International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

Multi-agent Data Fusion Architecture for Intelligent Web Information Retrieval

Amin Milani Fard, Mohsen Kahani, Reza Ghaemi, and Hamid Tabatabaee

***Abstract*—**In this paper we propose a multi-agent architecture for web information retrieval using fuzzy logic based result fusion mechanism. The model is designed in JADE framework and takes advantage of JXTA agent communication method to allow agent communication through firewalls and network address translators. This approach enables developers to build and deploy P2P applications through a unified medium to manage agent-based document retrieval from multiple sources.

***Keywords*—**Information retrieval systems, list fusion methods, document score, multi-agent systems.

I. Introduction

THE problem of finding information in scattered sources located on different servers has become more serious considering increasing number of databases on LANs and the Internet. The goal of distributed information retrieval (DIR) is to provide a single search interface that provides access to the available databases. This problem, also known as *federated search*, involves building resource descriptions for each database, choosing which databases to search for particular information, and merging retrieved results into a single result list [1]-[2]. Some applications of information retrieval from multiple sources include meta-search engines, distributed genomic search, newsletter gathering and etc.

DIR includes four sub-problems of resource description, resource selection, query translation, and result merging [1]. The resource description problem is how to learn and describe the topics covered by each different database. The resource selection means, given a query and a set of resource descriptions, how to select a set of resources to search. The query translation means mapping a given information need in some base representation, automatically to query languages appropriate for the selected databases. Finally, after result lists have been returned from the selected databases, result merging is used to integrate them into a single ranked list.

Multi-agent systems (MASs) as an emerging sub-field of artificial intelligence concern with interaction of agents to solve a common problem [3]. This paradigm has become more and more important in many aspects of computer science by introducing the issues of distributed intelligence and interaction. They represent a new way of analyzing, designing, and implementing complex software systems.

In multi-agent systems, communication is the basis for interactions and social organizations which enables the agents to cooperate and coordinate their actions. A number of communication languages have been developed for inter-agent communication, in which the most widely used ones are KIF (Knowledge Interchange Format) [4], KQML (Knowledge Query and Manipulation Language) [5], and ACL (Agent Communication Language) [6]. KQML uses KIF to express the content of a message based on the first-order logic. KIF is a language intended primarily to express the content part of KQML messages. ACL is another communication standard emerging in competition with KQML since 1995. Nowadays, XML (Extensible Markup Language) started to show its performance as a language to encode the messages exchanged between the agents, in particular in agent-based e-commerce to support the next generation of Internet commerce [7].

The idea of embedding intelligence in web environment was firstly introduced by Tim B. Lee, the web inventor, and named *Semantic Web*. This web would be an intelligent extension of the current web and aims to understand and percept human language and use *intelligent agents* to explore the web contents instead of users [8].

In this work we propose an intelligent multi-agent architecture for distributed information retrieval using fuzzy logic concept. The structure of the paper is as follows. Section II describes an overview of some previous researches. An investigation on multi-agent system design of our proposed approach is declared in section III, and finally concluded the work in the section IV.

II. Related Works

Let us consider *m* information servers, denoted by S_i and i=1...m; each of which provided with its own information retrieval system (IRS). Here we assume that the retrieval engines produce weighted lists of retrieved documents as an indicator of the estimated relevance of each document with respect to the query. In order to provide user with a single ordered list of relevant documents, the individual lists produced by IRSs must be fused [9]. This fusion is not an easy problem as IRSs produced their result with different criteria and statistics such as average document length, text frequency

Manuscript received April 8, 2007.

A. Milani Fard is with Department of Computer Engineering, Ferdowsi University, Mashhad, Iran (e-mail: milanifard@stu-mail.um.ac.ir).

M. Kahani, is with Department of Computer Engineering, Ferdowsi University, Mashhad, Iran (e-mail: kahani@um.ac.ir).

R. Ghaemi is with Department of Computer Engineering, Islamic Azad University, Quchan, Iran (e-mail: rezaghaemi@scientist.com).

H. Tabatabaee is with Department of Computer Engineering, Islamic Azad University, Quchan, Iran (e-mail: hamid.tabatabaee@gmail.com).

inverse document frequency, and etc [10].

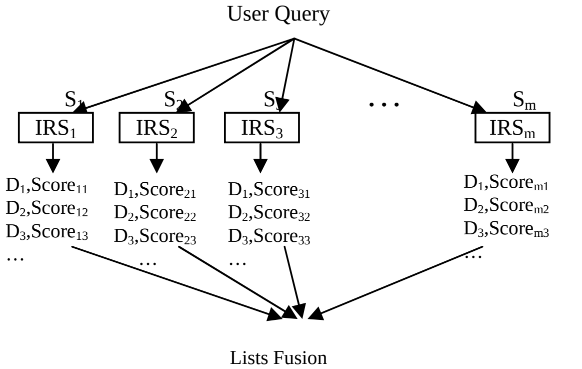
User Query
S_1
S_2
S_3
. . .
S_m
IRS_1
IRS_2
IRS_3
IRS_m
D_1,$Score_{11}$ D_2,$Score_{12}$ D_3,$Score_{13}$ …
D_1,$Score_{21}$ D_2,$Score_{22}$ D_3,$Score_{23}$ …
D_1,$Score_{31}$ D_2,$Score_{32}$ D_3,$Score_{33}$ …
D_1,$Score_{m1}$ D_2,$Score_{m2}$ D_3,$Score_{m3}$ …
Lists Fusion

Fig. 1 An overview of information retrieval from multiple sources

Several works have faced the problem of distributed information retrieval and fusion [11]-[12]-[13]-[14], in which essentially consist of a three step process. In the first step, a query *Q* is submitted to the *m* servers and consequently *m* result lists would be produced by each IRS. It is assumed that in these lists, each document has its score which indicates its relevance to the query *Q*.

In the second step, a procedure computes the number N_i of documents that each server has to provide in the list. Considering *N* as the total number of documents to in the final fused list, we have $N = \sum_{i=1}^{m} N_i$

Finally the third step performs fusion of those *m* individual ordered lists and shows the user one ordered result [9]. Some solutions to the problem of the lists fusion can be found in [11]-[12]-[13]-[14]-[15].

One strategy to decide N_i for each server is to evaluate the appropriateness of the server information to the given user query. This is a knowledge intensive approach as some information about the content of the server is needed. In [11]-[12] the authors have proposed an approach to determine N_i based on the use of training queries to build both the content and search behavior of each collection.

In [9]-[14]-[15] a knowledge based approach has been proposed. This approach is based on the computation of a fitness score for indicating appropriateness of each server with respect to the user query. In order to compute this score, firstly a set of query type is defined; with each query type a set of fitness score (one per server) is associated.

A query type is formalized by means of a rule in which the antecedent is the description of the query by means of a set of properties, and the consequent specifies the fitness of each server with respect to this query type. Formally, the properties useful to characterize the subject matters of queries are denoted as U_j (j=1..*s*); the fitness scores are the values associated with variables V_i (i=1..*m*), one for each server S_i. The fuzzy rules [9] identifying query types are formalized as:

if U_1 is A_{k1} and … and U_s is A_{ks} then *V* is B_k

in which k is the index of the rule, the A_{kj} are fuzzy subsets containing the values of variables U_j, V is an m dimensional vector whose components are the V_i, and B_k is also an m dimensional vector whose components B_{ki}, correspond to the fitness of server S_i as an appropriate source of document for the type of query described in the antecedent of rule k. Once a user query has been evaluated by the IRSs on the servers, a matching between it and the query type rules is performed, in order to compute the fitness scores for the considered query [9].

The first step is to obtain a unique fitness score per server for the desired query. To this aim the user query is also described in terms of properties U_j, a value a_j is associated with each U_j and for each rule *k* a firing level τ_k is computed: $\tau_k = min_j[A_{kj}(a_j)]$. To combine the fitness scores in the distinct rules, the obtained information is employed as follows in which p is the number of query type:

$$B^* = \frac{\sum_{k=1}^{p} \tau_k B_k}{\sum_{k=1}^{p} \tau_k} \tag{1}$$

The components b_i^* of vector B^* denote the fitness values of the servers for the considered query. Finally, to obtain the values N_i of documents to be contributed by each server, first the fitness values are normalized:

$$\alpha_i = \frac{b_i^*}{\sum_{i=1}^{m} b_i^*} \tag{2}$$

At this point the normalized values are used to compute the number of documents to be retrieved from each server:

$$N_i = \alpha_i N \tag{3}$$

The number of documents to be selected from each server is proportional to the fitness of that server to the query [9].

III. Proposed System Architecture

Our proposed multi-agent system architecture is based on *Java Agent DEelopment* (JADE) framework [16]. JADE is a software development framework aimed at developing multi-agent systems and applications in which agents communicate using FIPA[1] Agent Communication Language (ACL) messages and live in containers which may be distributed to several different machines. The Agent Management System (AMS) is the agent who exerts supervisory control over access to and use of the Agent Platform. Only one AMS will exist in a single platform.

Each agent must register with an AMS in order to get a valid AID. The Directory Facilitator (DF) is the agent who provides the default yellow page service in the platform. The

[1] Foundation for Intelligent Physical Agents (http://www.fipa.org)

Message Transport System, also called Agent Communication Channel (ACC), is the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms. The standard model of an agent platform, as defined by FIPA, is represented in the Fig. 2.

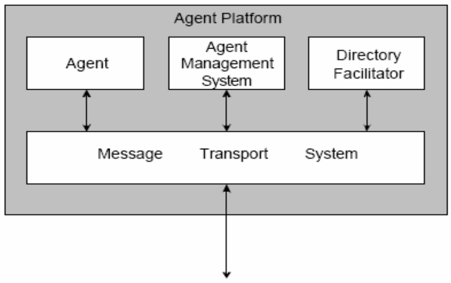

Fig. 2 Reference architecture of a FIPA Agent Platform

JADE is capable of linking Web services and agents together to enable semantic web applications. A Web service can be published as a JADE agent service and an agent service can be symmetrically published as a Web service endpoint. Invoking a Web service is just like invoking a normal agent service. Web services' clients can also search for and invoke agent services hosted within JADE containers.

The Web Services Integration Gateway (WSIG) [17] uses a Gateway agent to control the gateway from within a JADE container. Interaction among agents on different platforms is achieved through the Agent Communication Channel. Whenever a JADE agent sends a message and the receiver lives on a different agent platform, a Message Transport Protocol (MTP) is used to implement lower level message delivery procedures [18]. Currently there are two main MTPs to support this inter-platform agent communication - CORBA IIOP-based and HTTP-based MTP.

Considering large-scale applications over separated networks, agent communications has to be handled behind firewalls and Network Address Translators (NATs), however, the current JADE MTP do not allow agent communication through firewalls and NATs. Fortunately, the firewall/NAT issue can be solved by using the current JXTA implementation for agent communication [19].

JXTA is a set of open protocols for P2P networking. These protocols enable developers to build and deploy P2P applications through a unified medium [20]. Obviously, JXTA is a suitable architecture for implementing MTP-s for JADE and consequently JADE agent communication within different networks can be facilitated by incorporating JXTA technology into JADE [19].

In this work, the user sends a query to the search agents asking them to look for the answer using the submitted terms in the available ontologies. Agents may also communicate with other agents and exchange information to broaden the search results.

Our proposed architecture uses different types of agents, each having its own characteristics.

1. Broker Agent: After receiving the user query, the BA does a simple task sharing process for search agents. The BA can also create search agents if needed.

2. Search Agents: These agents will return an ordered list of search results to the response agent.

3. Response Agent: This agent has the responsibility to show the result of retrieved information. To do so, RA collects SAs results, ranks them and then writes them on the screen ordered by relation percentage.

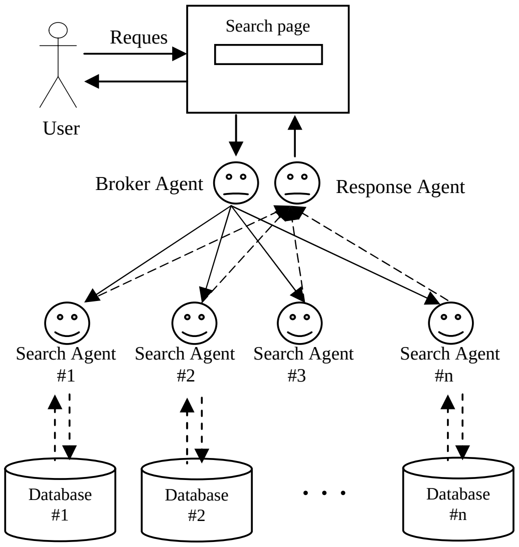

Fig. 3 Proposed system architecture

The *response agent* has the responsibility of computing the fitness scores of each server using a combination method of approaches presented in [9]-[21]-[22] which are based on associating each server with a fuzzy set of key terms; the key terms are those which best represent the topics in the considered archive, and their associated membership values are interpreted as their importance of the server.

In other words the weights represent the fitness of the server with respect to the topic expressed by the associated term. The fuzzy set can be interpreted as a *prototypal document* or *document type* constituting a representative of the documents in the archive of the server.

The terms and their fitness weights can be defined either manually or automatically computed. An automatic selection of the key terms can be obtained on the basis of their normalized occurrences in the archive on the considered server; the computed occurrence can then be used as terms membership values. A document is formally represented as a fuzzy binary relation

$$R_{di} = \sum_{(t)\in T} \mu_{di}(t)/t \tag{4}$$

where R_{di} is the representation of the archive of the server *i's* document $d \in D$, the set of documents $t \in T$, the set of index terms, and $\mu_{di}\, D\times T\rightarrow[0,1]$ the membership function of R_{di}. μ_{di} is again a dynamic function with $\mu_d(i,t,s)$ expressing the significance of the term t in section s of document d on the server i. On the other hand, for sections containing textual descriptions, $\mu_d(i,t,s)$ can be computed as a function of the normalized term frequency for that section as bellow

$$\mu_d(i,t,s) = TF(dst)_i * IDF(t)_i \qquad (5)$$

in which $IDF(t)_i$ is the inverse document frequency of the term t, and $TF(dst)_i$ is the normalized term frequency of the server i defined as

$$TF(dst)_i = \frac{OCC(dst)_i}{MAXOCC(sd)_i} \qquad (6)$$

$$IDF(t)_i = \log_2^{(n/DF_{ti})} \qquad (7)$$

where $OCC(dst)_i$ is the number of occurrences of term t in section s of document d in the server i, $MAXOCC(sd)_i$ is the highest number of occurrence among all terms in section s of document d in the archive of the server i, and DF_{ti} is the number of documents in the total number of n pages in the server i that contain term t.

To obtain the overall degree of significance of a term in a document, computed over all the sections, an aggregation scheme ordered weighted average (OWA) is used

$$\mu(i,d,t) = \mathrm{OWA}_{lq}(\mu_1(i,d,t),\ldots,\mu_n(i,d,t)) \qquad (8)$$

Parameters l, q are determined by users specified relative weights to the sections. A query $<t,w>$ is represented by terms t_j and the corresponding weights w_j.

The query is evaluated for a given document, and then aggregation operators are used. Thus the result of a query evaluation is represented as a fuzzy subset of the archived documents, given by

$$R_{di}(t) = \sum_{d\in D}\mu_W(i,d,t)/d \qquad (9)$$

This brings to light that fuzzy Boolean IR models are more flexible in representing both document contents and information needs. When a query has to be evaluated, the process of estimate of the appropriateness of each server as the controller of the relevant information for the query can be done on the basis of a procedure which performs a matching between the query and the fuzzy set of archived documents

The matching procedure of a Boolean query against the fuzzy term set proceeds in a bottom up way: first, the membership of each query term to the fuzzy set is computed. The obtained values (including the fitness of the server for each query term) are then aggregated by applying the connectives specified in the query (the *AND* is interpreted as *min*, the *OR* as *max*, and the *NOT* as a *complement*). Fig. 4 shows an example of computing b_i^*, regarding equation 1, of the server S_i for a given query Q.

$$\boxed{\begin{gathered} Q = \mathrm{t}_1\ AND\ (t_2\ OR\ t_3) \\ \Downarrow \\ B_i^* = min(\mu_w(i,d,t_1),\ max(\mu_w(i,d,t_2),\ \mu_w(i,d,t_3))) \end{gathered}}$$

Fig. 4 Sample score computation

IV. Conclusion and Future Works

A multi-agent architecture for distributed document retrieval on the web was proposed and a fuzzy logic based approach was used for result lists fusion mechanism. This model in contrast with other multi-agent models for web is considered to be more practical using hybrid JADE-JXTA framework which allows agent communication through firewalls and NATs. Some future works can be done on performance evaluation of the retrieval mechanism and meta-search engine development under this architecture seems to be a challenging matter. Also Grid data retrieval is a very good testbed for such an approach.

References

[1] J. Callan, “Distributed information retrieval”. In Advances in Information Retrieval, W. B. Croft, Ed. Kluwer Academic Publishers, 2000, pp. 127–150.

[2] L. Si, J. Callan, “A Semisupervised Learning Method to Merge Search Engine Results”, ACM Transactions on Information Systems, Vol. 21, No. 4, October 2003, Pages 457–491.

[3] M. Wooldridge, “An Introduction to Multiagent Systems” Published in February 2002 by John Wiley & Sons. ISBN 0 47149691X.

[4] M. R. Genesereth, R. E. Fikes, “Knowledge interchange format”, version 3.0. Technical Report 92-1, Stanford University, Computer Science Departament, 1992

[5] T. Finin, R. Fritzson, D. McKay, R. McEntire,”KQML as an Agent Communication Language”, Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94), ACM Press,Gaithersburg, MD, USA, editor N. Adam, B. Bhargava, Y. Yesha, pp 456-463, 1994

[6] Y. Labrou, T. Finin, Y. Peng, “The current landscape of Agent Communication Languages”, The current landscape of Agent Communication Languages, Intelligent Systems, volume 14, number 2, March/April 1999, IEEE Computer Society, 1999

[7] A. Korzyk, “Towards XML As A Secure Intelligent Agent Communication Language”, the 23rd National Information Systems Security Conference, Baltimore Convention Center, Baltimore, Maryland, SA, October 16-19, 2000

[8] T. B. Lee, J. Hendler, and O. Lassila, “The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”, Scientific America, May, 2001

[9] G. Pasi, and R. R. Yager, “Document retrieval from multiple sources of information”, in Uncertainty in Intelligent and Information Systems, edited by Bouchon-Meunier, B., Yager, R. R. and Zadeh, L. A., World Scientific: Singapore, 250-261, 2000.

[10] R. Baeza-Yates, B. Ribeiro-Neto, “Modern Information Retrieval”, Addison-Wesley, 1999

[11] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird, “The collection fusion problem”, Proceedings of the third Text Retrieval Conference (TREC-3), 95-104, 1994

[12] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird, “Learning collection fusion strategies”, Proceedings of the 18th ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, 172-179, 1995

[13] R. R. Yager, “On Ordered Weighted Averaging aggregation Operators in Multi Criteria Decision Making”, IEEE Trans. on Systems, Man and Cybernetics. 18(1), 183-190, 1998

[14] R. R. Yager, and A. Rybalov, “On the fusion of documents from multiple collection information retrieval systems”, Journal for the American Society for Information Science 49, 1177-1184, 1998.

[15] R. R. Yager, V. Kreinovich, "On how to merge sorted lists coming from different web search tools," Soft Computing Research Journal 3, 83-88, 1999.

[16] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, “JADE Programmer’s Guide”, 21-August-2006.

[17] Jade Board, “JADE WSIG Add-On Guide JADE Web Services Integration Gateway (WSIG) Guide”, 03-March-2005

[18] E. Cortese, F. Quarta, G. Vitaglione, P. Vrba. “Scalability and Performance of the JADE Message Transport System”. Analysis of Suitability for Holonic Manufacturing Systems, exp, 2002.

[19] S. Liu, P. Küngas, M. Matskin, “Agent-Based Web Service Composition with JADE and JXTA”, Proceedings of the 2006 International Conference on Semantic Web and Web Services, SWWS 2006, Las Vegas, Nevada, USA, June 26-29, 2006

[20] J. D. Gradecki, “Mastering JXTA: Building Java Peer-to-Peer Applications”, JohnWiley&Sons, 2002.

[21] G. Salton, C. Buckley, “Term-weighting approaches in automatic text retrieval”. *Information Processing & Management* 24(5): 513–523, 1988

[22] G. Pasi, G. Bordonga, “Application of fuzzy set theory to extend boolean information retrieval,” in *Soft Computing in Information Retrieval: Techniques and Applications*, F. Crestani, G. Pasi, Eds. Heidelberg, Germany: Physica Verlag, 2000, vol. 50, pp. 21–47.