Automatic Classification of Course Web Pages

Batu Inal **Cici Lu** **Jason Xie** **Nikita Bhutani**
University of Michigan
{batuinal, jiexilu, jasonx, nbhutani} @umich.edu

Abstract

In this project, we describe how to automatically classify and identify course web pages from a given collection of university web pages. In doing so, we explore different aspects of the classification task: data collection, data annotation/labeling, classification algorithm and evaluation. Data collection involves crawling university web pages and extracting relevant information. The extracted data needs to be labeled manually and then analyzed to design features. Depending on the dataset and features, an appropriate classification approach like Support-Vector Machines or Naive-Bayes will be used. A comparison of different classification approaches can give useful insight into the differences in the assumptions made by these approaches and the characteristics of the data.

1 Introduction[1]*

With the rapid development of World Wide Web (WWW), over 4 billion pages are accessible to the users today. As a result, getting relevant and interesting results quickly from the Web has become challenging and requires organization and categorization. Automatic classification of web pages into categories can help search engines find relevant results quickly. Text classification was traditionally done manually by domain experts. More recently, semi-automatic and automatic approaches for text classification have seen a rise as they are cheaper and faster than manual categorization. In this project, we propose to build a classifier to perform classification task on course web pages. Such a classifier will be helpful in building focused crawlers or search engines and in improving the quality of search results for academic queries.

Many learning-based approaches including k-Nearest Neighbor, Support-Vector Machines, Bayesian probabilistic approaches have been applied for the task of text classification. However, the classification accuracies of these algorithms depend on the choice of features. Web pages tend to contain many irrelevant and infrequent words that can reduce the performance of the classifier. It becomes important to carefully select and extract features that are representative of the content of the web page. Also, web page classification differs from text classification as a web page is more than just a plain text document. Every web page is associated with non-text components like title, hyperlinks, HTML tags and meta data. Intuitively, a classifier can benefit from the information available through these context features. The goal of this project is to identify features which can distinctively identify course web pages.

[1]* This document has been adapted from the instructions for earlier ACL and NAACL proceedings, including those for NAACL-HLT-12 by Nizar Habash and William Schuler, NAACL-HLT-10 by Claudia Leacock and Richard Wicen- towski, NAACL-HLT-09 by Joakim Nivre and Noah Smith, for ACL-05 by Hwee Tou Ng and Kemal Oflazer, for ACL-02 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats. Those versions were written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence*. This second version clarifies the procedure for submitting for double-blind reviewing.

2 Related Work

Web page classification is an essential part of many information mining and management systems. As such, many independent researches have been conducted regarding the topic. The research conducted by Qi and Davison [1] provides a detailed review of many useful features, algorithms, and other related issues specific to web page classification. More specifically, they point out that web page classification was traditionally treated as text classification while the visual information was ignored. A good set of features include title, headings, metadata, and main text on the page. Some of the studies which are based on visual analysis of web pages represent web page as a hierarchical visual adjacency multigraph and have proved to perform better than the traditional bag-of-words model. One interesting idea presented in the paper was to include features of neighboring pages to supplement information for categorization. Good algorithmic approaches rely on the type of features as well on the weighing scheme employed for classification. Dimensionality reduction by considering certain parts of the document more important than the others has proven to be useful in the literature and we intend to explore this approach in our work by combining content and link information in some form (for instance by emphasizing anchor text more than the text content). As pointed out in the paper, with the development of the semantic Web, ontology based approaches can prove useful where in the hierarchical information can be used to augment the knowledge available directly from the text content.

As previously mentioned, a large portion of pre-processing will consist of labeling or annotating our data. Standard classifiers require both positive and negative examples. In the case of web page classification, selecting a set of negative examples is both arduous and prone to bias since it must be representative of the universal set of web pages that are not course pages. Yu et al. propose a method [2] to learn decision boundaries using only positive examples and a set of unlabeled examples. The unlabeled examples can simply be any random sample of the universal set, even including positive examples. Using a series of linear kernel SVMs, test performance comparable to that of classic classifiers can be achieved. We may consider employing this method to avoid not only the extra time spent manually identifying and labeling the negative examples, but also any bias in selecting these examples.

Kan and Thi conducted research [3] in supplementing full text web page classification methods with additional features from the URL such as token precedence and length. Their results show an improvement on previous URL only methods by over 30%, and a small increase in performance when using full text supplemented with URL features. The features are almost trivial to extract but may significantly improve our results.

Shen et al. present a different perspective to web page classification by employing supervised and unsupervised summarization algorithms [4]. They extract most relevant contents from the Web pages and then pass them on to a standard text classification algorithm. They propose heuristics to identify information objects, navigation objects, interaction objects, decoration objects and special function objects on a web page and distinguish the noisy components on the web page which make the text summarization task difficult. We might consider exploring these heuristics to normalize the data set which so that the structural differences between the course web pages collected from different universities is minimized.

Zhu et al. follow the intuition that the content and linkage information of a web page can be collectively used for classification tasks. In particular, they proposed an algorithm [6] that derives a new representation of a web page in a low-dimensional space by carrying out a joint factorization on both the linkage adjacency matrix and the document-term matrix. They conducted their experiments on the WebKB data set consisting of about 6000 web pages from computer science departments of four schools (Cornell, Texas, Washington, and Wisconsin). They found average classification accuracy of about 81.5% while using SVM to categorize the data into seven categories. We intend to include a subset of this data set in our experiments so we have a baseline to compare our system with.

In a comparative study by Yang et al. [7] different term selection techniques like mutual information, document frequency, information gain, χ^2-test were discussed and compared. It was found that even simplest techniques like document frequency thresholding can be very aggressive yet effective and scalable. Since it is generally believed that low document frequency terms are more informative, in our work we intend to re-examine this assumption and explore how the classification accuracy is affected by number of features. Having fewer number of features allows exploration of more powerful and computationally intensive learning algorithms.

3 Data Pre-processing

A good classifier requires a large volume of labeled training data. We also need to select the features carefully; simply tokenizing the training data and using the terms as features will make the classification inefficient. In the following section, we describe the data pre-processing methodologies we intend to implement in this project.

3.1 Collection Methods

We have crawled over 3,000 potential course web pages from the following universities: Stanford University, Massachusetts Institute Of Technology (MIT), Caltech, Princeton University, Brown University, Yale University and University of Michigan. Initially, we used the university websites as seed for the crawler but that resulted in fewer pages related to academics and courses than pages related to admission, blogs, news and campus life.We then tried to include homepages of professors at different universities as these pages are more likely to be linked to course web pages. While it worked for some universities like University of Michigan, mostly the crawler ended up scraping information related to the research interests work of the professors since there is a significant number of professors who predominantly do research over teaching.

Finally, the course catalogs of the universities proved to be a good set of seed pages. While some of the universities like MIT have a course catalogue listing courses being taught across departments, others like Caltech and Princeton have course listing on the department websites. We ended up using a combination of the techniques discussed above and customized crawling for each of the universities listed above. We used python library called Scrapy to crawl the course web pages. While restricting the crawler to the university domain, URLs that included query string (*http://example.com/over/there?name=ferret*) or pointed to binary files (*http://example.com/over/there/winterForm.pdf*) were ignored. While we ignored the head element in the HTML pages that were crawled, we scraped the body of the HTML pages along with the title and URL of the page since we expected to find interesting structural features (besides the plain text content) in the page that would distinguish a course web page from other pages.

While doing research on literature on the subject, we found that similar researches to classify university web pages have been carried out at Carnegie Mellon University and University of Massachusetts Boston. By extensive search, we were able to obtain the data sets used in these research projects. These data sets can be found at: *http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/* and *http://www.cs.umb.edu/~smimarog/textmining/datasets/*. The data set was already annotated with the seven labels: course, department, faculty, other, project, staff and student. Our project can benefit from a controlled data-set and including a subset of this data set would provide a good balance of positive and negative examples in our data set.

3.2 Annotation Methods

Once the web pages were crawled, we manually annotated the web pages. The criteria used for annotating the web pages were:

a. A web page with course listings/catalogue is not considered a course web page.
b. A web page with information about the instructor, term offerings, course description, syllabus, assignment and project details is considered a potential web page.
c. A web page with information about a research project or faculty is not considered a course web page.
d. A web page with information about library resources, schedule of classes, registration information etc. is not a course web page.

Since we ignored binary files and also tuned the depth of the crawler, binary files related to homework, projects and exams were not scraped and hence could be excluded from the annotation criteria.

3.3 Data samples and statistics

The numbers of pages crawled from each university are listed in Table 1.

University/Department	Pages crawled	Course pages
Brown CS	251	244
Caltech CMS	43	10
Caltech Math	59	58
EDX	4	3
MIT OCW	2401	2339
Princeton CS	114	13
Stanford CS	71	43
Yale OYC	72	41
UMich EECS	135	53

The numbers of pages in each category of the WebKB dataset are shown in Table 2.

Category	No. of pages
Student	1641
Faculty	1124
Staff	137
Department	182
Course	930

Project	504
Other	3764

Since a large subset of pages crawled were from MIT, only a subset of these pages are included to avoid the classifier from being biased towards these pages. Also, only a subset of pages from project, faculty, department, student and other categories from the WebKB data set are included in the final data set to have a good balance of positive and negative examples.

Of over 6000 pages crawled and collected, about 1600 pages were annotated as course pages. In order to have a balanced data set for negative examples with no bias towards any specific category, 1600 pages were selected as negative examples from the crawled and collected data set. While selecting pages from collected data set, care was taken to have same percentage of pages from each of the categories (student, faculty, department, project) so that none of the categories dominate the negative examples set. Further, to avoid bias towards web pages from a particular university, we separate pages from a few universities and included them exclusively in the test set. The data set of about 3200 web pages was then partitioned; 75% as training set and 25% as test set for the classification task.

4 Method Description

In any classification task, some features in the data set may be noisy and/or irrelevant to the label or some subset of features may be adequate for labeling which causes the other features to become redundant. In such cases, determining and using a specific subset of features results in a faster and usually more accurate solution [5]. For instance, in classifying course web pages, while it is useful to include some features which are indicative of structural differences between course and non-course web pages, it is important to not include too many (and equally weighted) features that might over fit the classifier to a specific design of a web page. Consider a typical web page from MIT Open Courseware: *http://ocw.mit.edu/courses/architecture/4-001j-cityscope-new-orleans-spring-2007/*. There are very few keywords in the text indicating that it is a course web page. While the anchor text is more useful, some of these are more useful than others (see links at the bottom of the page). Consider a typical page from the Brown University: *http://blogs.brown.edu/csci-0160-s01/*. It follows a blog layout and anchors indicate all the information to classify it as a course web page. This indicates that we might need some kind of weighting scheme for our features, and a naive term frequency based approach might not work for this classification task.

4.1 Feature selection/Extraction

a. Frequency Threshold: If a word is occurring in different examples frequently enough, it is included in the feature set as a unique feature. Too infrequent and too frequent words can be ignored. Thresholding on document frequency is aggressive but has proven to be effective. We also intend to use χ^2-test to reduce the number of features. Furthermore, we plan to explore Term-Frequency as well as Term-Frequency-Inverse-Document-Frequency (TF-IDF) as feature weighing scheme.

b. Clustering: Another approach to reduce number of features is to find groups of similar terms. A group (or cluster) is then considered a feature. The resulting clusters can then be used to describe a web page. This technique might not be very helpful in this problem considering a course web page will have more proper nouns and hence cannot be reduced significantly if clusters were used. However, this approach can be explored in future works.

4.2 Evaluation Methods

a. Classification algorithms: We intend to use Logistic Regression, Naive Bayes and Decision Trees classification algorithms in our experiment. Comparing the performances of these algorithms will give an insight into the validity of the modeling assumptions of these algorithms in the context of course web page classification. We also plan to compare the performances of Multinomial and Bernoulli variations of Naive Bayes.

b. Effect of feature space reduction techniques on classification: To compare different feature set reduction methods, a single classification algorithm must be chosen to make results comparable. Impact of different reduction methods can studied as a function of feature set size.

c. Feature Design: Since most of the course web pages do not have a lot of text, we plan to see the effect of including special features, derived from the layout and metadata, in the feature set. For instance, the title of a course web page typically contains the course number/name which can distinguish it from other university web pages. Thus, by considering only the title of a page as source of information we want to see how accurately the classifier can identify it to be a course web page. Differential weighing of different tags (title, anchors and body) can also reveal how relevant these tags to classification.

   Furthermore, an important feature to consider is the relative count of relevant anchors. A relevant anchor is one which contains one of the following keywords: *'slide', 'handout', 'schedule', 'syllabus', 'homework', 'lecture', 'assignment', 'project', 'exam', 'midterm', 'final', 'notes', 'staff', 'hours', 'course-info', 'piazza', 'pdf'*. We expect that including this feature can boost the classifier performance since anchor urls typically get stripped off during tokenization.

   Intuitively, certain tags are more prevalent in a course web page than a non-course web page. By utilizing tags we can take advantage of the structural information embedded in the HTML files, which is usually ignored by plain text approaches. Therefore, we design a feature that considers relative count of *'li', 'ul', 'a', 'h1', 'h2', 'h3'* tags on a web page.

d. Limited vocab: Due to lack of time to explore how the system performance can be enhanced by using an ontology or lexicon, we intend to see the effect of using a limited domain vocabulary on classifier's performance. We carefully decided on a small set of keywords that could potentially distinguish a course webpage from a non-course web page: *'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'course', 'class', 'syllabus', 'handout', 'homework', 'lecture', 'notes', 'slides', 'solution', 'problem', 'program', 'instructor', 'information', 'project', 'paper', 'guide', 'study', 'activities', 'projects', 'professor', 'office'*.

e. Information gain: Information gain values of different features can be compared/enumerated to measure their relevance in classification.

Our dataset is not biased towards any class; we have equal number of positive and negative examples in the training and test data set. Therefore, we expect a baseline accuracy of 50% (which is when all pages in the test set are assigned the same class). Besides the classification accuracy, we would include F1-score as it considers both the precision and the recall of the test to compute the score.

5 Experiments

In our initial experiment, we used a naive approach to tokenize the web pages and use a bag of words model. During tokenization, html tags are removed thus no information about the layout of the web page is retained. In this process, the URLs in the anchors get removed. Further, stop words are ignored and stemming is used. No special features were used. Term frequencies were used as term weights. Logistic regression was used to classify the web pages. We obtained an accuracy of 78% on the test set. By including features from meta data - relative relevant URL count, relative relevant tag count, document length (before tokenization), we were able to increase the classifier accuracy to 78.5%.

Term Frequency-Inverted Document Frequency has proved to be more effective than Term Frequency for term weights in most IR tasks. Even though, it is computationally intensive to use tfidf weights, in all the future experiments tfidf is used for term weights instead of tf. With bag of words model and no additional features, the classifier achieved an accuracy of 78.5% which is a marginal improvement from the term frequency based term weights, suggesting that attenuating the effect of terms that occur too often in the collection to be meaningful for relevance determination has no bearing on the performance in this context.

Reducing number of features by thresholding document frequency seemed to increase the classification accuracy up to a certain point. We found having a threshold of 20 on document frequency gave highest classification accuracy as shown in Fig 1.

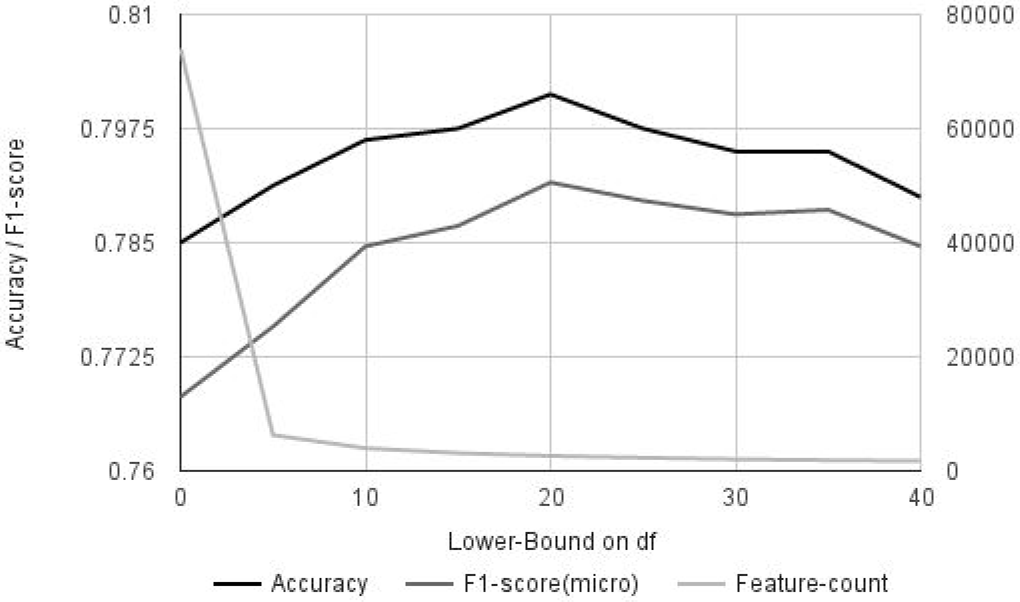

Fig 1: While the number of features dropped drastically, the classification accuracy showed marginal improvement. This suggests that words with low document frequency are perhaps noisy features, and removing those does not affect / improves the classification accuracy.

To explore alternate techniques to reduce the number of features, we experimented by thresholding on univariate linear regression test values. We used the select percentile module for feature selection, wherein the top k% features are selected based on their univariate linear regression test value.

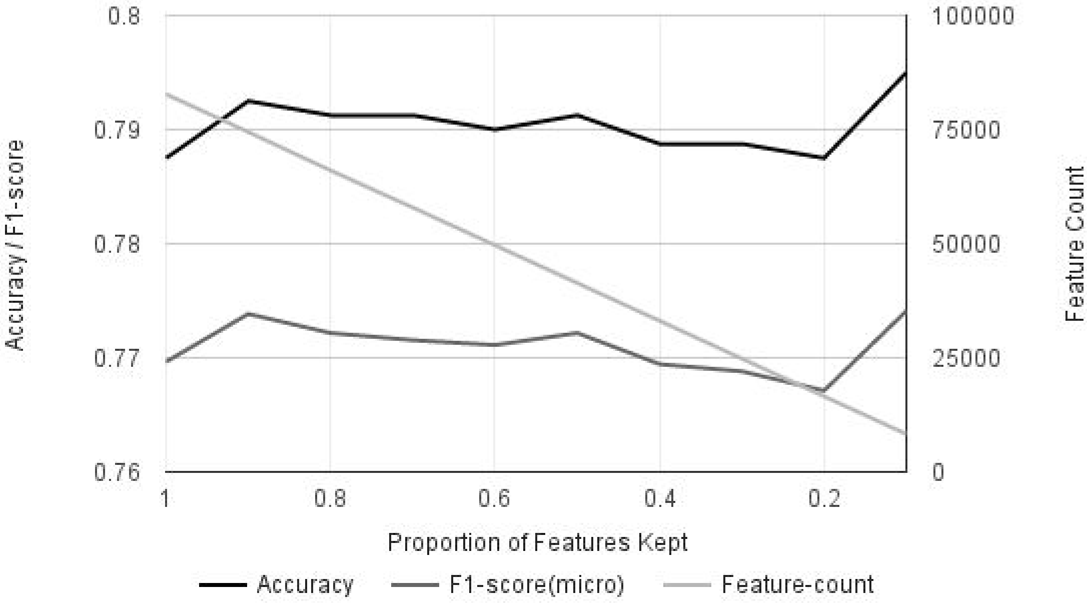

Fig 2: Effect of feature selection based on univariate linear regression test values. The classification accuracy remains nearly constant.

Next, to see the role of document structure, we experimented with using different weights for different tags. In these experiments, no special features were used. Term frequencies were used as term weights. The classification accuracies with logistic regression classifier are listed in Table 3. Differential weighing schemes based on the page layout seems to have little effect on the classification accuracy. However, using exclusively the titles of the web pages seems to do fairly well too even with just 4% of the original number of features when the content of the page is included.

Weight - Title	Weight - Anchors	Weight - Body	No. of features	Classification Accuracy (%)
1	0	0	2951	71.62
0	0	1	53684	62.87
1	1	1	74004	78.50
2	2	1	74004	74.25
3	3	1	74004	74.25
4	4	1	74004	74.12

The restricted vocabulary classifier, uses the frequency of the following words in a web page: *'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'course', 'class', 'syllabus', 'handout', 'homework', 'lecture', 'notes', 'slides', 'solution', 'problem', 'program', 'instructor', 'information', 'project', 'paper', 'guide', 'study',* *'activities', 'projects', 'professor', 'office'*. As expected, the classification accuracy dropped to 68.37% indicating that the features were too few to identify a course web page correctly.

Lastly, we compared the performances of different classifiers on this data set. The results are listed in Table 4.

Classification Algorithm	Classification Accuracy (%)	F1-score
Logistic Regression	78.5	0.7681
Naive Bayes - Multinomial	80.37	0.7942
Naive Bayes - Bernoulli	75.75	0.7252
Decision Trees	73.62	0.6902

While all classification algorithms seem to have similar performances, Multinomial Naive Bayes gave the highest classification accuracy. This is indicative of the fact that for a page to be identified correctly, the frequency of words on the page works better than the presence/absence of words. This seems intuitively correct as student or faculty web pages might have some words (for eg. course, lecture, project) in common with the course web pages, in which case the frequency of these words is a better indicative of whether the page is a course web page or not.

While using decision trees, the following features were found to have highest information gain: course, syllabus, instructor, hours, press, interests, homework, skip, assignments, prerequisite, science, supported, cis, advisory, csci, midterm, textbook, like, special, fall, favorite etc.

6 Conclusion

In this project, we built a classifier to categorize web pages into a set of predefined categories (course web page, non-course web page) based on labeled training data. We explored the reasons why unlike more general text classification, web page classification is more challenging and can take advantage of the semi-structured and metadata information on a web page. We also explored different ways to reduce the feature space without compromising the classifier performance. This gives opportunities to use more computationally intensive and powerful algorithms for the classification task and is suggests that the system is scalable. We also experimented by including novel features in the feature space to capture the semi-structured information on a web page. We also studied how information from different structural components of the page contribute to the accuracy of the classification task. A comparison of performances of different classification algorithms helped us better understand the assumptions made by each of these algorithms.

Individual Contributions

Batu Inal: Looked into various ways of data crawling. Found different references/annotations for our project. Found already well-known and reliable datasets, for e.g: WB->Kb. Helped write up the report and prepared the presentation. Tried to find additional features that we could add to our system rather than scikit but eventually ended up not working. I would like to thank Nikita for being a very good leader to our project and guiding our way of direction in the project from beginning to the end. It was a pleasure to work with Jason and Cici as well.

Cici Lu: Looked into the different data crawling ideas. After further research she directed the team away from looking at Professor websites and lead them to focus on university websites. In addition, she found patterns in course website URLs and suggested looking at course guides to gather course names and ids to use when looking for courses. She provided a diverse set of university websites to train the algorithm, and helped identify which websites were correct or not given the results of the crawler. She helped write the report on the ideas we found and assisted with writing parts of the code.

Jason Xie: Data annotation, lit reviews. Modified base classifiers to work with various data sets. Implemented univariate features selection implementation and performed/recorded experiments.

Nikita Bhutani: Data crawling. Data annotation (jointly with others). Data sampling. Two literature reviews. Implementation: tokenizer, classifiers (jointly with Jason), additional feature design, feature selection (jointly with Jason). Experimentation (jointly with Jason). Analysis of results. Report and presentation (jointly with others).

Acknowledgments

We would like to express our gratitude to Prof. Rada Mihalcea and Shibamouli Lahiri for their guidance and insightful discourse.

References

[1] Qi, X. and Davison, B. D. 2009. Web page classification: Features and algorithms. ACM Comput. Surv. 41, 2, Article 12 (February 2009)
[2] Yu, H., Han, J., Pebl, K.C.-C. Positive example-based learning for web page classification using SVM. In: Proceedings of ACM SIGKDD 2002 (2002)
[3] Min-Yen Kan , Hoang Oanh Nguyen Thi. Fast webpage classification using URL features. Proceedings of the 14th ACM international conference on Information and knowledge management, October 31-November 05, 2005
[4] Dou Shen , Zheng Chen , Qiang Yang , Hua-Jun Zeng , Benyu Zhang , Yuchang Lu , Wei-Ying Ma. Web-page classification through summarization. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, July 25-29, 2004
[5] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003
[6] Shenghuo Zhu , Kai Yu , Yun Chi , Yihong Gong. Combining content and link for classification using matrix factorization. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, July 23-27, 2007

[7] Yiming Yang , Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. Proceedings of the Fourteenth International Conference on Machine Learning, July 08-12, 1997