PSEFiL: A Personalized Search Engine with Filtered Links

Hajar Aghaiipour-Chafuchahi
Department of Software Engineerig, Islamic Azad University, Rasht Branch, Rasht, Iran

Fatemeh Ahmadi-Abkenari
Department of Software Enginnering and Information System, Payame Nour University (PNU), Iran

ABSTRACT
The dynamic nature of the World Wide Web and its growing dimension make retrieving the exact information a difficult task. Inaccurate answers delivered by search engines especially for the query phrases with different meaning makes the feeling of dissatisfaction in today's surfers who needs the specific answer for their information demand. Search engines nowadays tries to understand users' request through studying his/her search background or even make users participate in the search process in order to clarify what he/she really needs. This trend is part of the search engines' endeavors to become personalized.

One of the well-formed personalized search engines is SNAKET that employs the user participation for personalization process. In this paper based on the personalization algorithm of SNAKET, we propose an architecture of our personalized search engine of PSEFiL that filters out the links and delivers the answers to users with low or no content drift as a means of enriching the answer set. Furthermore, the answer set is robust because every existing link in result set either is highly ranked from other search engines or has no or less subject jumps with an exact manual scan process. Also every link is clearly classified to every fetched meaning of a query phrase. One objective of PSEFiL is to give the accurate answers not to populate the answer set with more links that may contain less or no accurate answers.

Keywords
Search engine, Search engine optimization, Search engine personalization, Web Structure Mining.

1. INTRODUCTION
WWW is a vast, diverse and dynamic environment where millions of users publish their documents and search for accurate documents corresponding to their needs. In recent years, efficient methods and techniques for accessing data, sharing data and mining data are on demand intensively. Effective data management and classification approaches are of high importance in order to effective analysis and employment of data for general users as well as the knowledge workers. In the meantime, the nature of the Web including its dynamic and semi structured shape bring many challenges that make it difficult to handle. Furthermore, the difficulty of finding the real data corresponding to the users demands due to the poor analytical precision of search engines, lack of data personalization, long response time for better algorithms and user dissatisfaction of the quality of the received responses are among other obstacles of the search engines functionalities.

Upon issuing a query by the user in a search engine, this Web application looks through its database and fetches the links related to user's written topic. Different methods are employed in order to retrieve data based on the content and structure of Web pages as the source data. Studies show that query words are short and each user has its own specific purpose of the same query. So search engines' result set may contain links that are not related to the users' information demands. Users have different preferences, but the search engine gives the same results for all of them. If search engine could guess user preferences by any means, search engines could deliver more accurate results. In fact, in such an intelligent environment, two users will get different results from the same issued query. One of the most popular discussions in the field of data retrieval is to identify user behavior and the use of behavioral background in order to represent more satisfactory results. In fact, the personalization process of search engines is still a search area in its infancy that attracts many researchers [1], [2], [3], [4].

One of the most useful tools in order to achieve the personalization of search engines is the usage of Web mining. Web mining is the discovery process of unknown information and useful knowledge from Web data as a specialized subcategory of data mining that is applied in a variety of areas. Actually, Web mining is the application of data mining techniques for data stored on the Web.

In the following, we first describe main Web mining processes, methods and tools. Then we will discuss the literature on search engine personalization. In the next step, the description on our conducted experiment will be presented through which upon a close analysis of SKANET search engine and employing its personalization idea, we will suggest a new filtered personalized result set based on a combined Web structure mining and Web content mining techniques.

2. RELATED WORKS

2.1 Web Mining
Web mining is the discovery process of useful and unknown knowledge of Web data and the application of data mining techniques to automatically discover and extract data from Web documents and Web services. In other words the field of Web mining aims to discover hidden knowledge from the Web data through employing data mining techniques. The process of converting data to knowledge is done in four distinct stages that include information retrieval, information extraction, pattern recognition and validation of the extracted patterns that are illustrated in figure 1.

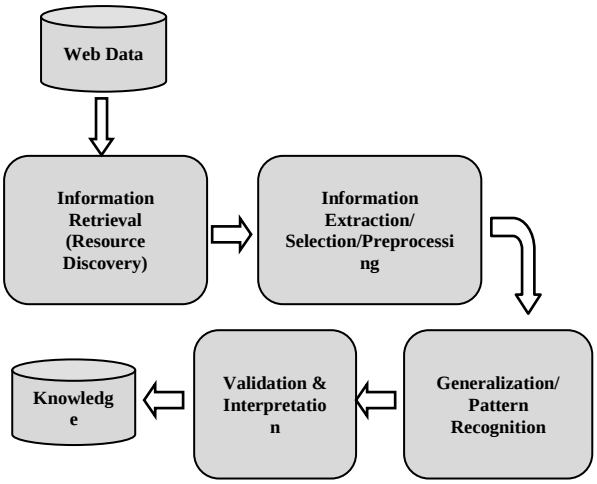

Fig 1: Web mining main processes

In the first step, various Web documents are retrieved. Then, the process of data preprocessing and features selection is required through which the words are reduced to their roots and redundant words are removed. In the third step, the general patterns in one or several Web sites are discovered by the use of data mining and machine learning techniques. The last step is to validate the interpretation and evaluation of the patterns obtained in the previous step. Web mining is applicable in areas such as e-commerce, e-learning, e-banking, digital libraries, knowledge management, e-government, etc. Some of methods and algorithms used in Web mining including decision tree, neural network, nearest neighborhood, maximum flow and average click [9], [10], [13].

Web mining consists of three subsets fields of content mining, structure mining and application mining as shown in Figure 2. Content mining is the process of extraction of text, image, video, audio, or structured records such as lists and tables of Web document contents. In order to structure mining, the Web should be as a graph whose nodes are documents and its edges are the links between the documents. It addresses the extraction of the data from this graph structure according to the links between documents. Web usage mining is to extract meaningful patterns from the data generated in the interactions between the client and server. The remainder of this paper will focus on the Web content mining and Web structure mining [5], [7].

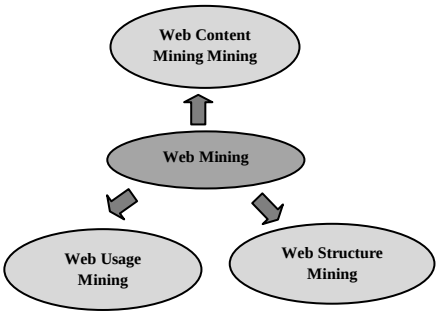

Fig 2: Web mining sub-classifications

2.1.1 Web Content Mining

As described earlier, data in Web content mining includes unstructured data such as free texts, semi-structured texts as HTML pages and structured texts including tables. Web content mining has two perspectives: 1- Data Recovery, which aims to improve the process of searching or filtering the data to the users and 2- Database formation which aims to provide a model of Web data by integrate them. Web content mining includes various activities such as [9], [10], [11], [12], [15]:

- Classification: The classification of documents, finding the appropriate topic category that indicates the main topic(s) of a document with the lowest error rate.
- Clustering: The process of grouping objects into classes of similar objects.
- Prediction and Estimation: For example, the income of people can be estimated according to the payment patterns and their age. In prediction, according to the patterns observed in a newspaper, we can predict the occurrence of future events.
- Determination of dependence and correlations: By using this type of search we can determine which goods to be purchased together.

2.1.2 Web Structure Mining

Exploring the Web structure is the process of extracting structural data from the Web. In order to apply Web structure mining algorithms, the representing models are used which could be 1- A graph-based model that consists of one or more nodes such as a single-node model that includes authority, hub and their combination. 2- The Marco model which is actually a Marco chain of rank M that changing the state of a system depends to the current state and the M-1 previous states. Web structure mining applications includes determining quality associated with a subject, page classification, Web navigation, finding Web communities, adaptive Web sites design and personalization [9], [10], [11], [12].

2.2 Search Engines Personalization

Search engines are programs that search the issued and converted keywords in an offline constructed databases. Search engine is composed of three parts of crawler, indexer and responder.

Different methods mainly based on content and structure mining are used to retrieve data. Studies have shown that query words are short and different and each user has its own specific purpose from a query. In fact, the presented results always do not correspond to the user demands. Users have different preferences from a same query but they all are provided with the same result set by the search engine. If user preferences can be used in search, certainly more satisfactory results will be presented to them. In a concrete definition, results personalization demonstrates appropriateness of the search engine results with user interest, knowledge and need. Personalization plays an important role in providing quick access and the required data to the users according to their interests. Personalization is performed in two processes, user modeling and the implementation of personalization system. Some researchers also classified the personalization process in three steps of user identification, user modeling and implementation of personalization system [4], [6], [8], [14].

SNAKET stands for Snippet Aggregation for Knowledge Extraction is one of the first complete and open source personalized search engines that offers hierarchical clustering and folder labeling with variable-length sentences close to Vivisimo, with a creative form of personalized ranking, privacy protection and scalability. SNAKET consists of a three phased algorithm: 1- Sentence selection and ranking 2-

hierarchical clustering and labeling 3- personalized rankings [6].

SNAKET extracts from enriched snippets, all pairs of words that occur in some fixed neighborhood windows. So these pairs using the engine based on DMOZ. The low ranking word pairs are disposed. The remaining pairs gradually merge together to form longer sentences. SNAKET neither use suffix tree nor suffix array to merge pair of words, because words of sentences may not be considered continuous in snippets. Therefore, its approach is based on a combination of inverted lists and bitmaps which is made quickly throughout the snippets. The longer sentence is maintained while the rank of this sentence is a function of the ranks of pairs of adjacent words and is calculated by the ranking engine, DMOZ. Lower ranked sentences are discarded and the process is repeated until there is no possible integration or the sentences based on the adjustable 8-words. All of the sentences that are discarded in the whole process provide candidate labels to annotate leaves of a folder hierarchy [6].

SNAKET uses a bottom-top hierarchical clustering algorithm whose aim is to create a folder hierarchy that is compact in terms of total number of folders. A snippet can cover multiple topics. The purpose of SNAKET also is allocation of folder labels according to the accurate snippet codes. Snippets are grouped according to the indented sentences (candidate) which have been shared. These folders provide leaves of our hierarchical, and their labels provide their annotations (called primary labels). We assume that snippet codes that share equal indented sentence deal with the same issue and therefore should be placed in the same folder [6].

Hierarchy making process is done in a bottom-top manner and consists of three main steps: parent formation, ranking and pruning. A parent folder *P*, for each group, *C1*, *C2*,. . . , *CJ* are created. The common sentence provides the primary *P* label. Secondary set of labels is formed by secondary labels from *Ci* which occur at least *c%* of *P* snippet codes. *Sig (P)* is obtained with the incorporation of *P* and *P* secondary labels [6].

After setting up all the parent folders and their labels, SNAKET ranks them by applying the folder labels rank of their children. In fact, the rank *P* is calculated from *Ci* labels rank. Then, SNAKET creates a weighted graph *G* in which the vertices set are presented by the parent folders and their children folders, edges indicate by parent-child mediator and the weight of a vertex is the corresponding folder rank [6].

The diagram is applied on next pruning step where the purpose is to erase the current level of folder hierarchy to match with stated targets. SNAKET takes two different pruning rules to skip some parent folders that are already formed. If two parent folders cover the same children folders, then SNAKET keeps parent folder which has the highest rank. If two parent folders are described with the same labeling words, then SNAKET holds the parent folder with highest rank. The second law keeps conciseness, accuracy and distinction of the label. Terms of the weighted graph are applied by a greedy method. Number of the folders that are set aside could not be negligible since eliminating them would lead to a general comprehensible and more compacted hierarchy. After pruning, the remaining parent folders provid the next level and the bottom-up process repeats. This process stops after three levels were made. Since the user will not consider a deeper hierarchy [6].

Link-based ranking methods tend to produce results to the most popular query. For example the search of the query of "Jaguar" on Google doesn't contain the related answer to the Mayan civilization in the first ten results. In contrast, SNAKET is able to capture some key concepts of Web snippet codes and by employing the user role in order to personalize the results from the major search engines. SNAKET applies the labeled folders hierarchy for modifying the query, clarification and knowledge extraction as sub-detail (Module of personalization engine in Figure 4) [6].

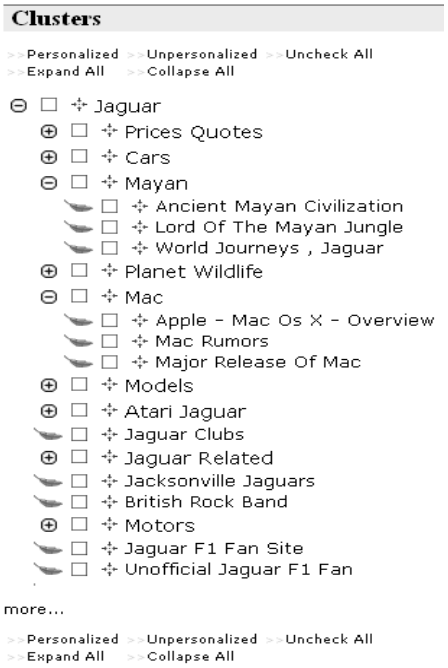

Fig 3: Knowledge extraction for "jaguar" [6]

Users can navigate the hierarchy to extract data. This extension is inexpensive or time consuming since it is done on the client side. This navigation can be seen as a form of knowledge extraction process which allows the user to specify his point of view in the form of 200 or more query results without seeing all of them. Figure 3 shows an example for the query of "Jaguar" where a user selects from the folder labels created for the this phrase. The word of "Jaguar" refers to an animal, a car, Mayan civilization, the rock bands and … . When the user looks at the folder hierarchy, he can decide to modify the query slightly. The users can select a set of L=$\{l_1, \ldots, l_f\}$ labels and request the snippet codes from SNAKET to filter the ranked list.

SNAKET personalization is fully compatible, scalable, and non-intrusive to the user. It is fully compatible and scalable, because it is not based on the profiles and users can adapt their choice based on personalization they do themselves. SNAKET also protect user privacy because it doesn't need explicit and clear log in, the user profile pre-formulation and tracking past searches. Note that the user can change the selected labels several times and thereby change his specified personalized results. Filtering the 200 (or more) returned snippet codes is done by the search engines. Briefly, SNAKET is a plug-in that can convert any impersonal meta-search engine to a personal meta-search engine.

We have noted that SNAKET provides a lightweight Web-client interface that does not need to maintain any data on the server. Expanding folders, browsing and personalization is scalable because they occur on the client side. In contrast,

VIVISIMO requires a connection between client and server to modify each folder. Figure 4 shows the architecture of SNAKET.

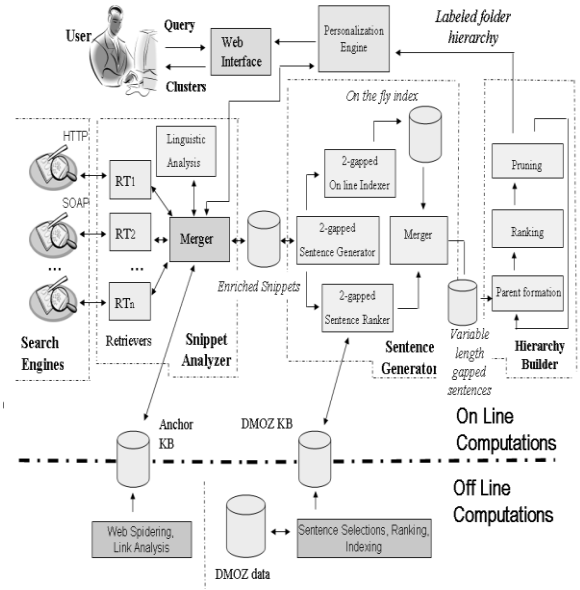

Fig 4: SNAKET Architecture [6]

3. EXPERIMENTAL RESULTS

The conducted experiment includes searching for different queries across the Google search engine. The queries have been chosen in two groups. The first group includes phrases or terms with different meaning such as "Java", "Beetle", "Puma", "Platform", "Jaguar" and the second group are phrases with multiple sub categories such as "Web Mining", "Operating System", "Neural Network", "Computer Architecture" and "Data Base". For example the user could expect search engine to deliver pages for the query of "Java" related to a programming language or pages related to an island. As an example from the second group, the user may expect the search engine to deliver links for the query of "Data Base" as a course, as a concept, as a technology or as commercial tools.

The download has been performed during November and December 2014. Issuing these queries to *Google* consist downloading the 300 pages for each queries. After manually omitting the ppt, pps, pptx, ppsx, pdf around fifty pages remains as clean links with out-links. Figure 5 illustrates the architecture of our PSEFiL stands for Personalized Search Engine with Filtered Links. To observe the quality of extracted links, we investigated out-links in terms of the number of broken links, related out-links and unrelated links with content drift or subject jumps. Then, the ratio of the number of related out-links to the sum of broken, related and unrelated links was calculated and the links with the value greater than zero were considered as nominated links to be included in the lower part of the answer set. This job is the responsibility of *Link Filtering* component as shown in figure 5.

In another parallel test, the queries have been issued to different search engines of *Google*, *Ask*, *Bing*, *Excite*, *Dogpile*. From each search engine, the first 20 links along with their snippet have been observed in order to categorize the totally 100 links into subcategories of each subjects. For this purpose we utilize the snippet analysis along with page content check for accurate categorization. This performance is the responsibility of the *Sub Category Maker* component that is shown in figure 5. This component forms the tree representation that the user must click on the desired sub category. Then each related Web page to each sub category along with its snippet is leaded into the database that is constructed in meta approach since the result is combined from the outputs of different search engines (*Database Loading* component in figure 5). Table 1 shows a part data for the link filtering for the query of "Java", table 2 for query of "Operating System", table 3 is for the query of "Web Mining" and table 4 for the query of "Beetle".

Table 1. An extract of the link filtering data for the query of "Java"

Relates/Sum	External			
	sum	Broken	Unrelated	Related
0.533	15	4	3	8
0.466	15	0	8	7
0.333	3	1	1	1
0	4	3	1	0
0	5	5	0	0
0.923	105	3	5	97
1	1	0	0	1
0	11	11	0	0
0	13	8	5	0
0.888	54	4	2	48
0.4	5	1	2	2
0.428	7	3	1	3
0.058	17	16	0	1

As shown in figure 5, the personalization of the search engine is dependent to the user participation. For this reason, after a user issues a query, the search engine shows the offline constructed sub category of that query in a tree form and waits for user click on one of the labels as shown in figure 6 in which the associated number represents the number of links in each category. For example if user issues the query of "Beetle", the search engine shows a tree with the labels "Insect", "Music Band", … . After user clicks on one label, the pages related to only that sub category will be represented along with the page snippet to the user. Figure 7, 8 shows the upper part of PSEFil result set for the sub category of "Course" and "Concept" of the category of "Operating System" respectively.

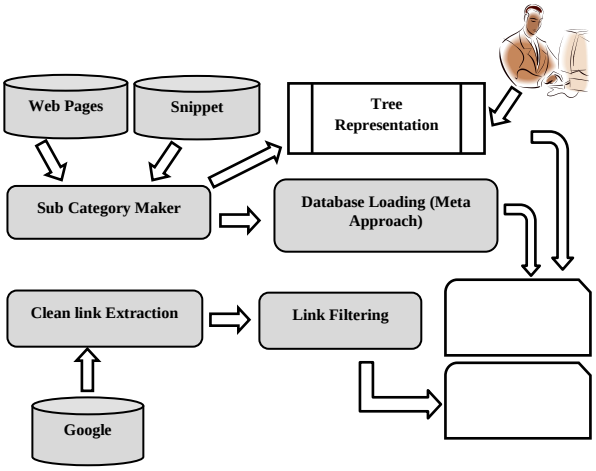

Fig 5: Architecture of PSEFiL search engine

Table 2. An extract of the link filtering data for the query of "Operating System"

Relates/Sum	External			
	sum	Broken	Unrelated	Related
0.593	32	3	10	19
0	6	6	0	0
0	11	10	1	0
0	26	20	6	0
0.125	16	11	3	2
0	0	0	0	0
0.030	66	1	63	2
1	1	0	0	1
0	8	4	4	0
0.333	6	0	4	2
0.8	5	0	1	4
0.15	20	7	10	3
0.142	7	5	1	1
0.666	3	0	1	2
0.1	10	9	0	1
0.666	3	0	1	2
0.519	52	8	17	27
0.090	11	8	2	1
0	4	3	1	0

The upper part of the answer set will be formed from the categorized link based on the answers gathered offline from different search engine (since they are highly ranked from different perspectives that these search engine utilize) and the lower part will be populated with filtered links with no or less content drift from the *Google* (since they are valuable links) with their associated snippet.

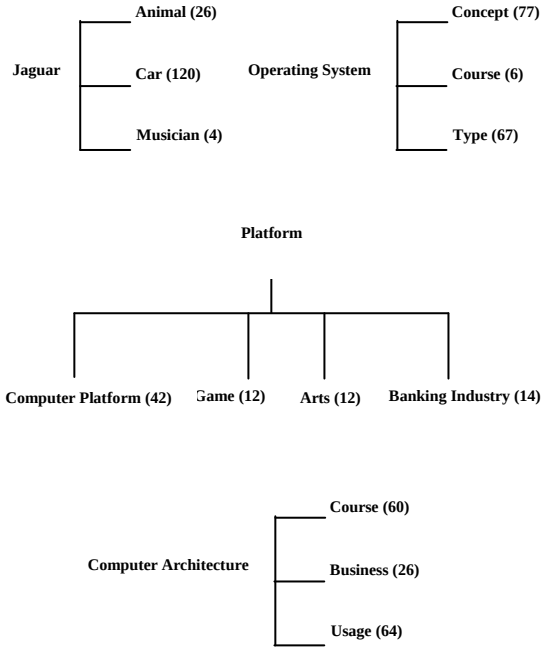

Fig 6: Tree representation of queries for user interference

Table 3. An extract of the link filtering data for the query of "Web Mining"

Relates/Sum	External			
	sum	Broken	Unrelated	Related
0. 666	18	5	1	12
0. 666	3	0	1	2
0	3	0	3	0
1	1	0	0	1
0.125	8	3	4	1
0	13	4	9	0
0.6	5	2	0	3
0	2	0	2	0
0	0	0	0	0
0	10	0	10	0

0.466	15	3	5	7
0.125	24	6	15	3
0	3	3	0	0
0.222	18	5	9	4
0.333	3	0	2	1
0.090	11	2	8	1

Table 4. An extract of the link filtering data for the query of "Beetle"

Relates/Sum	External			
	sum	Broken	Unrelated	Related
0.25	4	2	1	1
0	0	0	0	0
0.166	12	3	7	2
0.545	11	2	3	6
0.066	30	15	13	2
0.125	8	3	4	1
0	2	2	0	0
0	12	12	0	0
0.25	4	2	1	1
0.083	12	4	7	1
0	0	0	0	0
0	19	19	0	0
0	5	3	2	0
0	6	4	2	0
0	14	13	1	0
0	11	3	8	0
0	40	6	34	0

1. 23.672 - Legal Information Institute - Cornell University
www.law.cornell.edu/cfr/text/14/23.672
(b) The design of the stability augmentation system or of any other automatic or power-operated system must permit initial counteraction of failures without ...

2**. Computer Science - Building an Operating System for CS**
outlier.uchicago.edu/computerscience/OS4CS/ The "Building an Operating System for *Computer Science*" (*OS4CS*) study was ... Chicago, IL: *CEMSE*, *University of Chicago* with UEI, *University of Chicago*.

3. Computer Laboratory – Raspberry Pi: Baking Pi – Operating *www.cl.cam.ac.uk/projec*... Computer Laboratory, University of Cambridge It may be helpful to have some *programming* experience, but the course ... full details of creating an *Operating System*, these *tutorials* focus on achieving a few ...

4. Operating Systems - San Jose State University ***www.sjsu.edu/faculty/watkins/opsys.htm*** San Jose State University *SAN JOSÉ STATE* UNIVERSITY ECONOMICS DEPARTMENT Thayer *Watkins*. Nature and History of. Operating Systems for Computers. An operating system is ...

5. CS2510 | Department of Computer Science | University of
https://www.**cs.pitt.edu/grad/courses/cs2510.php**
University of Pittsburgh *Pitt* Home Dietrich School Apply Contact Us ... Summer 2014 · Fall 2014 · Spring 2015 · *Course* Descriptions; *Graduate* Degrees in Computer Science: ... *CS 2510*: Computer Operating Systems ... Suggested *Course*. *CS* 1550 or its equivalent.

Fig 7: An extract of PSEFil result set for the sub category of "Course" from "Operating System"

1. Operating system - Wikipedia, the free encyclopedia
An *operating system* (OS) is software that manages computer hardware and software resources and provides common services for computer programs.

2. What is Operating System (OS)? Webopedia
The *operating system* (OS) is the most important program that runs on a computer. Computers and mobile devices must have an *operating system* to run ...

3. Computer Basics: Understanding Operating Systems
www.gcflearnfree.org › ... › Computer Basics An *operating system* is the most important software that runs on a computer. It manages the computer's memory, processes, and all of its software and hardware.

4. Monitoring Systems Operated Remotelyelectrical .about.com /b/2011 /09 /24 /monitoring-systems-operated-remotely.htm How would you like to have a security system and temperature control system that can be operated from anywhere you can connect to the internet or somewhere you can make a phone call. That's right, there is such a device and it does the thinking and actions for you, no ... More »

Fig 8: An extract of PSEFil result set for the sub category of "Concept" from "Operating System"

4. CONCLUSION

Due to the rapid development rate and significant increase in the volume of Web data, the brand new algorithms are on demand in order to efficient and organized access to data. Traditional methods for data retrieval could not freely used in Web due to the semi-structured nature of the Web. One of the main obstacles in the search engine application job is their blind view to the users' query and simply answering them with the links that many of them are completely unrelated to user information demand. This unrelated links are mostly associated to other meaning of the query not with the one user exactly expect. In this paper, by utilizing SNAKET's approach toward making offline sub-categories for each word in order to make users to participate in search process, we filtered the result set with the most related links with less or no content drift. Other part of the result set will be populated in a meta trend with the answers received from other search engines. Our future work consists of developing new algorithms and approach for better sub categorization of each word or phrase and a more robust link filtering technique that leads to no links with content drift in answer set. So for future work we continue our research through focusing on the categorization algorithms and also developing robust approaches to identify and prevent content drifted links in search result set.

5. REFERENCES

[1] Ahmadi-Abkenari, F., Selamat, A. 2012. "An Architecture for a Focused Trend Parallel Web Crawler with the Application of Clickstream Analysis", International Journal of Information Sciences, Elsevier, Vol. 184, pp. 266-281.

[2] Ahmadi-Abkenari, F., and Selamat, A. 2013. "Advantages of Employing LogRank Web Page Importance Metric in Domain Specific Web Search Engines". JDCTA: International Journal of Digital Content Technology and its Applications. Vol. 7, No. 9. pp. 425-432.

[3] Ahmadi-Abkenari, F., and Selamat, A. 2012. "LogRank: A Clickstream-based Web Page Importance Metric for Web Crawlers". JDCTA: International Journal of Digital Content Technology and its Applications. Vol. 6, No.1. pp. 200-207.

[4] Alhalabi W., Kubat M. and Tapia M. 2006. "Search Engine Personalization Tool Using Linear Vector Algorithm". Proceedings of the 4th Saudi Technical Conference and Exhibition. pp. 336-344.

[5] Baeza R. and Boldi, P. 2010. "Advanced Techniques in Web Intelligence". Studies in Computational Intelligence. Vol. 311. pp. 113-142.

[6] Ferragina P. and Gulli A. 2005. "A Personalized Search Engine Based on Web Snippet Hierarchical Clustering". In proceedings of the World Wide Web Conference, Japan. pp. 801-810.

[7] Husin H.S, Thom J.A and Zhang X. 2013. "News Recommendation Based on Web Usage and Web Content Mining". Data Engineering Workshops (ICDEW), IEEE 29th. pp. 326 – 329.

[8] Kim K. J., Cho S.B. 2005. "Personalized mining of Web documents using link structures and fuzzy concept networks". Applied Soft Computing 7. Elsevier. pp. 398–410.

[9] Kumar S. and Devi N. 2010 "Learner's Centric Approach for Web Mining". International Journal of Computer Science and Information Technologies (IJCSIT). Vol. 1(2).

[10] Liu B., Mobasher, B. and Nasraoui O. 2011. "Web Data Mining Data-Centric Systems and Applications". pp. 527-603.

[11] Markov, L. 2007. "Data Mining the Web". Wiley Publication. Chapter 6, 7, 8. pp. 143-188.

[12] Nyein, S.S. 2011. "Mining Contents in Web page using Cosine Similarity". Computer Research and Development (ICCRD). IEEE. Vol.2. pp. 472 – 475.

[13] Sharma K., Shrivastava G. and Kumar V. 2011. "Web Mining: Today and Tomorrow". The 3rd International Conference of Electronics Computer Technology (ICECT), Vol. 1. pp. 399 – 403.

[14] Souldatos S., Dalamagas T. and Sellis T. 2006. "Captain Nemo: A Meta-Search Engine with Personalized Hierarchical Search Space", INFORMATICA. Vol. 30. pp.173-182.

[15] Srikantaiah K.C., Suraj M., Venugopal K.R., Iyengar S.S. and Patnaik L. M. 2012. "Similarity Based Web Data Extraction and Integration System for Web Content Mining". Advances in Communication and Computing. Springer. Vol. 108. pp. 269-274.