American Journal of Economics and Business Administration 3 (1): 213-218, 2011
ISSN 1945-5488

Categorizer Agent for Electronic Computer Science Academic Papers

Khalifa Chekima and Patricia Anthony
UMS-MIMOS Center of Excellence in Semantic Agent
School of Engineering and Information Technology,
University Malaysia Sabah, Sabah, Malaysia

Abstract: Problem statement: 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, at the same time downloading and sharing academic papers among them through WWW. Categorizing a document manually can take up considerable amount of user's time whereby user will have to read each of the documents to decide which category it is suitable. **Approach:** Our research study proposes the use of set of terms stored in a database to categorize computer science papers. The categorizer agent focuses on categorizing the text document into predetermined categories based on the extracted keyword. **Results:** 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 in the database. **Conclusion:** The Categorizer Agent proposed in this research paper is evaluated as a good approach to categorize electronic papers. Moreover, the results indicated that the use of this term database is a sustainable way to categorize computer science electronic documents.

Key words: Artificial intelligence, information retrieval, document categorization, data mining, Support Vector Machine (SVM), Artificial Neural Networks (ANN), Generalized Instance Set (GIS), Self Organizing Map (SOM)

INTRODUCTION

In the era of information technology, information like books, journals and articles are converted into electronic version such as e-book and e-journal; this can also be known as online text document. The number of online text document grows rapidly day after day. To manage this information, a manual assignment of text documents is the earliest systems used to categorize different types of items. The categorizing of text documents manually is done by looking at the overview of the content of the document and deciding to which category the document should belong to. As the volume of the text documents become larger, the process to decide which category the text document should belong to becomes more difficult. This makes maintaining of large electronic document time-consuming. Besides, the probability of assigning the wrong category of text document can occur since the assigning process is done by human based on the personal understanding and background. For that reasons, this has caught researches attention to look at document clustering and categorization. According to Sree *et al*. (2008), one of the way to enhance the quality of clustering is by using a Cellular Automata Classifier for information retrieval. In this study, we focus on developing a Categorizer Agent that can perform better and faster in categorizing Computer Science papers into subcategories. Document categorizer agent is proposed to help categorize different computer science papers into different sub-categories. We perform categorization process to make it easier for researchers to organize and search for documents. Once a document is categorized in the right category, a user would be able to open the relevant folder to find the target paper. Hence, to assist user to automate and speed up the categorization process we proposed a document categorizer agent.

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. In general, a software agent is a program that performs some information

Corresponding Author: Khalifa Chekima, UMS-MIMOS Center of Excellence in Semantic Agent,
School of Engineering and Information Technology, University Malaysia Sabah, Sabah, Malaysia

gathering or processing task in the background.

Typically, an agent is given a very small and well-defined task. There is no unique definition of what constitutes an agent, but according to Russell and Norvig (2009), “An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.” Whereas Maes (1995) described that "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment and by doing so realize a set of goals or tasks for which they are designed."

A software agent can be refined as a computer program which works toward goals in a dynamic environment on behalf of another entity, possibly over an extended period of time, without continuous direct supervision or control and exhibits a significant degree of flexibility and even creativity in how it seeks and attempts to transform goals into action tasks.

Agent characteristics: In a software agent domain, different characteristics are important for different domains of applications. Although there are several but not universally accepted about characteristics of agents, according to Wooldridge and Jennings (1995), among those characteristics typically most important terms are as following.

Autonomy: The autonomous agents operate with minimum intervention of humans or others and have some kind of control over their actions and internal state. Full autonomy may even be undesirable due to uncontrollable actions of the agent that may cause infinite costs and unpredictable consequences. Some control over agent’s behavior and reasonable restrictions would be preferable.

Pro-activeness: An intelligent agent is capable of exhibiting proactive behavior. It consists of pro-active purposeful that is goal-oriented. It attempts to accomplish its goals but does not simply acts in response to the environment. In other words, preconditions described which procedures need to be satisfied in order to be executed when writing a procedure for objectives design. When preconditions are met and procedures are executed correctly, then the post-conditions specified will be true. Besides, agents having pro-activeness are able to exhibit goal-directed behavior by taking the initiative.

Goal-directness: As epitomized via execution of a simple procedure has two inherent limitations. First, it assumes that while the procedure is executing, the preconditions remains valid which means the environment does not change. Second, it presupposes that the goal and the conditions for pursuing that goal remain valid at least until the procedure terminates. Both assumptions are not realistic in complex, dynamic and uncertain environments. Agents should not only blindly attempt to achieve their own goals, but they should able to perceive changes in environment and responds accordingly in time. This reactiveness characteristic involves sensing and acting.

Social ability: Typically agents may live and act in an environment along with other agents, human and artificial as well. Social ability means that agent is able to coordinate, cooperate, negotiate and even compete with others in order to achieve one’s objectives. It should be able to communicate with other agents, including people as well via some kind of agent communication language.

Types of agent: There are several types of software agents, including mobile agents, interface agents, collaborative agents, information agents, reactive agents and hybrid agents (Hyacinth, 1996).

Mobile agents: Mobile agents are processes dispatched from a source computer to accomplish a specified task (Chess *et al*., 1995). Mobile agent is a type of software agent with features of autonomy, learning, social ability and most distinguish and important is mobility. Example of mobile agent is Aglet. Aglets are Java objects that can move from one host on the Internet to another.

Interface agents: An interface agent could be considered to be a "robot" whose sensors and effectors are the input and output capabilities of the interface and for that reason are sometimes also referred to as "softbots" (Etzioni and Weld, 1994). Interface agents allow systems to monitor the user's actions, develop models of user abilities and automatically help out when problems arise. Example of interface agent is Open Sesame, an interface agent for the MacOS Finder which learns user behavior and offers automation and customization suggestions to the user. It can schedule both time and event-based tasks.

Collaborative agents: Collaborative agents interact with each other to share information or barter for specialized services to effect a deliberate synergism. While each agent may uniquely speak the protocol of a particular operating environment, they generally share a common

interface language which enables them to request specialized services from their brethren as required (David, 1997). A collaborative agent is a software program that helps users solve problems, especially in complex or unfamiliar domains, by correcting errors, suggesting what to do next and taking care of low-level details.

Information agents: An information agent is an agent that has access to at least one and potentially many information sources and is able to collate and manipulate information obtained from these sources in order to answer queries posed by users and other information agents (Papazoglou et al., 1992). The information sources may be of many types, including, for example, traditional databases as well as other information agents. Information agents perform the role of managing, manipulating or collating information from many distributed sources. In this study, we will be developing a specific information agent that is able to categorize the papers based on its content.

Reactive agents: Maes (1991) highlights the three key ideas which underpin reactive agents that include emergent functionality, task decomposition and operate on representations. Firstly, emergent functionality which means the dynamics of the interaction leads to the emergent complexity. Hence, there is no a priori specification or plan of the behavior of the set-up of reactive agents. Secondly, is that of task decomposition. A reactive agent is viewed as a collection of modules which operate autonomously and are responsible for specific tasks which may be sensing, motor control, computations and others. Thirdly, reactive agents tend to operate on representations which are close to raw sensor data, in contrast to the high-level symbolic representations that abound in the other types of agents discussed so far. Interactions between reactive agents are provided by signals, as stimuli-reactions.

Hybrid agents: Hybrid approach, according to Maes (1990), brought together some of the strengths of both the deliberative and reactive paradigms. Hence, hybrid agents refer to those whose constitution is a combination of two or more agent philosophies within a singular agent. These philosophies include a mobile philosophy, an interface agent philosophy, collaborative agent philosophy and others. Hybrid agents consist of an agent knowledge base and its associated control unit sitting on top of the perception-action component which also handles the low-level communications.

By looking at the agent characteristics such as autonomy, pro-activeness, social ability and goal directness, this can result in developing an intelligent agent that suites our system requirement.

In this study we would like to capitalize of the agent's functionalities to develop a categorizer agent that is able to demonstrate some of the characteristics described. The remainder of the paper is organized as follows. In next section, we describe the method that we use to design and develop our categorizer agent. This is followed by the discussion on the results that we have obtained. The related works are discussed next and the paper ends with the conclusion and future work.

MATERIALS AND METHODS

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.

In our study, the Document Categorizer 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, 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 Fig. 1.

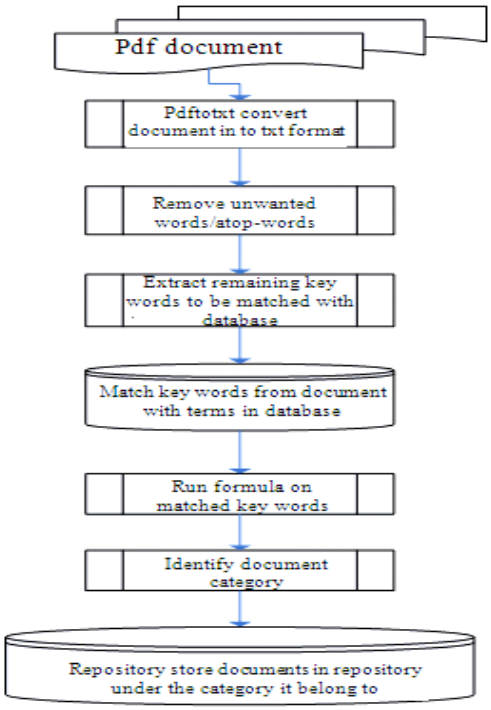

Fig. 1: Agent's performance against human's performance in categorizing computer science papers

It is assumed that the set of downloaded documents will be stored in a repository under a local folder called unCategory. The monitoring of the document downloading process is monitored by another agent called the User Monitoring Agent. This agent monitors the browsing behavior of the user and when it detects that a download process is taking place, the agent will copy a copy of the document being downloaded into the unCategory folder. This agent will then notify the Categorizer Agent to perform the categorization process for that document.

The Categorizer Agent will then read the document from the the unCategory folder. Before parsing the content of the document, the agent will match the title and type of the document with the documents in database to ensure that this document is a new document. If the term and title already existed in the database, the agent will not categorize the document. It will delete the document from the unCategory folder.. If the document does not exist in the database, the agent will proceed to 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. In this 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.

In the second stage, the agent will perform matching between the 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, "computers" becomes "computer". The agent will search the term "computer" in the database to ensure that "computer" is a valid term. If there is a match, the keyword "agent" and "agents" are considered to have the same keyword.

The terms stored in the database 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 terms database contains the terms of each predetermined category. For example, a paper categorized as Artificial Intelligence might contain terms such as "biorobotic", "decision support", "deduction", "learning" etc. Fig. 2 shows a snapshot of the terms dictionary that we have used. The frequency of these keywords will be captured and will be used to determine the document's category. We defined a formula to calculate the expected utility to decide which category the document should belong to. The categorization process has two possible outcomes. The first outcome is the terms that matches with the terms database fall in one category only and the second outcome is 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 terms matched. Hence, we assign the probability of 0.7 for the second outcome and 0.3 for the first outcome. Each category can then 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 in which the expected value is the highest.

Fig. 2: Terms in database

RESULTS

Initial experimental: To test the accuracy of our categorizer agent, 100 papers were collected from five different computer science subcategories namely, computer graphics, artificial intelligence, software engineering, computer networking and database management. For the same set of papers we asked a group of final year students to read the papers and categorize them into any of the five subcategories. To save time, we asked each student to read 10 papers each. We then run our categorizer agent, to categorize these 100 papers.

For the human users, the categorization is done manually. The student will have to read the paper's content and based on the reader's personal understanding, the reader will decide to which category that particular paper belongs to. The categorization accuracy of our agent is then compared with the accuracy of the students' categorization.

The result of this categorization process is shown in Fig. 3. It can be observed that the document categorizer agent performed much faster and better compared to the manual categorization performed by the students. The agent recorded an accuracy of 67.80% compared to the 66.60% obtained by the students. The agent performed slightly better than the students in that it achieved a higher accuracy and outperformed the students by 1.11%. This early result shows that by using a very simple technique, the agent is able to categorize the academic papers better and faster compared to the traditional way of categorizing by reading and analyzing manually. The performance of the agent can be further improved by refining the categorization technique. However, the result obtained is sufficient to show that agent can be utilized to perform document categorization in an automatic manner.

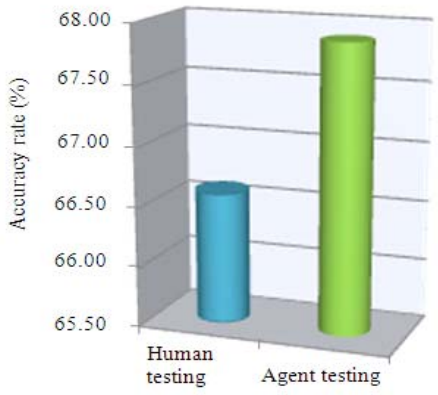

Fig. 3: Agent's Performance against Human's Performance in Categorizing Computer Science Papers

DISCUSSION

The process of categorizing academic papers using intelligent agent is one of the ways to help academicians in identifying academic papers related to their fields of interest. We have studied a few papers that helped us identify the important features that should be present in order to develop effective Categorizer Agent. As academic papers in .pdf format consist of text and images, the text features are believed to provide the primary content information about documents. So the simplest approach is to use word frequency. Depending on the context features used, we have divided 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 et al., 2006). There are five classification methods involved in this process. The first method considered text only document. The second method looked at the title and headings of the paper. 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 of title, headings, URL and anchor text. Based on the experimental evaluation, it was found that the fourth method has shown best categorization accuracy among all.

Another work used Artificial Neural Networks (ANN) to categorize documents (Miguel and Srinivas, 2001). In this study, 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.

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 (Han *et al*., 1998). The heart of the agent is the use 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.

CONCLUSION

In this study, we presented a categorizer agent that is able to categorize papers based on certain keywords that are found in the paper. Even though, we are using a very simple algorithm to categorize the papers, the experimental result has shown that the categorizer agent was able to perform better than the 10 final year project students. This brings us to the conclusion that, with the help of agent technology, we managed to improve the process of categorizing computer science electronic academic papers much faster and more accurate compared to manual categorization technique. 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 many improvements that need to be made. We need to improve the proposed algorithm to include more complex techniques such as the use 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. Besides, we would like to look at a wider electronic paper format such as .docx and .doc. We would also like to explore the possibility of using semantic technology to enhance the categorization's technique

ACKNOWLEDGEMENT

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

REFERENCES

Chess, D., B. Grosof, C. Harrison, D. Levine and C. Parris *et al*., 1995. Itinerant Agents for Mobile Computing. J. IEEE Personal Communi., 2: 34-49. DOI: 10.1109/98.468361

David, W.C., 1997. Intelligent Software Agents: Definitions and Applications. Analytic Services, Inc. http://alumnus.caltech.edu/~croft/research/agent/definition/

Etzioni, O. and D. Weld, 1994. A softbot-based interface to the internet. Communi. ACM., 37: 72-76. DOI: 10.1145/176789.

Fang, R., A. Mikroyannidis and B. Theodoulidis, 2006. A voting method for the classification of web pages. Proceeding of the 2006, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, Dec. 18-22, Hong Kong, pp: 610-613. DOI: 10.1109/WI-IATW.2006.23

Han, E.H., D. Boley, M. Gini, R. Gross and K. Hastings *et al*., 1998. WebACE: A web agent for document categorization and exploration. Proceeding of the 2nd International Conference on Autonomous Agents, (ICAA'98), ACM New York, NY, USA., pp: 408-415. DOI: 10.1145/280765.280872

Hyacinth, S.N., 1996. Software Agents: An Overview. Know. Eng. Rev., 11: 205-244.

Maes, P., 1990. Situated agents can have goals. Robotics Autonomous Syst., 6: 49-70. DOI: 10.1016/S0921-8890(05)80028-4

Maes, P., 1991. Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back. 1st Edn., The MIT press, London, ISBN-10: 0262631350, pp: 200.

Maes, P., 1995. Artificial life meets entertainment: Lifelike autonomous agents. Communi. ACM., 38: 108-114. DOI: 10.1145/219717.219808

Miguel, E.R. and P. Srinivas, 2001. Hierarchical Text Categorization Using Neural Networks. Inform. Retrieval, 5: 87-118. DOI: 10.1023/A:1012782908347

Russell, S. and P. Norvig, 2009. Artificial Intelligence: A Modern Approach. 3rd Edn., Prentice Hall, USA., ISBN-10: 0136042597, pp: 1152.

Sree, P.K., G.V.S. Raju, I.R. Babu and S.V. RajuImproving, 2008. Improving quality of clustering using cellular automata for information retrieval. J. Comput. Sci., 4: 167-171. DOI: 10.3844/jcssp.2008.167.171

Wooldridge, M. and N.R. Jennings, 1995. Intelligent agents: Theory and practice. Know. Eng. Rev., 10: 115- 152.