Document Categorizer Agent for Computer Science Academic Papers

Khalifa Chekima[1], Patricia Anthony[2]

[1]UMS-MIMOS Center of Excellence in Semantic Agent
School of Engineering and Information Technology,
Universiti Malaysia Sabah (UMS),
Sabah, MALAYSIA
hisham_chekima@yahoo.com

[2]UMS-MIMOS Center of Excellence in Semantic Agent
School of Engineering and Information Technology,
Universiti Malaysia Sabah (UMS),
Sabah, MALAYSIA
panthony@ums.edu.my

ABSTRACT

This paper presents Document Categorizer Agent that categorizes computer science academic papers in .pdf format such as journals and proceedings. In this paper, we propose the use of set of term stored in a database to categorize computer science papers. Few methods and algorithms from related work are considered in improving the categorization process. We have evaluated our document categorizer agent on a number of computer science papers. The categorization process is done by parsing the document, calculating the frequency of each term and matching the terms found with the dataset found in the database. We have shown that the use of this term database can be used to categorize documents. The categorizer agent focuses on categorizing the text document into predetermined category based on the extracted keyword. This can help in making the searching process more efficient and saves the user's time in searching for the desired document.

Keywords
Artificial intelligence, Information retrieval, document categorization, web mining, web classification, Ontology.

1.0 INTRODUCTION

With the rapid development of World Wide Web (WWW), a huge amount of information is now accessible to the web users. This phenomenon has attracted academic users to publish their research papers online and at the same time download and share academic papers among them through WWW. In this paper, we focus on the categorization problem of computer science academic papers. Document categorizer agent is proposed to help categorize different computer science papers into different categories. Document categorizer agent is a decision making agent that can make an intelligent decision. When a new document is downloaded, this agent will parse the content of document and categorize the document based on its keywords into the predetermined category. It can match the user query and returns a list of related documents to user. The problem faced here is to categorize a document which can take up a large amount of time by the user if it is done manually. The user will have to read each of the documents to decide which category is suitable. Moreover, categorizing manually by the user may result in a wrong categorization since the categorization is based on the understanding of the user. Each category has its own specific keywords in database. If the document's keywords match with the keyword in a given category, the document will be placed under that category. Before we developed the categorizer agent, we have reviewed few existing methods to try and capture the advantages of these methods and include them in our system to make it more effective and more reliable. The rests of the paper is organized as follows. In Section 2, we describe the related work in document and web classification. Our document categorizer agent is described in Section 3. The experimental evaluations are reported in Section 4, and finally the conclusion and future work are elaborated in Section 5.

2.0 RELATED WORK

Academic papers in .pdf format consist of both text and images. As the text features are believed to provide the primary content information about documents, the simplest approach is to use word frequency. Depending on the context features used, we divide the other works on documents classification into, Support Vector Machine (SVM) that uses individual features, Neural Networks, Multiple Similarity-Based Models and Data Summarization and Clustering.. In the Support Vector Machine (SVM), different context features are combined to improve the performance of the classifier (*Fang, Mikroyannidis & Theodoulidis)*. There are five classification methods involved in this process. The first method considered text only. The second method looked at the title and headings. The third method took into account the URL and headings. As for the fourth method, the text, title, headings, URL and the anchor text is considered. The last method made use oftitle, headings, URL and anchor text. Based on the experimental evaluation, it was found that the fourth method has shown best categorization among all.

Another work used Artificial Neural Networks (ANN) to categorize documents (Ruize & Srinivas). In this work, two ANN techniques Multilayer Perceptron and Self Organizing Map (SOM) are compared against symbolic machine learning algorithms, C4.5 decision tree and PART decision rules. The results obtained showed that MPL and SOM performed better in categorizing document compared to C4.5 and PART.

Lai and Laim (YEAR) proposed a meta-model framework which combines the strength of GIS algorithm as well as state-of-the-art existing algorithms using multivariate regression analysis on document feature characteristics. Generalized instance set (GIS) algorithm is an algorithm which combines the advantages of linear classifiers and k-nearest neighbour algorithm. This algorithm had shown that its performance is better than the other algorithms but it is limited to certain areas only.

WebACE is an agent that explores and categorizes document on the World Wide Web (PUT REFERENCE HERE). The heart of the agent is the usage of automatic categorization combined with a process for generating new queries used to search for related documents and filtering the related documents to extract the set of documents that are most closely related to the starting set.

3.0 THE DOCUMENT CATEGORIZER AGENT

In our work the Document Categorize Agent is only limited to parsing the document with .pdf format. The main reason for doing this is because most academic papers are in the form of a PDF file. Currently, the document categorizer agent is concerned with text document only. As such the images in the PDF document will be ignored since we would like to focus on parsing the content of the document. The categorization process is described in Figure 1.

It is assumed that the set downloaded documents will be stored in a repository under a local folder called unCategory. When a user wants to categorize the document, the document categorizer agent will prompt the user to key in the title and select the type of the document. This process can also be done automatically without the interaction between user and the system depending on the user' srequirement. Before parsing the content of the document, the agent will match the title and type of the document with the document database, if it already exists in the document database, the agent will not categorize the document and delete the document from the unCategory folder, and prompts the user of the existence of the document. This is to avoid duplication of document in the database.If the document does not exist in the document database, the agent will parse the content of the document. As the agent cannot read directly from the .pdf file, a pdftotxt software is used to convert .pdf documents into .txt file to allow word filtering process.

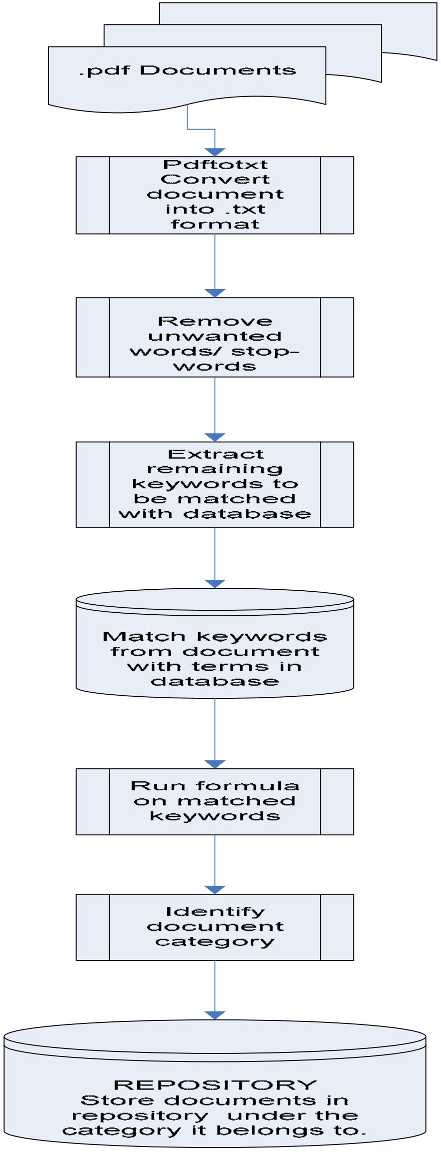
.pdf Documents
Pdftotxt Convert document into .txt format
Remove unwanted words/ stop-words
Extract remaining keywords to be matched with database
Match keywords from document with terms in database
Run formula on matched keywords
Identify document category
REPOSITORY
Store documents in repository under the category it belongs to.

Figure 1: The categorization Process

The agent will filter out all the high frequency words, stop-words and unwanted words, such as prepositions and conjunctions. After filtering out all the high frequency words, the agent will then extract all the remaining words in the document as the document keywords.

The second stage is to match singular and plural words. A technique called plural-matching rule is applied to these document's keywords before they are matched with the keyword database. The plural-matching rules delete the "s" only behind a keyword, for example, agents become agent. As a result, when a keyword matches the ontology database, the keyword "agent" or "agents" can be considered as the same keyword and are matched with the term in the database by using "agent" only. Thus, the term "agent" in the ontology database can recognize "agents" and will not just ignore it.

The terms database used is based on the ACM Computing Classification System, which is a subject classification system for computer science devised by the Association for Computing Machinery. The term database contains the terms of each predetermined category. We defined a formula to calculate the expected utility to decide which category the document should belong to. The classification has two possible outcomes. The first outcome is the number of terms that match with the term database in one category only, and the second outcome is the total number of the terms are matched in more than one category. Here, we assume that the total number of the terms matched is less important than the number of term matched for this agent. Hence, we assign the probability of 0.7 for the second outcome and 0.3 for the first outcome. Each category can theb be measured by using the method below:

Let,

O_1, O_2, O_3, …O_n represent the possible outcomes of an action.

P (O_n) = probability assigned to outcome O_n

V(On) = the value of outcome On

The expected value of an action A is:

$EU(A) = (V(O_1)*P(O_1)) + (V(O_2)*P(O_2)) + \ldots+(V(O_n)*P(O_n))$

The document will be stored in the category for which the expected value is the highest.

4.0 INITIAL EXPERIMENTAL EVALUATION

To test the accuracy of our categorizer agent, one hundred papers were collected from five different computer science subcategories, Computer Graphics, Artificial Intelligence, Software Engineering, Networking and Database Management. These same set of papers were categorized by 10 different common computer science users. We then run our categorizer agents, to categorize these 100 papers. The performance of our agent is then compared with the accuracy of the common computer science users.

The result of this categorization process is shown in Figure 2. It can be observed here that the document categorizer agent performed slightly better and compared to the manual categorization performed by humans. The agent recorded an accuracy of 67.80% compared 66.60% obtained by the computer users. This early results shows that it is possible to develop an agent that can be used to categorize academic papers.

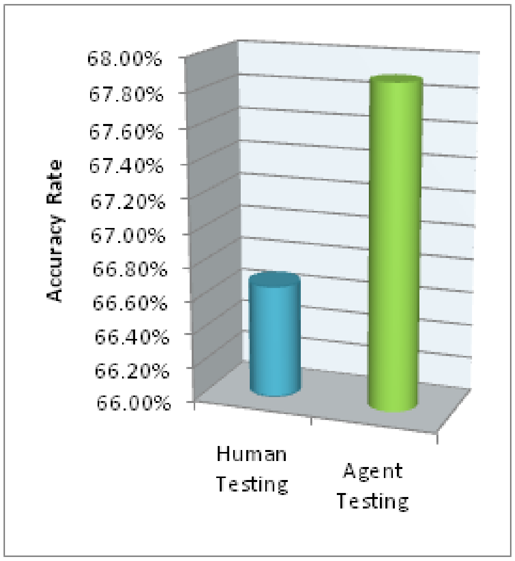

Figure 2: Agent's Performance against Human's Performance in Categorizing Computer Science Papers

5.0 CONCLUSION AND FUTURE WORK

In this paper, we have studied the the different techniques that can be employed to categorize academic papers. Even though, we are using a very simple algorithm to categorize the papers, the experimental result has shown the categorizer agent was able to performed better than the 10 computer users. At the moment, we are limiting the categorization to computer science papers only. However, we plan to extend this algorithm so that it can cater for any category of academic paper.

While our results are encouraging, there are still much improvements that need to be made. We need to improve the proposed algorithm to include more complex technique such as the used of DBPedia to assist in the categorization process. We would also like to combine techniques such as Hierarchical Agglomerative Clustering or K-Mean Clustering to produce a higher accuracy rate.

ACKNOWLEDGEMENT

We wish to acknowledge (Tey Lay Fun & Nelson Wi Weng Kim) for their contributions to this research project.

REFERENCES

Ruiz, M. E., & Srinivas, P. (2001). *“Hierarchical Text Categorization Using Neural Networks”*.

Fang, R., Mikroyannidis, A., and Theodoulidis, B. (2006). A Voting Method for the Classification of Web Pages Proceedings of the 2006, *IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*.

Sun, A. (2002). Web Classification Using Support Vector Machine. *Proceedings of the 4th international workshop on Web information and data management.*

Lai, K-Y., & Lam, W. (2001). Automatic Textual Document Categorization Using Multiple Similarity-Based Models, *SIAM.*

Han, E-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1998) WebACE: A Web Agent for Documents Categorization and Exploration. *Proceedings of the second international conference on Autonomous agents.*