2011 International Conference on Telecommunication Technology and Applications
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

WEB MINING-BASED UNIVERSITY SEARCH PORTAL

Robert Joseph Ssemmanda, Chien-Sing Lee and Lay-Ki Soon

Faculty of Information Technology, Multimedia University, Cyberjaya

Abstract. This paper intends to show how to mine a benchmark university's Web portal structure, classify the Web structure and use the Web structure as a model to mine the Web portals of other universities, integrating these structures into the benchmark university's Web structure. SVM classification algorithm is used to create a model from the four University data sets, which is then used to classify other web pages. The users' time is saved as they get their results already classified and integrated.

Keywords: Knowledge Management Systems, Information Retrieval and Recommender Systems, Web Mining University Structure.

1. Introduction

Eighty percent of the information on the World Wide Web is in the form of web pages, which in their raw form is text-based (Sanchez, Bautista and Blanco, 2006). This presents an opportunity through which Data mining and text mining can be applied. Patterns and useful information can be extracted from these pages.

The issue of getting information from the Internet is solved by search engines. These engines are mainly involved in comprehensive data indexing. However, it is harder for people to acquire information that they need accurately and quickly because these engines do not or hardly provide individual services, such as customized search queries for each of their users (Yin, Wang, and Zhang, 2007). The quality of the information acquired from these engines highly depends on the query that is entered in the system. In this paper, the query will simply be a university name.

The information acquired from the search engine is then downloaded, the links extracted from this data, and downloaded, mined and then classified. In order to classify the information, a model needs to be built. This model requires a training set. A bench mark data set with already classified web pages is to be used, in this case, the four universities data set (Rish, 2001). The student then will have relevant data, which is classified and categorized. This paper serves as a guide to building a Web-mining-based university search portal.

1.1 Problem Definition

Higher education is a very important aspect of today's employment world. More important is the quality of this education. As a result, students are interested in quality education and search for where they can get it. Thanks to the Internet, students can visit a number of Institutes of Higher Education's Web pages to learn more about them. However, there are a huge number of pages that they have to visit and some of them are relevant and some irrelevant. It is therefore important that not only do they find information on the Internet, but, that which is relevant, presentable and makes sense to them. Information from the Internet should thus be filtered and classified so that it is easy for users to pick what they want.

1.2 Scenario

Suppose a student in Africa is looking for a university to study in, in Malaysia. This student would like to know the courses offered and more information about them such as the syllabus of the university. He wants to compare with other universities. The student needs to identify useful information and extract what is necessary for him to make a sound decision.

As seen from the scenario above, the student needs to find information regarding a university or universities. The student needs to be able to acquire information from the huge amount of data available. Hence, the student needs to access the Internet, and then query for the information he/she is interested in. There are many ways to get the information that the student requires. But in the case of the Internet, the best alternative is a Web search engine, where the student will submit queries and results shall be returned. The application should be able to acquire this data and store it. This involves acquiring the search results and downloading the corresponding HTML pages to the URL links in the search results.

Looking at the data returned, it is just raw data, and the student might not make much sense of what he/she has, which might be a lot of information. It would be great if the student is able to extract only the relevant data that he/she needs. Hence, there is a need to look at information extraction or data mining. Once the data has been downloaded, the next task is to process this data into a format that can be mined, where useful information can be extracted.

1.3 Research Objectives

The objectives for this study are:

1. Mine a benchmark university's Web portal structures, classify the Web structure and use the Web structure as a model.
2. Mine the Web portal of other universities and integrate these structures into the benchmark university's Web structure.

2. Methodology

The first task is to acquire information from the Internet. This is achieved by using search engines. For this paper we used Google AJAX API, which can connect to the Google servers and return the search results for the entered query. The results returned from the Google servers are basically HTML links to the various matching categories of your search term or query. In this case, universities' websites are linked to the websites' titles. The next task is to download the pages that these links connect to.

Next, we convert these pages to text format. This process is known as HTML de-tagging. It involves removing HTML tags and syntax from the web pages such as "<html>", "</html>", "
" etc. This converts the pages to plain text files. Then we carry out text mining on these files. Text mining refers to the discovery of non-trivial, previously unknown and potentially useful knowledge from a collection of texts (Sanchez, Bautista and Blanco, 2006).

In order to carry out text mining on the pages downloaded, they must be in a proper mineable format. This involves removing stop words from the text such as "I", "and", "is". Stemming is then carried out. Stemming is the process of reducing inflected or derived words to their stem, base or root. For example, running is reduced to run. After this process is done, the pages are now ready to be mined. In this paper, WEKA tools are used to filter the data and classify it. The WEKA file conversion tool is used to convert the text files to the .arff format that is used by WEKA, then its filtering tool is used to carry out stemming and removal of stop words, on the downloaded web pages.

In order to classify the pages downloaded, we need to create and train a model. To do this, it is important to find a training set to train the model. The training set that we used is the benchmark Carnegie Mellon University (CMU) dataset referenced from (CMU, 2010), which includes among others, universities such as Cornell, Texas, Washington and Wisconsin Web pages collected from their computer science departments in January 1997 by the World Wide Knowledge Base (Web->Kb) project of the CMU text learning group. This benchmark dataset consists of 8282 web pages and others from miscellaneous sources. Out of the 8282 web pages, 867 are from Cornell, 827 from Texas, 1205 from Washington, 1263 from Wisconsin and 4129 from other universities. The CMU web pages were classified into seven categories, i.e., student, faculty, staff, department, course, project and other. There were 1641 pages for student, 1124 pages for faculty, 137 pages for staff, 182 pages for department, 930 pages for course, 504 pages for project and 3764 pages for other.

This data set was used to create and train the model as, what we needed was to have the web pages presented and categorized in any of the categories in Table 2, i.e., student, faculty, staff, department, course, project and others, as determined by the model. In order to create a model, a classification algorithm has to be used from those available i.e SVM (Support Vector Machine), Naive Bayes and Decision trees (Mitchell, 1997). From the research carried out by Chan and Wong (2010), the best algorithm to use for text mining was Support Vector Machine (SVM). Not only did it give the highest accuracy of correctly classified web pages, but also the best overall results in comparison to the other algorithms such as Naive Bayes, Decision trees and Neural networks.

According to Cortes and Vapnik (1995), Support Vector Machines (SVM) is a supervised learning method used for classification. It is the application of linear classification techniques on non-linear data. It includes an attribute (predictor variable) and a transformed attribute that is used to define the hyperplane (feature). The task of choosing the most suitable representation is known as feature selection. A set of features that describes one case is called a vector.. The vectors near the hyperplane are the support vectors.

Using the WEKA classification tool, SVM classification method and the four universities data set, a model was created. The WEKA filtering tool was used to convert the web pages that were converted to plain text, ready for text mining, into sparse ARFF files. Sparse ARFF are similar to ARFF files except that data values of 0 are not represented.

For the new data to be applied to the model, the testing set should be compatible with the training set. In other words, both the training set and testing set should have the same number of attributes and all the words appearing in the training set should appear in the test set. The words in the test set that do not appear in the training set should be removed.

In the sparse ARFF format, words have been converted to attributes. We used 1 to represent present attributes and 0 absent attributes. To ensure that the testing set is compatible with the training set, first, we added the classes we wanted the data classified in, i.e., staff, department etc. to the training set as the class, and removed the previous class. Then, we looped through the training set and removed all the attributes that were not in the testing set. After this was done, since the remaining attributes in the training set were in alphabetical order and the testing set was also in alphabetical order, we added the words with the same index from the testing set to the training set.

This model was then applied to the testing set. The model predicted the probability of the testing set falling into any of the classes, i.e., staff, department etc. The class with the highest probability was then assigned to the testing set. This set can then be displayed to the user in a tree form, i.e., each node of the tree showing a different class and the web pages that fall in that class, by title or HTML link.

3. Application Setup

The user can query the system, train the model, view the results, view the preview of a specific result webpage and open its URL in a browser. The steps are as follow:

Step 0: The user must load or train the model that would be used for classification.

Step 1: After the user has entered the University's name, the university's name is then sent to Google through the Google API so as to get the search results for this university.

Step 2: After the results from Google have been returned, the links are used to download the webpages related to each returned result link, then stored.

Step 3: The HTML tags in the downloaded pages are removed to convert the pages to normal text file and stored.

Step 4: The text files are converted to .arff files so that they can be compatible with the WEKA application.

Step 5: The text file data is converted into a sparse format so that it would be classifiable.

Step 6: Ensure compatibility with the training data set by eliminating terms and attributes in the downloaded pages that are not in the training set and adding attributes to the downloaded pages (testing set) that are in the training set but do not appear in the downloaded pages.

Step 7: The model created in step 0 is applied to the downloaded pages (the testing set). The result from this step would be the classified test set.

Step 8: The user is shown the results from the classification process, and the category in which each of the downloaded pages were categorized in relation to the classes in the training set, i.e., course, department, faculty, other, project, staff and student.

4. Testing

To test the system, we entered the universities' names into the system, and then measured the accuracy, precision and recall of the results returned by the system. In this way, we would be able to measure how accurate the system is. The results are shown in a contingency table (confusion matrix). An example of a contingency table is shown in Table 1.

Other than accuracy, we can calculate Precision, Recall, and F-Measure of each of the different categories. Precision is a ratio or percentage showing how many web pages were correctly identified out of the total number of pages available. Recall is the ratio or percentage showing how many pages in a class

were correctly matched with respect to the total number of pages in that class. F- measure is the harmonic mean of precision and recall. The outcomes of our simulation are shown in Tables 2 to 11.

TABLE 1. EXAMPLE OF THE EXPERIMENTS' RESULTS

Classified as	A	B	C	D	E	F	G	Explanation
A = Course	N1	N2	N3	N4	N5	N6	N7	The highlighted cells in Table 1 are the ones used to calculate the overall accuracy of the model. Accuracy is the ratio or percentage of pages that were correctly classified. Using course category as example: Accuracy = N1+N9+N17+N25+N33+N41+N49) / ($\sum_{K=1}^{49}$ Nk)) * 100
B = Department	N8	N9	N10	N11	N12	N13	N14	
C = Faculty	N15	N16	N17	N18	N19	N20	N21	
D = Other	N22	N23	N24	N25	N26	N27	N28	
E = Project	N29	N30	N31	N32	N33	N34	N35	Precision $_{course}$ = N1 /(N1+N8+N15+N22+N29+N36+N43)
F = Staff	N36	N37	N38	N39	N40	N41	N42	Recall $_{course}$ = N1 / (N1+N2+N3+N4+N5+N6+N7)
G = Student	N43	N44	N45	N46	N47	N48	N49	

Query 1: Multimedia University (MMU).

Table 2. CONFUSION MATRIX (MMU)

A	B	C	D	E	F	G	Classified as
0	0	0	1	0	0	0	A = Course
0	0	0	0	0	0	0	B = Department
0	0	1	5	0	0	0	C = Faculty
0	0	0	8	1	0	0	D = Other
0	0	0	0	0	0	0	E = Project
0	0	0	0	0	0	0	F = Staff
0	0	0	0	0	0	0	G = Student

Accuracy = 9/16 = 56.25 %

Table 3. ACCURACY BY CLASS (MMU).

Category	Precision	Recall
Course	N/A	0.00
Department	N/A	N/A
Faculty	1.00	0.17
Other	0.57	0.89
Project	0.00	N/A
Staff	N/A	N/A
Student	N/A	N/A
Average	**0.52**	**0.35**

Query 2: Stanford University (SU)

Table 4. CONFUSION MATRIX (SU)

A	B	C	D	E	F	G	Classified as
1	0	0	0	0	0	0	A = Course
0	0	0	0	0	0	0	B = Department
0	0	6	4	0	0	0	C = Faculty
0	0	2	3	0	0	0	D = Other
0	0	0	0	0	0	0	E = Project
0	0	0	0	0	0	0	F = Staff
0	0	0	0	0	0	0	G = Student

Accuracy = 10/16 = 62.5 %

Table 5. ACCURACY BY CLASS (SU).

Category	Precision	Recall
Course	1.00	1.00
Department	N/A	N/A
Faculty	0.75	0.60
Other	0.43	0.60
Project	N/A	N/A
Staff	N/A	N/A
Student	N/A	N/A
Average	**0.73**	**0.73**

Query 3: M.I.T

Table 6. CONFUSION MATRIX (MIT)

A	B	C	D	E	F	G	Classified as
0	0	0	0	0	0	0	A = Course
0	0	0	0	0	0	0	B = Department
0	0	1	1	0	0	0	C = Faculty
0	3	4	7	0	0	0	D = Other
0	0	0	0	0	0	0	E = Project
0	0	0	0	0	0	0	F = Staff
0	0	0	0	0	0	0	G = Student

Accuracy = 8/16 = 50%

Table 7. ACCURACY BY CLASS (MIT)

Category	Precision	Recall
Course	N/A	N/A
Department	0.00	N/A
Faculty	0.25	0.50
Other	0.88	0.50
Project	N/A	N/A
Staff	N/A	N/A
Student	N/A	N/A
Average	**0.38**	**0.50**

Query 4 : Universiti Malaya (UM)

Table 8. CONFUSION MATRIX (UM)

A	B	C	D	E	F	G	Classified as
0	0	0	0	0	0	0	A = Course
0	0	0	0	0	0	0	B = Department
0	0	2	4	0	0	0	C = Faculty
0	0	2	8	0	0	0	D = Other
0	0	0	0	0	0	0	E = Project
0	0	0	0	0	0	0	F = Staff
0	0	0	0	0	0	0	G = Student

Accuracy = 10/16 = 62.5 %

Table 9. ACCURACY BY CLASS (UM)

Category	Precision	Recall
Course	N/A	N/A
Department	N/A	N/A
Faculty	0.50	0.33
Other	0.67	0.80
Project	N/A	N/A
Staff	N/A	N/A
Student	N/A	N/A
Average	**0.00**	**0.17**

Query 5 : Carnegie Mellon University

Table 10. CONFUSION MATRIX (CMU)

A	B	C	D	E	F	G	Classified as
1	0	0	0	0	0	0	A = Course
0	0	0	0	0	0	0	B = Department
0	0	4	0	0	0	0	C = Faculty
0	1	1	9	0	0	0	D = Other
0	0	0	0	0	0	0	E = Project
0	0	0	0	0	0	0	F = Staff
0	0	0	0	0	0	0	G = Student

Accuracy = 14/16 = 87.5%

Table 11. ACCURACY BY CLASS (CMU)

Category	Precision	Recall
Course	1.00	1.00
Department	0.00	N/A
Faculty	0.80	1.00
Other	1.00	0.82
Project	N/A	N/A
Staff	N/A	N/A
Student	N/A	N/A
Average	**0.70**	**0.94**

The cells in the tables above with "N/A" indicate that, that the category was empty. In other words, of all the pages downloaded, none was classified to that class. The Google API restricted the result links returned to a maximum of 16 links. Hence for each query, there were 16 links returned and classified.

From the results above, the accuracy of the system ranged from 50% to 87.5%, precision from 0.38 to 0.73 and recall from 0.35 to 0.94. It should be noted that the accuracy increased significantly to 87.5% and recall to 0.94 when the university name queried was among the four universities that contributed to the training set. The more the similarity to the training set, the higher the accuracy of the application.

The overall accuracy of the program should be higher. However, the training set provided was already stemmed, and the stemming algorithm was not mentioned. In this case, the test set stemming algorithm did not work as well as that used in the training set. As a result, there was a loss of words during the standardization step, i.e., when the testing set was being edited to be compatible with the model and the stemming step. These words were very important because they were what the model relied on to make a prediction and classify the page. The ways in which this could be solved was either by making sure the training set and test set use the same stemming algorithm, or improve the stemming algorithm to be as good as the one which was used on the training set.

At the moment, due to the stemming algorithm used in the test set, accuracy was 0.63, precision 0.582 and recall being 0.618. We intend to improve on the stemming algorithm.

5. Conclusion

We have presented how to build a Web-based university search portal, i.e. by mining a benchmark university's Web portal structures, classifying the Web structure and using the Web structure as a model. We have also mined the Web portal of other universities and integrated these structures into the benchmark university's Web structure. Accuracy, precision and recall evaluation scores are promising.

6. References

[1] D. Sanchez, M. J. Bautista and I. Blanco. *Text knowledge mining: an alternative to text data mining.* University of Granda Spain, Department of Computer Science and A.I. 2006.

[2] S. Yin, G. Wangg, W. Zhang. *Research and Implementation of Classification Algorithm on Web Text Mining*. Faculty of Computer and Information Science, Southwest University Chongqing, China, 2007.

[3] I. Rish. An empirical study of the naive Bayes classifier. 2001 *Workshop on Empirical Methods in Artificial Intelligence*. International Joint Conferences on Artificial Intelligence (IJCAI)2001.

[4] Carnegie Mellon University's *The Four Universities Data Set*, retrieved October 28, 2010 from www.cs.cmu.edu/afs/cs/project/theo-20/www/data/.

[5] T. Mitchell. Machine Learning, McGraw Hill, 1997.

[6] K. X. Chan and P. W. Wong. *Web Page Classification*. Unpublished Final Year Project report. 2010.

[7] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20, 1995.