Suitability of Multiagents for Intelligent Distributed Searching

D. Indumathi[1], A. Chitra[2] and R. Lavanya[3]

ABSTRACT

Search engines have become an essential component of everyday life in modern society. Most of the applications involve some interaction with search engines one way or the other. The objective of this work is that an agent can "rapidly" customize a search result. An agent is a software or hardware that is capable of acting on behalf of its users to accomplish their tasks. An agent monitors the user's actions to prioritize the remaining query results. It learns based on the user's preferences and information content of the query. Fuzzy based computation is used in grouping / categorizing the files according to the relevancy. Based on the relevancy score calculated, the files are ranked. This is a powerful utility for fast file searching across the LAN and includes the ability to search for the file and its contents, given the keyword and ranks the result set based on the relevancy.

KEYWORDS : Intelligent agents, searching, Software agents, information retrieval

1. INTRODUCTION

Basically an agent is an active object with the ability to perceive reason and act. A simple agent is described in figure 1. The arrows show the characteristics autonomy, proactiveness, reactivity and social ability that an agent possess.

[1]Research Scholar, [2]Professor, [3]PG Student

Department of CSE, PSG College of technology, Coimbatore, India.

E-mail : [1]indumathi_d@yahoo.com

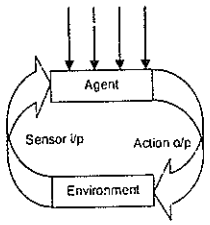


Figure 1: Simple agent

A Multi-Agent system (MAS)[1] is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. A MAS distributes computational resources and capabilities across a network of interconnected agents. Whereas a centralized system may be plagued by resource limitations, performance bottlenecks, or critical failures. A MAS is decentralized and thus does not suffer from the "single point of failure" problem associated with centralized systems. This allows for the interconnection and interoperation of multiple existing legacy systems. By building an agent wrapper around such systems, they can be incorporated into an agent society.

A MAS models[2] problems in terms of autonomous interacting component-agents, which is proven to be a more natural way of representing task allocation, team planning, user preferences, open environments, and so on. A MAS efficiently retrieves, filters, and globally coordinates information from sources that are spatially distributed. In general the MAS provide a distributed problem-solving environment; hence it can be used in LAN searching for efficient retrieval.

2. PROBLEM DEFINITION

In this research work the suitability of multiagents for distributed searching in a LAN environment is

investigated. Relevancy ranking is the method that is used to order the result list in such a way that the records most likely to be of interest to a user will be at the front. This makes searching easier for users as they won't have to spend as much time looking through records for the information that interests them. A good ranking algorithm will put information most relevant to a user's query at the beginning of the returned results. How relevant the record is, is given in relation to other sets within the search result space. Also the ranking varies with respect to the user's profile. So the problem is characterized with a complex and ubiquitous environment and requires reactive system to provide a solution to it. To handle complexity in ubiquitous systems and to deliver rational functionality the design representative bodies possesses the following[3]: **Autonomous**: given a vague and imprecise specification, it solves the problem, without constant guidance from the user. **Proactive:** it gives suggestions to the user. **Adaptive:** it knows the user's preferences and tailor interactions to reflect them.

3. Design Issues

The general agent based design[4] to solve any problem is given in figure-2.

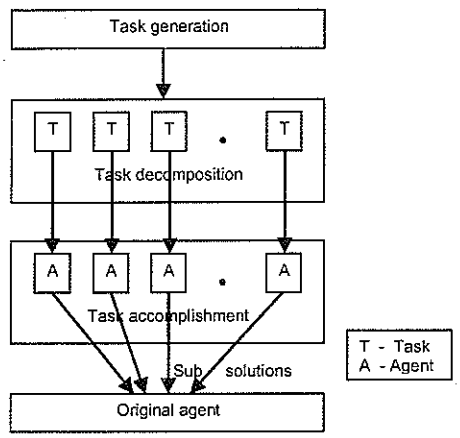


Figure 2: Agent based Solution

The system includes design of MAS communication model, computational model and the ultimate goal - smart model.

3.1 Communication Model

The figure-3 shows the MAS based communication model.

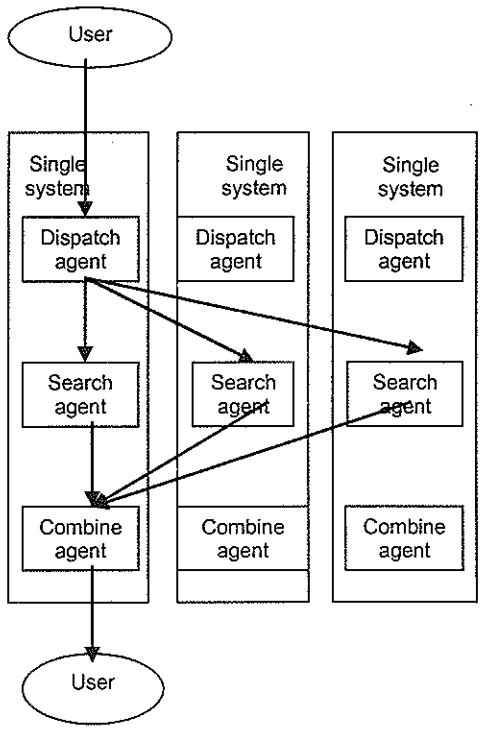


Figure 3: Distributed multiagent based communication model

The sequences of behaviors of the model are described

1) The user submits the keyword to the system using the graphical user interface
2) The dispatch agent decomposes the task and delegates it to other search agents in the network
3) The search agent is a work agent, which does the searching and computes the rank.
4) The combine agent takes care of combining the sub solutions and takes care of interacting back with the user along with the results obtained.

3.2 Computational Model

As shown in figure 4 the user interface agent gets the input and passes the searching request to software agents. The software agents contact the applications and retrieve the information and serve the result. This is in turn sent to the user.

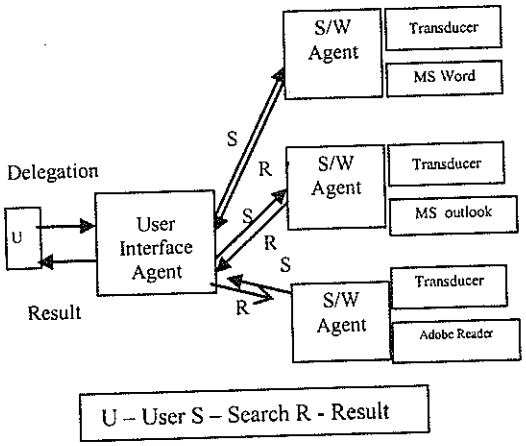


Figure 4: Computational model

3.3 Smart Model

The principle is in order to exactly retrieve the information we need to go for natural language processing agents.

The idea behind the smart model is the search results obtained from various application specific search agents are processed by NLPA and the ranking is to be done with reference to the context of the keyword.

Figure-5 shows the smart search agent model.

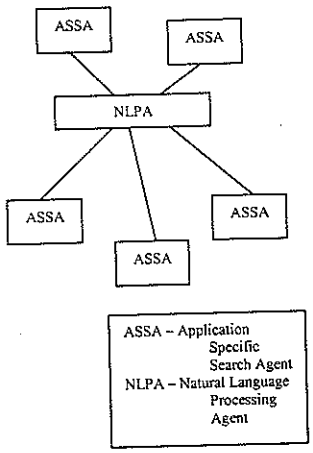


Figure 5: Smart Search Agent Model

4. Implementation

The project is implemented in Java.[10] The entire work is abstractly modularized into two horizontal pipelined blocks – data retrieval and information retrieval blocks.

4.1 Data Retrieval Block

The distributed multiagent based search system searches for the keyword in the following

- File and directory names
- Meta data, such as titles, authors, comments.
- File content.

This module is just pattern matching. The system implemented searches the exact byte stream of the file and returns the result to the information retrieval system.

4.2 Information Retrieval

In information retrieval module the files obtained as the result of data retrieval system are assigned some weightage[11]. The weightage is obtained by ranking the files. Figure 6 shows the hierarchical view of the components of ranking.

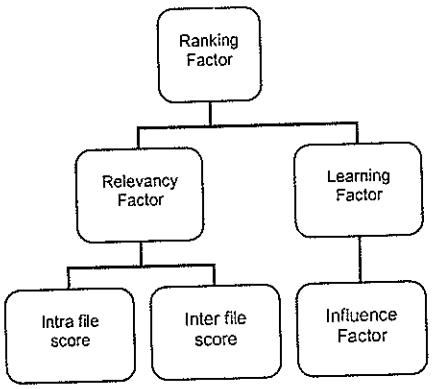


Figure 6: Hierarchical view of the components of ranking

4.3 File Score

The inter file score is based on the frequency of occurrence of the searched keyword in a particular file relative to the total number of occurrence of the keyword in the entire search.

Also known as frequency score since the scoring is based on the frequency.

More the occurrence of the keyword the more the relevancy is. So in the figure 7 shown, the file 2 gets higher credit than other files. The fuzzy score is assigned as shown in the table-1 .

Based on the position of occurrence of the keyword within the file, the relevancy is

Table -1 frequency score

High	$freq > 2(\sum_{i=1}^{n} freq)/3$
Medium	$\sum_{i=1}^{n} freq/3 < freq \leq 2(\sum_{i=1}^{n} freq)/3$
Low	$freq_i \leq \sum_{i=1}^{n} freq_i / 3$

calculated. Greater the proportion of text following the keyword the more is the intra file score score. This is also known as position score. In the figure 8, file 1 has more position score than file 2. The more the keyword concentrated in the upper block more the position score.

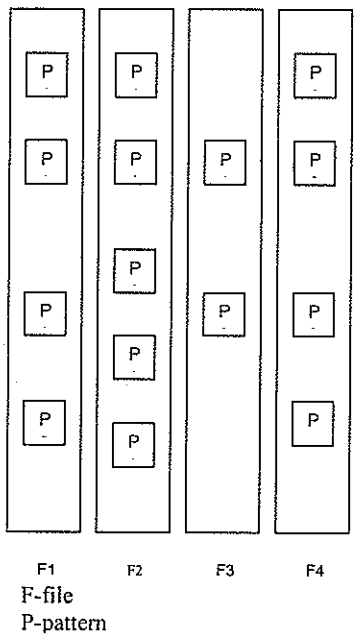


Figure 7 : Frequency score illustration

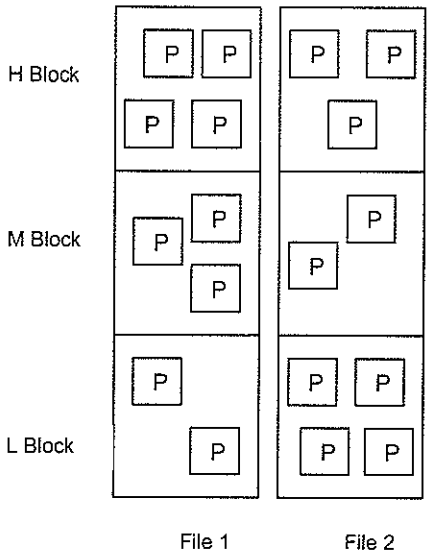


Figure 8: Position score illustration

4.4 Relevancy Score

The relevancy score is calculated based on the position and frequency score. The fuzzy computation of the relevancy score is shown in the table below.

TABLE -2 RELEVANCY SCORE

PS / FS	High	Medium	Low
High	High	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

4.5 Learning Score

The user's recently accessed files are maintained and those files are given recency score based on the time of file access. Ranking is more for the most recently accessed file and is displayed in the beginning of the result set.

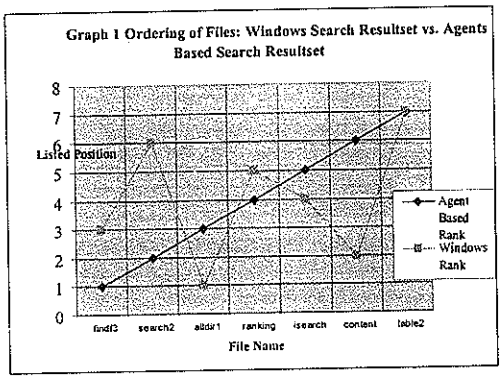
5. Experimental Results

The table 3 shows the search result for the keyword void.

Table 3 Details of agent based ranking

File name	Freq	Size (KB)	FS	PS	Rank
Alldir1	3	1	L	M	3
content	1	4	L	L	6
Findf3	9	3	H	H	1
isearch	1	1	L	L	5
ranking	3	3	L	L	4
search2	3	1	L	H	2
Table2	2	14	L	L	6

Graph - 1 shows the variations in the listing sequence between windows search resultset and agent based searching.



The existing system just groups all the files containing the keyword into one single category. It is based on simple 'if – then' rule. (i.e) If the given keyword is present in the file then list, group such files and display. Whereas, the agent based intelligent system takes dynamic decisions not just 'if – then' decision. Thus the use of agent technology in this domain serves better and it provides a powerful tool for fast file searching across a LAN.

6. Conclusion

The tool designed is a straightforward and easy to use interface. This intelligent utility works intuitively and ranks the files, which contain the keywords based on fuzzy based computational intelligence. This provides preliminary information retrieval system.

7. Future enhancements

Additional ways for improving the smart model are also being considered. A natural language processor can be incorporated in to the smart model and also file access patterns can be tracked to make better rationalized decisions. This model can be extended to web spiders or crawlers by building efficient indexing services.

8. References

[1] Wooldridge M. *"An introduction to multiagent systems"*. John Wiley & Sons, Chichester, February 2002.

[2] M. Wooldridge and N R Jennings. *"Intelligent agents: Theory and practice, The Knowledge Engineering"* Springer Review, 1995

[3] Jeffrey Bradshaw, *"Software Agents"*, MIT Press, 1997.

[4] Gerhard Weiss, *"Multi agent systems: A modern approach to distributed Artificial Intelligence"*, MIT Press, 1999.

[5] Franklin S. et al. *"Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents"* in Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

[6] Stuart Russell and Peter Norvig, *"Artificial Intelligence, A Modern approach"*, Prentice Hall, 2002.

[7] Matthias Klusch, Sonia Bergamaschi and Pete Edwards, *"Intelligent Information Agents"*, The Agentlink Perspective (Lecture notes in computer Science), Springer-Verlag, 2003.

[8] Nicholas R Jennings and Michael J Wooldridge," *Agent Technology: Foundations, Applications and markets"* Springer-Verlag, 1998.

[9] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, *'Placing search in context: The Concept revisited'*,

ACM Transactions on Information Systems, Vol 20, No.1, January 2002.

[10] Patrick Naughton and Herbert Schildt.,*Java 2 The Complete Reference*, Third Edition, Tata McGraw Hill, 2005

[11] Masoud Mohammadian, *computational intelligence techniques driven intelligent agent for web data mining and information retrieval'*.Idea Group Inc, 2004

Authors' Biography :

Ms. D Indumathi is doing MS (By research) in the area of computer science and Engineering. She is currently working as a lecturer in the department of Computer Science and Engineering in PSG College of Technology, Coimbatore, TN, India. Her field of interest includes Intelligent Agents, Analysis of Algorithms, Knowledge based systems. She is a life member of ISTE and ACS.

Dr. A Chitra is working as a professor in the department of Computer Science and Engineering in PSG College of Technology, Coimbatore, TN, India. She has completed her PhD in the area of Computer Science and Engineering. She has got 20 years of teaching experience in the field of Distributed Computing, Intelligent agents, Image processing, Soft Computing etc. She is a member of Board of Studies in PSG College of technology. She is a joint secretary of ACS, Bangalore and Secretary of ACS, Coimbatore Chapter. She is a life member of ISTE, CSI , ACS and IE. She also received Tamilnadu Young Women scientist award for the year 2004 in Engineering and Technology. She has published 3 technical books.

Ms. R Lavanya is doing M.E in Computer Science and Engineering in PSG College of Technology, Coimbatore, TN, India. Her interests include, Agent technology, Soft Computing, Programming and debugging.