Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

IJMTER

ISSN (ONLINE) : 2349 - 9745

ISSN (PRINT) : 2393 - 8161

A NOVEL TECHNIQUE FOR MAPPING USER QUERIES TO CATEGORIES IN PERSONALIZED WEB SEARCH

Rasika M. Kaingade[1], Sneha P. Sumare[2], Hemant A. Tirmare[3]

[1,2,3]Department of Technology, Shivaji University, Kolhapur

Abstract— Currently web search engines are widely used to find out the information on the web. A typical search engine provides similar set of results for different users without considering who had submitted the query. Different users have different needs. Hence, personalized web search provides the search results based upon user interests. A novel technique is proposed to map a user query to a set of categories, which represent the user's search intention. This set of categories can serve as a background to disambiguate the words in the user's query. The user's search history and a category hierarchy are used to learn the user profile and a general profile respectively. These two profiles are combined to map a user query into a set of categories. There are several learning and combining algorithms which are found to be effective and efficient.

Keywords— Personalization, Search Engine, User Profile, Category Hierarchy, Information Filtering.

I. INTRODUCTION

Web search engine is a tool which allows the web user for finding information from the World Wide Web. Many search engines like Google, Yahoo provide a relevant and irrelevant data to the user based on their search. To avoid the irrelevant data the technique called Personalized Web Search (PWS) were arise. Personalized web search is promising way to improve the search quality by customizing web search result for people with different information goals. When the same query is submitted by different users, a typical search engine returns the same result, regardless of who submitted the query. This may not be suitable for users with different information needs.

For example, for the query "java", some users may be interested in documents dealing with “java” as “programming language”, while other users may want documents related to java island in Indonesia. One way to disambiguate the words in a query is to associate a small set of categories with the query. For example, if the category "programming language" is related with the query "java", then the user's intention becomes clear. Current search engines such as Google or Yahoo! have hierarchies of categories to help users to specify their intentions.

A user may relate to one or more categories to his/her query manually. For example, before submitting the query a user may browse a hierarchy of categories and select one or more categories in the hierarchy. Making use of these categories, a search engine will return results that are more appropriate to the user. But the category hierarchy exposed to a user is generally large which may lead user to have difficulty in finding out the proper path to suitable categories. Also users are often too impatient to identify the proper categories before submitting the query.

Instead of browsing, user can obtain a set of categories directly by a search engine. But categories returned from a search engine are independent of a particular user which does not reflect the intention of the user.

This paper studies how to provide a small set of categories as a context for each query submitted by the each user, based on his/her search history. The following points show the approach:

- Collect the user's search history.
- Construct a user profile based on the search history and construct a general profile based on the *ODP* (Open Directory Project) category hierarchy.
- Assume suitable categories for each user query based on the user's profile and the general profile.

The categories obtained from the proposed method are likely to be related to the user's interest and, therefore, can provide a proper context for the user query. Suppose a mobile user wants to retrieve documents using his/her PDA. Since the bandwidth is limited and the display is small, it may not be practical to transmit a large number of documents for the user to choose the relevant ones. If it is possible to show the retrieved documents on one screen then these documents are not relevant to the user. By making use of proposed technique, a small number of categories with respect to the user’s query are shown. If none of the categories is desired, the next set of categories is provided. This is continued until the user clicks on the desired categories, usually one, to express his/her intention. Therefore, the proposed technique can be used to personalize web search.

II. LITERATURE REVIEW

Many techniques are used in modern search engines to provide more contexts for user queries. Search engine such as Yahoo, Google and ODP return both categories and documents. Northern Light [1] and WiseNut [2] cluster their results into categories, and Vivisimo [3] groups results dynamically into clusters. A lot of study in metasearch [4][5][6] also examine mapping user queries to a set of categories. But all above techniques return the same results for a given query, regardless of who submitted the query. This can be understood by having a general profile. The combination of a user profile and a general profile usually yields significantly higher accuracy than using a general profile or a user profile alone.

Several papers on information filtering [7][8][9][10][11] and intelligent agent such as WebWatcher [12], construct user profiles explicitly or implicitly and recommend documents using the profiles. Previous methods filter the documents, but the goal is to retrieve categories of interest for a user query. No general profile is used in information filtering.

Also the text categorization has been examined in detail. A comparison of various methods is given in [13]. In [14][15][16] categorization of web pages or collections of web pages has been studied. Use of a category hierarchy is derived from [16].

In [17] WebMate uses user profiles to refine user queries. In [18] Watson refines queries using a local context but does not learn the user profile. The paper [19] uses user’s preferences to choose data sources and refine queries but it does not have user profiles, and requires the users to provide their preferences of categories. The user profiles are learned from their surfing histories, and re-ranks/filters documents returned by a metasearch engine based on the profiles [20].

III. PROPOSED APPORACH

3.1. User Search History

A user’s search history is stored by the search engine to learn the user’s long-term interests. The information considered to represent a user’s search history is: queries, relevant documents and related categories.

One search record is generated for each user search session. A tree model of search records is shown in Figure 1. In this model, nodes are information items and edges are relationships between nodes. The root of a search record is a query. Each query has one or more related categories. Associated with each category is a set of documents, each of which is both relevant to the query and related to the category. For almost all queries, each query is related to only one or two categories.

For example, a user is interested in programming languages submits a query “java” to the search engine, and it returns the top 10 documents and 10 categories. The user clicks the 5th category “programming languages” and the search engine shows all documents that have been clustered into this category. Then the user clicks two documents about java programming language. When this search session is finished, a search record as shown in Figure 1 can be generated and saved for the user.

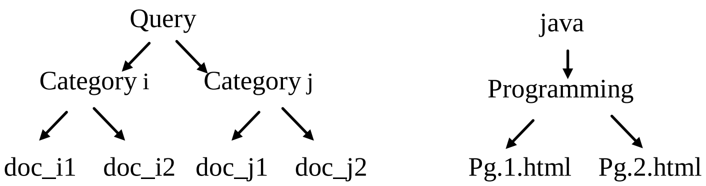


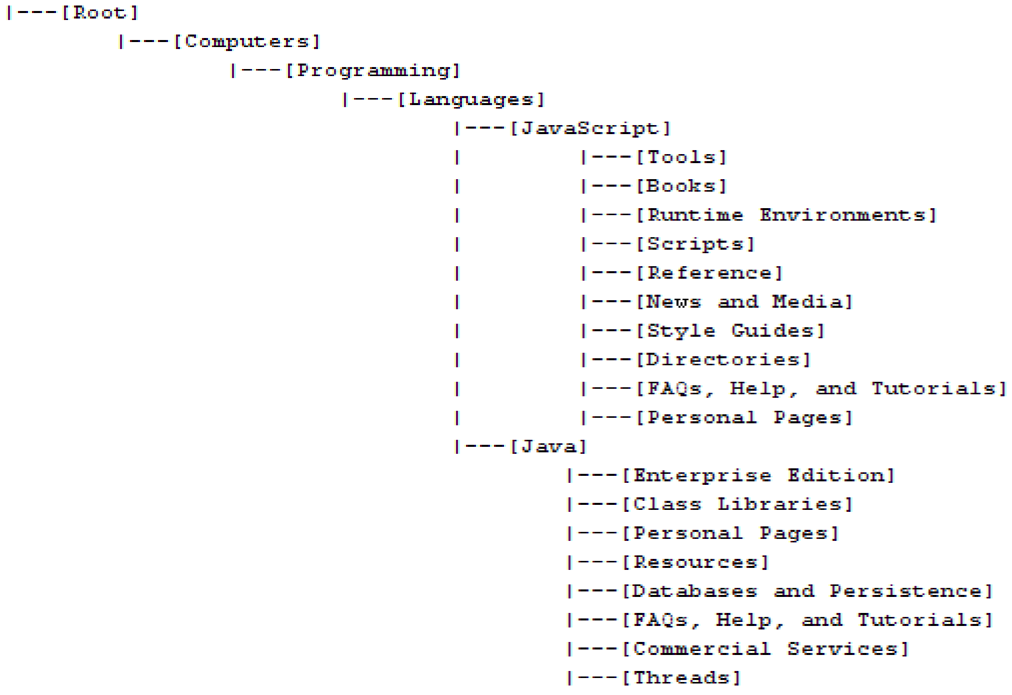
Figure 1: Example of a search record

3.2. User Profile

User profiles are used to represent users' interests and to assume their intentions for new queries. In this paper, a user profile consists of a set of categories and for each category, a set of keywords with weights. Each category represents a user interest in that category. The weight of a word in a category reflects the significance of the word in representing the user's interest in that category. For example, if the word "java" has a high weight in the category "programming language", then the occurrence of the word "java" in a future query of the user has a tendency to indicate that the category "programming language" is of interest. A user's profile will be learned automatically from the user's search history.

3.3. Category Hierarchy

We use some general knowledge which is applicable to all users. The reason for using the additional information is that the knowledge acquired from a user is often limited and may not be sufficient to determine the user's intention when a new user query is encountered. For example, a new query may contain terms that have never been used by the user before, nor appeared in any of his/her previous retrieved relevant documents. The general knowledge that our system utilizes is extracted from *ODP*. Specifically, we use the first three levels of *ODP*. The categories in the first two levels are used to represent the set of all categories. The terms appearing in these three levels of categories are used to represent the categories in the first two levels. We learn a general profile from the category hierarchy using a process similar to that for learning the user profile. Figure 2. shows a category hierarchy.



```
|---[Root]
        |---[Computers]
                |---[Programming]
                        |---[Languages]
                                |---[JavaScript]
                                |        |---[Tools]
                                |        |---[Books]
                                |        |---[Runtime Environments]
                                |        |---[Scripts]
                                |        |---[Reference]
                                |        |---[News and Media]
                                |        |---[Style Guides]
                                |        |---[Directories]
                                |        |---[FAQs, Help, and Tutorials]
                                |        |---[Personal Pages]
                                |---[Java]
                                        |---[Enterprise Edition]
                                        |---[Class Libraries]
                                        |---[Personal Pages]
                                        |---[Resources]
                                        |---[Databases and Persistence]
                                        |---[FAQs, Help, and Tutorials]
                                        |---[Commercial Services]
                                        |---[Threads]
```

Figure 2: Category Hierarchy

The mapping is carried out as follows. First, the similarities between a user query and the categories representing the user's interests are computed. Next, the categories are ranked in descending order of similarities. Finally, the top three categories together with a button indicating the next three categories are shown to the user. If the user clicks on one of these top three categories, then the user's intention is explicitly shown to the system. If the user's interest is not among the top three categories, then the button can be clicked to show the next three categories.

IV. ALGORITHMS USED TO LEARN PROFILES

4.1. pLLSF-based Algorithm

Given the a matrix of m-by-n called as document-term matrix DT and the m-by-p called as document-category matrix DC, the Linear Least Squares Fit (LLSF) method [21] computes a p-by-n category-term matrix *M*. Another alternative called "pseudo-LLSF" (pLLSF), is calculated in which the dimensions of DT are reduced. But the basic idea is that the noise in the original document-term matrix DT is removed by the dimension reduction technique.

4.2. bRocchio Algorithm

The simple version of bRocchio adopted in text categorization is used:

$$M(i,j) = \frac{1}{N_i} \sum_{k=1}^{m} DT(k,j) * DC(k,i)$$

Where M is the matrix representing the user profile, N_i is the number of documents that are related to the i-th category, m is the number of documents in DT.

4.3. kNN Algorithm

The k-Nearest Neighbor (kNN) does not compute a user profile. Instead, it computes the similarity between a user query and each category directly from DT and DC.

V. MAPPING USER QUERIES TO RELATED CATEGORIES

The following three processes show mapping a new user query to a set of categories.

5.1. Using User Profile Only

The similarity between a query vector q and each category vector c in the user profile M is computed by the Cosine function [22]. In kNN, the algorithm first finds the k most similar documents among all document vectors in DT using the Cosine function. Then, among these k neighbors, a set of documents, say S, which are related to a category c can be identified using DC. Finally, the similarity between q and c is computed as the sum of the similarities between q and the documents in S. This is repeated for each category. The following formula is used which is modified to some extent from [13]:

$$Similarity(q,c_j) = \sum_{\mathrm{d_i} \in kNN} Cos(q,d_i) * DC(i,j)$$

Where q is the query; c_j is the j-th category; d_i is a document among the k nearest neighbors of q and the i-th row vector in DT, $Cos(q,d_i)$ is the cosine similarity between q and d_i.

5.2. Using General Profile Only

To calculate the general profile pLLSF is used which has the highest average accuracy and even though it is computationally expensive, profile needs to be computed only once.

5.3. Using Both User and General Profiles

The three methods are combined and compared with the above two baseline cases. Let c^u and c^g be the category vectors for the user profile and the general profile respectively. The following computation is done for every category.

1. Using only user profile: $Similarity(q,c) = Similarity(q,c^u)$.

2. Using only general profile: $Similarity(q,c) = Similarity(q,c^g)$.
3. Combine technique 1:
$$Similarity(q,c) = (Similarity(q,c^u) + Similarity(q,c^g))/2.$$
4. Combine technique 2:
$$Similarity(q,c) = 1 - \left(1 - Similarity(q,c^u)\right) * \left(1 - Similarity(q,c^g)\right).$$
5. Combine technique 3:
$$Similarity(q,c) = \max\left(Similarity(q,c^u), Similarity(q,c^g)\right).$$

VI. CONCLUSION

In this paper, an approach for personalization of web search by mapping user queries to the categories is explained. Users search history is collected without direct user’s participation. The user’s profile is constructed automatically from the user's search history. The user's profile is enhanced by a general profile which is extracted automatically from a common category hierarchy. Also the categories that are likely to be of interest to the user are deduced based on his/her query and the two profiles (general profile and user profile). For the construction of the profiles, learning algorithms (pLLSF, kNN and bRocchio) are evaluated.

REFERENCES

[1] http://www.northernlight.com/
[2] http://www.wisenut.com/
[3] http://www.vivisimo.com/
[4] S. Gauch, G. Wang, M. Gomez, “ProFusion: Intelligent Fusion from Multiple, Distributed Search Engines”, Journal of Universal Computer Science, 2(9), 1996.
[5] A. E. Howe and D. Dreilinger, “SavvySearch: A meta-search engine that learns which search engines to query”, AI Magazine, 18(2), 1997.
[6] C. Yu, W. Meng, W. Wu and K. Liu, “Efficient and Effective Metasearch for Text Databases Incorporating Linkages among Documents”, ACM SIGMOD, 2001.
[7] J. Allan, “Incremental relevance feedback for information filtering”, SIGIR, 1996.
[8] U. Çetintemel, M. J. Franklin, and C. Lee Giles, “Self-Adaptive User Profiles for Large-Scale Data Delivery”, ICDE, 2000.
[9] W. Foltz and S. T. Dumais, “Personalized information delivery: An analysis of information filtering methods”, CACM, 1992.
[10] D. H. Widyantoro, T. R. Ioerger and J. Yen, “An adaptive algorithm for learning changes in user interests”, CIKM, 1999.
[11] T. W. Yan and H. Garcia-Molina, “SIFT -- A Tool for Wide-Area Information Dissemination”, USENIX Technical Conference, 1995.
[12] Joachims, T., Freitag, D., and Mitchell, T., “Webwatcher: A tour guide for the World Wide Web”, IJCAI, 1997.
[13] Y. Yang and X. Liu, “A re-examination of text categorization methods”, SIGIR 1999.
[14] D. Koller and M. Sahami, “Hierarchically classifying documents using very few words”, ICML, 1997.
[15] Y. Labrou and T. Finin, “Yahoo! as an ontology: using Yahoo! categories to describe documents”, CIKM, 1999.
[16] W. Meng, W. Wang, H. Sun and C. Yu, “Concept Hierarchy Based Text Database Categorization. International Journal on Knowledge and Information Systems”, March 2002.
[17] L. Chen and K. Sycara, “WebMate: A Personal Agent for Browsing and Searching. Autonomous Agents and Multi Agent Systems”, 1998.
[18] J. Budzik and J. K. Hammond, “Watson: Anticipating and contextualizing information needs. In Proceedings of the Sixty-second Annual Meeting of the American Society for Information Science”, 1999.
[19] E. Glover, G. Flake, S. Lawrence, W. Birmingham, A. Kruger, C. Giles, and D. Pennock, “Improving Category Specific Web Search by Learning Query Modifications”, SAINT, 2001.
[20] A. Pretschner and S. Gauch, “Ontology based personalized search”, ICTAI, 1999.
[21] Y. Yang and C. G. Chute. An example-based mapping method for text categorization and retrieval. TOIS, 1994.
[22] G. Salton and M. J. McGill. Introduction to Modern Information Retrieval, 1983.