# A Personal Search Agent System

**Ahmed Al-Zeyodi and Ali Al-Qayedi**

Etisalat University College, P.O.Box:980, Sharjah, UAE
asz1212@euc.ac.ae, alqayedi@euc.ac.ae

***Abstract* — This paper demonstrates a personal search agent system which enables the users to personalise their search and hence to save their time and effort. The system uses its own ranking algorithm and provides two types of search: a direct search (via a local database) and an indirect search (via third party search engines). The system achieves search personalisation through an initial user interaction that decides on what search criteria should be considered more important than others. The system also attempts to learn the user behaviour through a user feedback mechanism. This mechanism can cause two search attempts with the same query to return two different results; because of changing the user preferences even though the search is run on the same data set. Issues regarding the setup, implementation and experimental results of the system are illustrated here.**

***Index Terms* — Personal search agents, search engines, ranking algorithms.**

## I. Introduction

The World Wide Web has dramatically changed the way people live. More and more tasks can be performed on the Web now. However; searching the internet, using the well-known public search engines, remains to be one of the earliest activities people try when they first start using the Internet. In principle search engines do return the required results, however; they do not guarantee that those results would be upfront and sometimes they are not clear about their practices according to which the returned results were ordered or ranked. Also, nearly all public engines give preferences and advanced search options, which still do not meet the level of personalisation expected, especially the need for a mechanism that enables refining the search results returned for a given query so as to make closer to the user needs. These facts and others have introduced the need for personal agents that act as search assistants to Web users. A number of those were developed as standalone desktop applications, but still many do rely on public engines results and hence are incapable of satisfying the personalisation level needed. This paper describes a prototype desktop system termed Personal Search Agents (PSA), which attempts to solve some of the personalisation issues involved with search engines so as to enable users to retrieve results that are close to their own interests.

The paper is organised as follows: background on personal search engines is given in section 2. Section 3 describes the developed PSA system. Implementation issues and experimental results are given in sections 4 and 5 respectively. Those results are discussed in section 6 while conclusions with future recommendations are summarised in section 7.

## II. Background

There are numerous search personalisation systems available in the market, examples of these include: WebSeeker, Copernic and WebFerret [3, 4, 7]. All of these are desktop applications that provide indirect search (i.e. a search via remote search engines). Each of those applications provide a way of collecting results from various search repository and display them to the user, that is a Meta engine but on the user desktop rather than being a Web site. All of these systems attempt to improve the quality of their results by allowing the user to verify the existence of the returned links, in addition to the ability to modify their search and choose among a number of search filtering mechanisms.

The PSA system described in here provides not only an online indirect search, but also a direct search through a local database content that is pre-prepared off-line. It also focuses on improving the type of results that is displayed to the user beside its equality. The main contribution is the ability to personalise the search to the users through a feedback mechanism.

## III. Personal Search Agent System (PSA)

### *A. System Architecture*

Basically, as shown in Fig 1, the system consist of 5 main stages these are: firstly getting the user request, secondly searching and retrieving the results, thirdly analysing and ranking the results, then getting the user feedback and finally updating the search rules. First the user should enter the search keyword and the type of search. In case of a direct search the PSA system searches for the keyword in the local document database. However; for an indirect search the system passes the keyword to the specified set of search engines

to perform the search and retrieve results. The returned results will be ranked according to some metrics applied by the PSA system before it is shown to the user. If the results do not match the user interest he/she can change the rules of the search such as increasing or reducing the weight of a given word in the search query which causes the rules to be updated automatically according to the user feedback. The user needs to perform the search again and a new result will be shown to reflect the changes made.

*B. System Overview*

The PSA system attempts to personalise the search for the user by operating as follow:

1) Getting user request and retrieving the results

The system provides the user with two different search methods: a direct search and an indirect search. Indirect search takes the user query and sends it to a number of search engines including Google, Yahoo and Alltheweb. On the other hand, direct search takes the user query and search for it in the local document database. After getting the results the system displays it on both the graphical user interface and the default browser.

2) Analysing and Ranking results

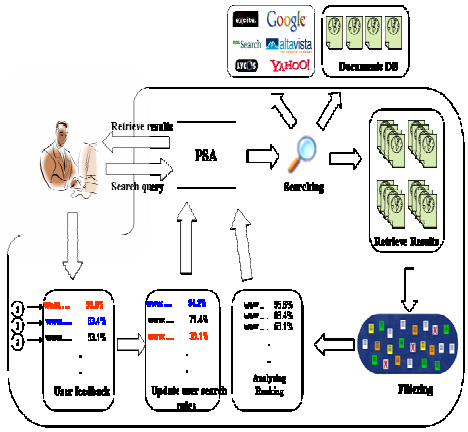
After fulfilling the search and retrieving the required data, the PSA system performs a ranking of the sites. Each element of the returned results is given a rating depending on some rules applied by the PSA system. Examples of these rules are:

- How many search engine among the list does index this site?
- How many times does the keyword appear in the title of the document/site?
- How frequent is the keyword in each document/site?
- Is the keyword important in the document/site? On other words is it emphasized? i.e. written in bold, italic or underlined.

During this stage the system also takes care of removing any duplication among the results. The returned links are then sorted according to their rank from highest to lowest. The result is finally displayed to the user in both the PSA window as well as the browser.

3) User feedback and Updating the user search rules

At first glance, the PSA system may not provide the required results to the user. It needs some time to acquire sufficient knowledge of the user interest and learn more about his/her likes and dislikes. This is achieved through the user feedback mechanism. Every time a result is shown by the system, the user is given a chance to rectify this result in future searches by updating their profile through increasing/decreasing the weight of some factors in the ranking formula or completely disabling some fields in it. The search rules will be updated automatically after getting any feedback from the user. It is through this user interaction




mechanism where the PSA system achieves its goals of personalising the search results.

Fig.1: System Architecture

*C. Algorithm*

1) Direct PSA Ranking Model

The direct PSA ranking model is the algorithm that is developed in this system for ranking documents returned by the direct search of the locally indexed database. The main reason for introducing such an algorithm is to overcome some of the problems encountered in both of the famous ranking algorithms: Pagerank[2][6] and Hyperlink Induce Topic Search (HITS) [1][5]. The ultimate aim is adding a user personalisation touch to the returned search results. This means that the anatomy of the Web that considers the in-links and out-links as the two main factors in PageRank and HITS will not be the only playing factors in the PSA algorithm. Instead other factors such as title weight, keyword repetition and keyword importance is also involved in the algorithm.

The direct PSA ranking algorithm can be worked out as follows:

```
Rank= TitleWeight*(TitleCount/Max TitleCount)
      +FrequencyWeight*(FrequencyCount/MaxFrequencyCount)
      +ImportanceWeight*(ImportanceCount/MaxImportanceCount)

DocumentRank=Rank/(Rank+1)
```

(1)

Where:

**TitleWeight**: the weight given by the user to the keyword that is found in the title.
**FrequencyWeight**: the weight given by the user to the repetition of the keyword in the documents.
**ImportanceWeight**: the weight given by the user to the Importance of the keyword in the document.
**TitleCount**: the number of occurrences of the keyword in the document title.
**MaxTitleCount**: the maximum number of keyword repetition in all titles.
**FrequencyCount**: the number of occurrences of the keyword in the document body.
**MaxFrequencyCount**: the maximum number of keyword repetition in all documents.
**ImportanceCount**: the number of times the keyword is emphasized i.e. made important in the document.
**MaxTitleCount**: the maximum number the keyword is emphasized i.e. made important in all documents.

2) Indirect PSA Ranking Model

The indirect PSA ranking algorithm is also introduced to rank the results of the Indirect search (Meta search). The main idea of this algorithm is to utilise the ranking returned by "Google", "Yahoo" and "Alltheweb" search engines. This algorithm combines the ranked results produced by the three engines and produces a newly ranked PSA result. For example; assuming that the results size is ten (10), so the first site in the results of all the three search engines is given a rank of ten. The second site rank is assigned a rank of nine (9) and so on until the last site in the results is given rank of one (1).

The Indirect PSA rank of page (A) can be calculated as follows:

$$\text{Indirect\_PSA}(A) = \text{GoogleRank} + \text{YahooRank} + \text{AlltheWebRank} \quad (1)$$

If any of the three search engines does not have page (A), its rank will be assigned to zero (0).

*C. User Interface*

The GUI of the PSA system is implemented using the Java language. The system functionalities were integrated into the interface incrementally. All the functionalities were verified successfully. Fig. 2 shows a snapshot of the interface.

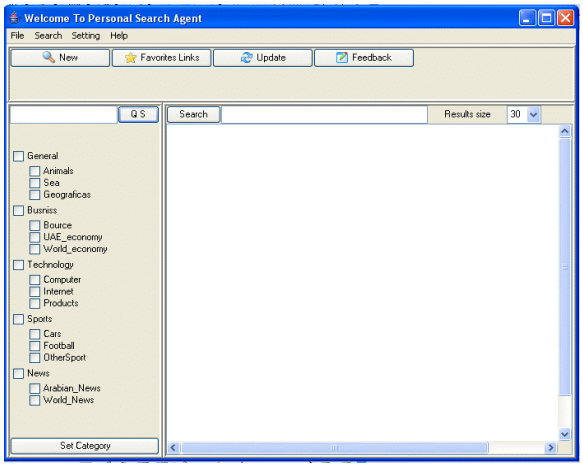



Fig. 2: GUI of the PSA system

## IV. Implementation Issues

Our system is a desktop application which was developed using the Java programming language under the Windows operating system. It is compiled using JCreator LE compiler. On the other hand, the system databases were implemented using MySQL.
The system is composed of a number of libraries and classes these include: a Swing library to implement the GUI and an SQL library to allow the interaction between the system and its databases.

## V. Experimental Results

The system has been tested singly and in comparison with the Google search engine. This section shows a sample scenario of the results returned by the system as an example. The returned search results for the term **Arabian News** after performing the search on Google only, on the PSA indirect search only and on the PSA indirect search with the user feedback are shown in Fig. 3, 4 and 5 respectively.

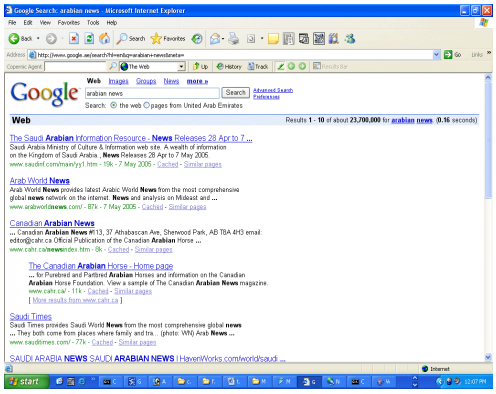



Fig. 3: Google results for "Arabian News"

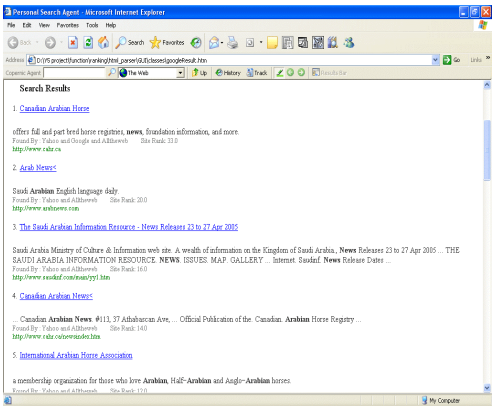



Fig. 4: PSA results for "Arabian News"

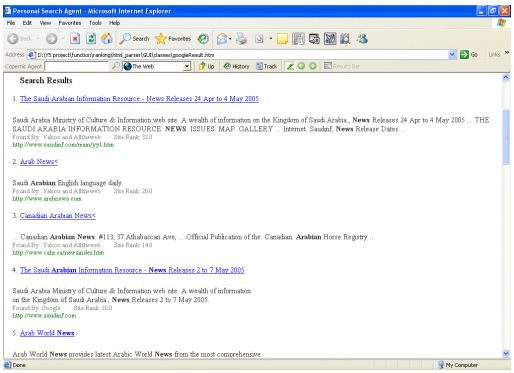



Fig. 5: PSA Results after user feedback

The three previous Figures show three different results as expected. The resulting links and their ranking in Fig. 4 are different from those in Fig. 3 since the weighting used by the PSA indirect ranking algorithm is based on three search engines Google, Yahoo and Alltheweb rather than on Google only as in Fig. 3.

Also Fig. 5 shows the result after the user has added the word "News" to his frequent keyword list by increasing its weight and giving it a higher rate as shown in Fig. 6. This has affected the returned results, so even though the user has attempted the search using the same search term i.e. **Arabian News**, the PSA indirect search algorithm has returned two different results (Fig. 4 and 5) due to the personalization element introduced by the user feedback.

## VI. DISCUSSION

The PSA system was tested thoroughly using different test scenarios and was found to meet the essential requirement which is to personalise the search results in accordance to the user needs. However, the implemented

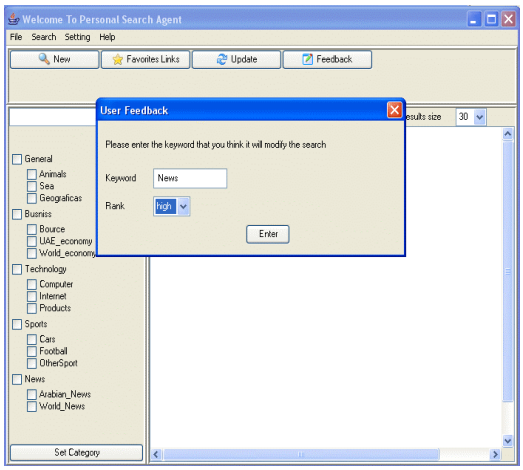



Fig. 6: User's feedback to the system

system remains a prototype and still has a number of limitations. Firstly, the PSA system provides a text-based search only and offers no multimedia search support at its current status. Secondly, the PSA system is a desktop application rather than a Web site, so it needs to be installed on the user system which has to have a Java Runtime Environment (JRE). Thirdly, the PSA direct search is limited to the content of the local documents database which is supposed to be updated frequently according to a parameter set by the user. Finally, the PSA indirect search currently interacts with three search engines only. This was found to be sufficient for the purpose of this application. More search engines can be easily added, however, this can affect the system performance.

## VII. CONCLUSION

This paper has described a Personal Search Agent system that adds an element of user preference to the existing conventional search engines. We show that our system is particularly useful as it includes a user interaction through a feedback mechanism. The system allows the user to get a new result that is closer to his/her needs without changing the search term which saves the user's time and effort.

## REFERENCES

[1] Amy N. Langvill and Carl D. Meyer, "The Use of the Linear Algebra by Web Search Engines", *Journal of the International Linear Algebra Society* No. 33, Dec., 2004, pp. 2-6.

[2] Brin S. and Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Proc. 7th Int. World Wide Web Conf.*, Brisbane, Australia, 14-18. April, 1998.

[3] Bluesquirrel web site, WebSeeker 5.0, http://www.bluesquirrel.com/products/webseeker/,.

[4] Copernic web site, “Copernic Desktop Search”, http://www.copernic.com/en/products/desktop-search/index.html

.[5] K. Baharat and M. Henzinger. "Improved algorithms for topic distillation in a hyperlinked environment", *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104-111, Aug. 1998.

[6] Wang, Dewitt, "Computing pagerank in a distributed internet search system", *Proceedings of the 30th International Conference on Very Large Databases*, Toronto, 2004.

[7] FerretSoft, “WebFerret: MetaSearch Utility for Windows”, http://www.webferret.com/learn.htm.