Egyptian Informatics Journal (2012) **13**, 191–198

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

Agent-based web search personalization approach using dynamic user profile

Ibrahim F. Moawad *, Hanaa Talha, Ehab Hosny, Mohamed Hashim

Faculty of Computer and Information Science, Ain shams University, Abbasia, Cairo 11566, Egypt

Received 29 April 2012; revised 31 August 2012; accepted 16 September 2012
Available online 25 October 2012

KEYWORDS

Web search personalization;
Dynamic user profile;
Query optimization;
Multi-agent system

Abstract The World Wide Web has become the largest library through the history of the humanity. Having such a huge library made the search process more complex as the syntactic search engines offer an overwhelming amount of search results. Vocabulary problems like polysemy and synonymy can make the search results of traditional search engines irrelevant to users. Such problems trigger a strong need for personalizing the web search results based on user preferences. In this paper, we propose a new multi-agent system based approach for personalizing the web search results. The proposed approach introduces a model to build a user profile from initial and basic information, and maintain it through implicit user feedback to establish a complete, dynamic and up-to-date user profile. In the web search process, the model semantically optimizes the user query in two steps: query optimization using user profile preferences and query optimization using the WordNet ontology. The model builds on the advantages of the current search engines by utilizing them for retrieving the web search results. We present a detailed case study and simulation results evaluation to illustrate how the proposed model works and its expected value in increasing the precision of the traditional search engines and solving the vocabulary problems.

1. Introduction

With the huge amount of information available on the World Wide Web, some Internet users may face information overflow problems, where there are a huge number of hosted documents on the web. The keyword-based search engines are unable to satisfy the user needs during his web search process. Furthermore, these search engines do not address vocabulary problems such as polysemy and synonymy. Polysemy refers to the existence of multiple meanings for a single word. For example, when a user searches for the word "Jaguar", the retrieved results may be related to jaguar as an animal or jaguar as a car brand. Synonymy refers to the existence of multiple words

* Corresponding author. Tel.: +20 1064577725.
E-mail addresses: ibrahim_moawad@cis.asu.edu.eg (I.F. Moawad), dr_hanaa_talha@hotmail.com (H. Talha), ehabhosny336@hotmail.com (E. Hosny), mhashem100@yahoo.com (M. Hashim).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

1110-8665
http://dx.doi.org/10.1016/j.eij.2012.09.002

with the same meaning. For example, when a user searches for the word "car", all results that are related to another word such as "vehicle" would be neglected although the two words are nearly having the same meaning [1].

Personalization of web search results can solve the mentioned problems by focusing on the most relevant results for the user query combined with his preferences-identified in his user profile. In addition, it reduces the user efforts during the web search process [1]. Recently, several approaches of web search personalization have been proposed. Some of the most successful approaches are based on building a user profile that represents user interests and using it in the web search process [2–5].

In this paper, a new multi-agent system based model called "Web Search Personalizer" is proposed for personalizing the web search results. The multi-agent system (MAS) technique is employed here to enhance the information retrieval precision and recall assessment criteria through building multiple agents for addressing the different personalization issues and phases. Also, using MAS technique, our model gains the advantages of interacting with the current search engines to retrieve and defuse the web search results. In the proposed model, a user profile is built from initial and basic information and maintained through the implicit user feedback extracted by the click-through technique [6–8]. Consequently, the model keeps the user profile complete, dynamic and up-to-date. In addition to keeping an up-to-date user profile, we also propose a semantic query optimization technique based on both query related user profile preferences and the WordNet ontology. WordNet is the largest lexical database containing words grouped into synsets (Synonym Sets) [9].

The rest of this paper is organized as follows: Section 2 reviews some related work. Section 3 describes the proposed model conceptual view, while Section 4 presents the model architecture in details. Section 5 presents a detailed case study and its simulation results evaluation. Finally, the research work presented in this paper is concluded in Section 6.

2. Related work

Different approaches were proposed by researches in the area of web search personalization. Some of these approaches are based on the user's geographical location considering the location factor only [10–12]. In such approaches, the retrieved results are related to the user's language and his demographic attributes, without considering any other user preferences. Although, these approaches may give better results than the traditional search engines, the users in the same geographical area will have the same results even if they have different preferences and interests.

On the other hand, other some approaches re-rank the retrieved search results from the traditional search engines based on the user preferences [13–15]. The main disadvantage of these approaches is that the search process relies on the original search query without taking the user preferences into consideration.

A user feedback can be an important factor to fine-tune the search results, thus another type of approaches employs the user implicit feedback to avoid direct user involvement [6–8]. These approaches rely on the user feedback only, so the search process takes long time and passes through multiple iterations.

Finally, the most famous and effective approaches build a robust user profile from different resources. This profile contains all user preferences, and hence it is used in web search personalization [2,4]. The main disadvantages of these works are either they ignore the vocabulary problems or involve the user in enhancing and maintaining his profile.

To tackle the above-mentioned disadvantages, we propose a multi-agent system based model called "Web Search Personalizer" (WSP) that builds and maintains the user profile automatically. The proposed model also presents a semantic-based optimization method for the user's query. The model uses the user preferences to overcome the polysemy problem, and employs the WordNet ontology to solve the synonymy problem. Moreover, the model updates the user profile through the user implicit feedback.

3. WSP model conceptual view

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.

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 WordNet 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,

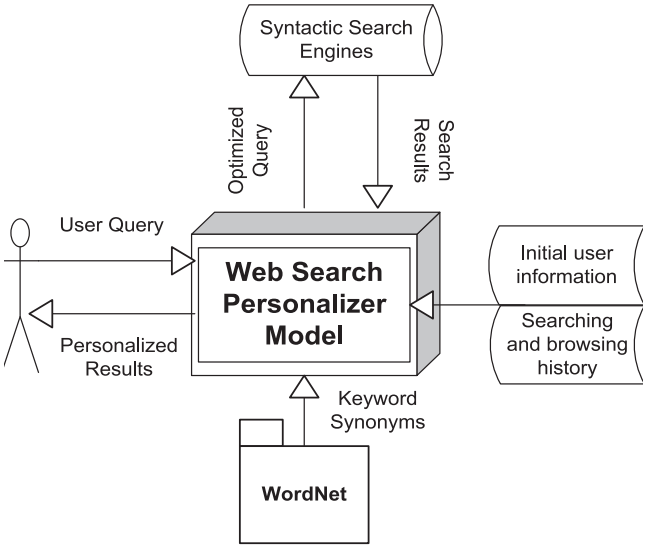

Figure 1 The WSP model conceptual view.

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. 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.

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) [16]. 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-elated 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.

4.1. The interface agent

The Interface Agent is a coordinator agent [17] 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

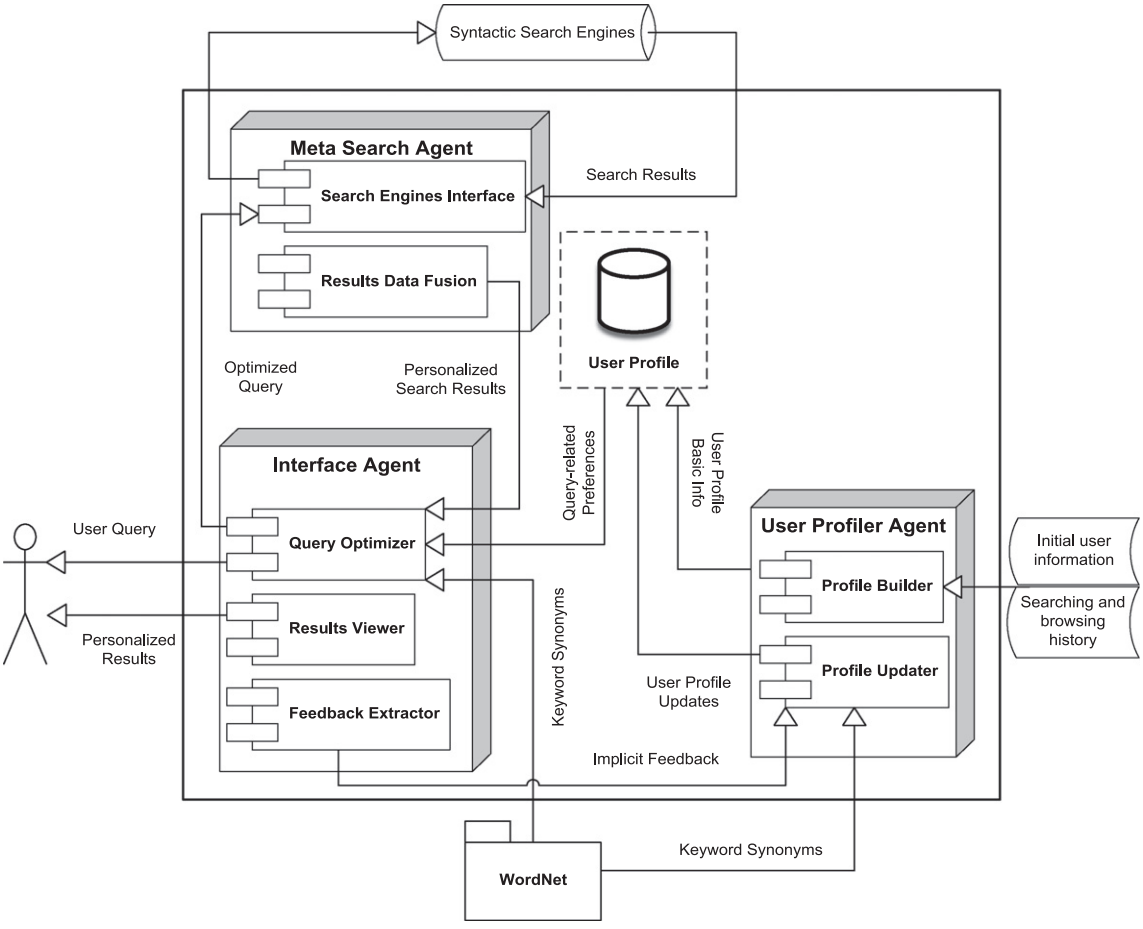

Figure 2 The WSP model architecture.

```
Input: Q(U_i) = {T_1, T_2, ..., T_n}
Output: Q_e(U_i)
Steps:
For Each T_i In Q(U_i)
        T_is = GetStem(T_i)
        D(T_is) = GetWordNetDomain(T_is)
        If (IsNotExist(D(T_is) , DC))
                Add(DC, D(T_is))
        Else
                IncreaseWeight(D(T_is))
        End If
End For
HDC = GetHighestWeightDomains(DC)
If (LengthOf(HDC) > 1)
        For Each HDC_j In HDC
                W_j = GetProfileDomainWeight(HDC_j)
                If W_j > W_j-1
                        D(Q(U_i)) = HDC_j
                End If
        End For
Else
        D(Q(U_i)) = HDC_1
End If
PP_i = GetRelatedTermsFromUserProfile(Q(U_i))
WS(Q(U_i)) = GetSynonyms(Q(U_i))
D(Q(U_i)) = GetNearestSuperClass(D(Q(U_i)), PP_i , WS(Q(U_i)))
Q_e(U_i) = Concatenate(Q(U_i), PP_i, D(Q(U_i)) , WS(Q(U_i)))
```

Figure 3 The query optimization algorithm.

a set of term stems $\{T_0, T_1, T_2, \ldots, T_n\}$, 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 $WS(Q(U_i))$.

$$Q_e(U_i) = \{Q(U_i)_U PP_{iU}\{D(Q(U_i))\}_U WS(Q(U_i))\} \quad (1)$$

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 [18].
- *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.

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.

4.2. The user profiler agent

User Profiler Agent is a reactive agent [17] 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.

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 $\{T_{i0}, T_{i1}, \ldots, T_{ij}\}$ and domains $\{D_{i0}, D_{i1}, \ldots, D_{ij}\}$ and each term stem has a weight $W(T_{ij})$, so P_i can be represented as a set of triplets: $\{(D_{i0}, T_{i0}, W(T_{i0})), (D_{i1}, T_{i1}, W(T_{i1})), \ldots\}$. 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.

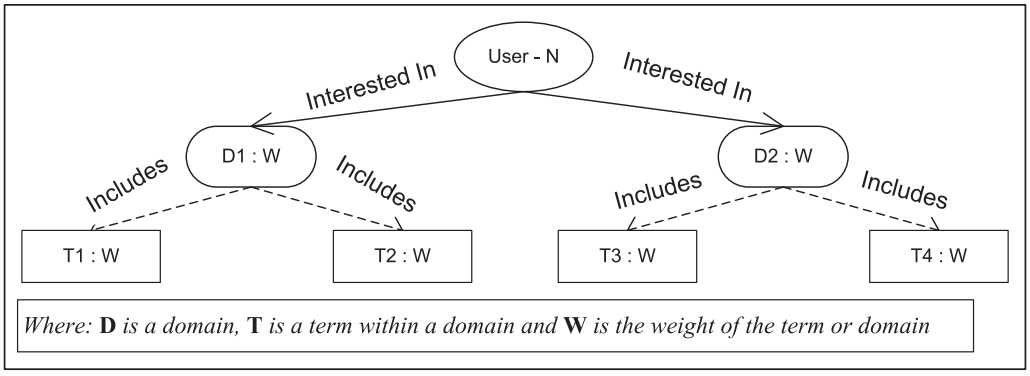

Figure 4 User profile representation.

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

Figure 5 The user profile updating algorithm.

$$W(D_{ij}) = \left(\sum_{k=1}^{n} W(N_{ik})\right) \Big/ n \tag{2}$$

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 [19] extracts the document main terms.
- *Step* 2: Then for each extracted term, the Porter's algorithm [18] 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 WordNet 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).

4.3. The meta-search agent

Meta-Search Agent is a reactive agent [17] 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 $\langle r_{i0}, r_{i1}, r_{i2}, \ldots, r_{in}\rangle$. 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 [20]. 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. Case study

Suppose that we have a user (A) and user (B). Each user has his own profile that contains his preferences, which are represented by a tree of domains and their related term stems as shown in Fig. 6. As we can see from this figure, user (A) is interested in "Movies" and "Cars" domains. He is interested especially in "Cartoon" movies for two cartoon characters: "Goofy" and "Pluto". He is also is interested in two car brands in "Cars" domain: "Jaguar" and "Ferrari". On the other hand, user (B) interested in two domains: "Animals" and "Planets". In "Animals" domain, user (B) interested in the "Jaguar" animal, while he is interested in "Pluto" planet in the "Planet" domain.

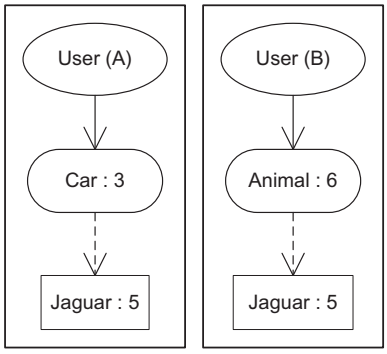

Figure 7 Query-related partial profiles for users (A) and (B).

If both users enter the same search query: "Jaguar Speed", the query optimizer component will access both WordNet ontology and the user profile to determine the query context domain and the query-related partial profile for reach user. As shown in Fig. 7, the query-related partial profiles extracted include only the domains related to the query context domain.

After that, the query optimizer component optimizes the query to generate the corresponding optimized queries for both users based on their context domain, user profile preferences (query-related partial profiles), query stems and query synonyms. The query optimization process passes through set of steps. Table 1 shows the intermediate results derived from these steps to infer the optimized query from the original query ("*Jaguar Speed*") for both user (A) and user (B). Then, the optimized query for each user is sent to the Meta-Search agent, which interacts with set of search engines to retrieve their search results.

Finally, all the retrieved search results from the search engines can be defused together to generate a single list of results to guarantee that no redundant search results will be displayed

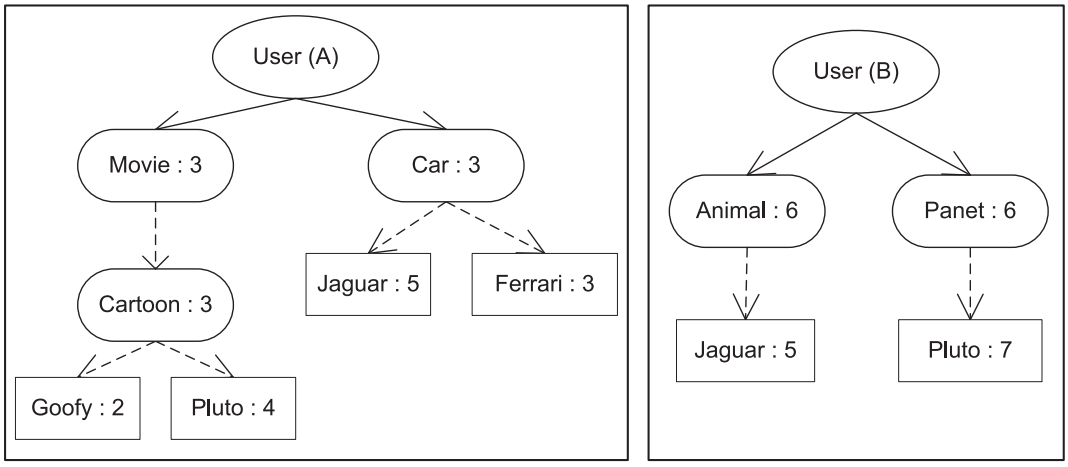

Figure 6 User (A) and user (B) profiles.

Table 1 The intermediate results of the optimization query steps.

Result/user	User (A)	User (B)
Query stems	{*Jaguar Speed*}	{*Jaguar Speed*}
Query domain context	{*Car*}	{*Animal*}
Query-related partial profile	{(*D* = "*Car*", *T* = "*Jaguar*", *W*(*T*) = 5)}	{(*D* = "*Animal*", *T* = "*Jaguar*", *W*(*T*) = "5")}
Query synonyms	{*Velocity*}	{*Cat*, *Velocity*}
Optimized query set	{*Jaguar Speed*} *U* {*Car*} *U* {*Car*} *U* {*Velocity*}	{*Jaguar Speed*} *U* {*Animal*} *U* {*Animal*} *U* {*Cat*, *Velocity*}
Optimized query	*Jaguar AND* (*Speed OR Velocity*) *AND Car*	(*Jaguar OR Cat*) *AND* (*Speed OR Velocity*) *AND Animal*

Table 2 Google and Bing results for original and optimized queries.

System/criteria	Number of relevant results	Number of irrelevant results	Precision (%)	Solving vocabulary problems
Google (original query)				
User (A)	41	9	82	No
User (B)	6	44	12	No
Bing (original query)				
User (A)	40	10	80	No
User (B)	4	46	8	No
Google (WSP-optimized query)				
User (A)	46	4	92	Yes
User (B)	47	3	94	Yes
Bing (WSP-optimized query)				
User (A)	49	1	98	Yes
User (B)	45	5	90	Yes

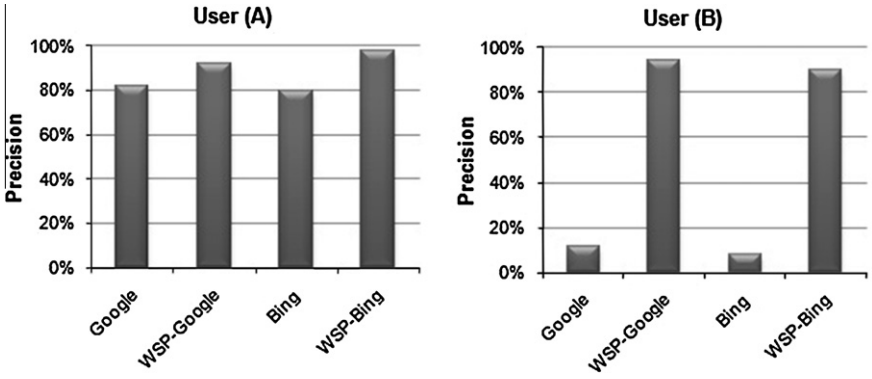

Figure 8 Original query precision versus original query precision.

for the user. The WSP model can also sense the click-through feedback to update the user profile for each user based on the taken actions on each link.

We assessed our model, WSP, versus both Google and Bing search engines that we also exploited them in our model. The original query of the above case study for both users has been passed to Google and Bing search engines to retrieve their results. On the other hand, we have passed the optimized query for both users to the same search engines. Table 2 shows the results for the top 50 retrieved links from Google (using original query), Bing (using original query), Google (using WSP-optimized query), and Bing (using WSP-optimized query) search engines.

As we can notice from Table 2 and Fig. 8, the precision of the retrieved search results from both search engines has been increased for both users when they search by the WSP-optimized query. The Google's precision has been increased from 82% to 94% for user (A), and from 12% to 94% for user (B). In the same way, the Bing's precision has been increased from 80% to 98% for user (A), and from 8% to 90% for user (B).

In addition, WSP model was also capable of handling the vocabulary problems (polysemy and synonym). For example, the case study results show that some of the retrieved documents contains the word "velocity", which is not exist in the original query, but it has the same meaning of the word "speed", which is exist in the original query.

6. Conclusion

In this paper, we presented a model for web search personalization based on multi-agent system technique. The model builds and maintains the user profile dynamically to keep it up-to-date. During the web search process, the query is optimized semantically using the user profile preferences and the WordNet ontology. Also, a detailed case study was presented to show how the proposed model increased the precision of both Google and Bing search engines when they were used by two users to search the web using the same query, but with different profiles. Besides, in the case study, both polysemy and synonymy problems were overcome by exploiting user preferences and WordNet ontology respectively.

References

[1] Micarelli A, Gasparetti F. Personalized search on the World Wide Web, the adaptive web. Berlin, Heidelberg: Springer-Verlag; 2007, p. 195–230 ISBN: 978-3-540-72078-2.

[2] Stermsek G, Strembeck M, Neumann G. User profile refinement using explicit user interest modeling. In: GI-Jahrestagung conference. Technical University in Berlin; 2007. p. 289–93.

[3] Min J, Jones GJF. Building user interest profiles from wikipedia clusters. In the workshop on enriching information retrieval (ENIR 2011) at special interest group on information retrieval (SIGIR), Beijing, China; July 2011.

[4] Ghosh R, Dekhil M. Discovering user profiles. In: Proceedings of the 18th international conference on World Wide Web. Polytechnic University in Madrid; 2009. p. 1233–4.

[5] Kovacikova T, Petersen F, Pluke M, Alonso Alvarez V, Bartolomeo G, Frisiello A, et al. Personalization and user profile standardization. European Telecommunications Standards Institute (ETSI) workshop on personalization and user profile standardization, at the ETSI Headquartes in Sophia Antipolis, France; 2009.

[6] Sun J, Zeng H, Liu H, Lu Y, Chen Z. CubeSVD: a novel approach to personalized web search. In: Proc of the 14th international World Wide Web conference, Chiba, Japan; 2005. p. 382–90.

[7] 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.

[8] 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.

[9] Princeton University. WordNet: A lexical database for English. <http://wordnet.princeton.edu/>; 2012.

[10] 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.

[11] Bouidghaghen 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.

[12] 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.

[13] LI L, YANG Z, Kitsuregawa M. Rank optimization of personalized search. Detroit Baptist Seminary (DBSJ) J 2009;4(3):666–71.

[14] Morris MR, Teevan J, Bush S. Enhancing collaborative web search with personalization: groupization, smart splitting, and group hit-highlighting. In: Proceedings of the 2008 ACM conference on computer supported cooperative work (CSCW), San Diego, California, USA; 2008. p. 481–4.

[15] Matthijs N, Radlinski F. Personalizing web search using long term browsing history. In: ACM international conference on web search and data mining (WSDM), Hong Kong; 2011. p. 25–34.

[16] Labrou Y, Finin T, Peng Y. Agent communication languages: the current landscape. J IEEE Intell Syst 1999;14(2):45–52.

[17] Breitman K, Casanova MA, Truszkowski W. Semantic web: concepts, technologies and applications. Springer-Verlag London Limited; 2007, p. 219–27 e-ISBN: 978-1-84628-710-7 [chapter 11].

[18] Willett P. The Porter stemming algorithm: then and now. Program: Electr Libr Inform Syst 2006;40(3):219–23.

[19] VU T, AW AT, Zhang M. Term extraction through unithood and termhood unification. In: Proceedings of the third international joint conference on natural language processing, Hyderabad, India; 2008.

[20] Lee JH. Combining multiple evidence from different properties of weighting schemes. Interest group on information retrieval (SIGIR). In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval, Seattle, Washington, USA; 1995. p. 180–8.