Chinese Abbreviation Identification Using Abbreviation-Template Features and Context Information*

Xu Sun and Houfeng Wang

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

Abstract. Chinese abbreviations are frequently used without being defined, which has brought much difficulty into NLP. In this study, the definition-independent abbreviation identification problem is proposed and resolved as a classification task in which abbreviation candidates are classified as either 'abbreviation' or 'non-abbreviation' according to the posterior probability. To meet our aim of identifying new abbreviations from existing ones, our solution is to add generalization capability to the abbreviation lexicon by replacing words with word classes and therefore create abbreviation-templates. By utilizing abbreviation-template features as well as context information, a SVM model is employed as the classifier. The evaluation on a raw Chinese corpus obtains an encouraging performance. Our experiments further demonstrate the improvement after integrating with morphological analysis, substring analysis and person name identification.

1 Introduction and Background

As a special form of unknown words, Chinese abbreviations are frequently used without being defined, which has brought much difficulty into natural language processing (NLP), especially for agglutinative languages such as Chinese, in that the problem is exacerbated by the lack of word boundaries. How to identify abbreviations[1] becomes a common problem within Chinese word segmentation, Chinese co-reference resolution, Chinese named-entity (NE) recognition, etc. Take for instance the NE recognition in which a large part of target NEs are abbreviated within the source texts, it is of necessity to retrieve NEs from those abbreviations. As a precondition for this task, however, there lies the above-mentioned more basic problem: How to retrieve abbreviations within agglutinative Chinese texts?

* Supported by National Social Science Foundation of China (No. 05BYY043) and National Natural Science Foundation of China (No. 60473138, No. 60675035).

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

Y. Matsumoto et al. (Eds.): ICCPOL 2006, LNAI 4285, pp. 245 – 255, 2006.

To a large extent, the success of identifying English abbreviations[2] goes to two aspects: First, most of the English abbreviations contain uppercase letters (e.g., 'DNA'). Second, lots of English abbreviations are marked by parentheses, namely the pattern 'abbreviation (definition)' or 'definition (abbreviation)'. Taghva (1999), Yeates (1999), and Byrd (2001) utilized the uppercase information for their abbreviation acquisition, while Schwartz (2003), Chang (2002), and Zahariev (2004) suggested the parentheses. Unfortunately, for Chinese texts, there is no 'uppercase', and very few abbreviations are explicitly marked by parentheses. Additionally, in Chinese this issue is exacerbated by the ambiguity of word boundaries. Thereby, it is of extraordinary difficulty to extend the abbreviation identification techniques from English to Chinese.

The literature on Chinese abbreviation identification is relatively small. Sproat (2002) and Sun (2002) introduced heuristics for this study. Such heuristics, however, can easily break. Of the more recent research in the area, important work is that of Chang (2004), who presented a hidden Markov model (HMM) based approach for abbreviation identification. In the experiment to guess the abbreviations from given definitions, the accuracy rate is 72%. Yet we can see that the definition information is still of necessity in the task.

Hence, in this study, our motivation is to investigate a definition-independent approach so that the abbreviations can still be identified in texts where the availability of definitions is not guaranteed. Instead of relying on abbreviation-definition mapping, which is a typical technique being vastly used in previous definition-required studies, we add generalization capability to the abbreviation lexicon by replacing words with word classes and therefore create abbreviation-templates to meet our aim of identifying new abbreviations from existing ones.

As has been mentioned, automatic abbreviation identification is a key component in systems that handle the various extended tasks, such as automatic abbreviation expansion, co-reference resolution, named-entity recognition or automatic query expansion. The extended tasks, however, are not the main focus of this paper.

The rest of this paper is organized as follows. In next section, the system architecture is described. In section 3 and section 4, abbreviation disambiguation techniques and improvement solutions are respectively presented. The remainder of this paper is experiment results and conclusion.

2 System Overview

In our study, word segmentation and abbreviation identification are integrated into a unified framework in order to automatically extract abbreviations from raw text. As described in Gao (2003), we define Chinese words in this paper as one of the following four types as well: (1) entries in a lexicon (lexicon words below), (2) morphologically derived words, (3) factoids, and (4) named entities, because these four types of words have different functionalities in Chinese language processing, and are processed in different ways in our system.

[2] Here, *English abbreviation* is a general denotation, containing *acronym*.

In this paper, we made the following assumptions about abbreviations:

(1) The abbreviation length should be between two and five characters. The single-character abbreviations will not be considered in our system, because the number of single-character abbreviations (e.g., '法' for '法国') is much less than multi-character abbreviations and that it is possible to be enumerated, so that the their identification can be resolved by simply using a single-abbreviation list. On the other hand, the abbreviation containing more than five characters is very rare (less than 0.1%).

(2) An abbreviation contains no lexicon words (otherwise it will be segmented during word segmentation). Though actually there are abbreviations containing lexicon words, our experiment performed on an abbreviation set containing 5,121 entries shows that this kind of abbreviation is few (about 4%).

In our system, the abbreviation identification process contains three steps: (1) word segmentation and text normalization, (2) abbreviation candidate search, and (3) abbreviation disambiguation. A lexicon containing around 100K entries is employed for word segmentation, and during text normalization, the factoids are normalized and replaced with tags (e.g., 十二点三十分 '12:30' will be replaced by 'F_TIME'.). In our study, word segmentation is employed for two reasons: First, based on abbreviation assumption (2), abbreviation contains no lexicon words so that abbreviation candidates will be collected from unknown character sequences, and word segmentation is required for the retrieve of unknown sequences. Second, word segmentation is required for providing context information during abbreviation disambiguation.

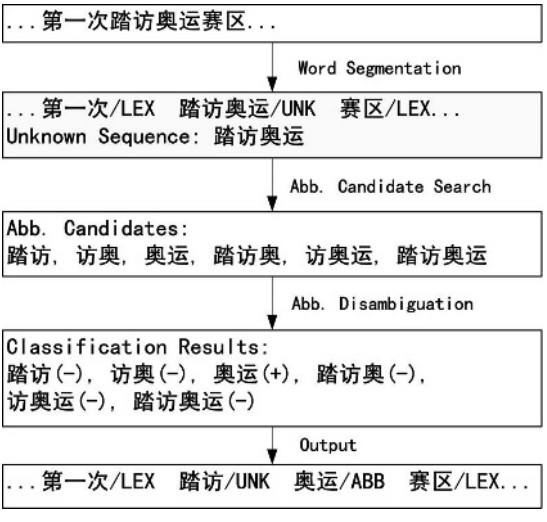

Fig. 1. Illustration of the overall system architecture

After unknown sequences are generated during word segmentation, searching abbreviation candidates is relatively simple. Based on abbreviation assumption (1), for

each substring within an unknown sequence, if it measures up to the length constraints, it will be collected as an abbreviation candidate.

An abbreviation candidate may also relate to (be a sub-sequence or super-sequence of) a named-entity, a misspelled word, etc, thus abbreviation disambiguation is crucial in our system. Since a range of weak evidence must be combined in order to make a judgment, statistical techniques are ideal for this environment. The details of abbreviation disambiguation will be presented in next section.

The system overview is illustrated in Fig. 1. As we can see, '踏访奥运' is marked as an unknown sequence during word segmentation because it contains no lexicon word. Totally 6 abbreviation candidates are collected according to abbreviation assumption (1), thereafter, candidates are classified by the abbreviation disambiguation model, and eventually only '奥运' (abbreviated from 奥林匹克运动会 'Olympic Games') is classified as 'abbreviation'.

3 Abbreviation Disambiguation

We use the SVM (support vector machines) model to disambiguate abbreviation candidates. The key to the classification is to select discriminative features that effectively capture the distinction between 'abbreviation' and 'non-abbreviation'. Based on our own investigation of the abbreviations, two groups of features are employed: abbreviation formation analysis (the conceptual formation of abbreviations are modeled by using a class-based language model) and context information analysis.

3.1 Abbreviation Formation Analysis

The abbreviation formation analysis is based on the assumption that a Chinese abbreviation is generated as follows: First, a person chooses a sequence of concepts to set up the concept structure of the abbreviation; then the person attempts to express each concept by choosing characters. We assume that different abbreviations may share a common concept structure. E.g., 上影厂 'Shanghai-Film-Studio' and 北工大 'Beijing-Technology-University'[3] is generated from the same concept structure 'location + category/industry + entity-postfix'. Therefore, although new abbreviations are constantly being created, their concept structure may remain the same. In our system, a concept is modeled by using a word class (in most cases, a word is represented as a character in the abbreviations, so in this paper it can be also called 'abbreviated-word class' or simply 'character class').

Word Class of Abbreviations

An efficient method for word clustering has been introduced in Och (1999) for machine translation, and its main idea is adopted for determining word classes in our study. We use a statistical language model to estimate the probability $P(w_1^N)$ of the

[3] The formal name is 'Beijing University of Technology'.

word sequence $w_1^N = w_1 ... w_N$ of an abbreviation. A simple approximation of $P(w_1^N)$ is to model it as a product of bigram probabilities $P(w_1^N) = \prod_{i=1}^{N} P(w_i \mid w_{i-1})$. Rewriting the probability using classes we arrive at the following probability model:

$$P(w_1^N \mid C) := \prod_{i=1}^{N} P(C(w_i) \mid C(w_{i-1})) \bullet P(w_i \mid C(w_i)) \tag{1}$$

where the function *C* maps words w to their class *C(w)*. In this model, we have two types of probabilities: the transition probability *P(C|C')* for class *C* given its predecessor class *C'* , and the membership probability *P(w|C)* for word *w* given class *C*. To determine the optimal word classes *C* for a given abbreviation corpus (a manually collected abbreviation set containing 5,121 entries is used as the training data), we perform a maximum-likelihood estimation:

$$C_{opt} = \arg\max_{C} P(w_1^N \mid C) \tag{2}$$

During the implementation, an efficient optimization algorithm for word clustering is the exchange algorithm (Martin et al., 1998). It is necessary to fix the number of classes in *C* in advance as the optimum is reached if every word is a class of its own. Considering the size of our training data, the default number of classes is set as 30 in our system.

Here two resulting word classes are selected for illustration. The first one is '班, 办, 部, 场, 厂, 池, 处, 局, 具, 圈, 势, 室, 署, 厅, 系, 校, 源, 院, 站'. As we can see, most of the entries are 'entity-postfix', which is frequently used as the last character in the formation of abbreviations. '埃, 北, 东, 低, 南, 西, 朝, 成, 川, 滇, 韩, 京, 兰, 黎, 柳, 闽, 欧, 葡, 深, 沈, 蜀, 湘, 亚, 燕, 粤, 云, 浙, 中' gives another example of our resulting word classes. As illustrated, most of these entries are 'location name'.

Abbreviation Formation Features

Three features are used for abbreviation formation analysis:

Abbreviation formation score: For an abbreviation candidate $w_1^N = w_1 ... w_N$, its abbreviation formation score will be estimated by using:

$$P(w_1^N \mid C_{abb}) := \prod_{i=1}^{N} P(C_{abb}(w_i) \mid C_{abb}(w_{i-1}), C_{abb}(w_{i-2})) \bullet P(w_i \mid C_{abb}(w_i)) \tag{3}$$

Where C_{abb} is the partition of word classes trained from the above-mentioned abbreviation corpus. Trigram probabilities are used to estimate the transition probability (while only bigram probabilities are used for determining word classes in Eq. (1), and the difference comes from different compromise between efficiency and exactness). The trigram probabilities can be calculated by training on the same abbreviation set, and to further deal with the data sparseness, we use a standard backing off schema (Katz, 1987).

Word penalty: This feature counts the length in words of the target abbreviation candidate to balance the abbreviation formation score. Without this feature, the final abbreviation produced tends to be too short, because shorter items tend to get higher probability based on the language model.

Numeric information: Although factoids have been normalized during preprocessing, there are still many abbreviation candidates containing numeric characters. Most of these candidates are noise, while some of them are not (e.g., 二战 'the Second World War'). This feature records the number of numeric characters inside a candidate as well as their character position: 'BEGINNING', 'MIDDLE', or 'END'.

3.2 Context Information Analysis

In our study, two features of context information are adopted to improve abbreviation identification:

Contextual words: A large part of abbreviations are from named entities, and their contextual words have its own traits. This feature is used to record the left/right word and their corresponding length (number of characters) surrounding the abbreviation candidate. Note that the contextual words are ambiguous when both sides of this abbreviation candidate are unknown sequence. In such a case, this feature will not be chosen.

Frequency feature: This feature counts the occurrence of abbreviation candidate in the local document. It is used to discriminate abbreviation from random noise, in that in many cases such noise will occur only once on the document level.

4 Improvement

The abbreviation disambiguation model introduced in section 3 is selected as our baseline model, in which the following deficiencies emerged during experimental evaluation (the experimental result will be shown in section 5): First, morphologically derived abbreviations are neglected. Second, experiments showed that some substrings of the abbreviations were mistakenly classified as 'abbreviation'. Third, we find lots of noise coming from named-entities, especially person names (PN). Then, we will provide solutions to the three problems respectively.

4.1 Morphological Analysis

As described in Gao (2003), the morphologically derived words are generated using 4 morphological patterns: (1) affixation: 朋友<u>们</u> (friend - <u>plural</u>) 'friends'; (2) head particle(i.e. expressions that are verb+comp): 走 'walk' + 出去 'out' ->走出去 'walk out'; (3) reduplication: 高兴 'happy' -> 高高兴兴 'happily' ; and (4) merging: 上班 'on duty' + 下班 'off duty' ->上下班 'on-off duty'.

Due to the reason that Chinese morphological rules are not as 'general' as their English counterparts, it is difficult to simply extend the well-known techniques from English (i.e., finite-state morphology) to Chinese. We use two different methods to solve those four morphological patterns: For morphological pattern of affixation, head

particle and reduplication, we simply use the solution of extended lexicalization suggested in Gao (2003). On the other hand, morphological pattern of merging is a special form of abbreviating and is called 'morph-abbreviation'. The identification of 'morph-abbreviation' is integrated into our unified abbreviation identification model, and a new feature is employed:

Morph-abbreviation identification: For the target candidate with the consecutive character string of ABC, we first extend it to ACBC. This feature then returns 'TRUE' if both AC and BC are proved being lexicon words and 'FALSE' otherwise. In the case of returning 'TRUE', it is of large probability that ABC is a morph-abbreviation merged from two lexicon words, namely AC and BC.

4.2 Substring Analysis

Examination shows that some substrings of the abbreviations are incorrectly classified as 'abbreviation'. Those errors come from the traits of the conceptual formation of abbreviations. For instance, the substring '影厂' of the abbreviation '北影厂' is sharing the same conceptual structure with another factual abbreviation '师大': 'category/industry + entity-postfix'. As a result, it is possible that the substring '影厂' will get a high score during abbreviation formation analysis. In order to address this issue, a 'super-sequence diversity feature' is developed based on the 'diversity' difference between a factual abbreviation and its substrings.

First, using an illustration we briefly define 'left-minimum super-sequence' (LMS) and 'right-minimum super-sequence' (RMS): In the consecutive sequence of ABCD, ABC is the LMS for BC and BCD is the RMS for BC. For the overall occurrences of an abbreviation in the local text, their LMSs and RMSs tend to be inconsistent. Yet for the overall occurrences of a substring, either their LMSs or RMSs tend to keep the same form, and vice versa, in that the substring itself is not an independent term for denotation. Therefore, it is possible to develop a rule combining LMSs and RMSs for discriminating real abbreviations from their substrings:

Super-sequence diversity feature: Formally, this feature is scored by using *Div(A)*, and larger value of *Div(A)* would represent a larger degree of this special form of diversity:

$$Div(A) \coloneqq \frac{type(LMS\_A) + type(RMS\_A)}{count(A)} \tag{4}$$

where the function *type(x)* represents the total kinds of inconsistent forms for *x*, and *count(x)* returns the overall occurrence-number of *x* on the document level.

4.3 Named-Entity Identification

We find that lots of noise comes from named entities, especially Chinese person names (CN) and transliterated foreign names (FN). The following heuristics are employed in our system to identify CNs and FNs inside the unknown sequences.

Chinese person names: As described in Gao (2003), Chinese PN consists of a family name F and a given name G, and is of the pattern F+G. Both F and G are of one or

two characters long. We only consider PN candidates that begin with an F stored in the family name list (which contains 297 entries in our system). High frequency used G character were also stored in a given name list.

Transliterated foreign names: As described in Sproat (1996), FNs are usually transliterated using Chinese character strings whose sequential pronunciation mimics the source language pronunciation of the name. Since FNs can be of any length and their original pronunciation is effectively unlimited, the recognition of such names is tricky. Fortunately, there are only a few hundred Chinese characters that are particularly common in transliterations. Therefore, an FN candidate would be generated if it contains only characters stored in a transliterated name character list (containing 472 entries).

It should be emphasized that there is no feature dimension increase during integrating named-entity identification: it is employed only for candidate pruning.

5 Evaluation

Our experiment data comes from the People's Daily corpus (http://icl.pku.edu.cn). The selected data contains 20,063 sentences from 4,769 documents, which are divided into two sets: one from 3,146 documents, for training; and the other from 1,623 documents, for testing. The number of abbreviation tokens in the testing corpus is 4,941. The abbreviation tokens within the corpus have been manually annotated. The original corpus is already segmented, and in order to get unsegmented raw corpus we have completely erased the segmentation marks. In practice, a lexicon containing around 100K entries is used to segment the raw corpus. This lexicon contains pure words and there is no additional tag to indicate extra information. To keep our training and testing outcomes justifiable, all abbreviations from experimental corpus are removed from the lexicon.

The SVM model is employed for abbreviation disambiguation. The commonly used SVM model is a machine learning paradigm based on statistical theory. The SVM model calculates separating hyperplanes that maximize the margin between two sets of data points. While the basic training algorithm can only construct linear separators, kernel functions can be used to calculate scalar products in higher dimensional spaces.

To evaluate the performance of our system, we use F-measure. Based on the precision P and the recall R, the F-measure is defined as follows:

$$F = \frac{2 * P * R}{P + R} \quad . \tag{5}$$

5.1 Comparison of Kernel Functions

During tuning the SVM model[4], 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

[4] We use the software SVM ^{light} (T. Joachims, 1999).

interesting to note that the linear kernel outperforms the Gaussian RBF kernel as well as the polynomial kernel, with the final F-measure of 73.7%. The reason might be the 'over fit' problem within the training of the RBF and polynomial kernels. Moreover, it should be emphasized that the linear kernel is efficient both in learning and classifying.

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

Table 1. Experimental results upon different SVM kernel functions[5]

Kernel Functions	P (%)	R (%)	F1 (%)	T-secs	C-secs
Linear Function	82.7	66.5	73.7	887	1
Radial Basis Function	83.2	65.3	73.2	16,550	517
Polynomial Function	82.4	66.1	73.4	14,863	473

5.2 Improvement Evaluation

We conducted incrementally the following four experiments:

(1) The SVM approach using features of abbreviation formation analysis together with context information analysis, which is selected as our baseline performance;
(2) Integrating the feature of morphological analysis (MA) into (1);
(3) Integrating the feature of substring analysis (SA) with (2);
(4) Integrating NE identification (NEI) with (3).

Both MA and SA will bring new features into our SVM classifier, while NEI is used only for candidate pruning. The details of incremental evaluation are shown in Table 2. As can be noticed, our baseline model reaches the F-measure of 64.0 %.

The integration of MA led to a slight better resulting F-measure. Primarily, the improvement results from the identification of morph-abbreviations. Unfortunately, we find that MA can sometimes make inaccurate identifications, which may undermine the improvement.

The system performance is also enhanced by the integration of the SA. However, this improvement is not as significant as our anticipation, the reason is that unfortunately some abbreviations occur only once throughout its context so that they can not be well discriminated from their substrings by the Super-sequence diversity feature employed by SA.

In our baseline model, the set of abbreviation candidates is large, and merely about 1/19 of them are real abbreviations. Thereby, the candidate pruning performed by NEI is crucial. In experiments, we found that by integrating the NEI, we not only achieved more efficient training and testing (in testing the abbreviation candidates are trimmed from 92,015 to 85,571 items), but also obtained significant higher F-measure. The

[5] *T-time* denotes *Training-time (s)*, and *C-time* denotes *Classifying-time (s)*. Experiments are performed on a 1.6G HZ CPU.

abbreviation recall rate increases from 55.5% to 66.5%. Its significant improvement is achieved through reducing the noise influence from NEs, in that the People's Daily Corpus is a news corpus, which tends to use a large number of Chinese person names and transliterated foreign names.

Table 2. The results of improvement evaluation

methods	Total Abbs	P (%)	R (%)	F1 (%)
Baseline	4,941	72.5	57.3	64.0
Baseline + MA		81.0	54.5	65.2
Baseline + MA + SA		81.4	55.5	66.0
Baseline + MA + SA + NEI		82.7	66.5	73.7

6 Conclusion and Future Work

In this paper, we proposed a supervised learning approach for automatic abbreviation identification in raw Chinese text. The definition-independent abbreviation identification problem is regarded as a classification problem in which an abbreviation candidates is classified into either 'abbreviation' or 'non-abbreviation' based on its posterior probability, and is integrated as part of a unified word segmentation model. The posterior probability is estimated by using abbreviation formation information and context information. It reaches an encouraging performance according to the experimental result upon the People's Daily Corpus.

Moreover, additional experiments further demonstrate the improvement after integrating with named entity identification (NEI), morphological analysis (MA) and substring analysis (SA). Morphological analysis helps the identification of morph-abbreviations, especially leads to a higher precision rate. The improvement of integrating SA is not as significant as our anticipation, because some abbreviations occur only once throughout its context so that the 'super-sequence diversity feature' becomes undiscriminating. On the other hand, significant improvement is obtained by integrating NEI.

In our future work, we will focus on fine-tuning our abbreviation formation model to further enhance its performance. Especially, we would like to investigate a more effective word clustering algorithm which enables a joint learning from both positive examples and negative examples, so that we can improve the quality of word classes and therefore generate more discriminating abbreviation-templates.

Acknowledgments

We would like to thank Galen Andrew for helpful suggestions on implementing word clustering algorithms and Sujian Li for helpful comments on earlier versions of this paper.

References

1. J.Chang, H.Schütze and R.Altman, Creating an online dictionary of abbreviations from MEDLINE, Journal of American Medical Information Association, 2002, 9(6), pp. 612-620.
2. Jianfeng Gao, Mu Li, and Changning Huang. Improved Source-channel Models for Chinese Word Segmentation. In Proceedings of the 41st Annual Meeting of Association for Computational Linguistics (ACL). July 8-10, 2003. Sapporo, Japan. pp. 272-279.
3. Jin-Shin Chang and Yu-Tso Lai. A Preliminary Study on Probabilistic Models for Chinese Abbreviations. In Proceedings of the Third SIGHAN Workshop on Chinese Language Learning, ACL, 2004, Barcelona, Spain, pp. 9-16.
4. Jian Sun, Jianfeng Gao, Lei Zhang, Ming Zhou and Chang-Ning Huang. Chinese Named Entity Identification Using Class-based Language Model. In Proc. of the 19th International Conference on Computational Linguistics, Taipei, 2002, pp. 967-973.
5. Katz, S.M. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE ASSP 35(3):400-401.
6. Och, Franz Josef. 1999. An efficient method for determining bilingual word classes. In EACL-99: Ninth Conference of the European Chapter of the Association for Computational Linguistics, pp. 71–76.
7. Richard Sproat, Chilin Shih, William Gale and Nancy Chang. 1996. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. Computational Linguistics. 22(3): 377-404.
8. Richard Sproat and Chilin Shih. 2002. Corpus-Based Methods in Chinese Morphology and Phonology. In: COLING-2002.
9. Schwartz, A. and Hearst, M. 2003. A simple algorithm for identifying abbreviation definitions in biomedical texts, Pacific Symposium on Biocomputing (PSB 2003), Kauai, Hawaii.
10. S.Martin, J.Liermann and H.Ney. 1998. Algorithms for Bigram and Trigram Word Clustering. Speech Communication, 24(1): 19-37, 1998.
11. Taghva, K. and Gilbreth, J. (1999), Recognizing acronyms and their definitions, International journal on Document Analysis and Recognition, pp. 191-198.
12. T. Joachims, Making large-Scale SVM Learning Practical. In: B. Schkopf and C. Burges and A. Smola (ed.), Advances in Kernel Methods - Support Vector Learning, MIT Press, 1999.
13. Yeates, S. (1999), Automatic extraction of acronyms from text. In Third New Zealand Computer Science Research Students' Conference, pp. 117-124.