UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

TOWARDS THE AUTOMATIC CLASSIFICATION OF DOCUMENTS IN USER-GENERATED CLASSIFICATIONS

Md.Ahsan-ul Morshed

January 2006

Technical Report # DIT-06-001

PhD Thesis Proposal

Towards the Automatic Classification of Documents in User-generated Classifications

Md.Ahsan-ul Morshed

XX Cycle

Academic Year: 2005-2006

Supervisor: Prof. Fausto Giunchiglia

Department of Information Technology & Communication,
University of Trento, Trento, Italy

Towards the Automatic Classification of Documents in User-generated Classifications

Ahsan-ul Morshed

Department of Information Technology, University of Trento, Trento, Italy

morshed@dit.unitn.it

Abstract. There is a huge amount of information scattered on the World Wide Web. As the information flow occurs at a high speed in the WWW, there is a need to organize it in the right manner so that a user can access it very easily. Previously the organization of information was generally done manually, by matching the document contents to some pre-defined categories. There are two approaches for this text-based categorization: manual and automatic. In the manual approach, a human expert performs the classification task, and in the second case supervised classifiers are used to automatically classify resources. In a supervised classification, manual interaction is required to create some training data before the automatic classification task takes place. In our new approach, we intend to propose automatic classification of documents through semantic keywords and building the formulas generation by these keywords. Thus we can reduce this human participation by combining the knowledge of a given classification and the knowledge extracted from the data. The main focus of this PhD thesis, supervised by Prof. Fausto Giunchiglia, is the automatic classification of documents into user-generated classifications. The key benefits foreseen from this automatic document classification is not only related to search engines, but also to many other fields like, document organization, text filtering, semantic index managing.

Keywords: Document classification, Semantic keywords, Categorization methods, Testing methodology.

1 Introduction

There is a huge amount of information scattered on the World Wide Web. As the information flows in the WWW at a high speed, there is a need to organize it in the right manner so that user can access it very easily. Most of the search engines classify the documents manually (e.g., Google [30], Yahoo [31], DMOZ [14] [3]). A good definition for text categorization can be found at Sebastini [1]: "*text categorization (TC-also known as text classification or topic spotting) is the task of automatically sorting a set of documents into categories (or classes, or topics) from a predefine set"*. Classification task is cross-pollinated between Information Retrieval field (IR), and Machine Learning field (ML) [1]. Text categorization is required in IR because when a user queries for information in a Web search engine, he/she wants information to be filed according to its relevance to his/her query and also to its contents. Text categorization also holds a great interest for researchers in the field of ML as it provides an excellent benchmark for their own techniques and methodologies [1].

We discuss two approaches in text categorization field: the manual approach, and the automatic approach. In the manual approach, human experts classify the document manually or use classifiers [12, 23, 25, and 28]. Although it gives quite accurate results, it is still very difficult to continuously update the information. Furthermore, it is expensive to maintain the classifier. Luhn [29] has shown that a statistical analysis of the words in the documents can provide some clues to its content. For example, a document that contains the words "boys", "girl" "teacher", "school", "arithmetic" "reading" etc., probably deals with education. He also uses the theme of probabilistic and human experts for classifying the documents [15, 12]. In the automatic approach, some well-known classification methods exist such as decision trees [23], decision rules [23, 24], k-Nearest Neighbor [23], Bayesian approach [12], vector-base [28] etc. Document classification systems such as Carrot^2 [19], Gammasite [20], Wordmap [13], IBM intelligent miner [27], Vivisimo [22] etc., are used for classifying the documents. These systems are not considering the semantic meaning of keywords extracted from documents. Woods [26] represents the conceptual indexing by considering the semantic meaning of words. We can get the keywords by using knowledge representation and natural language processing techniques. Then using these keywords we can extract the exact meaning from WordNet [6].We call these keywords as "*semantic keywords(S-keywords*)", as it contains some information regarding the context of our given keywords. These *semantic keywords* could be used for generating formulas. We could build the formulas based on the works of Magnini et al. [4] and Giunchiglia et al. [2]. We can translate natural language by mapping "parts of speech (POSs)" and their mutual syntactic relations. We can consider the formulas as logical disjunction ($\cup$), conjunction ($\cap$) and negation ($\neg$) of atomic concepts [2, 11] and we also use (#) symbol for distinguishing the senses of word. For example, we get a keyword "programming" from a document; we can go through this keyword in the WordNet [6] for extracting senses. We get the following senses: "programming#1", which could be interpreted as "*setting an order and time for planned events*" and "Programming #2", which could be interpreted as "*computer programming"*. Now we can translate these senses as a logical disjunction like (Programming#1 $\cup$ Programming#2). We take the most appropriate sense for building the formulas. Similarly, we can use conjunction and negation.

In my research, I will focus on automatic documents classification for user-generated classifications (we can call it as "predefined taxonomy"). Beyond its applicability in search engines, the scope of automatic document classification expands in many fields such as documents organization, text filtering, semantic index managing and so on. In our approach, we shall analyze the semantic meaning of keywords using WordNet [6] and build the formulas using natural language techniques. After building these formulas from the documents, we will run the S-Match/SAT [11] to match with existing formulas that we obtain from user-generated classification.

The structure of this work is as follows: Section 2 outlines the classification problem and evaluation methodology. Section 3 describes the existing methods and document classification systems. In section 4, the main objectives of the PhD thesis are defined. Section 5 illustrates the proposed solution which will be carried out during the research.

2 The Classification Problem and Evaluation Methodology

2.1 Motivating Example

We can examine the possible situations that can arise when we want to classify the documents under the user-generated classifications. Let us consider the DMOZ web page directory [14] as a user-generated classification. Suppose, we have the book: "Java Enterprise in a Nutshell, second edition". We need to classify this book under the user-generated classification of Fig.1. However, we only know about the title of book where one of extracted keyword is "Java".

If we consider this keyword in WordNet [6], we can extract three senses

1. Java is an island in Indonesia south of Borneo;
2. Java is a beverage consisting of an infusion of ground coffee beans;
3. Java is a simple platform-independent object-oriented programming language;

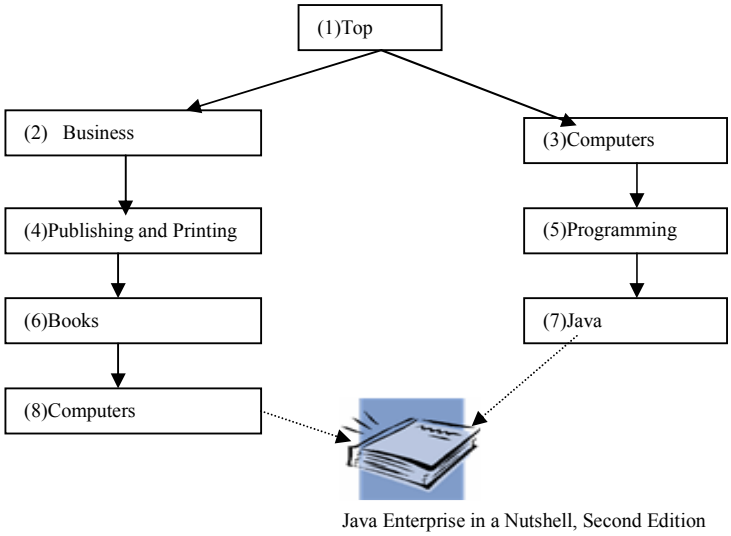

Fig. 1. A part of the DMOZ web directory (this picture was presented by Giunchiglia et al., [2]).

As a human expert, it is easy to classify documents in the user-generated classification of DMOZ [14]. Our book would be placed under node 7 titled "Java" [2]. We can also classify documents automatically using Machine Learning [24]. But in these techniques they use supervised and unsupervised classifier for classifying documents with a high level of human interaction [7, 8]. In our approach, we will reduce this human interaction by using the semantic keywords and build formulas and avoid using the classifiers.

2.2 Problem

In our motivating example, we have noted that the classification by a human agent can be performed easily. On the other hand a machine needs to understand the query word "Java Enterprise in a Nutshell, second edition" [2]. It is difficult for a machine to understand the exact meaning of the word automatically. But if we can provide the data in the right way then a machine can provide more accurate answer. In addition, the classification hierarchies are written in natural language. It is very hard to automate the classification task, and, as a consequence, standard classification approaches measure to manually classifying the object into classes. Let us take the example of open directory project *DMOZ, a human edited web directory, which "power the core directory services for the most popular portals and search engines on the web, including AOL, Netscape, Google, Lycos, DirectHit, and HotBot"* [14] *and the Dewey Decimal Classification System (DCS)* [2]. Although these are standards and universal techniques, they have a number of limitations [2]:

- The semantics of a given topic is implicitly codified in a natural language label. These labels must therefore be interpreted and disambiguated.

- Two nodes may also be ambiguous in the sense. A link connecting the parent node "programming" with child node "Java" may or may not mean that 1. The parent node means "computer programming" 2. The child node means "Java Programming".
- As a consequence of previous two items, the classification task also becomes ambiguous if we use a different classifier.

Now, a research question arises as "how can we classify the document automatically". We will see some techniques in the state-of-art. These techniques are appropriate for automatically classifying the documents requiring a high level of human interaction. In our approach we will overcome the limitation of hierarchical and manual classifications. We will give a new direction for classifying documents using semantic keywords and formulas.

2.3 Top-down and Bottom-up Approaches

Classification is not a recent problem. Researchers have been trying to solve it for years. Some of them are successful in getting the result by applying the "bottom-up" techniques. In these techniques, they mainly take details of data and start placing then in "buckets". This kind of approach is called an unsupervised approach. At the same time, some of them use supervised approach. In this approach, they train the classifier with labelled documents and then use this trained classifier to label an unseen document [7, 8].

On the other hand, in "top-down" techniques, domain experts analyze the subject matter at hand, and determine the category of a given document. For example, there are 10 general topics. Each document is then examined and placed in its appropriate pre-existing category. We can also consider the "top-down" approach where we can use user-generated taxonomy with the meaning of node. By this meaning, we can classify the documents [5, 9].

2.4 Classification Evaluation

The most well known mechanism that measures the performance of classification approaches are the calculation of precision, recall and F-measure [12],
Consider,

A is the number of documents known belonging to the category and assigned automatically to that category,

B is the number of documents known not belonging to the category but assigned automatically to that category,

C is the number of documents known belonging to the category but not assigned automatically to that category,

Based on A, B, and C, we measure Precision, and Recall as follows:

Recall = A/A+C

Precision=A/A+B

F-measure is a global measure of performance that contains Precision and Recall as follows:

$$F=\frac{2*\mathrm{Re}call*\mathrm{Pr}ecision}{\mathrm{Re}call+\mathrm{Pr}ecision}$$

In the computation of performance based measures, experts play the role of the measuring instruments, since they define the performance of document classifications. These measuring instruments are considered for execution time.

2.5 Testing methodology

From the document classification perspective, we distinguish the following categories of methodologies issue:
-Data set collection: This category includes the size of documents, the data collection methods.
- Result acquisition. This category includes the method of results, judgments and feedback.
-Analysis and interpretation: this category includes the comparison with other systems.

3 State-of-the-art

3.1 Categorization Approaches

The main two approaches are in the following:

Manual categorization: The human experts go through the whole process [3] .Although this gives more scalable results, it introduces inconsistency during the classification. More specifically, it does not keep changes as soon as new document arrive in the taxonomy because human experts are sometime lazy or concentrate other works.

Automatic categorization: In this case, the methodologies used in the automatic process of categorization are like wise not something we would know about at the beginning of our search. By learning more about the various methods of automatic categorization in Section 3.2, we can get better understanding of how each of the methodologies may be applied to our particular situations.

3.2 Categorization Methods

Some of the existing categorization methods include decision trees, decision rules, k-nearest neighbor, Bayesian approach, neural networks, regression-based methods, vector-base method etc. We illustrate some of them in the following sub-sections.

3.2.1 Decision Trees

We can build a manual categorization of the training document in a decision tree by representing a well defined true/false-queries where nodes represent questions and leafs represent the corresponding category of documents [23]. After having constructed the tree, a new document can easily be categorized by putting it in the root node of the tree and by letting it runs to through the query structure until it reaches a certain leaf. The main advantage is that the output tree is easy to understand by even those who not familiar with the details of the model. The risk is "overfitting" because there is an existence of an alternative tree that categorizes the training data incorrect way.

3.2.2 k-Nearest Neighbor

The previous method is based on a learning phase but k-nearest completely skips the learning phase and categorizes on-the-fly. The categorization is often performed by comparing the category frequencies of the k-nearest documents. The closeness of the documents can be evaluated by the calculating the Euclidian distance between the two vectors.

The method is simple. Hence, it does not need any resource for training the documents. The algorithm performs well even if the category of specific documents forms more than one cluster. The category may contain more than one topic [23], although there is a risk of inadequate categorization for different numbers of training documents per category.

3.2.3 Bayesian Approaches

A Bayesian categorization approach is quite challenging. We do not consider the computation part of these methods. We find two groups of Bayesian approaches in document categorization: First one is naïve Bayesian approaches and second one is non-naïve Bayesian approaches. The naïve approach uses the assumption of word (i.e. feature) independence, meaning that the word order is irrelevant and consequently that the presence of one word does not affect the presence or absence of another one. This assumption makes the computation of Bayesian approaches more efficient. Although the assumption is obviously violated in every language, it has been shown that the classification accuracy is not seriously affected by this kind of violations [25]. Nevertheless, several non-naïve Bayesian approaches eliminate this assumption [12]. A disadvantage of Bayesian approaches is trained data with supervised classifier required a higher level of human interaction [12].

3.3 Classification Systems

3.3.1 Document Classification Systems

Carrot^2: The Carrot^2 system [21, 19] is a document classification and clustering tool. It has similarity with Vivisimo Meta search engine [27]. Carrot^2 is used as a general search engine for collecting data source and clustered data by lingo rock method. It shows result by visualizing the component (dynamic tree in most cases).The search results come out as a group. Moreover, the process of discovering clusters is fully automated.

GammaSite: GammaSite [21, 20] is a document classification and categorization tool. It considers supervised machine learning techniques for creating the taxonomies and adds documents to categories. Dr. Yiftach Ravid[1], VP R&D of GammaSite, explains, "*Our algorithms automatically analyze and understand the content of documents and then automatically organize huge amount of textual data into pre -defined categories. Gamma Ware's strength lies not only in its high precision, but also in the minimal amount of manual effort required to achieve superb results for hierarchical categorization".*

IBM Intelligent Miner: This tool is used for document classifications. It uses the linguistic analysis technique, and vector-based method for clustering and classifying document by pre-determined categories [21, 27].

Vivisimo: This tool is used for clustering the documents. It uses a heuristic algorithm that puts documents together based on text similarity. "*Vivisimo does not use a predefined taxonomy or controlled, so every cluster description is taken from the search results within the cluster. It is not perfect. In fact, it is hard to know even perfection would even mean. Thus, sometimes an article will be in placed in a cluster which it does not really match, in the eyes of an expert. Or, two different meaning of a word or phrase may be conflated" [*21, 22].

Wordmap: This is a commercial tool for classifying documents. It uses Support Vector Machine (SVM) for statistical analysis and document classification. In this SVM converts the example documents into vectors that classify incoming document to a high level of accuracy. Common file formats such as MSWord, PDF, HTML and text are used in the classifier [21, 13].

[1]http://www.content-wire.com/Taxonomies/Index.cfm?ccs=82&cs=677

3.3.2 Keyword Extraction Systems

Kea: Witten et al. [10] implemented their methodology in Kea, an algorithm for automatically extracting keyphrases from the text. Kea identifies candidate keyphrases using pre-processing [11], calculates feature value for each candidate, and uses a machine learning algorithm to predict which candidates are good keyphrases. The Naïve Bayes machine learning scheme first builds a predication model using the training documents with known keyphrases, and then uses the model to find keyphrases in new documents. Two features are calculated for each candidate phrase and used in training and extraction. They are: TF × IDF, a measure of phrase's frequency in a document compared to its rarity in general use and first occurrence, which is the distance into the document of the phrase's first appearance. Kea´s effectiveness has been assessed by counting the keyphrases that were also chosen by the document's author, when a fixed number of keyphrases are extracted.

Barker Approach: Barker [16] describes a system for choosing noun phrases from a document as keyphrases. A noun phrase is chosen based on its length, its frequency and the frequency of its head noun. Noun Phrases are extracted from a text using a base noun phrase skimmer and an off-the-shelf online dictionary. Barker approaches involves human judgment for performing this experiments.

KPSpotter: KPSpotter is an algorithm implementing the methodology proposed by Song et al. [17].After classical pre-processing has been applied. The algorithm employs a technique that combines information gain and a data mining measure technique introduced in ID3 algorithm [18]. In this sense KPSpotter presents some resemblances with extractor. Both algorithms, in fact, use a learner belonging to the same family, i.e., the decision tree [18].

4. Objectives of Thesis and Research Challenges

In my thesis, I am proposing:

- Develop an algorithm for automatically generating semantic keywords(S-keywords).
- Develop an algorithm for generating the formulas from the semantic keywords(S-keywords).
- Implementation of both algorithms.
- Testing and Evaluation using classifications from Google, Yahoo and DMOZ.

We have discussed earlier about the problem of automatic documents classification in user-generated classifications. In previous section, we saw different methods and systems for solving the documents classification. Most of the systems classify documents manually or in a semi-automated way using a classifier. Also they handle multiple documents. In our approach, we will consider a single document and nation of logical AND, logical OR and logical Negations for building the formulas suppose,

1. Given a set of extracted keywords from documents, it is not clear if it should be logical AND or logical OR of these words. For instance, the keywords {skiing, trentino} may mean {skiing and trentino} or {skiing or trentino}
2. If a set of keywords being converted to a formula can be represented as [some formula or A], where A is an atomic concept which is not presented in the classification, then the documents will never be classified. Otherwise, it will be classified.
3. If we are using a predefined vocabulary, and if we can find some keywords from the vocabulary, then it is likely understood that the document is not about the rest of the keywords in the vocabulary. This is where we introduce negations.

5. The Proposed Solution

In Fig. 2**.** we will consider a single document as input for datamining tools and we get output as a set of keywords. After getting these keywords, we consider the semantic meaning of each keyword from WordNet and build the formulas. These formulas would be compared with already obtained formulas from the user-generated classifications by S-Match /SAT and classify that documents in the user-generated classifications.

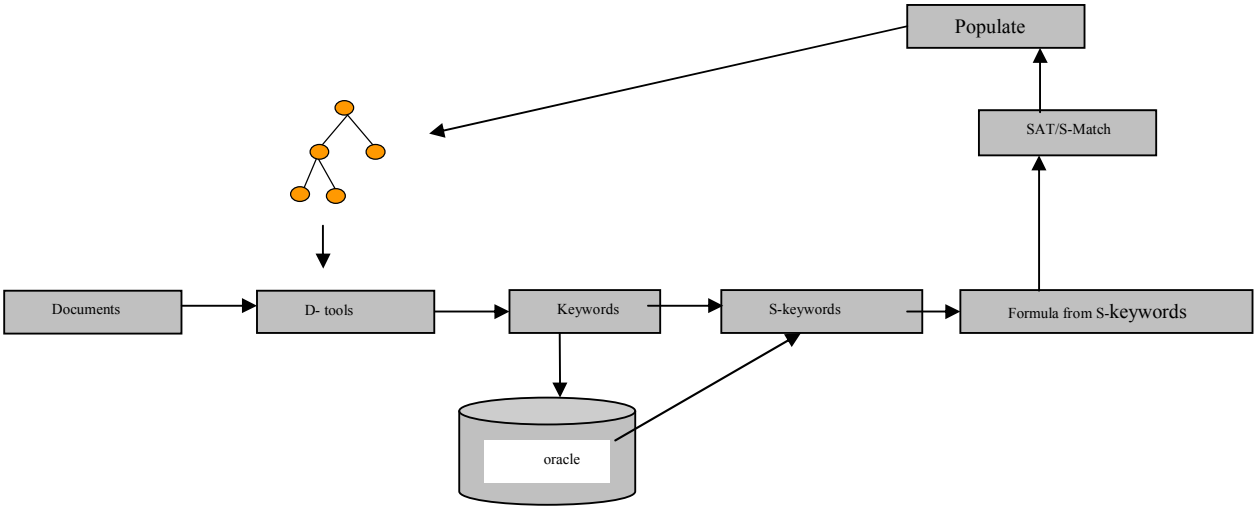

Fig. 2. An overview of document classification (proposed solutions).

We describe the following components in below:

User-generated classifications: At the beginning of our experiment, we start with well-known user-generated classifications from Google, Yahoo, and DMOZ etc.

Documents: In our section 3, we noticed that most of the systems available at present, are considering only multiple documents. We do not find any system which considers a single document. In our work, we will consider a single document for processing. During the first phase, we will consider only the text files. Later, we will expand our work with documents in the formats PDF, HTML, XML etc.

Data Mining tools (D-tools): There are some existing datamining tools like Kea, KPSpotter etc that can extract important keyphrases from the documents. These tools extract the important keyphrases with the help of a human expert. We will start our experiment with one of these tools, later we can develop our own data mining tools which can extract the keywords automatically from text without any help of a human expert.

Keywords: We will consider the stem word as a keyword, by removing any extra alphabets, but still retaining the context of the original keyword. For e.g. “animals” could be replaced with “animal”.

Oracle: We consider the WordNet as our oracle. We will extract the sense of our keyword by using this oracle.

Semantic Keywords (S-keywords): For instance, if our keyword is “automobile”, then we can get three senses from this word namely {car, bus, truck}

Formulas from S-keywords: After getting the S-keywords, we consider logical AND, OR or Negation for building formulas.

S-Match /SAT: An operator which matches the formula from documents and the formula from the user-generated classifications.

We exemplify our work as follows:

Develop an algorithm for automatically generating S-keywords:

All the existing tools use manually generated keywords for training the classifier. We will generate keywords automatically and check the oracle for getting appropriate meaning of the keywords that we consider as S-keywords. Actually this concept comes from conceptual indexing [26]. Example of S-keywords: Consider a keyword "automobile". We know three senses of automobile {car, bus, truck} from WordNet. S-keywords contain the context of given keywords.

Develop an algorithm for generating the formulas from S-keywords:

After extraction S-keywords, we will generate the formulas [2, 4, and 11]. For example: See our motivation example in Section 2.1

We can convert this title information by natural language processing as logical disjunction ($\cup$) and conjunction ($\cap$) Here, we can get the concept of book as java#3 $\cap$ Enterprise#2 $\cap$ book#1[2, 11]. This concept can be shown by propositional reasoning [2], that set of classification alternative includes all the nodes of corresponding user-driven classification. We are considering the nodes n7 and n8 for our discussion here. The discussion about the remaining nodes would be addressed in detail in the thesis. Now, we provide two formulas [2] in the nodes n7, n8 whose labels l_7^N = computer# $\cap$ programming# $\cap$ language# $\cap$ java# and labels l_8^N = business# $\cap$ (publishing # $\cup$ printing#) $\cap$ books# $\cap$ computers#

We can extract the following knowledge from WordNet: the *programming language Java is a kind of programming languages, and it is a more specific concept than computers is*; *books are related to publishing*; and *enterprising are a more specific than business* [11]. We can represent this knowledge by the following axioms [2]:
a_1= (java#3 $\subseteq$ pr_language#1), a_2= (java#3 $\subseteq$ computer#1); a_3= (book#1 $\subseteq$ publishing#1), a_4= (enterprise#3 $\subseteq$ business#2);

We then translate the axioms and the labels into the logical language as follows:

$(a_1 \wedge a_3 \wedge a_4) \rightarrow l_8^N$; $(a_1 \wedge a_2) \rightarrow l_7^N$;

Now, we will run the S-Match/SAT on above formulas, and classify the book under the given classification.

Acknowledgements:

I would like to thank Prof. Fausto Giunchiglia, Periklis Andritsos, and Ilya Zaihrayeu for many lessons on how to do research and write articles for their invaluable support and never-ending patience in guiding my entrance into AI domain and life in general.

References

1. Sebastiani, F.: Text Categorization. In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, 2005, pp109-129.
2. Giunchiglia, F., Marchese, M., Zaihrayeu, I.: Towards a theory of Formal classification. In Proc.of C&O-2005, Pittsburgh, Pennsylvania, USA.

3. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words: In Douglas H.Fisher, editor, proceedings of ICML-97,14th International Conference on Machine Leaning, pages 170-178, Nashville, US, 1997.Morgan Kaufman Publishers, San Francisco, US.
4. Magnini, B., Serafini, L., Speranza, M.: Making Explicit the semantics hidden in schema models. In Proc.of the workshop on Human Language Technology for Language Technology for the semantic Web and Web Services, held at ISWC-2003, Sanibel Island, Florida, October, 2003
5. Avesani, P., Adami, G., Sona, D.: Boostrapping for Hierarchical Document Classification. In Proc.of CIKM-03,12th ACM Int.Conf. on Information and Knowledge Management, pages 295-302, ACM Press, New York,US,2003.
6. WordNet :http://wordnet.princeton.edu/
7. Nigam, K., McCallum, A., Thrun, S., Mitchel, T.: Learning to Classify Text from Label and Unlabeled documents. In Proc.of AAAI-98,15th Conf.of the American Association for Artificial Intelligence, pages 792-799, Madison, US, 1998.
8. Liu, B., Lee, W., Yu, P., Li, X.: Partially Supervised Classification of Text Documents: In Proc of the 19th Int.Conf. on Machine Learning, Pages 387-394, 2002.
9. McCallum, A., Nigam, K.: Text Classification by Boostrapping with Keywords, EM and Shrinkage. In ACL'99 Workshop for Unsupervised learning in Natural Language Processing, pp.52-58.1999
10. Witten, I., Paynter, G., Frank, E., Gutwin, C., Nevill-Manning, C.: KEA: Practical Automatic Keyphrase Extraction .In ACM DL, pages 254-255, 1999.
11. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: An algorithm and an implemented of Semantic Matching. In Proc. of ESWS'04.
12. Lam, W., Low, K.: Automatic Document Classification Based on Probabilistic Reasoning. Model and Performance Analysis. In Proc.of the IEEE International Conference on Systems, Man and Cybernetics, Vol.3, p. 2719-2723, 1997.
13. Wordmap:http://www.wordmap.com/General/FAQ.html#connect_content_management
14. DMOZ: the Open Directory Project, see http://dmoz.org/
15. Borko, H., Bernick, M. : Automatic document classification. J.ACM, 10, (1963)
16. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases.In Proc. of the thirteenth Canadian Conference on Artificial Intelligence, pages 40-52, 2000.
17. Song, M., Song, I., Hu, X.: Kpsptter: A flexible information gain-based keyphrase extraction system. In Proc. of the fifth ACM international workshop on web information and data mangament, pages 50-53.ACM press, 2003
18. Quinlan, J.: Learning decision tree classifier.ACM Comput.Surv. 28(1):71-72, 1996.
19. Carrot^2 system: http://www.carrot2.org./website/xml/index.xml
20. GammaSite system:http://www.gammasite.com/unsupported_browser.html
21. Search tools:http://www.searchtools.com/tools/ibm-imt.html
22. Vivisimo: http://vivisimo.com/
23. Geiner, R., Schffer,J. (2001): AIxploratorium-DecisionTrees:http//www.cs.ualberta.ca/~aixplore/learning/Decision Trees
24. Apte, C., Damerau, F., Weiss, S.M.: Towards Language Independent Automated Learning of Text Categorization Models. In Proc.of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp23-30.
25. Domingos.P, Pazzani.M: On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, in: Machine Learning, Vol.29, No 2-3, pp103-130 on Information Systems.vol.12, No.3, pp253-277
26. Woods, W.: Conceptual indexing: A better way to organize knowledge.http://research.sun.com/techrep/1997/abstract-61.html
27. IBM Intelligent miner for Text: http://www.searchtools.com/tools/ibm-imt.html
28. Joachims, T.: Text categorization with support vector machines: Learning with Many Relevant Features. In Proc. of the 10th European Conference on Machine Learning, pp137-142.
29. Luhn, H.P 1957: A statistical Approach to Mechanized encoding and searching of literary information. IBM J. Res Dev, 1,309-317317
30. Google search engine: http:// www.google.com
31. Yahoo search engine: http:// www.yahoo.com