rank "lieutenant genera".*)

(1b) 信息 | 在 | 社会 | 发展
information in society development
| 中 | 将 | 起到 | 关键 | 作用 | 。
middle will play key role .
(Information will play a key role in society development.)

Covering ambiguities constitute about 10 percent of segmentation ambiguities in running Chinese texts (Liang, 1987). Their resolution is more difficult than that of overlapping ambiguities, -- it depends heavily on contexts, including syntactic, semantic and even pragmatic information. According to an evaluation held by the 863 High-Tech Project of China, the highest accuracy for covering

ambiguities is only 59.0% (Liu, 1997). Zheng (1999) presented a rule-based method, in which rules are prepared manually by making use of part-of-speech information and collocation of words, and got 85.0% accuracy in a close test. Xiao et al. (2001) regarded covering ambiguity resolution as an equivalent problem of word sense disambiguation (WSD). They exploited vector space model and TFIDF-like weighting scheme, and experimented on 20 ambiguous words.

The work here is in the line of Xiao et al. (2001). A variation form of TFIDF weighting is proposed, and a Chinese thesaurus is additionally utilized to cope with data sparseness problem. We select 90 frequent cases of covering ambiguities as the target. The experimental results show our model achieving 96.58% accuracy, outperforming the original form of TFIDF weighting as well as another baseline model, the hidden Markov model.

2 The proposed method for covering ambiguity resolution

2.1 Vector space model

Vector space model (VSM) and the relevant term weighting technique were initially developed in information retrieval (Salton and Buckley, 1988). Documents and queries are represented in a high-dimensional space, in which each dimension of the space corresponds to a term in the set of document collection (Manning and Schütze, 2000). This model has been applied to WSD thoroughly, in which the vector space is used to formulate contexts of polysemous words (Ide and Veronis, 1991; Yarowsky, 1992; Gale, 1993). As pointed out before, the contexts of ambiguous words are necessary for the resolution of covering ambiguities, thus the methods for WSD would also be rational for the issue of this paper.

2.2 The window of context

In the framework of VSM, all words co-occurring with an ambiguous word *w* could be extracted from sentences to form the vector of *w*, serving as its context. Xiao et al. (2001) found by experiments that it was appropriate for resolution of covering ambiguity if the window of context is restricted to ±3 words centered on *w*, i.e., three words preceding *w* and three words following *w* (Fig. 1). The position of a neighboring word of *w* is indicated in a negative number if it is on the left side of *w* and in a positive number if on the right side.

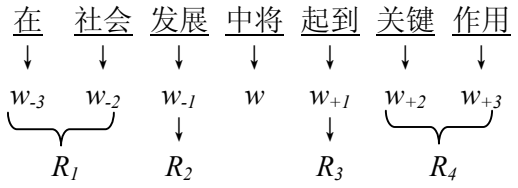

Fig. 1: The window of context of *w*

Xiao et al. (2001) suggested that it would be better for disambiguation if the six words are further divided into four regions, R_1 for w_{-3} and w_{-2}, R_2 for w_{-1}, R_3 for w_{+1}, R_4 for w_{+2} and w_{+3}.

We have simply accepted the above two conclusions.

2.3 TFIDF

TFIDF is a well-known formula in VSM (Salton, 1991). We re-state it here to fit our task.

Given an ambiguous word *w*, let:

The segmentation form *i* of *w*: *i* = 1 when *w* is in the combined form, 2 when *w* is the separated form;

d: The number of segmentation forms of *w*. It is always 2;

D_i: The collection of sentences containing *w* in the segmentation form *i* (the training set of *w*);

n: The number of distinct words in the union of D_1 and D_2;

tf_{ijk}: The frequency of word t_j in region R_k in collection D_i;

tf_{qjk}: The frequency of word t_j in region R_k in the input sentence *Q* containing *w*;

df_{jk}: The number of collections (D_1 and D_2) that contain t_j in region R_k. Its value ranges from 0 to 2;

idf_{jk}: $\log(d/df_{jk})$

To compute the weight of the word t_j in region R_k for the segmentation form *i* of *w*, we define the following general form (i=1,2; j=1,…,n; k=1,…,4):

$$d_{ijk} = TF_{ijk} \times IDF_{jk} \quad (1)$$

The original forms of TF and IDF are:

$$TF_{ijk} = tf_{ijk} \quad (2)$$

$$IDF_{jk} = idf_{jk} = \log(d/df_{jk}) \quad (3)$$

The similarity coefficient between w in the input sentence Q and the segmentation form i of w is defined as:

$$SC(Q,i) = \sum_{j=1}^{n} \sum_{k=1}^{4} tf_{qjk} \times d_{ijk} \quad (4)$$

The ambiguous word w in Q will take the segmentation form that maximizes the formula (4) over i.

2.4 Variation of TF

Variations of TF were studied to improve the performance of IR (Salton and Buckley, 1988). A typical one is using the logarithm of word frequency (Nie et al., 2000):

$$TF_{ijk} = \log tf_{ijk} + 1.0 \quad (5)$$

We made a minor modification on formula (5):

$$TF_{ijk} = \log(tf_{ijk} + 1.0) \quad (6)$$

2.5 Variation of IDF

We proposed a variation of *IDF* which is totally different from its original form given by formula (3), as follows:

$$IDF_{jk} = \exp(h \times \theta_{jk}) \quad (7)$$

where:

$$\theta_{jk} = \frac{\sigma_{jk}}{\mu_{jk}} \quad (8)$$

$$\mu_{jk} = \frac{1}{d} \sum_{i=1}^{d} tf_{ijk} \quad (9)$$

$$\sigma_{jk} = \sqrt{\frac{1}{d} \sum_{i=1}^{d} (tf_{ijk} - \mu_{jk})^2} \quad (10)$$

Note: μ_{jk} and σ_{jk} are the mean and the standard deviation of the frequency of t_j in region R_k in both forms respectively, θ_{jk} is the average σ_{jk} by μ_{jk}, h is a constant for adjusting the ratio of *IDF* to *TF* (The default value of h is 1.0).

The essence underlying formula (3) and formula (7) is quite similar, but the latter seems more adequate than the former in two aspects: it is more precise in describing the distribution of a word in both segmentation forms; and further, the exponential function has the effect of amplifying the difference.

2.6 Coping with data sparseness by using a Chinese thesaurus

The vector space is high-dimensional due to the large number of words in Chinese lexicon, so we encountered a serious data sparseness problem. To solve this, *TongYiCiCiLin*, a Chinese thesaurus (Mei et al., 1983) is used. The thesaurus has a three-level semantic coding hierarchy, for instance, the code of "国家"(country) is "Di02", where "D" stands for "abstract thing", "i" for "society, politics or law", and "02" for "country or district". A reasonable strategy here would be to replace low frequency words in vector space with their semantic codes. The details will be discussed in 3.4. A consequence of doing so is that the generalization capability of the model may also be improved to some extent.

3 The experimental study and results

3.1 Training set and test set

We select 90 frequent cases of covering ambiguities (i.e., 90 words with covering ambiguities), as listed in table 1, to be the target. For each ambiguous word w, a number of sentences containing it are randomly extracted from a large-scale Chinese corpus. The extracted sentences for all w form the data set as a whole. The data set is manually divided into two parts according to the segmentation forms of each w. we randomly sampled 80% of the data set, totally 77654 sentences, for training (the training set), and the rest 20%, totally 19242 sentences, for testing (the test set). On average, each ambiguous word has 863 sentences for training and 213 sentences for testing.

We notice that the distribution of randomly sampled sentences of w regarding two segmentation forms varies on the individual

basis. Suppose n_i is the number of sentences containing w in the training set with the segmentation form i. we can use

$$u = \frac{\max(n_1, n_2)}{\min(n_1, n_2)} \qquad (11)$$

to reflect the degree of imbalance of w in segmentation forms.

Moreover, we call the segmentation form with maximum n_i of w the major segmentation form of w. The remaining one will be the minor segmentation form of w.

u	Sub-total	Ambiguous word
$1 \leqslant u \leqslant 6$	26	成人,从小,大小,到底,的话,地点,多年,个人,家人,年内,人为,人心,上将,上来,上下,学会,一道,一点,一块,一行,在场,正当,正在,中将,着手,总会
$6 < u \leqslant 26$	22	才能,处在,创新,东西,都会,过后,会同,内在,年前,年中,前来,人生,日前,市区,是以,天下,一起,以为,因此,中长期,中等,走向
$u > 26$	42	比分,不断,部长,从中,存在,更新,国有,后来,即将,将来,可以,列车,马上,名将,名人,难以,年产,年会,前提,人才,人大,人均,上升,十分,所以,条约,下来,现在,行人,行为,要求,一定,一度,一生,一时,一下,已经,以前,中共,中学,自我,最近

Table 1: 90 ambiguous words grouped by u

Let us look at two examples. For ambiguous word "从小", n_1=612, n_2=141, so u=612/141≈4.34, and its major segmentation form is 1; for ambiguous word "才能", n_1=110, n_2=1347, so u=1347/110≈12.25, and its major segmentation form is 2.

Taking u as a normalization factor, formula (6) can be changed to its new form:

$$TF_{ijk} = \log(u^{\varepsilon_i} \times tf_{ijk} + 1.0) \qquad (12)$$

where:

$\varepsilon_i = 0$ if $n_i = \max(n_1, n_2)$

$= 1$ else

The experiment is conducted in two steps.

Step one

We choose ambiguous words with u as close as possible to 1 to design the core model, for two reasons: firstly, it is the most difficult case for disambiguation; secondly, the training data for two segmentation forms is roughly balanced. The core model to be determined in this section is based on 26 ambiguous words within a range $1 \leqslant u \leqslant 6$ (see sections 3.2, 3.3, 3.4, and 3.5).

Step two

The core model obtained in step one is then applied to other groups of ambiguous words to check its coverage. In cases when it is not feasible, we try other strategies (see section 3.6).

We define the accuracy of disambiguation for a collection of ambiguous words as the ratio of the number of times these words being correctly classified into their segmentation forms in sentences to the number of sentences containing these words in the test set.

3.2 Baseline — HMM

Sun et al. (1997) developed an integrated system, CSeg&Tag1.0, for doing word-segmentation and part-of-speech tagging simultaneously. In CSeg&Tag1.0, all possible word segmentation paths and all possible POS paths for an input sentence are expanded, forming a candidate space. Then the system uses HMM (Bigram) and dynamic programming to find the best path of word-segmentation and POS tagging. We have adopted this system as the baseline model. Two segmentation forms of each ambiguous word in a sentence are to be generated as part of

the candidate space, and the solution for the ambiguous word can be finally found in the best path. The accuracy of HMM for 26 words with $1 \leqslant u \leqslant 6$ is 68.48%. The result shows that HMM and adjacent POS information may be inappropriate for covering ambiguity resolution.

3.3 Results of TFIDF

Firstly, we compare the effects of different forms of *TF*. We fix formula (3) as *IDF*, and test *TF* in formulae (2), (5) and (12). The accuracy for the 26 words is 83.68%, 88.72% and 89.08% respectively. These results are much better than that of HMM. We thus regard formula (12) as a more effective form of *TF*.

Secondly, we compare the effects of different forms of *IDF*. We fix formula (12) as *TF*, and test *IDF* in formulae (3) and (7). The accuracy for the 26 words according to them is 89.08% and 92.85% respectively. Thus formula (7) outperforms formula (3).

Based on the results above, we finally choose formula (12) as *TF* and formula (7) as *IDF* in computing d_{ijk} :

$$d_{ijk} = \log(u^{\varepsilon_i} \times tf_{ijk} + 1.0) \times \exp(h \times \theta_{jk}) \quad (13)$$

3.4 Results of using Chinese thesaurus

As mentioned earlier, a rational way for settling data sparseness problem is to substitute low frequency words in vector space with their semantic codes. There are two related issues left: How to deal with polysemous words in the vector space in substitution? And, how about the threshold of frequencies for words in vector space under which the substitution should be done?

To the first question, we try three strategies: (1) simply discard all polysemous words; (2) replace each polysemous word with its semantic codes and assign each semantic code with the frequency of the word (though this seems not a sound strategy); and (3) replace each polysemous word with its semantic codes and assign each semantic code an average frequency, derived from the frequency of the word divided by the number of its semantic codes. The threshold of frequencies for both strategies (2) and (3) is setting at 10. The accuracy for the 26 words is 92.39%, 91.90% and 93.29% respectively. We confirm two points from the result: first, strategy (1) causes a reduction of the accuracy from 92.85%, by formula (13) directly without any substitution, to 92.39%, indicating that polysemous words should be kept in the vector space explicitly or implicitly; second, strategy (3) increases the accuracy from 92.85% to 93.29%, showing that it is adequate for frequency assignment to the expanded semantic codes of polysemous words.

We turn to the second question now. Different values of the threshold are tested. For any word in the vector space, if its frequency is below the given threshold, the substitution with semantic codes will be done on it. The results are shown in Fig. 2. As can be seen, the highest accuracy, i.e., 93.49%, for the 26 words is obtained when the threshold is at 35. Consequently, the dimension of vector space is decreased to 30% on average under this condition. Also note that as the threshold tends to infinite (i.e., all the words in vector space are transformed into their semantic codes), the accuracy is about 92.83%. This suggested that the vector space would like to be a mixture of high-frequency words and semantic codes derived from low-frequency words in well-balanced status.

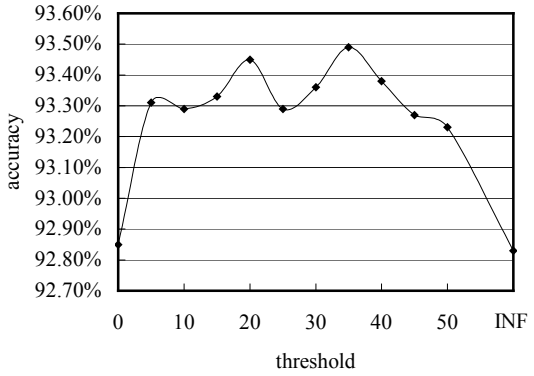

Fig. 2: Threshold for low-frequency word substitution with semantic codes

3.5 The core model

We are going to fix the last factor of the core model: the constant *h* in formula (13). We change *h* from 0.1 to 10. The highest accuracy, i.e., 93.58%, for the 26 words is obtained when *h* is 1.6 (Fig. 3).

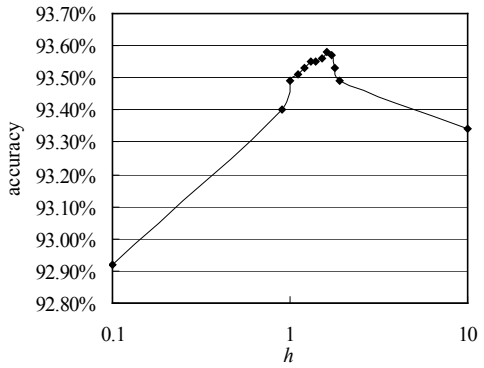

Fig. 3: Adjustment of *h*

Now we get the core model: We take formula (13) as weighting scheme, in which *h* is set to 1.6. Any polysemous word in the vector space, as well as any word with frequency less than 35 in the vector space (regardless if it is polysemous), shall be transformed to its semantic codes correspondingly. Each semantic code is then assigned an average frequency derived from the frequency of the word divided by the number of its semantic codes.

3.6 Applying the core model to all ambiguous words

The core model is constructed in the case of $1 \leqslant u \leqslant 6$. We now apply it to other two cases in table 1, i.e., $6 < u \leqslant 26$ and $u > 26$. We compared our core model to a very simple model that takes the major segmentation form straightforwardly as the solution (we call it the default model). The results are listed in table 2.

u	Accuracy of the core model	Accuracy of The default model
$1 \leqslant u \leqslant 6$	93.58%	67.92%
$6 < u \leqslant 26$	95.92%	92.49%
$u > 26$	97.40%	98.77%

Table 2: Comparison between the core model and the default model

Note that in case of $u > 26$, the default model outperforms the core model.

It is time for us to provide the total solution: using the core model when *u* is smaller than 26, otherwise using the default model (table 3).

u	The proposed model
$1 \leqslant u \leqslant 6$	The core model
$6 < u \leqslant 26$	The core model
$u > 26$	The default model

Table 3: The proposed model

We obtained 96.58% accuracy on average for all 90 ambiguous words by using the proposed model. The average accuracy for the default model and HMM (baseline) is 88.32% and 68.48% respectively. The improvement is rather significant (table 4).

Accuracy of the proposed model	Accuracy of the default model	Accuracy of HMM
96.58%	88.32%	68.48%

Table 4: Comparsion among the proposed model, the default model and HMM

The processing speed is quite fast: it takes about six minutes to train the model with 77654 sentences, and takes two minutes to test the model with 19242 sentences, on a personal computer with P4 and 128M memory.

4 Conclusion

We assume the resolution of covering ambiguities in Chinese word segmentation is an equivalent problem of word sense disambiguation. Consequently we make use of the vector space model to formulate the contexts of ambiguous words. A variation form of TFIDF weighting is proposed and a Chinese thesaurus is additionally utilized to cope with data sparseness problem. We selected 90 high frequency covering ambiguities as our target. The experimental results show that our model has achieved 96.58% accuracy, outperforming the original form of TFIDF weighting. The significant difference between the performance of our model and that of the baseline model, i.e., the hidden Markov model, which is based on adjacent syntactic information in nature, provides additional evidence that our assumption, i.e., an WSD-like approach to covering ambiguity resolution in terms of semantic information, is workable.

The proposed model here may serve as a general method for covering ambiguity resolution in Chinese word segmentation. Future work includes further validating and refining the model by enlarging the size and scopes of experimentations.

Acknowledgements

This research is supported by the National Plan for the Development of Key and Basic Research of China under grant no G1998030507 and the National Natural Science Foundation of China under grant no 60083005.

References

Ide N. and Veronis J. (1991) *Introduction to the special issue on word sense disambiguation: The state of the art*. Computational Linguistics, 24/1, pp. 1-4.

Gale W., Church K. and Yarowsky D. (1993) *A method for disambiguating word senses in a large corpus*. Computers and the Humanities, 26/1-2, pp. 415-439.

Liang N.Y. (1987) *CDWS: A word segmentation system for written Chinese texts*. Journal of Chinese Information Processing, 2, pp. 44-52.

Liu K. Y. (1997) *On evaluation techniques for word segmentation of contemporary Chinese*. Applied Linguistics (Beijing), 1, pp. 101-106.

Liu K. Y. (2000) *Word segmentation and part-of-speech tagging for Chinese texts*. The Commercial Press, Beijing.

Manning C. D. And Schütze H. (2000) *Foundations of Statistical Natural Language Processing*. MIT press.

Mei J. J. et al. (1983) *TongYiCiCiLin (A Chinese Thesaurus)*. Shanghai Cishu Press, Shanghai.

Nie J. Y., Gao J. F., Zhang J. and Zhou M. (2000) *On the use of words and N-grams for Chinese information retrieval*. Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages, Hong Kong.

Salton G. (1991) *Developments in automatic text retrieval*. Science, 253/5023, pp. 974-980.

Salton G. and Buckley C. (1988) *Term weighting approaches in automatic text retrieval*. Information Processing and Management, 24/5, pp. 513-523.

Sun M.S., Shen D.Y. and Huang C.N. (1997) *CSeg&Tag1.0: A practical word segmenter and POS tagger for Chinese texts*. Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington D.C., pp. 119-126.

Xiao Y., Sun M. S. and Benjamin K Tsou (2001) *Preliminary study on resolving covering ambiguities in Chinese word segmentation by contextual information in vector space model.* Computer Engineering and Application (Beijing), 37/19, pp. 87-89.

Yarowsky D. (1992) *Word sense disambiguation using statistical models of Roget's categories trained on large corpora*, Proceedings of COLING-92, Nantes, France, pp. 454-460.

Zheng J. H. and Wu F. F. (1999) *Study on segmentation of ambiguous phrases with the combinatorial type.* Collections of papers on Computational Linguistics, Tsinghua University Press, Beijing. pp. 129-134.