A Novel Model of Autonomous Intelligent Agent Topic Tracking System for the World Wide Web

Ejiofor Christopher
University of Port-Harcourt, River State, Nigeria
ejioforifeanyi@yahoo.com
Osuagwu O.E
Federal University of Technology, Owerri, Nigeria
drosuagwu@yahoo.com
Nwachukwu E.O.
University of Port-Harcourt, River State, Nigeria
enoblesciences@yahoo.com
Theo van der Weide
Radboud University, Nijmegen, the Netherlands
tvdw@cs.ni.nl

Abstract

Autonomous agents are software systems situated within and a part of an environment that senses stimuli in that environment, acts on it, over time, in pursuit of its own agenda so as to effect what it senses in the future. Autonomous agents take action without user intervention and operate concurrently, either while the user is idle or taking other actions. The internet encompasses a large number of documents to which search engines try to provide access. Even for many narrow topics and potential information needs, there are often many web pages online. The user of a web search engine would prefer the best pages to be returned. The use of autonomous intelligent agent topic tracker will help to make decision on behalf of the user, by narrowing the search domain and decreasing the human computer interaction, phenomenally. Previous research works on information retrieval system usually consists of long list of results containing documents with low relevance to the user query. Thus, the goal of this paper is to build an Intelligent Agent Topic Tracking System, that employs document concepts to track identical document related to the researcher's needs within a publication topic development. The system solely refines the user query as well as retrieving the result from a search engine with the help of Google API and refines the noisy result produced using Document-document Similarity model and the Document Component model to find similar topic documents in the document pool indexed by the search engines. In addition, the Web Structure Analysis model will use the hub and authority algorithm to evaluate the importance of web pages or to determine their relatedness to a particular topic. Finally, clustering is used to automatically group document pool into similar topics.

Keywords: Intelligent agent, document retrieval, search engine, topic tracking

Introduction

Document retrieval has been the subject of debate for several years because of its importance within the academic field. Research work has been hectic and unreliable since the search engines used are ineffective and time consuming. The search engines act as the foundation of interaction with voluminous knowledge stored in the World

Wide Web[23]. It has been associated with several disadvantages such as generation of too much noise and creation of long lists of results containing documents with low relevancy to the user query.

Many researchers nowadays spend a lot of time; approximately 70% locating materials related to their research work so as to study the state of art on the topic using the current search engines. The engines get thousands of results by deviating from the basics and focusing on quantity rather than quality (Blankson, 2008). This situation makes research work more complicated by making the researcher look for more information from documents than he or she actually need. Since research work and retrieval processes play a significant role in the productivity of a researcher, he or she is obliged to work tirelessly locating relevant documents containing information. He spends a lot of time and resources looking for materials related to the research instead of focusing more on innovation.

It is these problems that prompted the need to build an Intelligent Agent Topic Tracking System (IATTS), which employs document concepts to track data identical or related to the researcher's needs and requirements within a publication (Spink, Michael & Zimmer, 2008). IATTS is able to check and maintain the track of the dynamic nature of the Web and any advancement or changes made in relation to a user's interests and his or her satisfaction (Chen et al., 1998). The paper explores the features of IATTS contained within current search engines and tries to solve the problem by suggesting its introduction. It is more efficient and effective advanced search engine that makes research work easier and time saving together with exactness[23].

The current load of conducting a research work calls for a more and a proper search engine. Therefore, a solution to the search engine problem is necessary. One of the solutions would be to develop a new framework that will refine the noisy results produced by the current search engines during research work. Secondly, since research is very important to the society, learning should be made more personalized by application of intelligent agent which can assist the user refinery (Semantic Web project, 2006). Finally, the problem could be rectified by generating user reviewed results. The introduction of an intelligent agent topic tracking system is a major boost in solving the various research related problems.

The system solely refines the user query as well as retrieving the result from a search engine with the help of API. In relation to the problem, the IATTS tackles the shortfall using clustering. During collections of documents associated with a researcher's query, clustering is applied to categorize topics or research areas and later assist in tracking of the development of the topic of interest on the Web (Crestani, 2002). This assist researcher in checking track of the development of the research problem together with sections of development interests for growth and future discovering of fresh ideas related to the topic.

Related Research

Agents are software or hardware entities that perform a set of tasks on behalf of a user with some degree of autonomy (Russell and Norwig, 1995; Hoavar, 1998). In order to do this, an agent has to embody a certain amount of intelligence, which includes the ability to choose among alternative courses of action, to plan, to communicate, to adapt to changes in the environment and to learn from experience (Eissa and Alghamdi, 2010). The design of such mechanism for intelligent agent with differing degree of intelligence has been proposed in literature. These include: *reactive agents* which respond reactively to the changes that they perceive in their environment, *deliberative agents* that plan and act in a goal-directed fashion, *utility-driven* *agents* that act in ways designed to maximize a suitable utility function, *learning agents* which modify their behavior as a function of experience, and agents that combine different modes of behavior (Russell and Norwig, 1995; Hoavar, 1998).

Intelligent Information search and retrieval processes play a vital role in the productivity of a researcher. Every knowledge researcher, has to do extensive searches at some point in time to find information that may help or show that certain aspect of a research topic have already been covered before. Search engines provide the basic means of interaction with the massive knowledge base available on the World Wide Web (Zaka and Maurer, 2007). Ideally this will lead to agents that can adjust themselves to what a user wants and wishes, and what he or she is (usually) looking for, by learning from performed tasks and the way users react to the results of them. Some examples include WebWatcher (Joachims et al., 1997), Personal WebWatcher (Mladenic, 1996), Fab (Balabanovic, 1997; Balabanovic and Shoham, 1997) which learn user interest using user feedback and recommend web pages for users; and software agents for mail handling and electronic news filtering (Maes, 1997). These agents have been implemented using several methods such as inverted index, Boolean querying, knowledge base, Neural Network, probabilistic retrieval, genetic algorithm and machine learning approach (Baker and McCallum, 1998; Salton and Buckly 1988; Baeza-Yate, and Ribeiro-Neto, 1999; Frakes and Baeza-Yates, 1992; Belkin and Croft, 1992) to retrieve information from the World Wide Web.

Meta-Search Engines

Meta-Search engines or meta-crawlers are sites that take queries in form of keywords or natural language, send them to a large number of search engines and return the result to user. Meta-Search engines use three methods to search the web: Direct list of search engines, Sequential Search and Concurrent Search.

The Direct list of search engines sends the user query directly to a list of search engines and acquires their result for the query as if the user directly posed his query in each of the search engines in isolation (Barfourosh, 2011). This approach saves the user’s time and may also cover some search engines that the user has never tried. The results from the search engines are ranked using parameters such as search engine popularity, query terms and so on.

The Sequential Search allows user to select some search engines from a list and send the user query to these selected search engine. These meta-search engines wait to receive all the results from the search engines and then display them; these make the entire process too slow because the search engines have different capability and speed.

The Concurrent Search is similar to sequential search method, but it does not wait to receive the whole result from all search engines. Concurrent Search take inputs that are supported by all search engines that is uses, or it must convert the user’s query into a standard form supported by every search engine, which implies that the lowest search engines features will determine what user can enter. The results that are received first from any of the search engines are displayed for the user, while the subsequent results received will be gradually added to the received result (Barfourosh, 2011). The primary motivation being that the web is huge and most search engines in isolation cover only a small fraction of the web and have low recall and precision in their search result. This approach decreases the time before the user sees the first result from the search engine, but the transformed query may never satisfy the user intention.

Since meta-search engines do not allow for input of many search variables, their best use is to find hits on obscure items or to see if something is on the web or increase recall and precision. but the real convenience is finding the best result quickly and not getting the largest number of bad results. However, most meta-search engines results as in Dogpile (Kleinberg, 1999), Mamma (Jaczynski et al., 1999), Metacrawler (Yang et al., 2000) and Askjeeves (LookOff, 2000) represent cumulative search results over other search engines, but they still do not cover the entire web.

Framework

High Level Model of the Proposed Solution

The web is similar to a graph, in that links are like edges and web pages are like nodes. Several approaches have been proposed to overcome the current limitations of web Crawlers. Some approaches use web structure (relation between links) to guide web Crawlers in finding their path through the web and some approaches use web content (text within each page) to perform the same thing. We shall use a combination of these two aspects of web search to improve the functionality of web crawling strategies.

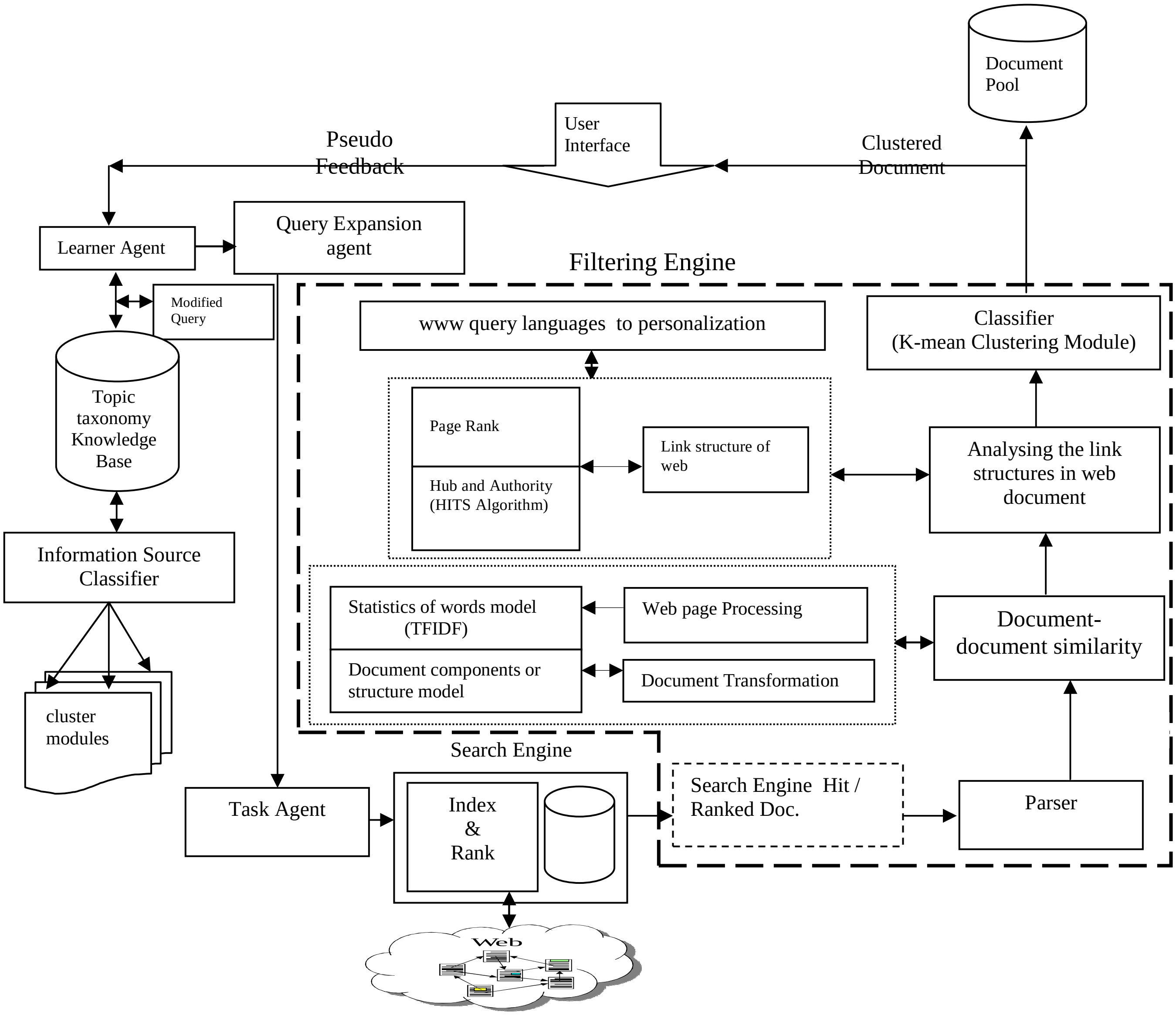
Document Pool
User Interface
Pseudo Feedback
Clustered Document
Learner Agent
Query Expansion agent
Filtering Engine
Modified Query
Topic taxonomy Knowledge Base
www query languages to personalization
Classifier (K-mean Clustering Module)
Page Rank
Hub and Authority (HITS Algorithm)
Link structure of web
Analysing the link structures in web document
Information Source Classifier
Statistics of words model (TFIDF)
Web page Processing
Document components or structure model
Document Transformation
Document-document similarity
cluster modules
Search Engine
Task Agent
Index & Rank
Search Engine Hit / Ranked Doc.
Parser
Web

Figure 1.0 The Architecture of the Proposed Intelligent Agent Topic Tracking System (Fact-Finder)

Properties of the High Level Model

The architecture of IATTS has many components with specific function to ease document retrieval. The intelligent agent topic tracker will run directly with any chosen search engine. The system will use search engine API to retrieve query result from search engine, which will be further classified based on the document data content and the structural relationship of each document with the query.

The system consist of the following components: User Interface, Knowledge base, Information source, Query processor, Learner agent, a document-document similarity measure, document link structure similarity measure, a clustering module and document pool

(i) User Interface :- Is the contact point between the user and the agent. It receives user data in form of query and presents the relevant search results (Halavais, 2008). Consequently, it observes actions and behaviour of the user and relays the pseudo-feedback on the search results to the machine learner. It also retrieves user preferences in terms of search engines to be used during research, visualization together with clustering parameters (Ramos and Cote, 2008)

(ii) Knowledge Base (KB) :- It includes varied sub-symbolic representations of topic categories. Each of the representations is got through analyzing the co-occurrence chance of main words in document within a given topic. These would propose significant terms used by the ML to reorganize exact term in relation to certain characteristics of a given subject domain [9].

(iii) Information Sources (IS) :- refers to the totality of data sources within the internet and an example includes database[9].

(iv) **The Query Expansion Model**

Query expansion is a technique, widely used in information retrieval, for obtaining additional terms relevant to a given query (search keywords). It is usually used to help information searchers express their intentions more accurately and increase the precision of search results. In Fact-Finder, however, its main purpose is to evaluate the relevance of search keywords to the document topic concept retrieved from search engine. Figure 2.0 shows a query expansion algorithm in Fact-Finder. Its general framework is as follows:

Getting relevant terms from the Web dynamically, Fact-Finder does not use any special dictionaries for query expansion, but it uses the Web (the existing Web documents) as the source of relevant terms. As shown in step 1 of Figure 2.0, it finds the Web documents relevant to the user query dynamically by submitting that query to a general Web search engine. The relevant terms are extracted from those documents. Since there is an immense corpus on the Web, terms relevant to any kind of search keywords can be obtained, even peculiar proper nouns, technical terms, etc.

Co-occurrence-based evaluation of term relevance..

The mutual relevance of terms is evaluated on the basis of their co-occurrence in the documents. In steps 2 and 3, the co-occurrences of the search keywords and other terms are counted in 30 documents retrieved by the general search engine in step 1. That is, the system lists all distinct terms contained in 30 documents, and counts for each term the number of documents that contain both the search keyword and that term. To reduce the computational time, Fact-Finder handles a pair of a page title and a snippet in the search result as a single document and does not download the actual documents.

Using a Pseudo-Feedback Technique

It is difficult to determine the term relevance from only the results of a single document search on the general search engine. Even closely relevant terms often have few co-occurrences in the 30 documents of the first search. Fact-Finder, therefore, re-evaluates such low co-occurrence terms: selecting terms to be re-evaluated from the first search results (steps 4 and 6), formulating new queries by adding the selected terms to the original query (steps 5 and 7), and performing the co-occurrence-based evaluation again for each formulated query (steps 8 and 9). The pseudo-feedback process treats topic terms as follows. First, as shown in step 4, low co-occurrence topic terms in the first search results are selected for re-evaluation prior to other non-topic terms.

Steps 6 and 7 are also important for topic terms. In these steps, non-topic terms are added to the original user query for the pseudo-feedback. However, the main purpose of this is to get new topic terms that were not obtained through the first search rather than to re-evaluate the added non-topic terms because search results can be improved in many cases by using additional terms. The improved search results are more likely to contain good terms like topic terms.

```
1. Get a document set $D_0$ relevant to a user query $Q_0$, where search keywords are $w_{01}$ ,... , $w_{0n}$,
       by sending $Q_0$ to a general search engine.
2. Count co-occurrences of search keywords and other terms in the document set $D_0$.
3. Let $W_{H0}$ and $W_{L0}$ be a set of terms whose co-occurrences exceed a certain threshold and a
       set of  the other terms, respectively. $W_{H0}$ is considered relevant to the query $Q_0$ and will be a
       part of the query expansion result.
4. Pick up at most four topic terms $w_{t1}$-$w_{t4}$ from $W_{L0}$.
5. Formulate four queries $Q_{T1}$-$Q_{T4}$ by combining $w_{t1}$-$w_{t4}$ with $Q_0$ (for example, $Q_{T1}$="$w_{01}$ ...
       $w_{0n}$  $w_{t1}$").
6. Clustering all terms in $D_0$ to at most three clusters: $W_1=\{w_{11}, \ldots, w_{1m}\}$, $W_2=\{w_{21}, \ldots, w_{2k}\}$
       and  $W_3=\{w_{31}, \ldots, w_{3j}\}$.
7. Formulate three queries $Q_1$-$Q_3$ by combining $W_1$-$W_3$ with $Q_0$ (for example, $Q_1$="$w_{01}$ ... $w_{0n}$
       $w_{11}$ … $w_{1m}$").
8. Get document sets $D_{T1}$-$D_{T4}$ and $D_1$-$D_3$ by sending $Q_{T1}$-$Q_{T4}$ and $Q_1$-$Q_3$ independently to a
       general search engine.
9. Count co-occurrences in $D_{T1}$-$D_{T4}$ and $D_1$-$D_3$. Sets of high co-occurrence terms $WTH_1$-$WTH_4$
       and $WH_1$-$WH_3$, as well as $WH_0$ in step 3, are query expansion results.
```

Figure 2: An Algorithm for Fact-Finder Query expansion procedure.

(v) **Web Structure Analysis**

Web structure is linked and is used to identify connectivity of the web page to a given topic. This is possible since web pages have in-links and out-links. It helps in ranking of sites and pages and their significance from the perspective of people citation irrespective of the topic. Under Web structure analysis, there exists Hub and Authority Pages. The significance of web pages can be obtained from link structure of the web where two kinds of pages are acknowledged from the page links within intelligent agent topic tracking system, the hubs are good sources of links while authorities are good sources of content. Figure 3 is a diagram illustrating the hubs and authorities. In clustering, a superior hub page points many authorities while a good page is that pointed by a number of good hubs

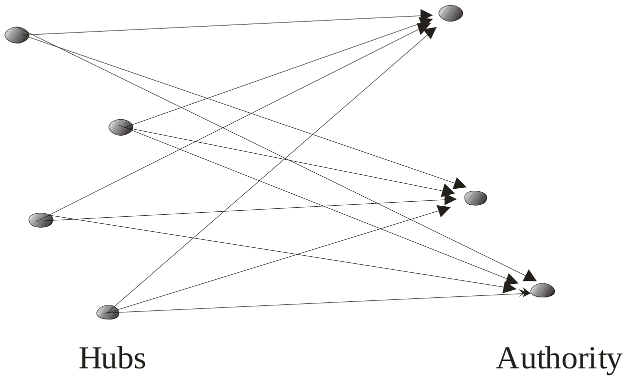

Figure 3 Hubs and Authority

The HITS Algorithm

The basic idea of the HITS algorithm is to identify a small subgraph of the Web and apply link analysis on this subgraph to locate the authorities and hubs for the given query. The subgraph that is chosen depends on the user query.

Identifying the focused subgraph

The focused subgraph is generated by forming a root set R - a random set of pages containing the given query string, and expanding the root set to include the pages that are in the “neighborhood" of R. The algorithm for computing the focused subgraph is as follows:

1. R ← set of t pages that contain the query terms (using the text index).
2. S ← R.
3. for each page p Є R,
   (a) Include all the pages that p points to in S.

(b) Include (up to a maximum d) all pages that point to p in S.
4. The graph induced by S is the focused subgraph.

The algorithm takes as input the query string and two parameters t and d. Parameter t limits the size of the root set, while parameter d limits the number of pages added to the focused subgraph. The expanded set S should be rich in authorities since it is likely that an authority is pointed to by at least some page in the root set. Likewise, a lot of good hubs are also likely to be included in S.

Link Analysis

The link analysis phase of the HITS algorithm uses the mutually reinforcing property to identify the hubs and authorities from the expanded set S. (Note that this phase is oblivious to the query that was used to derive S.) Let the pages in the focused subgraph S be denoted as 1, 2, . . ., n. Let B(i) denote the set of pages that point to page i. Let F(i) denote the set of pages that the page i points to. The link analysis algorithm produces an authority score a_i and a hub score h_i for each page in set S. To begin with, the authority scores and the hub scores are initialized to arbitrary values. The algorithm is an iterative one and it performs two kinds of operations in each step, called I and O. In the I operation, the authority score of each page is updated to the sum of the hub scores of all pages pointing to it. In the O step, the hub score of each page is updated to the sum of authority scores of all pages that it points to. That is,

$$\text{I step:}\quad a_i = \sum_{j \in B(i)} (h_j)$$

$$\text{O step:}\quad h_i = \sum_{j \in F(i)} (a_j)$$

The I and the O steps capture the intuition that a good authority is pointed to by many good hubs and a good hub points to many good authorities. Note incidentally that a page can be, and often is, both a hub and an authority. The HITS algorithm just computes two scores for each page, the hub score and the authority score. The algorithm iteratively repeats the I and O steps, with normalization, until the hub and authority scores converge:

```
1. Initialize $a_i$, $h_i$ ($1 \leq i \leq n$) to arbitrary values.
2. Repeat until convergence,
    (a) Apply the I operation.
    (b) Apply the O operation.
    (c) Normalize $\sum_i (a_j^2) = 1$ and $\sum_i (h_j^2) = 1$
3. End.
```

(vi) **Unsupervised Learning**

The clustering approach in the hypertext context entails issuance of the learner with a set of hypertext documents from the analysis in the link structure module. The learner is then expected to establish a hierarchy according to the documents along the hierarchy. The K-Means is the basic and good method since it collects documents together in or near the leaves of a hierarchy and dissimilar nodes joined near the root of the hierarchy. Within the K-Means, the frequency of k within K-Means Clustering displays the preferred number of cluster of the document. Each of the seed document assigns a cluster in a repeated process until the seed stabilizes and up to a point when no document remains. Thus as shown in Figure 2.0, the terms in the document set D_0 retrieved in response to the original user query Q_0 are clustered for the preparation of the pseudo-feedback process (step 6). The algorithm is as follows:

1. Pick up the term *wmax1* with the highest co-occurrence in the document set D_0 obtained in step 1 of the query expansion algorithm in Figure 2.0. Let D_{01} be a set of documents containing *wmax1*.
2. Pick up the highest co-occurrence term *wmax2* in a set of documents not containing *wmax1*. Let D_{02} be a set of documents containing *wmax2* and not containing *wmax1*.
3. Let D_{03} be a set of the other documents.
4. Terms that appear in D_{01}, D_{02}, and D_{03} would be the clusters of terms *W1*, *W2*, and *W3* in step 6 of Figure 2.0, respectively.

Effectiveness of the intelligent agent topic tracking system

The system uses clustering application which aims at reducing the user effort in locating an exact web document between the several documents returned by the search engines at a common query. The framework displays clear steps, function and steps within the clustering process. The aim of the stages is to ensure delivery of appropriate and precise topic clusters for researcher using search engine crawlers index. The process is quick and precise since it reveals exact information at a convenient time. The intelligent agent topic and development process ease research work and reduces time wastage and unnecessary expenses. It also minimizes environmental pollution in form of noise which can interfere with normal learning of the researchers.

Conclusion

Intelligent agent has been around for years, but the actual implementation is still in its early stages. The definition and analysis of implementations concerning autonomous agents is one of the most complex topics that are faced by network intelligence specialists, with the greatest interest in the topic being the creation of seeming intelligence that is capable of collecting and storing user browsing preferences for future use. As agents gain a wider acceptance and become more sophisticated, they will become a major factor in the future of the Internet. Through a thorough review of the literature that is available on the topic, an analysis of autonomous tracking search agents is created, guiding the manner through search engines retrieval processes and effectively reducing the search engine noise by an acceptable fraction. Intelligent agents will not completely replace surfing altogether, but they will make information gathering much easier for the users or researchers. Instead of searching through lists and lists of unwanted documents, the user could ask their agent to search for a particular document of topic interest, and in a few moments, it come back with the information that is needed immediately with few precise information that will drastically reduce information overload.

References

[1] **Blankson, S. (2008).** *Search Engine Optimization*. London: Blankson Enterprises Ltd.

[2] **Chen, H., Houston, L.A., Sewell, R. R., Bruce L. Schatz, B. L.(1998).** Internet browsing and searching: User evaluations of category maps and concept space techniques. *Journal of the American Society of Information Science*, 49(7):582-608.

[3] **Crestani, F. (2002).** Spoken Query Processing for Interactive Information Retrieval. *Data and* semantic analysis. *Journal of the Society for InformationScience*, 41(6):391-407.

[4] **Croft, B. W., Metzler, D., Strohman, T . (2010).** *Search engines: information retrieval in practice.* Boston: Addison-Wesley.

[5] **Deerwester, S., Dumais, T. S., Landauer, T. K., Furnas, F. W., Harshman, R. (1990).** Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391-407.

[6] **Glasersfeld, E. (2006).** *Database and Data Communication Network Systems*. Cambridge: Cambridge University Press.

[7] **Halavais, A. (2008).** *Search Engine Society*. Michigan: Polity.

[8] **Henning, J. (2009).** Voice of the Customer (VOC) Techniques & Technologies. London: John Wiley & Sons.

[9] **Jansen, B. J., Amanda, S., Taksa, I. (2008).** *Handbook of Research on Web Log Analysis*. Philadelphia: IGI Global snippet**.**

[10] **Jones,** ***B. K. (2010).*** *Search Engine Optimization.* New Jersey: Wiley Publishing Inc.

[11] **Kent, P. (2011).** *Search Engine Optimization For Dummies.* Indianapolis: Wiley Publishing Inc.

[12] **Kleinberg, J. (1999).** Hubs, Authorities, and Communities. In *ACM Computing Surveys*, 31(4).*Knowledge Engineering, 41(1):105-124.*

[13] **Levene, M. (2010).** *An Introduction to Search Engines and Web Navigation*. New Jersey: John Wiley & Sons Inc.

[14] **Meghabghab, G., Kandel, A. (2008).** *Search engines, link analysis, and user's Web behavior*. Berlin: Springer-Verlag Berlin Heidelberg.

[15] **Mooney, R., Bunescu, R. (2005).** Mining Knowledge from Text Using Information *Processing*, 7(1):p.3-10.

[16] **Morrison, J (2007).** *Analysis on Quality Function Deployment*. Upper Saddle: Cengage.

[17] **Notess, G. R. (2006).** *Teaching Web search skills: techniques and strategies of top trainers*. New Jersey: Information Today, Inc.

[18] **Ramos, A., Cota, C. (2008).** *Search Engine Marketing*. New York: Mc GrawHill Companies.

[19] **Robertson, S. E., Walker, S., Hancock-Beaulieu, M. (2000).** Experimentation as a way of life: Okapi at TREC. *Information Processing and Management 36(1):95-108.*

[20] **Savoy, S. (2003).** Cross-language information retrieval: experiments based on CLEF 2000. Corpora. *Information Processing and Management* 39(1).

[21] **Semantic Web project. (2006).** http://www.semanticweb.org/ Accessed: Web 2/10/2011.

[22] **Spink, A., Michael T. Zimmer, M. T. (2008).** *Web search: multidisciplinary perspectives***.** New York: Springer

[23] **Thurow, S. (2008).** *Search engine visibility*, *Part 2.*California: New Riders Publishing.