Categorizing Unknown Text Segments for Information Extraction Using a Search Result Mining Approach

Chien-Chung Huang[1], Shui-Lung Chuang[1], and Lee-Feng Chien[1,2]

[1] Institute of Information Science, Academia Sinica, Taiwan
{villars,slchuang,lfchien}@iis.sinica.edu.tw
[2] Department of Information Management, National Taiwan University, Taiwan

Abstract. An advanced information extraction system requires an effective text categorization technique to categorize extracted facts (text segments) into a hierarchy of domain-specific topic categories. Text segments are often short and their categorization is quite different from conventional document categorization. This paper proposes a Web mining approach that exploits Web resources to categorize unknown text segments with limited manual intervention. The feasibility and wide adaptability of the proposed approach has been shown with extensive experiments on categorizing different kinds of text segments including domain-specific terms, named entities, and even paper titles into Yahoo!'s taxonomy trees.

1 Introduction

Many Information extraction (IE) systems extract important facts [8], such as people names and event titles, from documents. However, given nowadays sentence analysis technology, it is not easy to understand the semantics of such word strings in a non-controlled subject domain. To extract more information from these extracted facts, a possible solution is to categorize the extracted facts into a well-organized topic taxonomy, and, based on the categories assigned, to find out their more semantically-deep meaning.

When applying text categorization techniques to complex domains with many categories, extremely large quantities of training documents are often necessary to ensure reasonable categorization accuracy [4]. Creating these sets of labeled data is tedious and expensive, since typically they must be labeled by a person. This leads us to consider an alterative approach that requires not much manual effort. Combining Web mining technique and text categorization technique, the proposed approach is efficient and highly accurate in categorization, and, most important of all, it can be easily adapted to different tasks and thus can be employed to design more advanced IE systems.

For general applications, an important fact extracted by an IE system is defined as a text pattern, which is a meaningful word string containing a key concept of a certain subject domain. More specifically, text segments can be domain-specific terms, named entities, natural language queries, or even paper title. Our task is to categorize these extracted text segments into appropriate categories.

Conventional text categorization techniques are often utilized to analyze relationships among documents, while both the aim and the skill of text pattern categorization

K.-Y. Su et al. (Eds.): IJCNLP 2004, LNAI 3248, pp. 576–586, 2005.

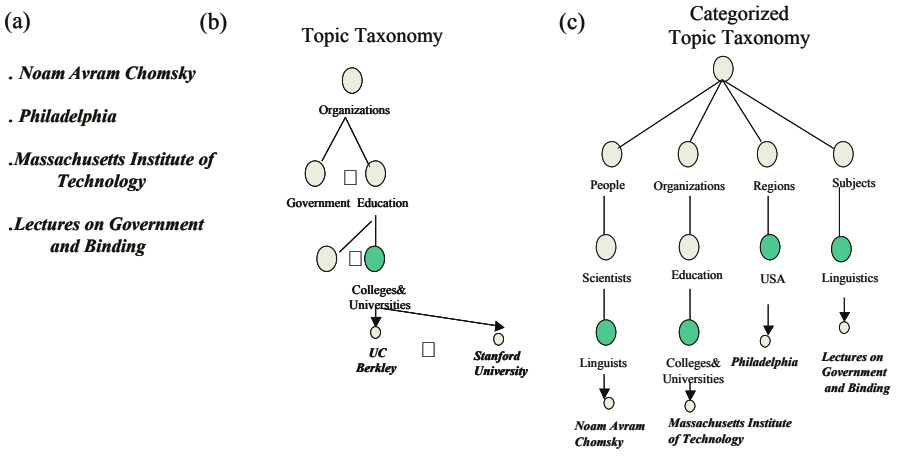

Fig. 1. A set of text segments and example topic taxonomy extracted from Yahoo!.

are quite different. Documents normally contain much more information than text segments. Therefore, the similarity between a document and a target class can be in some degree estimated with the difference of distribution of of words contained in the document itself and the training set of the class; on the contrary, the similarity between a short text pattern and a target class cannot be estimated in like manner. Obviously, if one wishes to categorize an unknown pattern, one has to use an external resource to give the unknown text pattern more features.

The Web is becoming the largest data repository in the world [11]. The rich Web corpus undoubtedly offers much possibilities for designing new applications. In this research, we intend to organize the text segments into a taxonomy, in which an overall topic structure of the extracted facts can be observed and deeper analysis performed. The taxonomy consists of two parts: keyword groups representing individual topic classes and a hierarchy of such topic classes. The topic taxonomy looks like a classification tree in which each node is associated with a keyword to represent a certain concept. Each non-leaf node (interior node) normally stands for a topic class, and its concept is composed of the concepts of its child classes in the hierarchy. For illustration, an example topic taxonomy extracted from Yahoo! is depicted in Figure 1, which includes (a) a set of example text segments, (b) a part of the topic taxonomy that can be used for categorizing the segments, and (c) the part of the taxonomy with segments categorized. From this figure, we could imagine that the concept of the Colleges & Universities class is composed of the concept of existing keywords such as “UC Berkeley” and “Stanford University.” Suppose there is an unknown text pattern “Massachusetts Institute of Technology”, it is expected that it is to be categorized into this class and offer more concept for it at the same time. For another example, the concept of the Education class is composed of the concept of the Colleges & Universities class along with the concepts of other sibling classes. Reserving the branches with segments categorized, a taxonomy tree as shown in (c) can provide another view for understating the unknown segments.

Normally, it is not too difficult for a human expert to construct a classification tree for a certain IE task, but there are unavoidably two challenge issues: (1) whether the constructed taxonomy is well-organized; (2) how to collect sufficient corpus to train a statistical model for each class. Our research is focused on dealing with the second issue. The proposed approach utilize real-world search engines to train our model. Our main idea is to employ highly ranked search result pages retrieved by the unknown segments and the keywords of the target classes as a source for feature extraction. This approach not only reduces manual labor but also supplements the insufficient information of unknown segments. The feasibility of the proposed approach has been shown with extensive experiments. We believe the proposed approach can serve as a basis toward the development of advanced Web information extraction systems. In the rest of this paper, we first review some related work and introduce the proposed approach; we then present the conducted experiments and their results; finally, we discuss some possible applications and draw conclusions.

2 Related Work

Word Clustering and Named Entity Identification. A number of approaches have been developed in computational linguistics for clustering functional-similar words and identifying named entities. These approaches relied on analysis of the considered objects' contextual information obtained from tagged corpus [3, 9]. Instead of using tagged corpus for categorizing word- or phrasal-level objects, the proposed approach is extended to fully exploit Web resources as a feature source to categorize text segments, which might be longer in length, into hierarchical topic classes. At the current stage of our research, we assume that the text segments are formed with a simple syntactic structure and containing some domains-specific or unknown words. Conventional syntactic sentence analysis might not be appropriate to be applied under such circumstances.

Complete grammatical sentence analysis is assumed inappropriate to be applied under this circumstance.

Web Mining. Our research is related to text mining [8], which concerns the discovery of knowledge in huge amounts of unstructured textual data from the Web. A variety of related studies have focused on different subjects, such as automatic extraction of terms or phrases [7, 2], the discovery of rules for the extraction of specific information segments [11], and ontology construction based on semi-structured data [1]. Different from these previous works, the proposed approach is to categorize text segments via mining of search result pages.

Text Categorization. As mentioned in Section 1, conventional text categorization techniques are often utilized to analyze relationships among documents [4], and there is much difference between document categorization and text pattern categorization. The latter seemed relatively little investigated in the literature. The work most closely related to ours in methodological aspect is [5], in which the named entities are categorized into three major types: *Organizations*, *Persons*, and *Locations*. Their work also use unlabel documents to help the process of categorization. Their idea is mainly like this: for the extracted text segments, the unlabeled data themselves offer useful contextual informa-

tion, which, if properly exploited by statistical or machine-learning techniques, can ensure high accuracy of categorization. The difference between our work and theirs is not only that we use Web corpus while they do not, but also we use structural information of topic taxonomy rather than contextual information contained in unlabeled documents to train classifiers. Comparatively, in finding out the information about text segments, the information we extract are semantically deeper. Also, our approach, thanks to Web corpus, are more flexible and can be easily adapted to other applications.

3 The Proposed Approach

The diagram depicted in Figure 2 shows the overall concept of the proposed approach, which is composed of three computational modules: context extraction, model training and pattern categorization. The approach exploits the highly ranked search-result pages retrieved from online search engines as the effective feature sources for training the topic classes contained in the taxonomy and each unknown text pattern. The context extraction module collects features both for the topic classes and for the unknown text segments, the model training module utilizes the feature sources to train statistical model for the topic classes, and the pattern categorization module determines appropriate classes for the unknown text pattern.

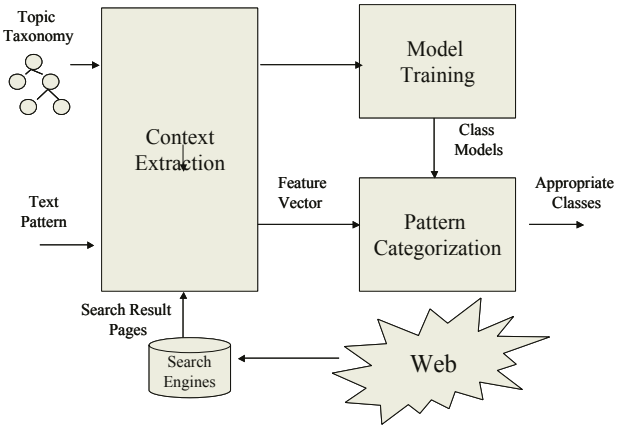

Fig. 2. An abstract diagram showing the concept of the proposed approach.

3.1 Context Extraction

We adopt the vector-space model as the data representation for both unknown text segments and target topic classes. The contexts of a text pattern are obtained from the highly ranked search-result pages (document snippets) returned by Web search engines, e.g., the titles and descriptions of search-result entries, and the texts surrounding the matched text segments. The features for a text pattern are then extracted from the returned snippets. The same procedure is used to collect document sets for training the topic classes in a predefined topic taxonomy.

Using Web search engines as information sources has both disadvantages and advantages. Web contents are usually heterogeneous and noisy, and need careful treatment. However, with the presentation schemes of most search engines, the neighboring contents surrounding a matched query (pattern) in Web pages are selectively shown in the returned snippets. Therefore, features are extracted from the corresponding text pattern's contexts instead of the whole Web page. Further, a huge amount of pages have been indexed, so most text segments can get sufficient results. As a result of recent advances in search technologies, highly ranked documents usually contain documents of interest and can be treated, at least in a certain amount of situations, as an approximation of the text segments' topic domains.

Representation Model. Suppose that, for each text pattern p, we collect up to N_{max} search-result entries, denoted as D_p. Each text pattern can then be converted into a bag of feature terms by applying normal text processing techniques, e.g., removing stop words and stemming, to the contents of D_p. Let T be the feature term vocabulary, and let t_i be the i-th term in T. With simple processing, a text pattern p can be represented as a term vector v_p in a $|T|$-dimensional space, where $v_{p,i}$ is the weight t_i in v_p. The term weights in this work are determined according to one of the conventional *tf-idf* term weighting schemes [10], in which each term weight $v_{p,i}$ is defined as

$$v_{p,i} = (1 + \log_2 f_{p,i}) \times \log_2(n/n_i),$$

where $f_{p,i}$ is the frequency t_i occurring in v_p's corresponding feature term bag, n is the total number of text segments, and n_i is the number of text segments that contain t_i in their corresponding bags of feature terms. The similarity between a pair of text segments is computed as the cosine of the angle between the corresponding vectors, i.e.,

$$sim(v_a, v_b) = \cos(v_a, v_b).$$

For the purpose of illustration, we define the average similarity between two sets of vectors, C_i and C_j, as the average of all pairwise similarities among the vectors in C_i and C_j:

$$sim_A(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{v_a \in C_i} \sum_{v_b \in C_j} sim(v_a, v_b).$$

3.2 Model Training

In this research, we consider using a Yahoo!-like topic taxonomy for the problem of text-pattern categorization. A Yahoo!-like taxonomy is a natural hierarchy of topics, in which most of non-leaf classes contain an appropriate number of child classes. In most cases, manually constructing such a topic hierarchy for a certain IE task is not too difficult. For example, as will be shown in Section 4, based on the Yahoo! directory, we can easily construct a taxonomy tree composed of topic classes including People, Event, Time, and Place. For training a categorization model for each topic class, such a taxonomy tree is useful, as its child classes offer ample information to characterize its concept. For example, In the People class, there are 55 sub-classes at the second level,

e.g., a Scientists class and a Politicians class; and there are about hundreds of sub-classes at the third level, nine of which, e.g., Mathematicians and Physicians, are the sub-classes of the Scientists class; these sub-classes enrich the concept of the scientists class.

In current stage of research, only non-leaf classes are considered as the target classes for categorization. For each non-leaf class, its training set is the union of the document snippets obtained by sending to search engines the class name and its child class names as queries. In our experiments, at most 100 document snippets could be extracted as the feature source for a specific query. Again, using the Scientists class as an example, there are totally 1000 (100+9*100) relevant document snippets that can be used to train its corresponding concept. Usually, it is not easy to obtain such amount of information from a corpus without extremely large sets of training documents.

On the other hand, the uniqueness and coverage of the child class names, however, might greatly affect the performance of the class model. If a class does not contain enough child classes or if many of its child class names are not meaningful, the features extracted from the retrieved document snippets might not be effective enough to characterize the concept of the class. Obviously, not all semantic classes can use this kind of approach to train their class models. Also, not every unknown text pattern can be categorized with the proposed approach. In fact, the proposed approach is more suitable to categorize the text segments that are more specific in their meaning and retrieve more contextual information from the Web. Therefore, the proposed approach prefers categorizing text segments into specific topic classes. It might not perform so well when categorizing common text segments into a broader class.

3.3 Pattern Categorization

Given a new candidate text pattern p, pattern categorization is to determine a set of categories C_p that are considered as p's related categories. With the same scenario stated previously, the candidate pattern p is represented as a feature vector v_p. For this categorization task, we here adopt a kNN approach.

kNN has been an effective classification approach to a broad range of pattern recognition and text classification problems [6]. By kNN approach, a relevance score between p and candidate cluster C_i is determined by the following formula:

$$r_{kNN}(p, C_i) = \sum_{v_j \in R_k(p) \cap C_i} sim(v_p, v_j)$$

where $R_k(p)$ are p's k most-similar objects, measured by sim function, in the whole collection.

The categories that a pattern is assigned to are determined by either a predefined number of most-relevant clusters or a threshold to pick those clusters with scores higher than the specified threshold value. Different threshold strategies have both advantages and disadvantages [12]. In this study, for evaluating the performance, we select the five most-relevant categories as candidates.

4 Experiment

To assess the performance of our approach, we have conducted several experiments. The Yahoo!'s taxonomy tree is used as our benchmark as it is readily available and well organized.

4.1 Domain-Specific Term Categorization

We first confined our attention to a specific domain, computer science, and conducted an experiment to observe how well our approach could be applied. In the Yahoo! Computer Science taxonomy tree, there are totally 36 first-level, 177 second-level, and 278 third-level classes. We used the first-level classes, e.g., "Artificial Intelligence" and "Linguistics," as the target classes and attempted to classify the class names at the third level, e.g., "Intelligent Software Agent," onto it. For each target class, we took its class name and child class names at the second level, e.g., "Machine Learning," "Expert System," and "Fuzzy Logic," as the seed instances for model training. These class names can be taken as a kind of domain-specific facts extracted in an IE task. Table 1 shows the result of the achieved top 1-5 inclusion rates, where top n inclusion rate is the rate of test segments whose highly ranked n candidate classes contain the correct class(es). To realize the effect of using second-level class names as seed instances in model training, the result is separated into two groups: with and without seed training instances.

Table 1. Top 1-5 inclusion rates for categorizing Yahoo!'s third-level CS category names.

Yahoo! (CS Class Names)	Top-1	2	3	4	5
KNN – With Seed Training Instances	.7185	.8841	.9238	.9437	.9636
KNN – Without Seed Training Instances	.4172	.6026	.6788	.7285	.7748

4.2 Paper Title Categorization

Besides using the third-level class names of the Yahoo!'s CS taxonomy tree, we also used another testing set. We collected a data set of the academic paper titles from four named computer science conferences in year 2002 and tried to categorize them into the 36 first-level CS classes again. Each conference was assigned to the Yahoo! category to which the conference was considered to belong, e.g., AAAI'02 was assigned to "Computer Science/Artificial Intelligence," and all the papers from that conference unconditionally belonged to that category. Table 2 lists the relevant information of this paper data set. Notice that this might not be an absolutely correct categorization strategy, as some papers in a conference may be even more related to other domains than the one we assigned them. However, to simplify our experiment, we make this straightforward assumption. Table 3 lists the experiment result. The purpose of this experiment is to examine the performance of categorizing longer text segments. Table 4 lists the categorization results of several miss-categorized paper titles. It can be observed that though these papers failed to be correctly categorized, they are conceptually related to

Table 2. The information of the paper data set.

Conference	# Papers	Assigned Category
AAAI'02	29	CS:Artificial Intelligence
ACL'02	65	CS:Linguistics
JCDL'02	69	CS:Lib. & Info. Sci.
SIGCOMM'02	25	CS:Networks

Table 3. Top 1-5 inclusion rates for categorizing paper titles.

Conference Paper	Top-1	2	3	4	5
KNN – With Seed Training Instances	.4628	.6277	.7181	.7713	.8085
KNN – Without Seed Training Instances	.2021	.2872	.3457	.3777	.4255

Table 4. Selected miss-categorized examples for categorizing paper titles.

Paper title	Conference	Target Cat.	Top-1	2	3	4	5
A New Algorithm for Optimal Bin Packing	AAAI	AI	ALG	AI	MOD	COLT	DNA
(Im)possibility of Safe Exchange Mechanism Design	AAAI	AI	NET	SC	LG	DB	MD
Performance Issues and Error Analysis in an Open-Domain Question Answering System	ACL	LG	AI	LG	ALG	DC	SC
Active Learning for Statistical Natural Language Parsing	ACL	LG	AI	LG	NN	COLT	ALG
Improving Machine Learning Approaches to Coreference Resolution	ACL	LG	AI	LG	ALG	FM	NN
A language modelling approach to relevance profiling for document browsing	JCDL	LIS	AI	UI	LG	LIS	ALG
Structuring keyword-based queries for web databases	JCDL	LIS	AI	LIS	DB	ALG	ARC
A multilingual, multimodal digital video library system	JCDL	LIS	LG	UI	LIS	ECAD	NET
SOS: Secure Overlay Services	SIGCOMM	NET	SC	NET	MC	OS	DC

Abbreviation List:
AI :Artificial Intelligence
ALG :Algorithms
ARC :Architecture
COLT:Computational Learning Theory
DB :Databases
DC :Distributed Computing
DNA :DNA-Based Computing
ECAD:Electronic Computer Aided Design
FM :Formal Methods
LG :Linguistics
LIS :Library and Information Science
MC :Mobile Computing
MOD:Modeling
NET :Networks
NN :Neural Network
OS :Operating Systems
SC :Security
UI :User Interface

those top-ranked categories to some degree. It is worthy of notice that the promising accuracy also shows the great potential of the proposed approach to classifying paper titles with Yahoo!'s taxonomy trees. From these tables, we can draw our preliminary conclusions: First, for very specific topic domains and text segments, our technique can obtain a rather high accuracy. Second, the seed instances used in model training have a crucial influence on the classification result. Without them, the performance drops significantly. Third, the result has shown that the proposed approach is also feasible for longer and specific text segments.

4.3 Named Entity Categorization

To observe how our technique performs in other circumstances, especially for named entities, we conducted another three experiments, i.e., using the sub-trees of "People" (People/Scientist), "Place" (Region/Europe), and "Time" (History-time Period) in Yahoo! as our testing beds. For these three cases, we randomly picked up 100, 100, and 93 class names, which can be considered as a kind of named entities, from the bottom-level and assigned them onto the top-level classes likewise. Tables 5 and 6 respectively list the relevant information of the sub-trees employed and some samples of the test named entities. It could be observed that in the "People" and "Place" cases, our technique got very satisfactory results, while in the "Time" case we did not get similar good result.

Table 5. The information of the three topic taxonomy trees extracted from Yahoo!.

Taxonomy Tree	# 1st-level Classes (Target Classes)	# 2nd-level Classes (Training Instances)	# 3rd-level Classes (Test Segments)
People (People/Scientist)	9	156	100
Place (Region/Europe)	44	N/A	100
Time (History-time Period)	8	274	93

Table 6. Top 1-5 inclusion rates for categorizing Yahoo!'s People, Place, and Time class names.

	Top-1	2	3	4	5
Yahoo! (People)	.8558	.9808	.9808	.9904	.9904
Yahoo! (Place)	.8700	.9500	.9700	.9700	.9800
Yahoo! (Time)	.3854	.5521	.6354	.6562	.6562

The reason of its degradation seems that the concept of a time period, such as "Renaissance" and "Middle Ages", is too broad and too much noise is contained in the returned snippets, thus lowering the precision of our categorization.

5 Information Extraction Applications

As we have shown in previous sections, as long as there exists a well-organized and reasonably-constructed taxonomy tree, we can categorize the unknown text segments onto it. Now suppose there is an article of unknown nature, we can extract some facts from it and classify them. Doing this, we may grasp their intended meaning and thereby have a clearer understanding of the whole article. We here use an example to illustrate our point. Randomly selecting several biography pages of scientist (these pages can be fetched from Yahoo!), we then extracted some facts (keywords) from them manually and attached them onto the taxonomy tree composed of the sub-trees of Computer Science, People, Place and Time that we have organized in Section 4. Note that a named entity may be categorized into multiple classes. After stemming the branches without attached facts, the taxonomy tree with the categorized facts can reflect a topic structure of the information contained in the facts. As shown in Figure 3, this kind of taxonomy tree offers a new perspective to understand the article.

Conventionally, when confronting a new document, one would try to classify the document and judge its content by the nature of the assigned class. In our case, we categorize its important facts. Although it is not fair to compare the effect of document categorization and that of text pattern categorization, we would like to point out that the benefit of text pattern categorization: (1) Text segments are more specific, giving a more concrete concept and are thus more suitable to be further exploited to develop advanced applications. (2) It is usually easier to categorize text segments than categorize the whole documents, since the latter often contains a lot of irrelevant information (features) and this may lead to poor performance in categorization.

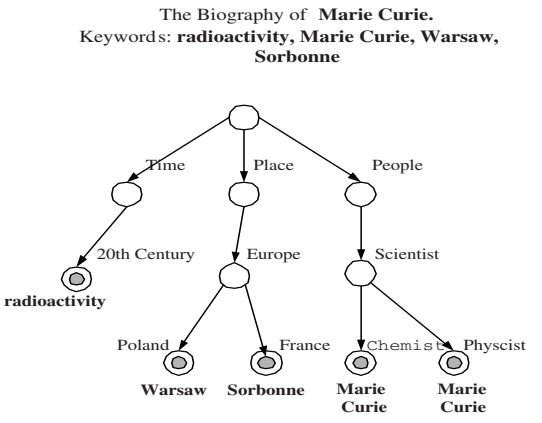

Fig. 3. A taxonomy tree with categorized facts.

6 Concluding Remarks

In this paper, we proposed a search result approach to categorizing unknown text segments for information extraction applications. Compared to conventional text categorization techniques, the proposed approach requires little manual effort and has few domain limitations. The feasibility of the proposed approach has been shown with extensive experiments. We believe the proposed approach can serve as a basis toward the development of more advanced Web information extraction systems.

References

1. E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the www. In *Proceedings of ECAI 2000 Workshop on Ontology Learning*, 2000.
2. H. Ahonen, O. Heinonen, M. Klemettinen, and A. Verkamo. Finding co-occurring text phrases by combining sequence and frequent set discovery. In *Proceedings of IJCAI'99 Workshop on Text Mining: Foundations, Techniques and Applications*, pages 1–9, 1999.
3. P. Brown, S. D. Pietra, V. D. Pietra, and R. Mercer. Word sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 264–270, 1991.
4. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–315, Zürich, CH, 1996. ACM Press, New York, US.
5. M. Collins and Y. Singer. Unsupervised models for named entity classiffication, 1999.
6. B. V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classiffication Techniques*. McGraw-Hill Computer Science. IEEE Computer Society Press, Las Alamitos, California, 1991.
7. R. Feldman, Y. Aumann, A. Amir, W. Kloesgen, and A. Zilberstien. Maximal association rules: a new tool for mining for keyword co-occurrences in document collections. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 167–170, 1997.

8. M. Hearst. Untangling text data mining. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.
9. S. Johansson, E. Atwell, R. Garside, and G. Leech. THE TAGGED LOB CORPUS users' manual, 1986.
10. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.
11. S. Soderland. Learning to extract text-based information from the world wide web. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 251–254, 1997.
12. Y. Yang. A study on thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 137–145, 2001.