Intelligent Agents for Bilingual Information Retrieval from the World Wide Web

Yehia Helmy[1], Mohamed Nour[2], Atef Ghalwash[1], Mai Hamdallah[1]
The Faculty of Computers and Information, Helwan University[1]
The Electronics Research Institute, Cairo[2]

Abstract

This paper presents a proposal of a multi-agent information retrieval system to help users in finding the required documents on the WWW. The proposed system architecture contains several components mainly: the interface agent, learning agent, filtering agent, searching agent, user profile, text information extractor, and others. The evolutionary mechanism of the proposed system is based on the concepts of genetic algorithms (GAs). New filtering agents are created by the genetic operators such as crossover and mutation. The filter agent filters the obtained results and lists to the user a reduced number of documents with high probability of being relevant to him/her. That filtering is based on a user profile such that the learning agent builds by analyzing the document examples presented by the user.

Keywords:

Intelligent Agents, Search Engines, Information Retrieval, and Genetic Algorithms

1. Introduction

Nowadays, there is a huge amount of data on the web. This means that there is information on almost any topic available online. Search engines are a popular way to locate information. In some cases, the information provided is too general to efficiently solve individual user's information needs [1]. Although users differ in their specific interests, they tend to provide short queries that do not adequately describe the specific information they seek. The same results are shown to all users even though they may be looking for different types of information [1]. Users of online search engines often find it difficult to express their need for information in the form of a query. If users can identify examples of the kind of documents they require then they can employ a technique known as relevance feedback [2]. Relevance feedback covers a range of techniques intended to improve a user's query and facilitate retrieval of information relevant to a user's information need. A user must revise a big number of uninteresting documents and consult several search engines before finding relevant information. To alleviate this problem, personalization becomes a popular remedy to customize the web environment towards a user's preference [3].

Moreover, intelligent software agents can be used to find information several ways: 1) by browsing, 2) by sending a query to a search engine or 3) by following existing categories in search engines. Intelligent agents are programs that act on behalf of their human users to perform laborious information-gathering tasks [3].

[4] presented an intelligent agent for information retrieval. That agent uses document references gathered by a user to build a hierarchical model. The agent then uses the model to interpret the queries and to filter the results returned by the search engines of the web. It also uses information gathered while observing the ongoing activities of the user to select the most relevant parts of this profile.

[5] mentioned that, genetic algorithms (GAs) constitute a different solution to profile adaptation. Typically, a population of agents is maintained that collectively represent the user interests.

[6,7] presented that, a significant problem in many information filtering systems is the dependence on the user for the creation and maintenance of a user profile. 'News Weeder' is a net-news filtering system that addresses that problem by filtering the user rate his/her interest level for each article being read, and then learning a user profile based on these ratings.

The vast increase of multilingual content on the Internet has created the need for information access across languages and cultures. Cross Language Information Retrieval (CLIR) is an important area of research and development that aims to overcome the cross-lingual access

problem. This is done by enabling the users to retrieve documents written in one language – often called the *target* language - based on queries typed in another - often called the *source* or *query* language [8,9].
This paper is organized as follows: section 2 presents the objectives and related work. Section 3 presents the proposed multi-agent architecture. The implementation work and experimental results are presented in section 4 while section 5 concludes the remarks.

2. Objectives and Related Work

A large number of papers describe systems that are related to what we are attempting here. The most relevant ones are assistants for information retrieval, using mechanisms based on filtering, learning or relevance feedback. These systems are using a model to represent documents, abstracting the information that they may enclose [11]. Some developments include [3, 4, 7, 10, 12, 13].
The main objective of this research work is to propose a multi-agent system for information retrieval. The proposed system is different from those presented by others. Letizia, Syskill&Webert, and Webnaut assist users browsing the Web using TF-IDF vectors [11] and a naive Bayes classifier respectively. TF-IDF stands for term frequency and inverse document frequency relationship. Syskill&Webert's users use a three-point scale to rate pages on a specific topic. The proposed system uses a five point scale to gather as information about the user's opinion in the documents presented to him/her.
The WebMate system as an example generates personal newspapers based on multiple TF-IDF vectors each of them representing a topic of interest. Webnaut is an intelligent agent system that uses a genetic algorithm to collect and recommend Web pages. A feedback mechanism adapts to user interests as they evolve. Webnaut uses a metasearch agent to search five search engines with a single query while the proposed system allows more than one query modeling user interest to be used at the same time widening search range. NewsWeeder [7] is a news filtering system which prioritizes USENET news articles for a user using the *minimum description length algorithm*. The development of these agents was focused in the accomplishment of some specific tasks, such as browsing assistance and news filtering, but these techniques can not be reused in further agent developments with additional requirements. Instead, the user profiling architecture we have presented supports the development of capabilities shared by the majority of textual-based agents. This can determine the relevance of a given piece of information, pro-actively suggest new information and establish common information interests among a group of users.
AIRA [13] uses the mass of information residing on the user's workstation to initialize a user profile. This approach is accurate as asking the user to select documents s/he find interesting as proposed by our system. *PersonalSearcher* [4] defines the user profile by using a textual case-based reasoning approach in order to detect specific subjects that the user is interested in and organizes them in a hierarchy.
All of the above works are monolingual (English). Our proposed system facilitates search in both English and Arabic without burdening the user. This is done by translating the user profile generated from the presented documents.

3. The Proposed Multi-Agents System Architecture

In this work, a multi-agent information retrieval system is proposed. The system elements have the capability to support a user of extracting and retrieving information in both English and Arabic languages. In this system, each user is assigned four distinct types of agents: interface agent, learning agent, filtering agents, and search agents. Moreover, multiple users can use the system at the same time but no interaction between them is permitted. The system consists of several components as illustrated in Figure 1. Each component is described in details as shown in the following sections. Before discussion, it is important to conduct the web document representation as follows:

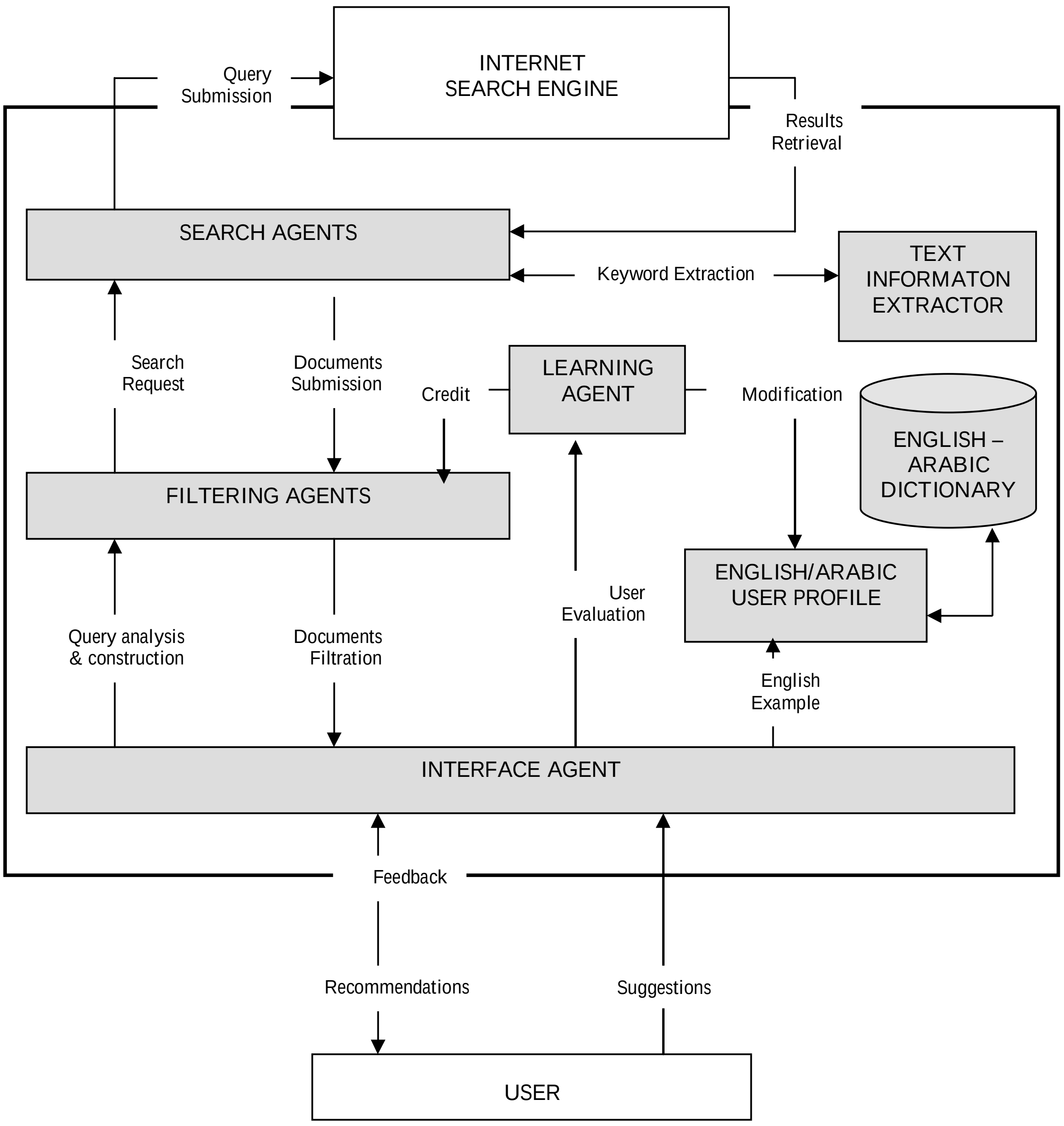


Figure 1: An Overview of the Proposed System Architecture

3.1 <u>Web Document Representation</u>

The vector space model is used here to represent information included in Web documents. The basic representation for the filtering agents as well as for the parsed HTML files is the weighted keyword vector. When an HTML file is processed by the Text Information Extractor, a clear-text version is generated. The text keywords can be weighted by using formula (1)[12,18].

$$w_{i=}\frac{\left(\frac{tf_i}{tf_{\max}}\right)}{\sqrt{\sum_{j=1}^{N}\left(\frac{tf_j}{tf_{\max}}\right)^2}}$$

(1) where

- § tf_i is the frequency of the keyword i in all texts in which it appears (term frequency);
- § $tf_{\max}$ is the maximum keyword frequency of all keywords in the user profile; and
- § N is the number of keywords in the user profile.

[12] stated that the term frequency tf and the inverse document-frequency idf weighting schema *tf-idf* is popular in similar applications, but a collection of documents is necessary be applied. To implement the *tf-idf* weighting, the user must create a collection of documents. A sufficient number of examples must be provided before the learning agent can begin the process of searching for new information.

3.2 Text Information Extractor

This component, which consists of a *lexical analyzer* and a *stop-word removal* algorithm, analyzes HTML pages producing a keyword vector. The lexical analyzer tokenizes the input HTML page in three steps: It selects all hyperlinks in the document, removes the HTML tags, and finally removes any script language commands. The stop-word algorithm removes all high-frequency words such as “for” and “the,” all common words such as days and months, and all numbers. (Common words are included in a word list file.) Then the extractor creates a weighted keyword vector for the HTML page on hand.
Figure 1 visualizes the architecture of the proposed system. The user gives examples that embody his/her interest and is then presented with a group of sites by the interface agent. S/he then provides a feedback by rating the sites included. Based on the user’s ratings, credit is assigned to the related filtering agents and the user profile weights are adjusted.

3.3 User Profile

User’s interests are extracted simply from documents presented by the user as examples. [14] quoted that “*A list of keywords is one of the most primitive representations of information needs*”. In information gathering, it is assumed that contents of a document can be approximately expressed by a set of terms excerpted from the document. Additionally, all words included in a document set are extracted to create a document vector (User Profile), who’s every element consists of both a keyword and its weight. Using formula (1), user profile keyword weights are also calculated.
The user profile which is maintained by the learning agent can be considered as a model of what documents the user finds relevant. The user profile is considered a virtual document that includes only those keywords of interest to the user.
User profiles are manipulated in one of two ways: First, the relevant documents are provided as training examples by the user. Second, the user prompts the agent with how relevant an encountered document is. The agent applies adjustments to the user profile according to the relevance feedback provided. Moreover, if the user is interested in finding Arabic documents matched with the presented document example, an Arabic user profile is built by translating each keyword in the User Profile keyword vector through the English/Arabic dictionary. The Arabic profile is used to match Arabic documents found on the WWW.

3.4 Filtering Agents (FAs)

A filtering agent is based on a keyword vector which is used to create the search query posted to the search agents. In addition to a keyword vector, a filtering agent also contains other information such as whether it was created by the user explicitly or not. If it was, it is treated more favorably by giving it a relatively initial high fitness. Moreover, the genotype of the information filtering agents is essentially a keyword vector. Each keyword is assigned a randomly generated operator (AND, OR, NOT).
The filtering agents are considered filters that allow only the documents that are close to the user profile’s weighted keyword vector to pass through. Each filtering agent selects the documents that is closest to the user profile and calculates how confident is that the specific document will interest the user.
In order for a filtering agent to assess its similarity to a given WWW page, it has to compare the user profile to the vector representation of the text inside that page. As mentioned above, each document is represented by a weighted keyword vector.
When two keyword vectors are compared for similarity, the cosine angle between them is computed. Formula (2) is used to calculate the similarity between documents D_S and D_D [12].

$$sim(D_D, D_S) = \frac{\sum_{k=1}^{N} W_{Dk} W_{Sk}}{\sqrt{\sum_{k=1}^{N} W_{Dk}^2 \sum_{k=1}^{N} W_{Sk}^2}}$$ **(2)** **where**

§ *WDk* and *WSk* are their weighted keyword vectors; and
§ *N* is the number of keywords.

3.5 Evolutionary Mechanisms

The evolution of the fitness agents is controlled by their individual fitness. The population is split into two parts. A variable number of the top ranked (the best performers) of the whole population in one part and the rest in another. The rank of an agent is based solely on its fitness. Each part is allowed to produce offspring separately [9,15].

Moreover, new filtering agents are created by crossover or mutation. All operators are applied to the evolvable part of the agents, the genotype. The other part of the agents, the phenotype contains information that should not be evolved, usually instructions on how to handle the evolvable part. Figure 2 illustrates the process of this mechanism.

The genetic operators applied on the FAs' populations are:

a. ***Crossover:*** The one point *crossover* operator, given two filtering agents returns two new agents that inherit a part of the keyword vectors of the parents. This operator randomly splits the two keyword vectors, and then recombines them to produce two new keyword vectors, precisely, creating two new agents. Keywords from the two keyword vectors are exchanged if and only if the same words don't appear in the new sets more than once.

b. ***Mutation:*** The *mutation* operator takes the genotype of an agent as argument and creates a new agent that is a randomly modified version of its parent. The new "mutated" keyword is a randomly selected keyword from the user profile.

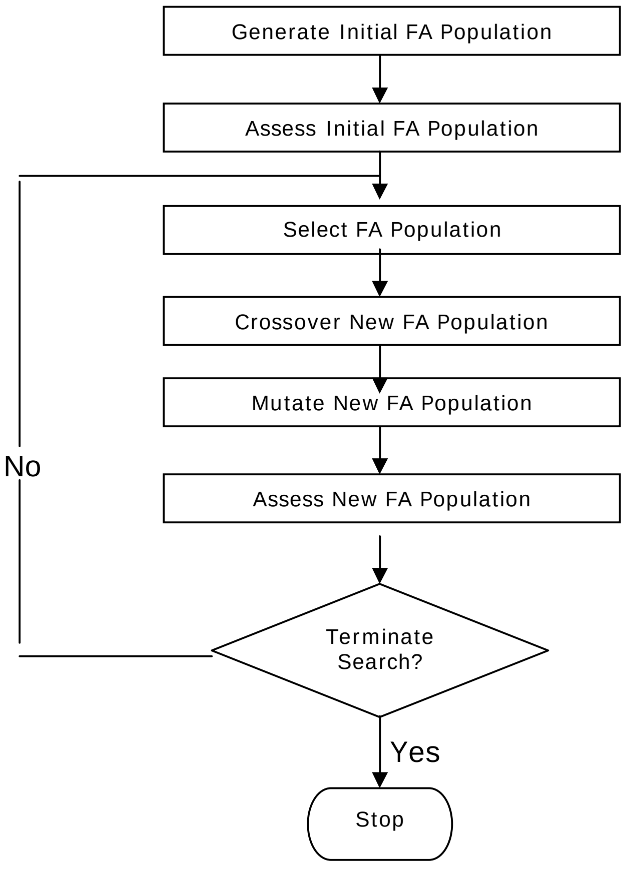


Figure 2: The Process of the Evolutionary Genetic Algorithms

The system keeps track of all the fitness agents generated. No two agents, in all the generated populations, can have the same genotype. So, if during crossover, any of the new offspring's genotype is identical to another agent (in the same generation or in a previous one), the agent's genotype is mutated to ensure that the generated agent is unique.

3.6 Search Agents (SAs)

Each filtering agent issues *requests* to search agents about the type of documents they are interested in finding. Each request includes a search query (list of keywords) and the ID of the filtering agent that made the request. Each Search Agent randomly selects which Filtering Agents' requests to fulfill. This happens in the following way: All Filtering Agents' requests are placed on a blackboard. When a Search Agent selects a request, that request is erased from the blackboard. Then the system proceeds to the next Search Agent, until all Filtering Agents' requests are fulfilled.
A search agent is based on a query string that it should utilize when querying the WWW search engines (Google in this case). A distinct characteristic of search agents include that they are parasitic in the sense that they are utilizing existing WWW search engines to find information and not dig it up on their own.
Search Agents are responsible for posting queries to Google, collect the results and present them to the filtering agents that requested them. If a link is already visited by the user or already proposed to him or her, it is not considered again. For each of the remaining links, the system fetches its HTML contents, processes them and saves them for consideration by the related filtering agent later on. All retrieved URLs are stored in a database. The database is used for keeping track of the URLs previously presented to prevent presentation of duplicate information to the user.

3.7 Interface Agents (IAs)

Not all documents introduced by filtering agents are presented to the user. The interface agent decides if the filtering agent is going to present something to the user by ranking the proposed documents using formula (3) (confidence level) [15,16,17]:

$$C_i = sim(D_D,D_S) \,.\, F_i \qquad \textbf{(3)}$$ **where**

- § *i* is the document number; and
- § *F* is the fitness of the filtering agent that proposed the document.

Only those documents that the interface agent is confident that they would resemble the user interest according to their confidence level then introduced to the user. The top *n* documents are selected from the ranked list, where *n* is a number that indicates the amount of items that the user is interested in including in one result page.
Highly relevant documents are presented through a simple interface that lets the user rate the results according to the categories in Table 1. This direct feedback ensures system responsiveness to changes in user navigation behavior as users select links to topics not included on their original agendas.
One would notice that if an information filtering agent doesn't present anything to the user then its credit would remain constant. To diminish the fitness of inactive agents, the fitness of those agents is adjusted according to a linear decay function.
We suggested the values which are similar to the ***AND*** column in Table 1 as the difference between each user feedback situation and the other is recognizable opposite to the values in the ***OR*** and ***NOT*** columns.

User's feedback	r				
	Query keywords			Terms not part of the query	Fitness agent that provided the document
	AND	OR	NOT		
Very interesting	2	1	-2	2	2
Interesting	1	1	-1	1	1
Indifferent	0	0	0	0	0
Irrelevant	-1	-1	0	-1	-1
Very irrelevant	-2	-1	0	-2	-2

Table 1: *The scale of values for grading documents. The keywords forming the query, terms that are not part of the query, and the Fitness Agent that presented the document.*

3.8 Learning Agents (LAs)

Other than building the user profile, the Learning agent alters the frequencies of words that are in the User Profile and filtering agents according to the user's rankings on specific documents. Changes are applied according to the formula (4) [12,17,18].

$$Tf_D = Tf_D + \left(\left(\frac{r}{100}\right) * Tf_R\right) \quad (4)$$

where

- § Tf_D is the word frequency in the Dictionary;
- § Tf_R is the word frequency in the recommended URL; and
- § r is the user's evaluation in the range -2 to 2.

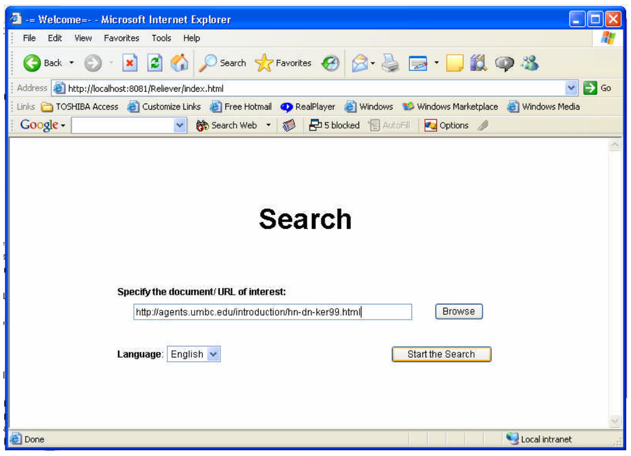
The exact combination of words and logical operators that brings about each recommended URL is known so as the Filtering agent that obtained it. As a result, formula (4) is applied not only to common words as described above, but also to the words that form the exact query. Using formula (4), the learning agent recalculates the weight of each word in the User Profile as well as the fitness of the specified filtering agent.

4. Implementation

The proposed multi-agent system architecture discussed above was developed, implemented, and applied. The development was done on an IBM/PC supported by the Microsoft Windows XP operating systems. Several software tools were used during the implementation mainly: Java software development kit JSDK1.5, Java Server Pages (JSP), Google API1s, and Hypertext Markup Language (HTML). To run the developed system, the environment should be supported by Tomcat, Apache Server, and JDK1.4.1.

Moreover, the system permits the emulation of Tomcat, Jakarta, and Apache. Figure 3(a) represents a snapshot of the system's input form. Here, the user specifies the path of the document example that embodies his/her interest whether on disk or on the WWW. The search language is also specified. Figure 3(b) provides an example of the system's results page. The top 10 documents presented to the user are those that the system is confident that they are closest to the user's interest. For each document, the following information is available to the user: its title, a snippet, its URL, and a combo box for the user to provide the degree of interest in this particular document.

An example of part of the document the user gave as input to the system is shown in figure 4.



-= Welcome=- - Microsoft Internet Explorer

Address: http://localhost:8081/Reliever/index.html

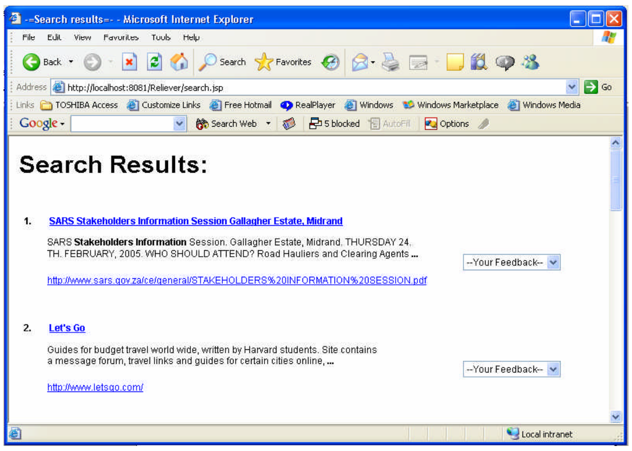
Search

Specify the document/ URL of interest:

http://agents.umbc.edu/introduction/hn-dn-ker99.html

Language: English

(a)



-=Search results=- - Microsoft Internet Explorer

Address: http://localhost:8081/Reliever/search.jsp

Search Results:

1. SARS Stakeholders Information Session Gallagher Estate, Midrand

   SARS **Stakeholders Information** Session. Gallagher Estate, Midrand. THURSDAY 24. TH. FEBRUARY, 2005. WHO SHOULD ATTEND? Road Hauliers and Clearing Agents ...

   http://www.sars.gov.za/ce/general/STAKEHOLDERS%20INFORMATION%20SESSION.pdf

   --Your Feedback--

2. Let's Go

   Guides for budget travel world wide, written by Harvard students. Site contains a message forum, travel links and guides for certain cities online, ...

   http://www.letsgo.com/

   --Your Feedback--

(b)

Figure 3: System snapshots (a) Input form (b) Results presentation and user evaluation collection

Genetic Algorithms were invented to mimic some of the processes observed in natural evolution. Many people, biologists included, are astonished that life at the level of complexity that we observe could have evolved in the relatively short time suggested by the fossil record. The idea with GA is to use this power of evolution to solve optimization problems. The father of the original Genetic Algorithm was John Holland who invented it in the early 1970's. ……

Figure 4: Part of the Document the User Gave as Input to the System

4.1 Experiments Results

Concerning the experimental work, figure 5 demonstrates a sample of queries randomly generated from the user profile. The size here is five keywords per query. Figures 5(a) and 5(b) show examples of both English and Arabic queries respectively.

offspring OR solution OR problem OR fitness NOT population
crossover AND population OR parent OR genetic NOT selection **(a)**
population individual OR fitnes OR selection NOT problem

(b)

• •• AND •• • ••• OR

Figure 5: Sample of Queries Generated

Table 2 shows part of the English user profile created from the file in Figure 4. Table 3 shows a division of the Arabic user profile generated when a user chose to search for Arabic documents providing the file in figure 4 (English text) as input (user interest).

Keyword	Frequency	Weight
Genetic	23	0.362711
algorithm	27	0.425791
problem	6	0.09462
Search	13	0.205011
Space	7	0.11039
selection	11	0.173471
.	.	.
individual	25	0.394251
generation	13	0.205011
population	18	0.283861
solution	18	0.283861

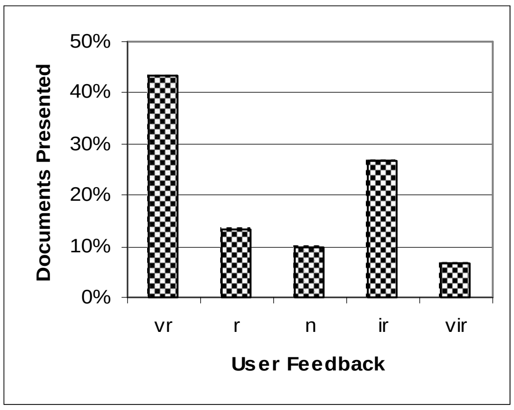
Table 2: English User Profile

The translation of the user profile was carried out using the English-Arabic online Dictionary ectaco. Available at www.ectaco.co.uk

Keyword	Frequency	Weight
••••••	23	0.362711
•••••	27	0.425791
••••••	6	0.09462
• ••	13	0.205011
•••	7	0.11039
••••••	11	0.173471
.	.	.
•••	25	0.394251
• ••	13	0.205011
• •••	18	0.283861
••	18	0.283861

Table 3: Arabic User Profile

In our set of experiments, the system was setup to work with four Filtering agents. Each Filtering agent has five terms per query. The system was run for three times with the same user profile. The average number of documents presented by each generation of the three times is presented in Figure 6(b). Figure 6(a) shows that 43% of the documents presented to the user are *very relevant* (vr), 13% are *relevant* (r), 10% are *neutral* (n), 27% are *irrelevant* (ir) , and 7% are *very irrelevant* (vir) based on user feedback. The changes in the fitness of filtering agents as a result to user feedback in each generation are shown in Figure 7.



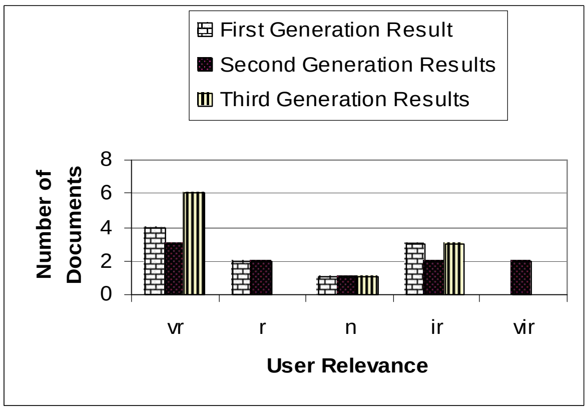


Figure 6: (a) The percentage of documents returned in each relevance group, (b)The number of documents presented in each generation grouped by relevance value.

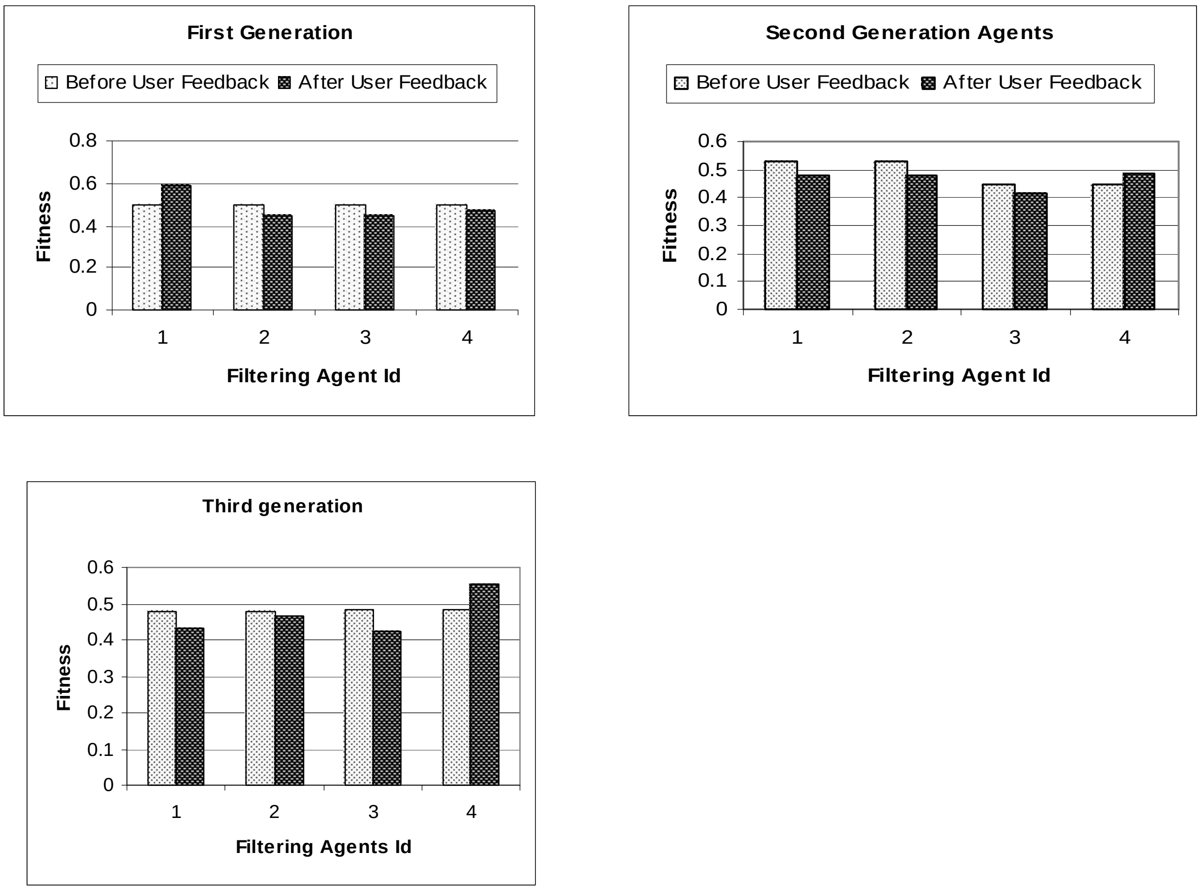


Figure 7: Changes Applied to Filtering Agents' Fitness According to User Relevance Feedback

5. Conclusions and Recommendations

We have presented a multi-agent information retrieval system, capable of generating user queries to apply search and evaluating the results according to a user's interests.

Moreover, this work discussed the idea of using evolving populations of agents for personalized information filtering and search. In particular, we introduced the idea of integrating different types and populations of agents; Interface, Learning, Filtering, and Search Agents into an ecosystem. Different agents compete and cooperate as the ecosystem evolves towards better fitness levels. Based on the proposed architecture, we have built a working system that provides to its users personalized information from the World Wide Web. Agents that are of service to users or other agents will run more often, reproduce, and survive while incompetent agents will be removed from the population. We have also shown that the system can apply a bilingual search covering a wider range of user needs. The experiments showed that the system can indeed converge to areas interesting to its users providing elite results. Several experiments were applied and run and the results were close to those presented before.

One important aspect of our design is the separation of agents. We are hoping to extend our experiments in multiple user domains where each user will have its own set of filtering agents but they will be able to share their search agents. In this case, a new user would not have to train its own search agents but utilize the existing ones that were already trained by the other users. Further more the separation of filtering and search agents provides the ability to support real online operation in the future, where users connect to their Internet service provider, send out requests and get the results by the time they reconnect. The search agents would collect all the documents requested and the information filtering agents present them to the user upon re-connection to the network. The user's filtering agents would do then all filtering and generate the digest. We should also note at this point that although we concentrated on document retrieval and evaluation, the same approach can in principle be applied to other media like image and audio, for which features can be automatically extracted. Such a research direction would further increase the system's scope.

Our system models a user's changing information need within a search session. We would like to expand our research to handle longer term models and implicit feedback from a user's browsing patterns between sessions. Further more, expanding dictionary terms and languages handled in it will help bridging the gap made due to language constraints and serve a wider range of people working in different domains with different languages.

6. References

1. Trajkova, J., and Gauch, S., " Improving Ontology-Based User Profiles", The Proceedings of RIAO 2004, University of Avignon (Vaucluse), PP. 380-389, France, 2004.
2. Shahabi, C. and Chen, Y., "Web Information Personalization: Challenges and Approaches", The 3nd International Workshop on Databases in Networked Information Systems (DNIS 2003), Aizu-Wakamatsu, Japan, 2003.
3. Chen, L. and Sycara, K., "WebMate: A Personal Agent for Browsing and Searching", The Proceedings of the 2nd International Conference on Autonomous Agents and Multi Agent Systems, AGENTS '98, ACM, PP.132-139, 1998.
4. Daniela, G.and Analía, A., "PersonalSearcher: An Intelligent Agent for Searching Web Pages", The Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI: Advances in Artificial Intelligence, Lecture Notes in Computer Science; Vol. 1952, PP.43-52, 2000.
5. Douglas, W. O. and Bonnie, D., "Evaluating Cross-Language Text Retrieval Effectiveness", The Cross-Language Information Retrieval, Gregory Grefenstette, ed., Kluwer Academic, PP. 151-161, 1998.
6. Lieberman, H., "Letizia: An Agent that Assists Web Browsing", The Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95), PP. 924–929, 1995.
7. Lang, K., "NewsWeeder: Learning to Filter News", The Proceedings of the 12th International Conference on Machine Learning (ICML-95), Lake Tahoe, CA., PP.331-339, 1995.
8. Demetriou, G., Skadina, I., Keskustalo, H., Karlgren, J., Deksne, D., Petrelli, D., Hansen, P., Gaizauskas, R.and Sanderson, M., " Cross-lingual Document Retrieval, Categorisation and Navigation Based on Distributed Services", The First Baltic Conference. Human Language Technologies- The Baltic Perspective. Riga, Latvia. PP.107-114, 2004.
9. Weiguo, F., Michael, D. G., and Praveen, P., "Discovery of Context-Specific Ranking Functions for Effective Information Retrieval Using Genetic Programming", The IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, PP. 523-527, 2004.
10. Pazzani, M., Muramatsu, J., and Billsus, D., "Syskill & Webert: Identifying Interesting Web Sites", The Proceedings of the 13th National Conference on Artificial Intelligence. Vol 1, PP. 54–61, 1996.

11. Salton, G. and McGill, M.J., "Introduction to Modern Information Retrieval", McGraw-Hill Inc., 1983.
12. Zacharis, Z. N. and Panayiotopoulos, T., "Web Search using a Genetic Algorithm", The IEEE Internet Computing, Vol. 5 No. 2:, PP.18-26, 2001.
13. Bottraud, J.C., Bissom, G., Bruandet, M.F., "An Adaptive Information Retrieval Research Personal Assistant", The Workshop (AI2IA) of Artificial Intelligence and Information Access and Mobile Computing – The International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.
14. Lingras, P.J. , "Evolutionary Information Retrieval", The Proceedings of the Fifth Joint Conference on Information Sciences, Vol. 1, PP.166-169., 2000.
15. Alexandros, M. and Pattie, M., "Amalthaea: An Evolving Multi-Agent Information Filtering and Discovery System for the WWW, Autonomous Agents and Multi-Agent Systems, Vol.1 No.1, PP.59-88, 1998.
16. Schiaffino, S., and Amandi, A., "User - Interface Agent Interaction: Personalization Issues", The International Journal of Human-Computer Studies, Vol.60 No.1, PP.129-148, 2004.
17. Zhiming, J., (1999): "Survey Report on Cross- Language Text Retrieval", 1999.
18. Monika, H., "The Past, Present, and Future of Web Information Retrieval", The Proceedings of the SPIE, Vol. 5296, PP. 23-26, 2003.