BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Artificial Intelligence Advances
https://ojs.bilpublishing.com/index.php/aia/index

ARTICLE

A New Approach of Intelligent Data Retrieval Paradigm

Falah Al-akashi[1*] Diana Inkpen[2]

1. Faculty of Engineering, University of Kufa, Iraq
2. Faculty of Engineering, University of Ottawa, Canada

ARTICLE INFO

Article history
Received: 11 May 2021
Accepted: 16 June 2021
Published Online: 15 July 2021

Keywords:
Intelligent agents
Ranking schema
Distributed approach
Vector space model

ABSTRACT

What is a real time agent, how does it remedy ongoing daily frustrations for users, and how does it improve the retrieval performance in World Wide Web? These are the main question we focus on this manuscript. In many distributed information retrieval systems, information in agents should be ranked based on a combination of multiple criteria. Linear combination of ranks has been the dominant approach due to its simplicity and effectiveness. Such a combination scheme in distributed infrastructure requires that the ranks in resources or agents are comparable to each other before combined. The main challenge is transforming the raw rank values of different criteria appropriately to make them comparable before any combination. Different ways for ranking agents make this strategy difficult. In this research, we will demonstrate how to rank Web documents based on resource-provided information how to combine several resources raking schemas in one time. The proposed system was implemented specifically in data provided by agents to create a comparable combination for different attributes. The proposed approach was tested on the queries provided by Text Retrieval Conference (TREC). Experimental results showed that our approach is effective and robust compared with offline search platforms.

1. Introduction

With the growing number of information sources available via the internet, the problem of how to combine distributed and heterogeneous information sources is becoming increasingly difficult [1]. Large area, high bandwidth network such as World Wide Web creates a number of opportunities and challenges for effective retrieval of information. Such a network makes users to gain access to huge amounts of data in a wide variety of types. In realistic setting, without effective information management tools most of this data are worthless since users are unable to find data of interest [6,7]. Currently, the available source of information in World Wide Web is classified domains into particular topics, including: health, travel, shopping, etc. The classical contents, such as: URL, title, headers, and document body; as well as the traditional ranking, such as: global rank, local rank, insight rank, knowledge bases rank, and so forth are important contents making information distributed resources in agents are comparable. In recent years, traditional approaches to building distributed or aggregated systems do not scale well [5]. Current systems e.g. search engines or topic directories on the World Wide Web provide limited capabilities for locating, combining, processing, and organizing information. The advent of large area networks connecting

**Corresponding Author:*
Falah Al-akashi,
Faculty of Engineering, University of Kufa, Iraq;
Email: falahh.alakaishi@uokufa.edu.iq

 DOI: https://doi.org/10.30564/aia.v3i2.3219

many diverse repositories of data creates the challenge of finding particular data of interest easily and timely manner. More contrast, providing access to the large number of information sources and organizing them into a network of information agents is a big challenge. Each agent provides expertise on a specific topic and sometimes drawing on relevant information from other information agents. To build such topographical network, we need an infrastructure of a single agent system that can be instantiated to provide accessing to multiple agents. Information mediator that provides access to heterogeneous data and knowledge base is an aspect of our previous work [4]. We need to consider a unique aspect that is critical for any agent-based system: how to draw and simulate knowledge in the network and how to categorize or classify data in the network based on their topic similarity; and finally how to combine them. Formulating the results retrieved from distributed agents and computing the federated rank based on that aspect is the last goal of our approach.

However, the standard form of searching topics in the network is to use vector space model and inner product for similarity [2]. The inner product syntax composed of atoms that retrieve documents from index file, and inner operators work on their operands to retrieve set of documents that compromise a set of vectors. On another hand, keywords in a query can be expressed in a syntax tree where leaves are the basic query keywords and the internal nodes are the operators. The operators that commonly used in inner expression could select documents satisfying different features. Besides selecting the appropriate documents based on particular features, the retrieval approach might also add other criteria to enhance the initial rank if some similar features detected in some documents in similar resources. To provide an efficient information retrieval approach in large distributed resources, it is important to process resources and commonly build a desired index [11]. Realistically, the professional and significant prototype of index structure in the distributed agents is a memory allocation table or so-called 'hash table', which is a data structure that stores data in an associative manner. Data is stored in an array format, where each data value has its own unique index value. Access of data becomes very fast if we know the index of the desired data. In turn, search system collects, parses, and stores data in the hash content in order to facilitate the process of fast indexing and retrieving.

The rest of this manuscript is organized as follows: Section 2 will outline some related work. Section 3 will discuss the architecture of our proposed approach. Section 4 will demonstrate the query processing and documents ranking procedure. Result analysis will be discussed in Section 5, while Section 6 will outline the comparison between our proposal and some related work. Experimental results and interface will be discussed in Section 7, and finally, conclusions are highlighted in Section 8.

2. Related Work

With the vast amount of information resources available today, a crucial problem is how to locate, retrieve and process information in that resources. It would be impractical to build a single unified system that combines all of these information resources. A more promising approach compromises between agents in a network of information retrieval agents when needed. There has been much recent work in a similar vein where the aim is to build a distributed index that share different tasks. Researchers [8] proposed an automated image information retrieval system that helped internet users to find the required information with high performance. Experiments showed that gathering images potentially takes a long time with a single-threaded information crawler on a single computer. Deploying parallel information crawler and distributed across many machines was addressed that problem. Another researcher [9] attempts to use a thesaurus to provide meaningful search of the query to help the distributed model to utilize for document retrieval. A scalable agent-based information retrieval engine [10] deployed intelligent software agents, natural language understanding, and conceptual search techniques to support public accessing to the data over the distributed resources. The researcher had demonstrated the feasibility of multi-agent architecture to support intelligent information accessing and retrieving. A so-called "just-in-time" information retrieval agent [11] was another model that proactively retrieves and presents information based on a person's local context. Researchers showed that users of agents are not merely more efficient at retrieving information, but actually retrieve and use more information than they would with traditional search engines. Multi-agent information retrieval system based on ontology was proposed by researchers [12]. They proposed a multi-agent information retrieval system based on ontology. Researchers showed that introducing ontology on information retrieval system can realize knowledge domain-expression in order to provide users with the increment of information service and refine the initial query.

3. Components of the Proposed Approach

In this section, we describe the key components of our proposed information retrieval model. As the number of documents that represents the relevant content in our distributed resources has increased, the system for searching relevant documents for each query is

DOI: https://doi.org/10.30564/aia.v3i2.3219

needed and must be built efficiently. We will discuss our documents indexing algorithm in each agent represented by local indexes connected by a centralized index. It is worth mentioning that the key note for rebuilding index table is only when a new set of documents is added to the collection or when there is substantial changing in the data. In this section we will describe our key components of our proposal approach, as shown in Figure 1.

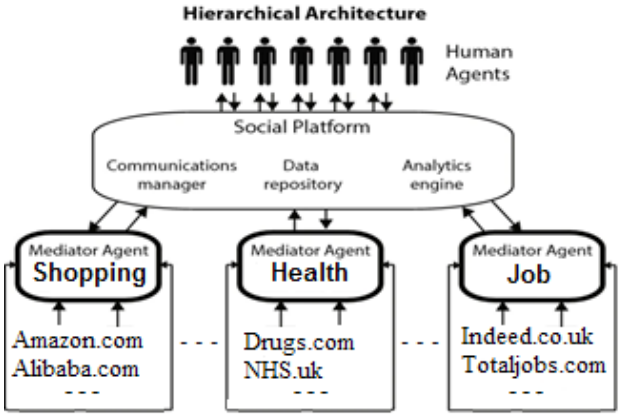

Figure 1. The architecture of our distributed model

3.1 Distributed Indexes

According to the requirements of distributed agents, centralized index in each agent normally gathers documents from the distributed resources devoted similar topic. The information in remote documents returned to a searcher implies two types of data: Meta and regular. We aimed to work with the textual information as text fields in a hash table format in order to allow more flexible indexing and retrieving. Some textual indexing format has limited matching capabilities. Hash table emphasizes simplicity, generality, and usability across our distributed resources. Although the design of hash table focuses on limited capabilities, it is widely used for the representation of arbitrary data structures; such as those used in web services. Data is strongly structured and stored in hash table and text is quite restrictive [13]. Our document representations have a fixed set of attributes; some fields are textual content while others are not. The attributes are in a fixed order in documents; such that, if document does not contain enough text, it does not classify under any attribute because attributes are not allowed to be nested or overlapped. As a result, the retrieval data restricted to specifying that giving basic evidence is only for a given attribute. This is more reasonable when collection has a fixed structure e.g. indexing meta-data without tokenization. As shown in the example below, data is structured into many attributes and evidences. Each set is for an identical agent; and thus, set of resources with similar task are aggregated in indexing file.

Currently, we have approximately of 24 agents and different number of resources for each, as shown in Table 1, each resource in agent advert some searchable parameters grouped in similar task. The principle for categorizing agents and how tasks involved for each were mentioned in [4].

Table 1. Agents categories and number of resources used for each

Agents	Topics
Indeed.co.uk	Jobs
Totaljobs.com	Jobs
Optnation.com	Jobs
Monster.co.uk	Jobs
Wikicfp.com	Academic
Dblp.org	Academic
Citeseerx.edu	Academic
Univerzities.com	Academic
Conferencealert.com	Academic
Researchgate.com	Academic
Taste.com	Recipes
Simplerecipes.com	Recipes
Saveland.ca	Shopping
Amazon.com	Shopping
Ebay.com	Shopping
Alibaba.com	Shopping
Answers.com	Q/A
Answers.yahoo.com	Q/A
Wikihow.com	Q/A
Drugs.com	Health
Everydayhealth.com	Health
NHS.uk	Health
Patient.info `	Health
derm101.com	Health
Cntravel.com	Health Travel
Hotelscombined.com	Travel
Tripsavvy.com	Travel
Tvguide.com	Shows
Hollywoodlife.com	Shows
Imdb.com	Shows
Thespruce.com	Home
Investopedia.com	Local
Mapquest.com	Local
Embassypasges.com	Local
Zoominfo.com	Local
Foursquare.com	Weather
Openweathermap.org	Encyclopedia
Wikipedia.org	Encyclopedia
Nationsonline.com	Encyclopedia
Gov.uk	Encyclopedia
Geonames.org	Encyclopedia
Nationsonline.org	Encyclopedia
Reference.com	Encyclopedia
Thoughtco.com	Encyclopedia
NPR.org	News
News-medical.net	Health
Uptodate.com	Health
Verywell.com	Health
Mayoclinic.org	Finance
Thebalance.com	Finance
Historynet.com	History
Twitter.com	Social
Facebook.com	Social
Wordpress.com	Blog
Thefreedictionary.com	Dictionary
Merriam-webster.com	Dictionary
Cambridge.org	Dictionary
Google.com	Scholar
Worldcat.org	Scholar
360daily.com	Video
Vimeo.com	Video

 DOI: https://doi.org/10.30564/aia.v3i2.3219

3.1.1 Features Selection

We need to extract some features (attributes) from resources to be evaluated in a heuristic search, and to provide a basis for the learning component. The most promising schema uses vector space information retrieval paradigm, where documents are represented as vectors [2]. In our paradigm, we use features that are related to the resources that devoted locally and globally. As a result, each set of similar resources is specifically located to a topical agent. Below is an example of a document retrieved from a resource "drug.com" corresponding to an agent "Health". The features we proposed are: "URL", "title", "snippet", "IDF", "PageRank", "Global Rating", "Local Rating", "date", "time", and "home". The example below shows the proposed attributes and the corresponding values.

<ROW>
Agent: Health;
Resource: drug.com;
URL: http://www.drugs.com/cg/lipoma.html;
Title: "Lipoma » What You Need to Know";
Snippet: "A lipoma is a benign (non cancer) tumor made up of fat tissue. Lipomas can form anywhere in your body, but are usually found on the back, shoulders, neck, and head. The cause of lipomas is unknown. Some types of lipomas may run in families".

IDF: 0.7;
PageRank: 0.4;
Wiki: 1;
Agent: 0.3;
Resource: 0.7;
Home: 1;
Expiration: 1-15 second;
Type: info, video, image, social, link, web, article;
</ROW>
<ROW>
……
……..
</ROW>
……..

3.1.2 Features Weighting

The proposed schematic weights in our algorithm are influenced from our previous experiments using eight schematic weights: (1) '*URL*', '*title*', and '*snippets*' are attributes used to force a high impact weight for document relevancy. We assume that query terms in 'snippets' are more scattered than in 'URL' and in 'title'; in which, we assigned a maximum weight as 0.2, 0.1, 0.1 to URL, title, snippets, respectively. (2) The *PageRank* attribute plays a significant algorithm for moving the relevant pages on the top of the result list [25-27]. The Alexa networking was used to collect PageRank data for each domain. (3) The *Inverse Document Frequency* (IDF) emphasis on the number of documents involved in each resource that contains similar query terms, the value is computed as follows:

$$IDF_k = 1 + \log_2 \frac{n}{d_k} \quad (1)$$

where d_k denotes the document frequency of term k, where as n denotes the number of documents in the collection.

(4) The '*Local Ranking in Agent*" and "*Local Ranking in Resource*" attributes assume that at most we retrieve 10 resources from each agent and 10 documents from each resource. Therefore, if a document retrieved from an agent n and a resource m, the rank is $n+m$. Experimentally, we ordered resources in an agent based on the user's preferences. (5) In terms of the '*Wikipedia*' attribute, our perspective is that not all terms involved in the '*URL*' are labeled as domain-specific terms unless they shared with the Wikipedia content. For instance, if documents retrieved from a particular resource and such document is referenced by Wikipedia, the documents will be ranked high "0.10"; otherwise "0.0". (6) The *Date* attribute is used to give a high weight for a document that created recently in its resource, in which, 0.05 is assigned for document has been created at the moment, e.g. publicizing a link to such document in Twitter or news websites. (7) Similarly, if a document is referenced in a home-page, it will be ranked high in a "*Home*" attribute; the order of its link in the home page will determine the value of relevancy. That means, the first document is ranked "0.5" and the second document is ranked "0.4", and so on. (7) The "*Expiration*" attribute is parametric rather than metric; which means, if an agent retrieves a document from a resource within the expiry time, the document will be declined unless the search process is still active and the number of returning documents is still few. (8) Likewise, "*Type*" attribute is used for categorizing the results on the user's frontend. Means, assigns high priority to the results returning from Twitter, Wikipedia and so on. The following table shows the types of features and the schematic weighting at each one.

DOI: https://doi.org/10.30564/aia.v3i2.3219

Table 2. Features, types, and ranking schema

Attribute Keywords (T_1, T_2,..., T_n)	*Weight* 1.00
URL	0.0 - 0.20
Title	0.0 - 0.10
Snippet	0.0 - 0.10
PageRank	0.0 - 0.10
Inverse Document Frequency (IDF)	0.0 - 0.10
Local Ranking in Agent	1.0 - 0.10
Local Ranking in Resource	0.0 - 0.10
Wikipedia	0 or 0.10
Expiration	0.0 - 0.05
Type	0.0 - 0.05

3.2 Centralized Index

Often, indexing model shares similar document statistics with other models when data derived from different relevance matching assumptions①. The indexer distills information in the corpus of documents in the network into a format that is suitable for quick access by the query processor. This typically extracting document features by breaking documents down to their constituent terms, as well as extracting any statistics related to the terms in the documents of related resources, and then calculating any query related evidences represented in its attributes, as mentioned in the previous section.

As our approach runs in real-time and the proposal index table is quite small, the index built instantly in the memory instead of in the disk space, for the following reasons:

1) To overcome the problem of disk overhead.

2) To avoid the network delay, accessing the index through the internet for every query searching is fairly expensive to the users.

3) Accessing the data in the memory is mostly 10 times faster than any other medium.

Using the index can only determine whether searching string exists in a particular document or not. Because the index stores information regarding the degree of matching and other attributes, it is therefore considered to be an informational and knowledgeable index. Such an index can be used to determine not only document matching a query, but also can be used to compute the rank of documents.

Because not all documents in an agent satisfy all attributes, the index table is dispersed a bit in its content [3]. This is why we composited the index table into a structured form of hash table to scan the index linearly. This process is commonly referred to gripping through text. Gripping linearly through the text can be very effective process, especially influences the speed of modern computers. Depending on whether we look at the table rows (documents) or table columns (attributes) we can have a vector for each attribute or a vector for each document.

$$A_{Shopping} \rightarrow R_1[<d_1...><d_2...>...<d_a...>]R_2[<d_1...><d_2...>...<d_b...>]...R_m[<d_1...><d_2...>...<d_n...>]$$

4. Query Processing and Document Ranking

Intelligent retrieval systems are capable of understanding user intentions and use this knowledge in the information retrieval process. To support our intelligent retrieval model, advanced navigational techniques such as relevance ranking, natural language queries, and concept searching are required in evolving information retrieval systems [10].

Our agent-based search architecture comprises three types of local agents: dispatch, display, and search engine interface, while a set of distributed agents incorporated globally. In local agents, the interface agent uses REST API to communicate with remote global agents which are type of Web-accessible services; and in turn, agents communicate with each other to process the required task. The purpose of the interface agents is to facilitate and incorporate a variety of search services; whereas, each interface agent is designed to accommodate its target remote agents and failure modes. However, all these principal components are stimulated and controlled by a query processor which, in turn, deployed for document ranking to compute the degree of document relevancy [7]. For instance, given a query and a set of documents, a scoring function is usually utilized to determine the relevance degree of a document with respect to the query. Then a ranking list is produced by sorting in descending order of the relevance score. Generally, the ranking schema incorporates one of two types of evidences: dynamic or static. In dynamic, ranking exploits the heuristic forms of document evidences; whereas in static, ranking assigns a particular value for the evidence. As we experienced earlier in our previous approaches [4], we assigned a static value for an attribute in the document. We proposed a fixed schematic value of ranking for attribute presented in the documents as well as in its agents, including the highest rank if all query terms are available and the lowest if otherwise.

More contrast, given query 'Q' composites from three terms, the first step is to search those terms in the index file in resource '*R*' in an agent '*A*'. The

① The matching assumptions are also defined by agents.

 DOI: https://doi.org/10.30564/aia.v3i2.3219

corresponding posting list of matched documents is fetched and transferred to another table in the memory. The posting ranks for all selected attributes are summed linearly to represent the initial rank for each document. Thereafter, the posting list of documents will be ordered in initial order, and in some cases, set of documents may incorporate equal ranks. Based on our ranking schema, if set of documents retrieved from set of resources and corporate equal ranks, it must be re-ranked again using user preferences, e.g. homepage, social pages, news, etc. The final results are merged in the forward table before displayed to the user. First of all, *Jaccard Coefficient* is used to compute the similarity between the user's query '*Q*' and document attributes (URL, title, snippets):

$$\text{Similarity}(D,Q) = \sum_a \frac{|D \cap Q|}{|D| + |Q| - |D \cap Q|} = \frac{\sum_i^n a_i q_i}{\sum_i^n a_i^2 + \sum_i^n a_i^2 - \sum_i^n a_i q_i} * N(a) \quad (2)$$

where n is the number of terms in the query (Q), N is a normalized value for an attribute a

Then, the similarity weight is accumulated with the schematic weights of other attributes to represent the final rank of document D, as shown below:

$$Rank(D,Q) = Similarity(D,Q) + \sum_a Nomalize(a) \Lambda E > 0.6 \quad (3)$$

The pseudo code below shows the proposal ranking algorithm.

```
Initialize
Loop for each document 'D' in the collection;
C1→URL, title, snippet, C2→PageRank, Local, Wikipedia, IDF;
   Loop for all attributes 't' in document 'D';
      If C1 and attribute 't' = = C1 then increment rank 'R' to document 'D' by C1;
      Else If C1 and attribute 't' != C1 then assign rank 'R' to document 'D' by '0';
      if C2 and attribute 't' == C2 then increment rank 'R' to document 'D' by C2;
   End of loop for all attributes in document;
  Forward document D and its rank in table 'T';
End of loop for each document in collection;
  Sort documents in table 'T' Ascending;
   Loop for all documents in table 'T';
      Loop for each rank 'R' in table 'T';
        Sort rank 'R' alphabetically;
      End of loop for each rank;
    End of loop for all documents;
Display result.
```

5. Result Analysis and Discussion

Though our model is for selecting the appropriate schematic rank for some attributes proposed by our model, specifying the relevant agent for a user query topic and deterring the appropriate resources in right order is the main goal for our approach. By submitting a preliminary run to this system, the runs were validated by checking if they adhere to the TREC format, and the main evaluation metrics were returned. The evaluation metrics returned were based on 10 test queries (fully annotated but not used for the actual evaluation). Figure 2, 3, and 4 showed the main evaluation metrics (F1[①] for Agent Selection, and nDCG@5, nDCG@10, and nDCG@20 for both Resource Selection and Results Merging) for validating run among the online trial submissions. These metrics are the results with respect to the 50 evaluation topics, not including the 10 test topics for which the participants received the intermediate results (and towards which their systems might have been tuned). We often submitted consecutive runs to the model, either for a range of different techniques, e.g. ordering the resources in the agent using different schematic preferences, or maybe to determine suitable values for modeling hyper parameters. For the Agent Selection task, our approach has a substantial increasing in effectiveness over the systems submitted to TREC. For the Resource Selection task, we have a good improvement over other systems [4]. For the Results Merging task, we have better results over other systems. The test and training sets of queries involved for two tasks (Adhoc and Diversity) as well as some relevancy complexity. F-measure and Normalized Discounted Cumulative Gain (nDCG) are computed as follows:

$$F = \frac{2\ .\ Precision\ .\ Recall}{Precision + Recall} \quad (4)$$

The DCG accumulated at a particular rank position p {\displaystyle p} is defined as:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \quad (5)$$

The *nDCG* values for all queries can be averaged to obtain measure of the average performance of ranking algorithm. A perfect ranking algorithm, the DCG_p will be producing an *nDCG* of 1.0. All *nDCG* calculations are involved on the interval 0.0 to 1.0 cross-query comparable (Figure 2 and 3). Figure 4 and 5 are showed the accuracy of agent and resource selection, respectively; whereas Figure 6 shows the accuracy of result merging. Table 3 shows the Discounted Cumulative Gain for each attribute, in which, Wikipedia's attribute is identical and comparable with others.

① Usually F-measure called F1 because recall and precision are evenly weighted.

DOI: https://doi.org/10.30564/aia.v3i2.3219

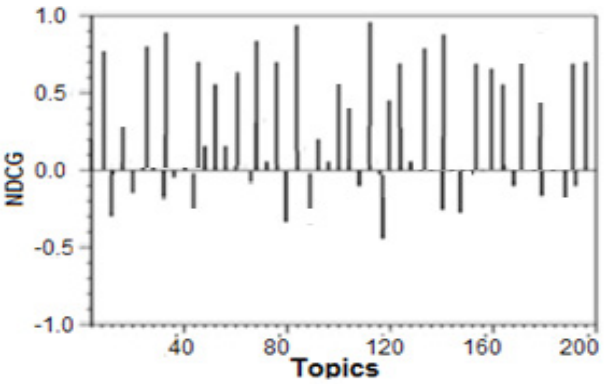

Figure 2. Normalized Gain NDCG @5 per topic

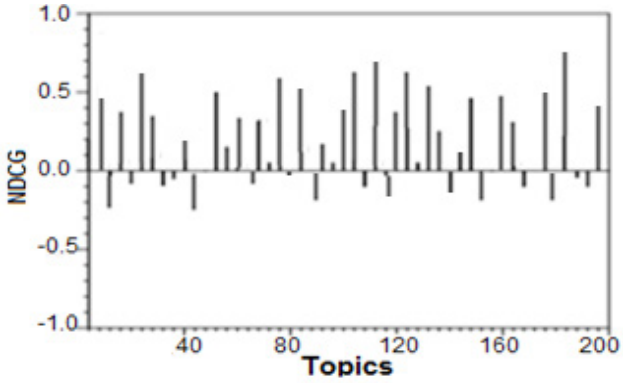

Figure 3. Normalized Gain NDCG @10 per topic

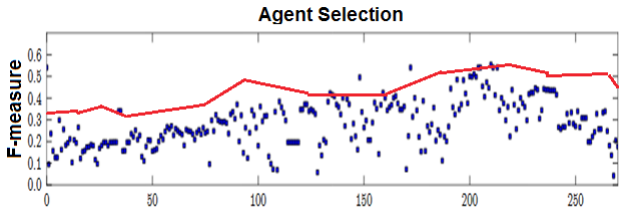

Figure 4. Accuracy for Agent Selection

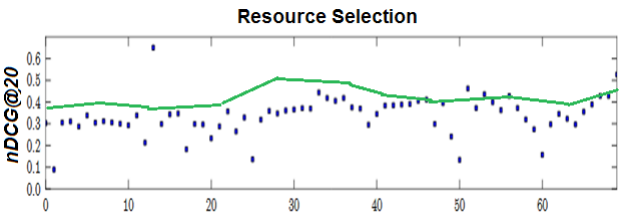

Figure 5. Accuracy for Resource Selection

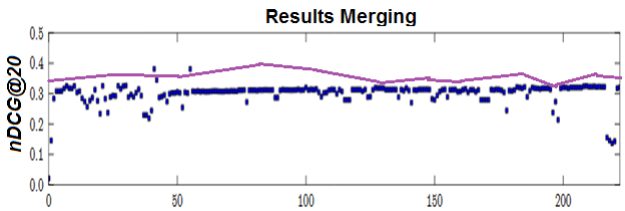

Figure 6. Accuracy for Results Merging

Table 3. Discounted Cumulative Gain @20 in Results Merging using individual attributes

Attributes	*nDCG@20 compared with baseline value*
Query,T_1 ,T_2.., T_n	Average query length = 3
URL	0.21651108
Title	0.31872399
Snippet	0.25437600
PageRank	0.35986654
Inverse Document Frequency (IDF)	0.37864310
Local Ranking in Agent	0.13982340
Local Ranking in Resource	0.17442129
Wikipedia	0.41865477
Date	0.11232875
Home	0.39543768

6. Comparisons & Outlook

The best performing methods, as shown in the Table 4, rely on indices based on single documents (rather than snippets) and combine evidence from classical retrieval algorithms such as variations on TF.IDF and language modeling. The best running models did not use external resources such as Wordnet and Wikipedia, but we showed how Wikipedia was played a very important feature for enhancing overall performance. A notable exception was the RS_clue web baseline, which used the collections' snippets in combination with the ClueWeb'09 collection to make size estimates. We assume that for a given query all engine results are readily available but more realistic scenario would be to first make a selection of a small number of promising engines, and to retrieve and re-rank this set of results. As shown in the Table 4, the best run performed nDCG@20 score 0.439. The organizers' baseline runs used the static rankings from the corresponding size-based resource selection baselines (RM_clueweb and RM_query pools). The results of the top 5 ranked resources were combined using a round-robin merge. In our perspective, querying the TREC FedWeb 2013 collection is different from such realistically created test collections, in which, it provides the real results of 157 real web search engines and each provides its own retrieval method and heterogeneous content types including images, PDF, plain-text, video, etc. Also, querying lots of resources e.g. 157 and selecting only 5 resources resulting network overload

 DOI: https://doi.org/10.30564/aia.v3i2.3219

which was not applicable in the real-time algorithm. Appendix (A) shows the 50 test queries, whereas Appendix (B) shows the ClueWeb19 collection used by other models.

Table 4. The performance of related approaches

Group ID	Run ID	nDCG@20
CMU_LTI	googTermWise7	0.286
	googUniform7	0.285
	plain	0.277
	sdm5	0.276
ECNUCS	basedef	0.289
ICTNET	ICTNETRM01	0.247
	ICTNETRM02	0.309
	ICTNETRM03	0.348
	ICTNETRM04	0.381
	ICTNETRM05	0.354
	ICTNETRM06	0.402
	ICTNETRM07	0.386
SCUTKapok	SCUTKapok1	0.313
	SCUTKapok2	0.319
	SCUTKapok3	0.314
	SCUTKapok4	0.318
	SCUTKapok5	0.320
	SCUTKapok6	0.323
	SCUTKapok7	0.322
ULugano	ULugFWBsNoOp	0.251
	ULugFWBsOp	0.224
dragon	FW14basemR	0.322
	FW14basemW	0.260

7. Experimental Results

The implementation is performed on 3.06 GHz Pentium Dual Core computer with 3 GB RAM, running on Windows 7. Java scripting programming language is used; since it is a web platform and an Object Oriented Language which has security packages. The proposed approach is based on a normalization scheme which uses mean value of our previous approaches. We have implemented the proposed approach on the "http://site.uottawa.ca/~falak081" website at University of Ottawa. Figure 7 shows our output interface for a query 'madam walker' (some results are beyond the scope of this research paper).

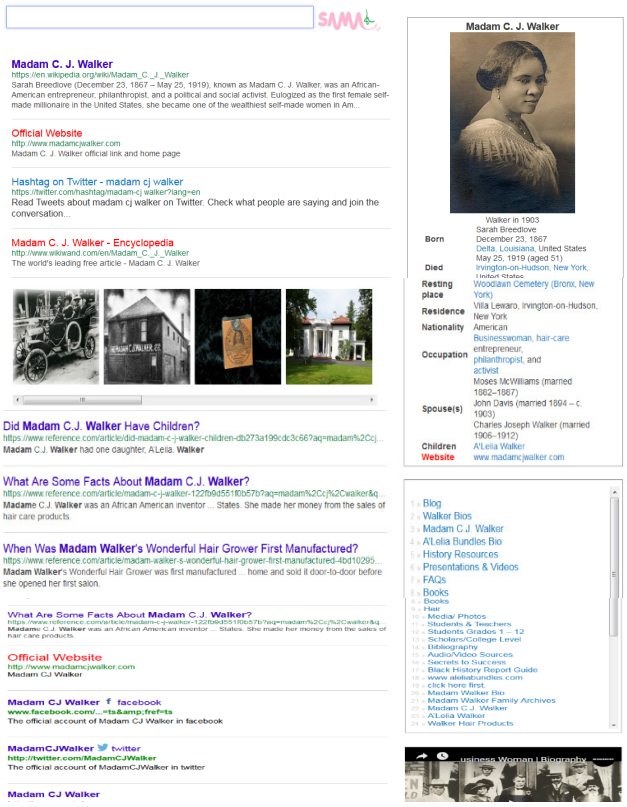

Figure 7. The interface of our approach

8. Conclusions

Distributed indexing is crucial to find relevant information on the network. Various indexing methods are used in a wide range of applications. In this research paper, we described the idea behind our indexing approach for extending the agent information retrieval system using multi-search criteria. We proposed a natural language processing approach for the keyword search in order to allow better matching with the textual descriptions. We implemented a combination scheme for matching multiple criteria that are formally structured in many attributes in the repository. The proposed approach is robust for determining the appropriate attributes. The proposed index is structured as a memory-based hash file that able to index the information of fifty resources distributed topically in several agents that totally satisfy most information retrieval and web search topics. Currently, hash files can be used only for indexing limited size of data. Hash index schema is robust if used in real-time environment since it comprises very fast search and query response time applicable. We believe that the traditional approaches based on the data crawled index are not applicable so far since data becomes diverse and grows impressively; especially when it is incorporated

DOI: https://doi.org/10.30564/aia.v3i2.3219

with social media.

Acknowledgments

This research was developed at the University of Ottawa as part of "SAMA" search engine①. We'd like to thank all the mobile assessors who participated in our user study, as well as the TREC at NIST for providing us by the training and testing data.

References

[1] YIGAL, A., CHIN, C., CHUN, H., and CRAIG, K. (1993). "Retrieving and Integrating Data from Multiple Information Sources". Intelligent and Cooperative Information Systems, Vol. 2, No. 2, pp. 127-158.

[2] Gerard, S., Wong, A., and Yang, C. (1975). "A Vector Space Model for Information Retrieval". Communications of the ACM, 18 (11): pp. 613-620.

[3] Gasser, L. (1988). "Large-scale concurrent computing in artificial intelligence research". In Proceedings of the third conference on Hypercube concurrent computers and applications.

[4] Al-akashi, F. (2014). "Using Wikipedia Knowledge and Query Types in a New Indexing Approach for Web Search Engines". PhD dissertation, University of Ottawa.

[5] Lenat, D., and Guha, V. (1990). "Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project,Addison-Wesley, Reading, Mass.

[6] Knoblock, A. and Ambite, L. (1997). "Agents for Information Gathering," in Software Agents, J. Bradshaw, ed., AAAI/MIT Press.

[7] Knoblock, A. (1996). "Building a Planner for Information Gathering: A Report from the Trenches," Proc. Third Int'l Conf. AI Planning Systems, AAAI Press, pp. 134-141.

[8] Hazem, M., Alaa, R., Ahmed, A., Sameh, G., and Nikos, M. (2015). "A New Automated Information Retrieval System by using Intelligent Mobile Agent". RECENT ADVANCES in ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES.

[9] Dhanapal, R. (2008). "An intelligent information retrieval agent". Knowledge-Based Systems, Volume 21, Issue 6, pp. 466-470.

[10] Jidé, O., David, K., Stuart, W., Nagi, W., Sadanand, S., Chris, G., and JoAnna, G. (1997). "SAIRE - A SCALABLE AGENT-BASED INFORMATION RETRIEVAL ENGINE", Proceedings of the first international conference on Autonomous agents.

[11] Rhodes, B. and Maes, P. (2000). "Just-in-time information retrieval agents". IBM Systems Journal , Volume: 39 Issue: 3.4.

[12] Qian, G. and Young-Im, C. (2013) . "A Multi-Agent Information Retrieval System Based on Ontology". Advances in Intelligent Systems and Computing book series (AISC, volume 194). Intelligent Autonomous Systems 12, pp. 593-602.

[13] Norbert, F., Mounia, L., Saadia, M., and Zoltan, S. (2004) "Advances in XML Information Retrieval". Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX, Springer.

[14] Ambite, J. and Knoblock, C. (2000). "Flexible and Scalable Cost-Based Query Planning in Mediators: A Transformational Approach", Artificial Intelligence Journal.

[15] Pang, L., Lan, Y., Guo, J., Xu, j., and Cheng, X. (2017). "Deep Rank: A new Deep Architecture for Relevance Ranking in Information Retrieval". In Proceeding of CIKM.

[16] Diaz, F., Arguello, J., Callan, J., and Crespo, J. (2009). "Sources of evidence for vertical selection". In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09).

[17] Arens, Y., Knoblock, A., and Shen, M., (1996). "Query Reformulation for Dynamic information Integration," J. Intelligent Information Systems, Special Issue on Intelligent Information Integration,Vol. 6, Nos. 2-3, pp. 99-130.

[18] MacGregor, R., (1990). "The Evolving Technology of Classification-Based Knowledge Representation Systems," in Principles of Semantic Networks: Explorations in the Representation of Knowledge, pp. 385-400.

[19] Finin, T. (1994). "KQML as an Agency Communication Language," Proc. Third Int'l Conf. Information and Knowledge Management, ACM Press, pp. 456-463.

[20] Arens, Y., and Knoblock, C. (1994). "Intelligent Caching: Selecting, Representing, and Reusing Data in an Information Server,"Proc. Third Int'l Conf. Information and Knowledge Management, Nat'l Inst. of Standards and Technology, pp. 433-438.

[21] Hsu, C. and Knoblock, C. (1996). "Using Inductive Learning to Generate Rules for Semantic Query Optimization," in Advances in Knowledge Discovery and Data Mining, G. Piatetsky-Shapiro et al., eds., AAAI Press, pp. 201-218.

[22] Owen, K., Alistair, M., Tim, S., and Justin, Z. (1998). "Methodologies for Distributed Information

① http://eecs.uottawa.ca/~falak081

 DOI: https://doi.org/10.30564/aia.v3i2.3219

Retrieval". In Proceeding of ICDCS of the 18th International Conference on Distributed Computing Systems, pp.66.

[23] Duhan, N., Sharma, K., and Bhatia, K., (1009). "Page Ranking Algorithms: A Survey", Proceedings of the IEEE International Conference on Advance Computing, 978-1-4244-1888-6.

[24] Brin, S., and Page, L., (1998). 'The Anatomy of a Large Scale Hypertextual Web Search Engine', Computer Network and ISDN Systems, Vol. 30, Issue 1-7, pp. 107-117.

[25] Brin, S., and Page, L., Rajeev, M., Terry, W., (1998). 'The PageRank Citation Ranking: Bring Order to the '. Technical report in Stanford University.

APPENDIX (A): TEST QUERIES

ID	Query
7001	LHC collision publications
7003	Male circumcision
7004	z-machine
7007	Allen Ginsberg Howl review
7009	linkedin engineering
7018	audiobook Raymond e feist
7025	M/G/1 queue
7030	Lyrics Bangarang
7033	Porto
7034	sony vaio laptop
7039	import .csv excel
7040	vom fass gent
7042	bmw c1
7046	tuning fork
7047	Dewar ask
7056	ROADM
7067	used kindle
7068	Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
7069	Eames chair
7075	zimerman chopin ballade
7076	Bouguereau
7080	lord of the rings hobbits theme
7084	Burn after reading review
7087	Jonathan Kreisberg discography
7089	varese ionisation
7090	eurovision 2012
7094	calculate inertia sphere
7096	touchpad scroll dell latitude
7097	best dum blonds
7099	lecture manova
7103	cysticbrosis treatment
7109	best place to eat pho in new york
7115	pittsburgh steelers news
7124	yves saint laurent boots
7127	which cities surround long beach ca
7129	avg home edition
7132	massachusetts general hospital jobs
7145	why do cats purr
7209	crab dip appetizer
7258	swahili dishes
7348	map of the united states
7404	kobe bryant
7406	does my child have adhd
7407	kim kardashian pregnant
7415	most anticipated games of 2013
7465	xman sequel
7485	bachelor party jokes
7504	leiden schools
7505	ethnic myanmar
7506	I touch myself singer dead

APPENDIX (B): TRAINING RESOURCES

ID	Name	Categories	ID	Name	Categories
e001	arXiv.org	Academic	e099	Bing News	News
e002	CCSB	Academic	e100	Chronicling America	News
e003	CERN Documents	Academic	e101	CNN	News
e004	CiteSeerX	Academic	e102	Forbes	News
e005	CiteULike	Academic	e103	Google News	News
e006	Economists Online	Academic	e104	JSOnline	News
e007	eScholarship	Academic	e106	Slate	News
e008	KFUPM ePrints	Academic	e107	The Guardian	News
e009	MPRA	Academic	e108	The Street	News
e010	MS Academic	Academic	e109	Washington post	News

DOI: https://doi.org/10.30564/aia.v3i2.3219

ID	Name	Categories	ID	Name	Categories
e011	Nature	Academic	e110	HNSearch	News,Tech
e012	Organic Eprints	Academic	e111	Slashdot	News,Tech
e013	SpringerLink	Academic	e112	The Register	News,Tech
e014	U. Twente	Academic	e113	DeviantArt	Photo/Pictures
e015	UAB Digital	Academic	e114	Flickr	Photo/Pictures
e016	UQ eSpace	Academic	e115	Fotolia	Photo/Pictures
e017	PubMed	Academic,Health	e117	Getty Images	Photo/Pictures
e018	LastFM	Audio	e118	IconFinder	Photo/Pictures
e019	LYRICSnMUSIC	Audio	e119	NYPL Gallery	Photo/Pictures
e020	Comedy Central	Audio,Video	e120	OpenClipArt	Photo/Pictures
e021	Dailymotion	Audio,Video	e121	Photobucket	Photo/Pictures
e022	YouTube	Audio,Video	e122	Picasa	Photo/Pictures
e023	Google Blogs	Blogs	e123	Picsearch	Photo/Pictures
e024	LinkedIn Blog	Blogs	e124	Wikimedia	Photo/Pictures
e025	Tumblr	Blogs	e126	Funny or Die	Video,Photo/Pictures
e026	WordPress	Blogs	e127	4Shared	Audio,Video,Books,Photo/Pictures
e027	Columbus Library	Books	e128	AllExperts	Q&A
e028	Goodreads	Books	e129	Answers.com	Q&A
e029	Google Books	Books	e130	Chacha	Q&A
e030	NCSU Library	Books	e131	StackOver ow	Q&A
e032	IMDb	Encyclopedia	e132	Yahoo Answers	Q&A
e033	Wikibooks	Encyclopedia	e133	MetaOptimize	Academic,Q&A
e034	Wikipedia	Encyclopedia	e134	HowStu Works	Kids,Q&A
e036	Wikispecies	Encyclopedia	e135	AllRecipes	Recipes
e037	Wiktionary	Encyclopedia	e136	Cooking.com	Recipes
e038	E? Online	Entertainment	e137	Food Network	Recipes
e039	Entertainment Weekly	Entertainment	e138	Food.com	Recipes
e041	TMZ	Entertainment	e139	Meals.com	Recipes
e042	The Sun	Entertainment,Sports,News	e140	Amazon	Shopping
e043	Addicting games	Games	e141	ASOS	Shopping
e044	Amorgames	Games	e142	Craigslist	Shopping
e045	Crazy monkey games	Games	e143	eBay	Shopping
e047	GameNode	Games	e144	Overstock	Shopping
e048	Games.com	Games	e145	Powell's	Shopping
e049	Miniclip	Games	e146	Pronto	Shopping
e050	About.com	General	e147	Target	Shopping
e052	Ask	General	e148	Yahoo? Shopping	Shopping
e055	CMU ClueWeb	General	e152	Myspace	Social
e057	Gigablast	General	e153	Reddit	Social
e062	Baidu	General	e154	Tweepz	Social
e063	CDC	Health	e156	Cnet	Software
e064	Family Practice notebook	Health	e157	GitHub	Software
e065	Health Finder	Health	e158	SourceForge	Software
e066	HealthCentral	Health	e159	bleacher report	Sports
e067	HealthLine	Health	e160	ESPN	Sports

 DOI: https://doi.org/10.30564/aia.v3i2.3219

ID	Name	Categories	ID	Name	Categories
e068	Healthlinks.net	Health	e161	Fox Sports	Sports
e070	Mayo Clinic	Health	e162	NBA	Sports
e071	MedicineNet	Health	e163	NHL	Sports
e072	MedlinePlus	Health	e164	SB nation	Sports
e075	U. of Iowa hospitals and clinics	Health	e165	Sporting news	Sports
e076	WebMD	Health	e166	WWE	Sports
e077	Glassdoor	Jobs	e167	Ars Technica	Tech
e078	Jobsite	Jobs	e168	CNET	Tech
e079	LinkedIn Jobs	Jobs	e169	Technet	Tech
e080	Simply Hired	Jobs	e170	Technorati	Tech
e081	USAJobs	Jobs	e171	TechRepublic	Tech
e082	Comedy Central Jokes.com	Jokes	e172	TripAdvisor	Travel
e083	Kickass jokes	Jokes	e173	Wiki Travel	Travel
e085	Cartoon Network	Kids	e174	5min.com	Video
e086	Disney Family	Kids	e175	AOL Video	Video
e087	Factmonster	Kids	e176	Google Videos	Video
e088	Kidrex	Kids	e178	MeFeedia	Video
e089	KidsClicks?	Kids	e179	Metacafe	Video
e090	Nick jr	Kids	e181	National geographic	Video
e091	Nickelodeon	Kids	e182	Veoh	Video
e092	OER Commons	Kids	e184	Vimeo	Video
e093	Quintura Kids	Kids	e185	Yahoo Screen	Video
e095	Foursquare	Local	e200	BigWeb	General
e098	BBC	News			

DOI: https://doi.org/10.30564/aia.v3i2.3219