Chinese Abbreviation-Definition Identification: A SVM Approach Using Context Information*

Xu Sun, Houfeng Wang, and Yu Zhang

Department of Computer Science and Technology
School of Electronic Engineering and Computer Science
Peking University, Beijing, 100871, China
sunxu@pku.edu.cn, wanghf@pku.edu.cn, sdrzy@pku.edu.cn

Abstract. As a special form of unknown words, Chinese abbreviations represent significant problems for Chinese text processing. The goal of this study is to automatically find the definition for a Chinese abbreviation in the context where both the abbreviation and its definition occur, enforcing the constraint of one sense per discourse for an abbreviation. First, the candidate abbreviation-definition pairs are collected, and then a SVM approach using context information is employed to classify candidate abbreviation-definition pairs so that the pairs can be identified. The performance of the approach is evaluated on a manually annotated test corpus, and is also compared with two other machine learning approaches: Maximum Entropy and Decision Tree. Experimental results show that our approach reaches a good performance.

1 Introduction

The presence of unknown words makes the natural language text not always an ideal material for text processing applications. By unknown words, we mean words and symbols that are not listed in our system dictionary, including proper nouns, abbreviations, misspelled words, etc. For modern Chinese language, a large number of abbreviated words and their definitions are used interchangeably, in order to save space and give readers an intensive impression. Moreover, new abbreviations, such as '非典' for '非典型性肺炎' (Severe Atypical Respiratory Syndrome), are constantly being created, which justifies the need for software tools that can automatically identify abbreviation definitions in documents.

The goal of this study is to automatically find the abbreviation-definition[1] pairs, which is a special subtask for lexical cohesion analysis. It is also a key component for coreference resolution of named entities. The paper views abbreviation-definition identification as a classification task and a machine learning method using context information is applied to classify potential abbreviation-definition candidates.

* Supported by National Social Science Foundation of China (No. 05BYY043) and National Natural Science Foundation of China (No. 60473138).

[1] In this paper, the word *abbreviation* will always stand for *Chinese abbreviation* if there is no specific indication.

We define a Chinese abbreviation as a short representation of a multi-character word or more frequently a sequence of compounding words. E.g., the abbreviation '高校' is from '高等/学校' (Institutions of Higher Education), which contains two words. Chang (Chang et al., 2004) excluded two abbreviation formation regularities:

(1) Mapping substring to null string E.g., '欧洲/经济/与/货币/联盟' (European Economic and Monetary Union) to '欧盟'.

(2) Changing character order E.g., '第一/核能/发电厂' (First Nuclear Power Plant) to '核一厂', as illustrated in Fig.1.

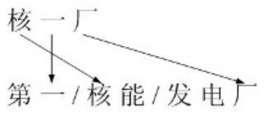

Fig. 1. An example of disordered abbreviation formation

However, this two regularities cover more than 20 percents of factual abbreviations. Thus in our study we made the following assumptions about abbreviations:

(1) An abbreviation includes fewer characters than its original definition, and an original definition can be a word or a phrase containing more than one word.

(2) The characters used in an abbreviation are a subset of those appearing in the original definition, but the character order can be different.

(3) We further define an abbreviation as a word that neither its whole sequence nor its substring is registered in our system vocabulary (otherwise it will be segmented during the word segmentation process). Though actually there are abbreviations containing 'real' words, our experiment performed on an abbreviation set containing 5121 entries shows that this kind of abbreviation is few (about 4%).

The previous related work of Chinese abbreviation expansion is relatively few. Chang et al. presented a HMM based error-recovery approach for abbreviation identification and root word recovery (Chang et al., 2004). Compared with the generative model introduced by Chang et al., the SVM approach employed in our system directly models the posterior probability of abbreviating and can get more flexibility to combine extra information, e.g., the similarity between the context of an abbreviation and that of its expansion candidate can be treated as an additional feature and then be directly combined into the SVM model.

This paper is organized as follows: Next section describes the system architecture. Section 3 and section 4 present the candidate search process and the feature set respectively. The remainder of this paper is experimental results and our conclusion.

2 System Description

Our abbreviation-definition identification system is made up of candidate search and abbreviation-definition disambiguation. The candidate search component is designed to collect both the abbreviation candidates and then the potential definition candidates, thus two sub-systems are employed: abbreviation candidate collector and definition candidate collector. The next task is to determine the most appropriate definition candidate for each given abbreviation, or it may find that none of the definition candidates is appropriate for the abbreviation candidate, that is, the system decides that the abbreviation candidate is not a real abbreviation. A SVM based machine learning method is used in the disambiguation process. The detailed descriptions of candidate search and abbreviation-definition disambiguation are respectively presented in section 3 and section 4.

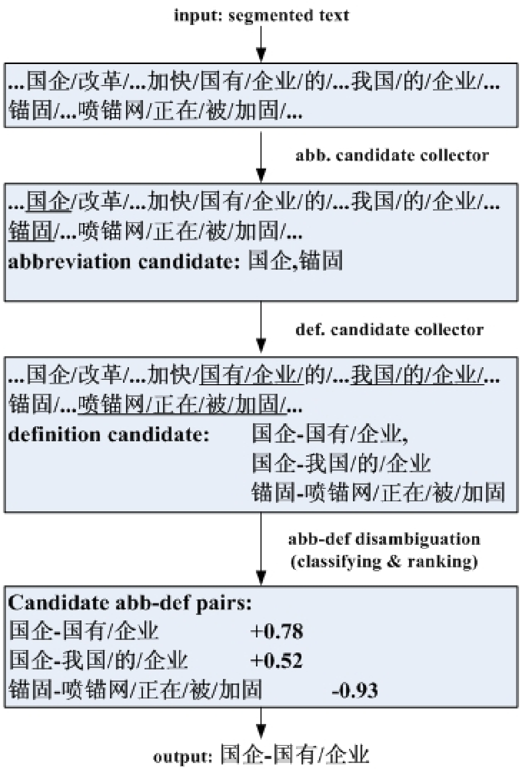
input: segmented text
...国企/改革/...加快/国有/企业/的/...我国/的/企业/... 锚固/...喷锚网/正在/被/加固/...
abb. candidate collector
...国企/改革/...加快/国有/企业/的/...我国/的/企业/... 锚固/...喷锚网/正在/被/加固/...
abbreviation candidate: 国企,锚固
def. candidate collector
...国企/改革/...加快/国有/企业/的/...我国/的/企业/... 锚固/...喷锚网/正在/被/加固/...
definition candidate: 国企-国有/企业, 国企-我国/的/企业 锚固-喷锚网/正在/被/加固
abb-def disambiguation (classifying & ranking)
Candidate abb-def pairs:
国企-国有/企业 +0.78
国企-我国/的/企业 +0.52
锚固-喷锚网/正在/被/加固 -0.93
output: 国企-国有/企业

Fig. 2. System overview

The abbreviation-definition disambiguation component is the core of the system, in that it determines both the final outcome of abbreviation-identification task and definition-identification task (abbreviation-identification determines whether or not an abbreviation candidate is a real abbreviation, while definition-identification finds the corresponding definition for a given abbreviation). In this way, the tasks of abbreviation-identification and definition-identification are not isolated in the system architecture.

The system overview is illustrated in fig. 2. As we can see, two abbreviation candidates ‘国企’ and ‘锚固’ are collected from the source text (where only ‘国企’ is a real abbreviation and ‘锚固’ is not). Thereafter, corresponding definition candidates are collected from context, and three abbreviation-definition candidate pairs are formed: ‘国企—国有/企业’, ‘国企—我国/的/企业’ and ‘锚固—喷锚网/正在/被/加固’.

The three candidate pairs are classified and ranked by the abbreviation-definition disambiguation component, and only the abbreviation-definition pair '国企—国有/企业' is outputted as 'abbreviating'. Although the candidate pair '国企—我国/的/企业' is classified as POSITIVE (abbreviating), it is discarded in that it shares the same abbreviation of another candidate pair '国企—国有/企业' and only the one with the highest score will be considered as the final abbreviation expansion result in our system. On the other hand, the candidate pair '锚固—喷锚网/正在/被/加固' is classified as NEGATIVE (non-abbreviating), and thus is abandoned.

3 Candidate Search

Both of abbreviation candidate collector and definition candidate collector are based on segmented Chinese text. Based on assumption (3) in Section 1, unknown character sequences and their substrings will be considered as potential abbreviation candidates.

When an OOV (out of vocabulary) word appears, it can also be a proper noun, a misspelled word, or a number rather than an abbreviation, thus in order to refine the abbreviation candidates, we implemented heuristics to exclude noise. Based on the observation that the number of single-character abbreviations (e.g., '法' for '<u>法</u>国') is much less than multi-character abbreviations and that it is possible to be enumerated, we use a single-character abbreviation list which is automatically collected from training corpus as a portion of our abbreviation candidate search heuristics. Besides, most of human names and numbers can be successfully identified and be excluded (a Chinese family name list is used to identify Chinese person names, and a transliterated name character list is used to identify transliterations of foreign names). However, inevitably, noise can still remain, and the final decision of abbreviation identification will be made by abbreviation-definition disambiguation, which is beyond the scope of candidate search and thus it will be presented in next section.

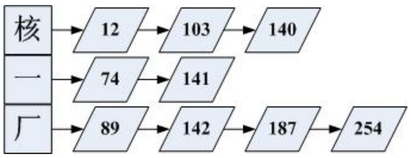

Fig. 3. An illustration of character based inverted file index

After abbreviation candidates are collected, the next task is to collect definition candidates for each abbreviation from the entire local document rather than only from local sentence or paragraph. Based on assumption (1) and (2) in section 1, we implement an inverted file index based algorithm to fulfill the collection task. Fig. 3 illustrates an inverted file index example for abbreviation '核一厂', where each abbreviation character is linked with its offsets in the text. An abbreviation-definition matching algorithm shown in Fig. 4 is used to collect definition candidates, where DIST is a threshold indicating the tolerable distance (number of characters) between adjacent characters (which are from given abbreviation) in the definition candidates, and it allows a trade-off between efficiency and quality. In our system, DIST is set as 6.

```
INPUT:  an abbreviation A, its inverted-index IDX, and local document D
WHILE  IDX is not completely searched
    TEMP:=GetAnOffsetSequence(IDX)
    TEMP:=Sort(TEMP)
    IF  MaxAjacentDistence(TEMP)<=DIST
        OFFSET_MIN:= GetMin(TEMP)
        OFFSET_MAX:=GetMax(TEMP)
        DEF_CAND:=GetString(D, OFFSET_MIN,OFFSET_MAX)
        IF  Length(DEF_CAND)>Length(A)
            DefCandCollection.Add(DEF_CAND)
OUTPUT:  DefCandCollection
```

Fig. 4. The abbreviation-definition matching algorithm

4 Abbreviation-Definition Disambiguation

We use SVM to disambiguate abbreviation-definition pairs. The key to the classification is to select appropriate features. We try to employ those features that effectively capture the distinction of the AC-DC pairs (AC indicates an abbreviation candidate, and DC indicates a corresponding definition candidate). Several features, including 'context similarity', are inspired by previous studys (Zahariev, 2004; Akira et al., 2001; Toole, 2000), while most of others such as 'position relation' and 'distance' are derived from our investigation of abbreviation formation regularities. The following features are considered in our system, including local features (without context information) and global features (using context information):

4.1 Local Features

(1) Difference in Length. This feature specifies the difference between the number of characters within AC and the number of words within DC. The value of this feature depends on the number of words in DC which maps to null string in AC.
(2) Final character. For the abbreviations of organization names, the final character contains rich information. E.g., the '院' (institute/academy) of both abbreviation '中科院' and its definition '中国/科学/院' (Chinese Academy of Sciences) is usually a symbolic character for an organization name.
(3) Character selection. In many cases, the first character of a word in the definition tend to be selected into the its corresponding abbreviation. E.g., '北京/大学' (Peking University) to '北大'. This feature records the position information of characters.
(4) Word order. This feature specifies whether or not the word order has been changed during abbreviating. In most cases, the word order will be kept the same.
(5) Frequency selection. This feature compares the universal frequency between selected characters and non-selected characters (obviously there are both selected characters and neglected ones within the DC, because AC is shorter than DC.).

4.2 Global Features

(1) w_{+1} and w_{-1} of AC. A large part of abbreviations are from named entities, and their contextual words have its own trait. This feature is used to record the left/right word of AC and to investigate the above mentioned contextual trait.
(2) Length of AC's w_{+1} and w_{-1}. This feature specifies the length of AC's contextual words.
(3) w_{+1} and w_{-1} of DC. From an identical motivation form global feature (1), this feature is used to record the left/right word of DC.
(4) Length of DC's w_{+1} and w_{-1}. This feature specifies the length of DC's contextual words.
(5) Frequency of AC. This feature represents the frequency of AC in local document. To deal with the sparseness, only 4 values are defined: '1', '2-3', '4-5' and 'MORE THAN 5'.
(6) Frequency of DC. This feature represents the frequency of DC in local document.
(7) Position relation. This feature identifies the position relation between AC and its DC. Generally speaking, if an abbreviation and its definition cooccur in a document, it would be more probable that the definition precedes the abbreviation.
(8) Context similarity. This feature estimates the similarity between AC context and DC context. Words appearing before and after an AC/DC are collected as context information. We use a window of size 5, and the window size need to be adjusted when sentence boundary is met. To measure the vector similarity, we use the cosine metric (Akira et al., 2001).
(9) Distance. This feature is used to represent the distance (number of words) between AC and its DC. An AC usually has more than one corresponding DC, and in general, the DC with minimum distance is more probable to become the definition.

5 Evaluation

Our experiment data comes from the People's Daily corpus. The selected data contains 361479 sentences from 9117 documents, which is divided into two sets: one from 6078 documents, for training; and the other from 3039 documents, for testing. The annotated corpus is required both for training and testing, and we annotated the abbreviation-definition pairs in three steps: first, annotate abbreviations manually. Second, generate abbreviation-definition candidate pairs using the abbreviation-definition matching algorithm, which is described in section 3. Third, pick out the correct abbreviation-definition pairs manually.

For some abbreviations, there might be more than one correct definition. E.g., '人大'can respectively be abbreviated from both '人民/代表/大会' (National People's Congress) and '中国/人民/大学' (Renmin University of China). However, this multi-sense case seldom occurs in a single document so that we enforce the constraint of one sense per discourse for each abbreviation.

5.1 Experiment

To evaluate the system, we use precision, recall, and F-measure. They are respectively defined as follows (Akira et al., 2001):

$$P = \frac{w}{w + x + y} \tag{1}$$

$$R = \frac{w}{w + x + z} \tag{2}$$

$$F = \frac{2 * P * R}{P + R} \tag{3}$$

where w is the number of abbreviations expanded correctly; x is the number of abbreviations expanded incorrectly; y is the number of abbreviation candidates that were not abbreviations but were incorrectly expanded by the system; and z is number of abbreviations not detected as abbreviations by the system[2].

In the SVM approach, data consisting of two categories is classified by dividing space with a hyperplane. For extended versions of the method, in general, the inner region of the margin in the training data can include a small number of examples, and the linearity of the hyperplane is changed to non-linearity by using kernel functions. (Vapnik, 1995; Cristianini et al., 2000)

During the tuning of the SVM model[3], we select a linear function as the kernel function according to the experimental statistics, a part of which is shown in Table 1. It is interesting to note that the linear kernel function outperforms the radial basis kernel as well as the polynomial kernels, with the final F-measure of 84.3%. One possible reason might be the 'over fit' problem of training for the radial basis and polynomial kernels. On the other hand, the linear kernel function is extremely efficient both in training and classifying, which can also be noticed in Table 1.

In order to deal with data sparseness problem, we discard the features occurring only once in the training data.

Table 1. Experimental results upon a variety of SVM kernel functions[4]

Kernel Functions		P (%)	R (%)	F (%)	T-secs	C-secs
Linear Function		87.2	81.5	84.3	2.7	0.1
Radial Basis Function		88.8	76.6	82.3	26.2	8.0
Polynomial Functions	$(\mathbf{x} \bullet \mathbf{y} + 1)^2$	85.0	82.1	83.5	20.9	5.0
	$(\mathbf{x} \bullet \mathbf{y} + 1)^3$	77.7	87.8	82.5	19.3	6.4

[2] Some abbreviations do not have corresponding definitions in local document, they would not be counted into z.

[3] We used the software SVM $^{\text{light}}$ (T. Joachims, 1999) developed by T. Joachims as the support vector machine.

[4] Where the *T-secs* is an indicator for *CPU seconds of training*, and *C-secs* stands for *CPU seconds of classifying*. Both the training and classifying are performed on a CPU of 1.6G HZ.

5.2 Comparison to Other Methods

To assess the performance of our SVM model, we implemented two other machine learning methods, which is described as follows:

(1) Decision Tree. The decision tree paradigm constructs classifiers by dividing the data set into smaller and more uniform groups, based on a measure of disparity which usually is entropy. The best choice of variable and threshold is the one that minimizes the disparity measures in the resulting groups. We used the C4.5 decision tree toolkit (http://www.rulequest.com) for our simulations.
(2) Maximum Entropy. The main idea of maximum entropy model (MEM) is to select probability distribution with highest entropy value based on the constraints. Highest entropy distribution gives the model minimum prejudice and maximum impartiality subject to the constraints, because any other selection would bring extra reasonless assumptions. We use the Generalized Iterative Scaling (GIS) algorithm to train feature weights and an additional Exponential Smoothing algorithm (Joshua Goodman, 2004) to penalize large weights.

The experimental results are shown in Table 2. For our abbreviation expansion task, the performance order of these methods is as follows:

F-measure: SVM > MEM > DT
Precision: DT > SVM > MEM
Recall: MEM > SVM >DT

Since they use exactly the same feature set, the comparison was strict. The number of abbreviation candidates that were incorrectly expanded is much less in Decision Tree method, thus it achieved the highest Precision of 88.8%. On the other hand, Maximum Entropy method achieved the highest Recall of 83.9%, with a dramatic increase of 168 on correct abbreviation expansions. With the F-measure of 84.3%, Support Vector Machine method presented a good compromise between Precision and Recall.

Table 2. Experimental results upon a variety of methods

Method	w	P (%)	R (%)	F (%)
Decision Tree	1467	88.8	75.6	81.7
Maximum Entropy	1627	83.0	83.9	83.5
Support Vector Machine	1580	87.2	81.5	84.3

Compared with the DT method, the improvement of MEM and SVM framework comes from their ability to more robustly combine features that do not form partitions of the event space, but instead overlap in arbitrarily complex ways. Compared with the MEM approach, the performance of the SVM approach is slightly superior within our abbreviation-definition disambiguation task, and the slight superiority might relate to our specific feature set. In some other fields of NLP (natural language processing) such as WSD (word sense disambiguation), similar comparison result between the SVM approach and the MEM approach can be observed as well (Wu et al., 2004).

We also test how the size of training data affected our experimental results. Fig. 5 shows that the difference of the F-measure rates between the MEM and the SVM was larger when the size of the training data was decreased especially from 2/15 data-size to 1/15 data-size. This indicates that when the size of the training data is smaller, the SVM approach has a more stable performance on our abbreviation-definition disambiguation feature set. On the other hand, the curved line of DT goes down rapidly when the size of the training data is increased from 1/15 data-size to 1/5 data-size, which is abnormal. It indicates that the performance of decision tree method on our feature set is volatile in a relatively small size of training data.

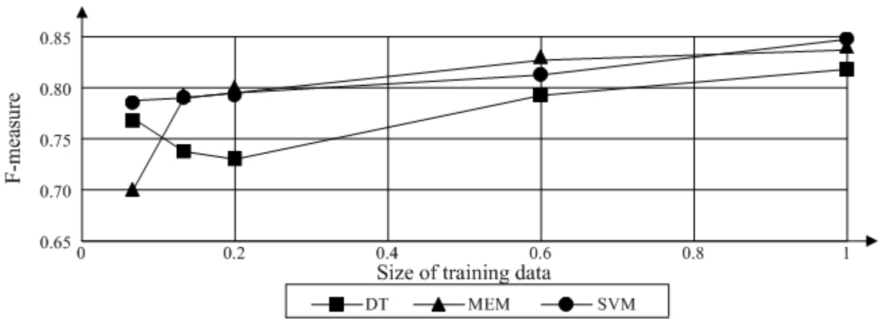

Fig. 5. Experimental results when changing the size of the training data

Chang presented a HMM based error-recovery model (Chang et al., 2004) for abbreviation identification and root word recovery, which is introduced in section 1. In Chang's study, the performance of guessing definitions from abbreviations is about 51%. However, it needs to be taken into account that Chang uses a different corpus.

6 Conclusion and Future Work

In this paper, we proposed a SVM based framework for automatic abbreviation-definition identification in Chinese text. It reaches an encouraging performance according to the experimental result upon the People's Daily Corpus. Moreover, using exactly the same feature set, additional experiments show that SVM slightly outperforms MEM and DT methods, and it is more effective and stable when the size of training data is small.

Feature design is a crucial step of supervised abbreviation-definition identification. Experimental result indicates that our feature set utilizing context information has discriminative power. For the portability concern, only general characteristics of context information are employed in our system.

Our future work will consist in abbreviation expansion using supervised learning method for those abbreviations whose corresponding definitions are not in context.

Acknowledgments

We would like to thank Galen Andrew, Wuguang Shi and Yun Xing for helpful comments on earlier versions of this paper.

References

1. Vladimir N. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.
2. T. Joachims. Making large-scale support vector machine learning practical. In Scholkopf et al., 1999, pages 169-184.
3. Nello Cristianini and John Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.
4. Jin-Shin Chang and Yu-Tso Lai. A Preliminary Study on Probabilistic Models for Chinese Abbreviations. ACL, 2004.
5. Zahariev, M. A (Acronyms). Ph.D. thesis, Simon Fraser University, 2004.
6. Akira Terada, Takenobu Tokunaga and Hozumi Tanaka. Automatic expansion of abbreviations by using context and character information. In Proc. of the Sixth Natural Language Processing Pacific Rim Symposium, 2001.
7. Joshua Goodman. Exponential Priors for Maximum Entropy Models. In Proc. of HLT-NAACL, 2004, pages 305-312.
8. Toole Janine. A hybrid approach to the identification and expansion of abbreviations. In Proc. of RIAO, 2000, pages 725–736.
9. Dekai Wu, Weifeng Su, and Marine Carpuat. A Kernel PCA method for superior word sense disambiguation. In Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics, 2004.