

WSP model on Users personalized web search & Data Security

Gurram Shireesha¹ & Dr. Shaik Abdul Muzeer²

M-Tech Dept. of CSE Megha Institute of Engineering & Technology for Women
Professor & principal Dept. of CSE Megha Institute of Engineering & Technology for Women

ABSTRACT-

Today's generic search engines are using search as you type technique while searching the information in order to retrieve information faster as compared to keyword search. But generic search engine does not distinguish different users need. It shows same result to different types of users without bothering their needs. Most of the personalized search engines retrieve the specific information but uses keyword search technique. This paper proposes personalized web search using domain knowledge with search as you type technique in order to retrieve specific information and compare the results with keyword search.

Keywords: - personalized web search; search as you type; domain knowledge; user profile

1. INTRODUCTION

With the growth of World Wide Web, web search engines have contributed a lot in finding information from the web. They help in searching information on the web robust and easy. But there is still room for improvement. Current web search engines do not consider particular needs of user and serve each user equally. Archetypal search engines are following the "one size fits all" model which is not suitable to individual users. It is difficult to let the search engine know what we the user actually wants. When different users give same query, same result will be returned by atypical search engine, no matter which user submitted the query. This might not be purloining for users which require different information. While searching for the information from the web, users need information based on their interest. For the same keyword two users might require different piece of data. This fact can be explained as follows: a biologist and a programmer may need information on "virus" but their fields are entirely different. Biologist is searching for the

"virus" that is a micro-organism and programmer is searching for the malicious software. For this type of query, a number of documents on distinct topics are returned by Archetypal search engines. Hence it becomes difficult for the user to get the pertinent content. Moreover it is also timeconsuming. Personalized web search is considered as a promising solution to handle these problems, since different search results can be provided depending upon the choice and information needs of users. It exploits user information and search context to learning in which sense a query refer. In order to perform Personalized Web Search it is important to model User's need/interest. User profiles are constructed to model user's need based on his/her web usage data. This Enhanced User Profile will help the user to retrieve concentrated information. This paper proposes architecture for constructing user profile and enhances the user profile using background knowledge. It can be used for suggesting good Web pages to the user based on his search query and background knowledge. Demolition of user profile is an important part for



personalized web search. The paper is organized as follows: Section 2, gives there late work focusing on personalized search systems. Section 3, proposes the framework for personalized web search that satisfies each user's information need by enhancing the user's profile without user's strain. Next section, we presents the experimental results for evaluating our proposed approaches. Finally, we conclude the paper with a summary and directions for future work in following sections.

2. RELATED WORK

Previous related work can be grouped into three categories: (1) general search personalization efforts; (2) query log analysis with the goal of long-term user modeling; (3) investigations of user expertise and content readability.

2.1 Search personalization

A growing number of data sources, such as search history, manually or automatically created preference profiles, and social network information, are being exploited for personalizing the selection of results for users. Approaches to search personalization vary in types of features considered (e.g., language models, topical categories, links or other metadata such as reading level), the time frame chosen (e.g., short-term or long-term profiles), and how the profiles are used (e.g., for ranking or recommendations). Several researchers have shown how profiles that consist of topical representations of users' search interests can be used to personalize search. Gauch and colleagues learned topical user profiles based on browsing history [11] or search history, and Ma et al. used topical profiles that users specified explicitly. In all cases, user profiles were compared with those of search results and used to re-order search results for individuals. Bennett et al. [5] have recently shown how topics from the Open Directory Project's (ODP) topology of the

Web1 could be used to personalize search ranking for individuals. Expanding on the Page Rank algorithm, Haveliwala proposed a topic-sensitive modification to allow for a direct, more focused scoring of the Web graph given a query's topic or a user's topical preference. Queries and Web content were automatically categorized using the ODP hierarchy in order to facilitate topic-sensitive scoring. Sugiyama et al. employed collaborative filtering on users' observed Web search histories for profile building. They compared their approach to exclusively using browsing history and implicit feedback mechanisms, finding significant merit in the use of profile expansion via their proposed method. Teevan et al. investigated the potential of re-ranking the top 50 search engine results based on previous user profiles. In particular, the authors explored alternative document representations for search personalization, finding that full-text representations outperformed priority-driven selective keyword models. Li et al. developed a dynamic graph-based adaptation scheme modeling a user's general preferences for search personalization, while accounting for changes of interest by incorporating short-term browsing information. In a recent study, Goel et al. [12] analyzed the U.S. market's large-scale consumption of movies, music, and Web search results in order to quantify the importance of the 'long tail' of items in those respective categories of popular media. The authors found that the majority of users largely displayed standard tastes in most categories but showed some degree of eccentricity in choices. In this work, we will investigate a related notion, namely, that of atypical search sessions: cases in which users occasionally stray from their personal previous mainstream.



2.2 Long-term user modeling

A special subclass of research on user modeling and search personalization is based on long-term profiles rather than focusing only on the user's immediate history. While being noisier than short-term profiles, this approach has the advantage of being able to detect niche interests or those that surface in long cycles. Matthijs and Radlinski captured users' 3-month Web history across multiple search engines and sites via a browser plug-in. The full resulting log files were used for result re-ranking and showed significant performance improvements over the native ranking of popular search engines. Furthermore, long-term user profiles served as reliable general descriptors of a user's interests. Tan et al. presented a language modeling approach that interpolated immediate search history and long-term user profiles in order to improve retrieval performance. They found that short-term profiles contained more useful clues as to the current query's intent, but that adequately-weighted long-term information introduced further performance gains. White et al. investigated the usefulness of short-, mid-, and long term profiles for the task of predicting user interest in Web sites. The authors demonstrated that, depending on the type of information being profiled as well as the type of information need, different profiling durations could be optimal. Finally, Bennett et al. [5] showed how long- and short-term profiles could be optimally combined for effective search personalization. They found that long-term models provided the most benefit at the beginning of a session, while short-term models became more important for longer sessions.

2.3 Expertise

Studies of Web search often distinguish between two types of expertise - search expertise

(reflecting knowledge of the search process) and domain expertise (reflecting knowledge of the domain or topic of the information need). In one of the early comparisons of Internet information search behavior and success, Holster and Strube examined both search and domain expertise. They reported that search experts displayed a richer set of skills, such as selection of tools, query formulation and relevance judgment than novice searchers. Also, experts were found to navigate search interfaces more efficiently. Beyond search skills, Thatcher showed that experts and non-experts followed different strategies to obtain search results, depending on the task. White and Morris [3] conducted a large-scale log analysis of the differences in search behaviors and success of search experts and novices. The authors found that experts generated different types of queries, had shorter and less branchy post-search browse trails, and were generally more successful than novices. More recent work has tried to model strategies of successful searchers. Ageev et al. [1] exploited this expertise-dependent difference in search behavior by using a Markov chain approach to predict search success for a range of pre-defined search tasks based on the sequence of actions the searcher had undertaken in the session. One of their main findings was that searchers who are more successful are generally more active (e.g., more queries issued and results clicked) in a given time window. Aula et al. [3] analyzed different characteristics of successful and unsuccessful search sessions. Based on a small qualitative lab study and a subsequent large-scale evaluation, they established a range of indicators for user frustration during search sessions that were not yielding the desired results. Most saliently, the authors report longer sessions, question-type queries, the use of advanced query operators and aimless scrolling on the results page



for failing searches. We will revisit these findings in Section 4 to employ them for identifying atypical sessions. Beyond the effect of general search expertise on success rates, other recent work has considered the searcher's familiarity with the search topic. Based on a large-scale query log analysis, White et al. [11] found significant differences between the search behavior of domain experts and non experts within the domain of their expertise (but not outside of the domain). The authors found that domain experts generated longer queries with more technical terms, had longer search sessions with more branches, and had greater success in satisfying their information needs than novices. Collins-Thompson et al. [6] investigated the use of reading level metadata for search personalization, finding that search ranking could be improved by taking into consideration the user's previous reading level preferences as well as the reading level coherence between a Web page and its result snippet. Kim et al. [8] followed up in this direction by jointly modeling reading level and topic preferences to describe users. Their so-called RLT profiles were used to distinguish domain experts from non-experts as well as to identify occurrences of 'stretch' reading behavior, i.e., when users go beyond their usual preferences to satisfy information needs. Their work is central to ours as it observes users temporarily diverging from their profiles to solve particular tasks. In this work, we focus on the in-depth analysis and support of such cases. In a similar effort, Tan et al. [9] exploit notions of reading level and text comprehensibility for ranking popular answers on the Web portal Yahoo! Answers. According to the searcher's degree of domain expertise, simple vs. more technical answers were ranked higher. The research we present in this paper extends previous

results by: characterizing the extent to which searchers diverge from their long-term search profiles, and demonstrating how the ability to detect such atypical sessions can be used to improve search personalization.

3 PROPOSED SYSTEM

In proposed system we are improving the quality of search engine by suggesting some relevant pages of user interest. Here we consider the users profile to suggest the results. The history and the domain knowledge of the user navigation are used to store the categories information. The query submitted by user is send to the query optimizer and history of the system. Then it is use with previous user profile to improve it. The domain knowledge is also use with them to produce new enhanced user profile. Then query and user profile is send to the search engine. The output of search engine is re-ranked with the help of user profile and domain knowledge. Then the final improved search result and the suggestions are return to the user.

4. Experimental Results

The main objective of the Web Search Personalizer (WSP) model is to retrieve the best search results that meet the user's preferences using his up-to-date user profile, which is being built and updated regularly. Fig. 1 shows the conceptual view of the WSP model and its interactions with the external entities.

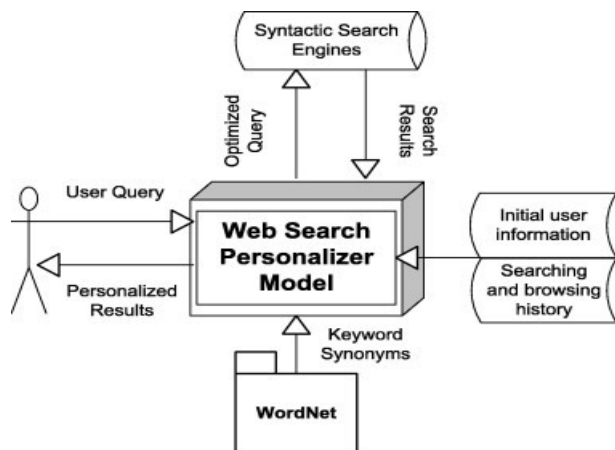


Figure 1. The WSP model conceptual view.

As shown in Fig. 1, the user interacts with the WSP model by entering a user query, which is then semantically optimized to produce an optimized query. The query is optimized based on the user's profile preferences and the query-related synonyms from the Word Net ontology. After that, the optimized query is sent to a set of syntactic search engines for retrieving the related search results, which are then defused to produce the final personalized results. Finally, WSP model extracts the user feedback implicitly by the click-through technique to update the user profile.

In order to build and update the user profile, WSP model interacts with the user to gather the static user profile part and interacts with published resources to gather the dynamic user profile part. The static user profile part represents the basic user information such as username, birth date, location, hobbies and other personal information. On the other hand, the dynamic user profile part represents the published searching and browsing history data, which are obtained automatically and updated periodically to keep the user profile up-to-date.

4.1 WSP model architecture

The Web Search Personalizer (WSP) model architecture employs the multi-agent system technique that can be easily extended and

maintained. It also enables asynchronous communication among agents, which means that they can work in parallel without interrupting or delaying one another. As shown in Fig. 2, WSP consists of three agents: Interface Agent, User Profiler Agent and Meta-Search Agent.

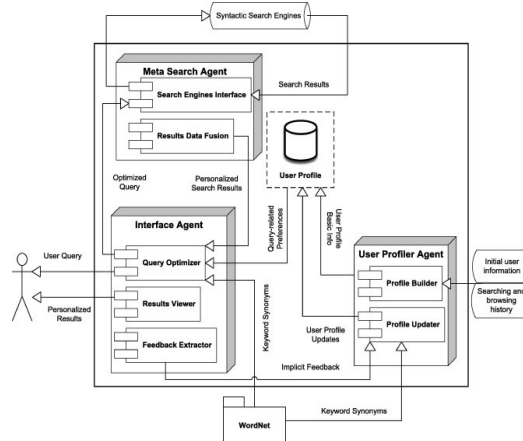


Figure 2. The WSP model architecture.

To generate the final personalized search results, the three agents are interacting together through well-defined interfaces by exploiting one of the standard Agent Communication Languages such as the Knowledge Query Manipulation Language (KQML) [12]. The user interacts with the Interface Agent by providing the search query and retrieving the personalized search results. In order to optimize the user query semantically to be sent to the Meta-Search agent, the Interface agent accesses the user profile to retrieve the user query-related preferences and the WordNet to retrieve the related synonyms. Once the Meta-Search agent receives the optimized query, it interacts with a set of syntactic search engines such as Google, Bing and Yahoo to retrieve and defuse the search results, and then it sends them back to the Interface agent. The final personalized search results are displayed to user by the Interface agent, which senses the user clicks to extract the user feedback implicitly. Finally, the Interface agent sends the implicit feedback to the

User Profiler agent to update the user profile. The following sections illustrate the internal components of each agent and show how they interact together.

4.2. The interface agent

The Interface Agent is a coordinator agent between the user and the other two agents. It is responsible for interacting with the user to acquire the user query such as “Ferrari Cars” or “Mango Trees”. Also, it is responsible for displaying the final search results to the user, and then extracting the implicit user feedback. Therefore, the Interface Agent has three main components: Query Optimizer, Results Viewer, and Feedback Extractor. The Query Optimizer is responsible for semantic query optimization based on the query context domain, the weight of each query term stem within the user profile (query-related preferences) and the WordNet synonyms for each query term stem.

For example, suppose that there are a user U_i who has a profile P_i (set of preferences), a search query $Q(U_i)$, which is a set of term stems $\{T_0, T_1, T_2, \dots, T_n\}$, and the query context domain $D(Q(U_i))$. As shown in formula (1), the corresponding optimized query $Q_e(U_i)$ is the union of the user query $Q(U_i)$, the query-related preferences PP_i , the query context domain $D(Q(U_i))$, and the WordNet synonyms of the query term stems $WS(Q(U_i))$.

equation(1)

$$Q_e(U_i) = \{Q(U_i) \cup PP_i \cup \{D(Q(U_i))\} \cup WS(Q(U_i))\}$$

Fig. 3 shows the proposed algorithm exploited in the query optimization process. The following steps illustrate how the algorithm works.

- Step 1: The Query Optimizer processes the user query to get the stem of each term within the user query using the Porter’s algorithm.
- Step 2: After that, the query optimizer accesses the WordNet ontology to retrieve the context domain of the query based on the query stems by navigating the “Hypernym” relationship in WordNet. The query context domain is identified by the most common domain among the query stems.
- Step 3: In case we have multiple context domains, the domain with the highest priority in the user profile (based on their weights) is selected.
- Step 4: According to the query context domain, the user profile is accessed to retrieve the related preferences.
- Step 5: For each query stem, its synonyms are retrieved from the WordNet ontology.
- Step 6: Finally, the Query Optimizer concatenates all of these items (query context domain, query stems, preferences and query stem synonyms) to generate the optimized query, which is then sent to the Meta-Search Agent.

```

Input: Q(Ui) = {T0, T1, ..., Tn}
Output: Qe(Ui)
Steps:
For Each Ti In Q(Ui)
    Tst = GetStem(Ti)
    D(Tst) = GetWordNetDomain(Tst)
    If (IsNotExist(D(Tst), DC))
        Add(DC, D(Tst))
    Else
        IncreaseWeight(D(Tst))
    End If
End For
HDC = GetHighestWeightDomains(DC)
If (LengthOf(HDC) > 1)
    For Each HDCj In HDC
        Wj = GetProfileDomainWeight(HDCj)
        If Wj > Wj+1
            D(Q(Ui)) = HDCj
        End If
    End For
Else
    D(Q(Ui)) = HDC1
End If
PPi = GetRelatedTermsFromUserProfile(Q(Ui))
WS(Q(Ui)) = GetSynonyms(Q(Ui))
D(Q(Ui)) = GetNearestSuperClass(D(Q(Ui)), PPi, WS(Q(Ui)))
Qe(Ui) = Concatenate(Q(Ui), PPi, D(Q(Ui)), WS(Q(Ui)))
    
```

Figure 3. The query optimization algorithm. After the personalized search results list comes from the Meta-Search Agent, they are passed to the Results Viewer component that displays them to the user. During viewing the results, the user

may do some actions on the same document such as clicking one of the displayed links, bookmarking some page, and copying some text. These actions are sensed by the Feedback Extractor component, which translates them into a set of pairs $\{(Document D, \{Action A\})\}$, and then sends it to the Profiler Agent.

4.3. The user profiler agent

User Profiler Agent is a reactive agent that is responsible for creating and updating the user profile, which is represented in a tree data structure of domains and preference stems. Fig. 4 shows a profile sample for user N.

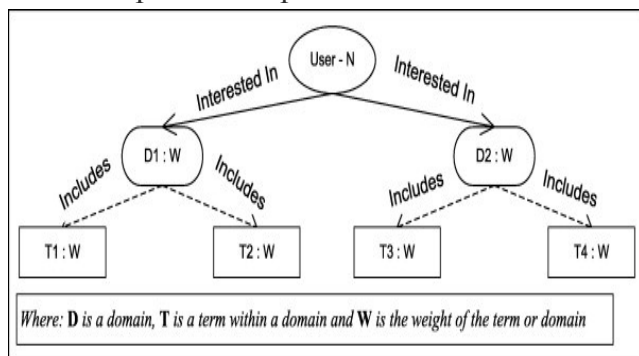


Figure 4. User profile representation.

The user profile tree root is a dummy node, which represents the user identifier. The tree root child nodes are the most abstract domains (Ds). Each domain has its own weight (W) that is calculated based on the child preferences (Ts). Each domain may contain other sub-domains and/or set of preferences. These preferences are semantically associated with their parent domain based on the WordNet ontology. For example, the user profile P_i for user U_i contains set of term stems $\{T_{i0}, T_{i1}, \dots, T_{ij}\}$ and domains $\{D_{i0}, D_{i1}, \dots, D_{ij}\}$ and each term stem has a weight $W(T_{ij})$, so P_i can be represented as a set of triplets: $\{(D_{i0}, T_{i0}, W(T_{i0})), (D_{i1}, T_{i1}, W(T_{i1})), \dots\}$. The weight of each parent domain $W(D_{ij})$ is calculated based on the average of its direct child node weights as shown in formula (2), where $W(N)$ is

the weight of each direct child node N (Term stem or Domain), and n is the number of direct child nodes for this domain. equation(2)

$$W(D_{ij}) = \left(\sum_{k=1}^n W(N_{ik}) \right) / n$$

The User Profiler Agent consists of two components namely Profile Builder and Profile Updater. The Profile Builder component is responsible for building the basic user profile through two steps. Firstly, it creates an initial user profile from the user basic information, which is provided by the user such as his name, gender, and location. Secondly, it asks the user to determine a list of domains to be ordered according to his interests. These domains represent the basic domains, which may be augmented gradually later during the search sessions according to the extracted implicit feedback. Then for each stem, the domain is obtained from the WordNet ontology. On the other hand, the Profile Updater component is responsible for updating the user profile based on the implicit feedback coming from the Interface Agent in the form of set of pairs $\{(Document D, \{Action A\})\}$. In addition, it also participates in building the initial user profile by updating the newly created profile with the extracted term stems and domains from the user browsing history documents that are stored on the local machine. The following steps briefly describe how the user profile is updated by the algorithm shown in Fig. 5.

- Step 1: For each document received, the “TREM” algorithm extracts the document main terms.
- Step 2: Then for each extracted term, the Porter’s algorithm is employed to derive the corresponding stem.
- Step 3: Assign weight initial value equal 1 to all document term stems.

•*Step 4:* Based on the term location in the document, the term stem weight may vary. For example, the weight of the terms located in the document header or meta-data have higher weights.

•*Step 5:* The actions taken on the document also affect the term stem weight. Therefore, if the user bookmarked the document or saved it, the weight of each extracted term stem from this document is increased.

•*Step 6:* After calculating the term stem weight for each term in each document, the nearest super class/domain is retrieved from the Word Net ontology to construct the user profile updating set, which has triplet elements: a term stem, its domain, and its weight.

•*Step 7:* For all term stems in the user profile updating set, the algorithm takes one of three actions. In case that the stem exists under its associated domain, the algorithm updates the stem weight. In case that the domain exists only, the algorithm adds the stem under this domain. In case that neither the domain nor the stem exist, the algorithm adds both of them.

•*Step 8:* Finally, the algorithm recalculates the domain weights using formula (2).

```

Input: DS = {(D1, {A11, A12, A1m}), (D2, {A21, A22, A2n})...}
Output: Updated User Profile
Steps
For Each Di In DS
    TE = ExtractTerms(Di)
    TEs = GetTermsStems(TE)
    For Each Tj In TEs
        W(Tj) = 1
        If (IncludedIn(DocumentLocation(Tj), LOI))
            Vi = GetLocationWeight(DocumentLocation(Tj))
            W(Tj) = W(Tj) + (W(Tj) * Vi)
        End If
    End For
    For Each Ai In {A}
        If(IncludedIn(Ai, AOD))
            Va = GetActionWeight(Ai)
            W(Tj) = W(Tj) + (W(Tj) * Va)
        End If
        AddInDT(Di, Tj, W(Tj))
    End For
End For
SC = GetWordNetHypernym(DT)
US(U) = ConstructUpdateSet(SC, T, W(T))
FOR USi In US(U)
    If(TermDomainExists(Pi, USi))
        Wp(Tk) = GetProfileTermWeight(Pi, USi)
        UpdateSavedTermWeight(Pi, Wp(Tk), USi)
    Else if(DomainExists(Pi, USi))
        AddTerm (Pi, USi)
    Else
        AddTermAndDomain (Pi, USi)
    End If
End For
RecalculateAllDomainWeights()

```

Figure 5. The user profile updating algorithm.

4.4. The meta-search agent

Meta-Search Agent is a reactive agent that reacts to the requests coming from the Interface Agent. It acts as a Meta-Search engine, where it sends the optimized user search query to several traditional search engines, and then it merges the results into a single list. It includes two components: Search Engines Interface and Results Data Fusion components. The Search Engines Interface component interacts with the search engines through an Application Programmable Interfaces (APIs) to send the optimized query and receives the search results. After that, the Results Data Fusion merges the results received from the search engines into a single list. For each search engine, there is a retrieved search results $r_i(Q_e)$, which is represented in a sequence $\langle r_{i0}, r_{i1}, r_{i2}, \dots, r_{in} \rangle$. The sequences of search results retrieved from the search engines are merged into a single sequence $R_m(Q_e)$. In our model, the CombSum method can be employed to merge the search results. This method sums up all the similarity scores of a document and the query, and also normalizes the similarity scores of the documents. This process is terminated when all results are retrieved and defused to generate the final single list of the search results.

5. CONCLUSION

We propose a model for personalized web search which will consider individual's interests and past browsing history into mind. For this purpose, we'll use domain knowledge in background. Using user's interest and browsing history, different domain will be incorporated in domain knowledge. Using domain knowledge, we'll create user profile. Once the user profile will be created, model will take user query and will suggest the relevant web pages with respect to query. In this model, we propose search as you



type method instead of keyword search. In relevant domain, the information will be searched as user will type character by character. We'll perform some experiments in order to compare the retrieval effect using search as you type with keyword search in particular domain. Acknowledgement With the cooperation of my guide, I am highly indebted to Prof. A.P. Bodkhe , for his valuable guidance and supervision regarding my topic as well as for providing necessary information regarding materials.

6. REFERENCES

- [1] Rakesh Kumar & Aditi Sharan, "PERSONALIZED WEB SEARCH USING BROWSING HISTORY AND DOMAIN KNOWLEDGE", Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference Page(s): 493 – 497, 2014
- [2] M Speretta and S Gauch, "Personalized Search Based on User Search Histories", Proceeding Of International Conference on Web Intelligence, pp. 622-628, 2005.
- [3] S.Vanitha, "A PERSONALIZED WEB SEARCH BASED ON USER PROFILE AND USER CLICKS", International Journal of Latest Research in Science and Technology Volume 2, Issue 5: Page No.78-82, September October 2013
- [4] F Liu, C Yu and W Meng, "Personalized Web Search for Improving Retrieval Effectiveness", IEEE Transactions On Knowledge And Data Engineering, pp. 28-40, Volume 16, 2004.
- [5] Susmitha Dey, Siby Abraham, "User Interface For A Search Engine: A Customized and Multi-domain Approach", International Journal of Computer Information Systems and Industrial Management Applications. Volume 4 (2012) pp. 169-179
- [6] Devang Karavadiya, Purnima Singh, "User Specific Search Using Grouping and Organization", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume 1, Issue 4, November – December 2012 Carman MJ, Crestani F, Harvey M, Baillie M. Towards query log based personalization using topic models. In: Proceedings of the 19th ACM conference on information and knowledge management (CIKM) 2010 Toronto, Ontario, Canada; 2010. p. 1849–52.
- [7] Lv Y, Sun L, Zhang J, Nie J, Chen W, Zhang W. An iterative implicit feedback approach to personalized search. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics (ACL), Sydney, Australia; 2006. p. 585–92.
- [8] Princeton University. WordNet: A lexical database for English. <<http://wordnet.princeton.edu/>>; 2012.
- [9] Leung KW, Lee DL, Lee W. Personalized web search with location preferences. In: IEEE international conference on data engineering (ICDE), Long Beach, California; 2010. p. 701–12.
- [10] Boudghaghen O, Tamine L, Boughanem M. Personalizing mobile web search for location sensitive queries. In: Proceedings of the 2011 IEEE 12th international conference on mobile data management, vol. 01, Lulea, Sweden; 2011. p. 110–8.

[11] Weber I, Castillo C. The demographics of web search. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval. Special interest group on information retrieval (SIGIR), UniMail, Geneva, Switzerland; 2010. p. 523–30.

Author Profile



Dr. Shaik Abdul Muzeer

Professor & Principal

Megha institute of Engineering & Technology for Women

Dr.S.A.Muzeer, at present working as a principal of Megha institute of engineering & Technology has completed his PG and P.HD in Electronics & Communication Engineering and published around 25 Papers in National & International Journals. His area of research is Digital signal processing and Bio-medical engineering