IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Deducing User Seek Goals with Feedback Series Using Fuzzy Self Constructing Algorithm

R.Keerthika[1], Prof.G.Ilanchezhiapandian[2]

[1]P.G Student, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore-632102, Tamil Nadu, India

[2]Dean (academic) & HOD CSE, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore-632102, Tamil Nadu, India

Abstract

An ambiguous query, various users may have distinct search goals when they enter in to a search engine. The deducing and analysis of user search goals can be very useful in improving search engine relevance and user experience. In this paper, I propose a novel approach to deduce user search goals by analysing search engine query logs. First, I propose a framework to discover various user search goals for a query by clumping the proposed feedback sessions. Feedback sessions are constructed from user click-through logs and can efficiently reflect the information needs of users. Second, I propose a novel approach to generate pseudo-documents to better represent the feedback sessions for clumping. Finally, I propose a new criterion "Classified Average Precision (CAP)" to evaluate the performance of deducing user search goals. Experimental results are presented using user click-through logs from a commercial search engine to validate the effectiveness of our proposed methods.

Keywords- *User search goals, feedback sessions, pseudo-documents, restructuring search results, classified average precision*

1 INTRODUCTION

Accurately measuring the semantic similarity between words is an important problem in web mining, information retrieval, and natural language processing. Web mining applications such as, community extraction, relation detection, and entity disambiguation, require the ability to accurately measure the semantic similarity between concepts or entities. In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Efficient estimation of semantic similarity between words is difficult for various natural language processing tasks such as word sense disambiguation (WSD), textual entailment, and automatic text summarization. Semantically related words of a particular word are listed in manually created general-purpose lexical ontologies such as WordNet. In WordNet, a synset contains a set of synonymous words for a particular sense of a word. However, semantic similarity between entities changes overtime and across domains. For example, apple is frequently associated with computers on the web. However, this sense of apple is not listed in most general-purpose thesauri or dictionaries. A user who searches for apple on the web, might be interested in this sense of apple and not apple as a fruit. New words are constantly being created as well as new senses are assigned to existing words. Manually maintaining ontologies to capture these new words and senses is costly if not impossible. I propose an automatic method to estimate the semantic similarity between words or entities using web search engines. Because of the vastly numerous documents and the high growth rate of the web, it is time consuming to analyze each document separately. Web search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most web search engines. Page count of a query is an estimate of the number of pages that contain the query words. In general, page count may not necessarily be equal to the word frequency because the queried word might appear many times on one page.

In this paper, I aim at discovering the number of diverse user search goals for a query and depicting each goal with some keywords automatically. I first propose a novel approach to deduce user search goals for a query by clumping our proposed feedback sessions. The feedback session is defined as the series of both clicked and unclicked URLs and ends with the last URL that was clicked in a session from user click - through logs. And Then, I Propose a novel optimization method to map feedback sessions to pseudo-documents which can efficiently reflect user information needs. At last, I cluster these pseudo documents to deduce user search goals and depict them with some keywords. Since the evaluation of clustering is also an important problem, I also propose a novel evaluation criterion classified average precision (CAP) to evaluate the performance of the restructured web search results. And finally I demonstrate that the proposed evaluation criterion can help us to optimize the parameter in the clustering method when inferring user search goals.
To sum up, our work has three major contributions as follows:

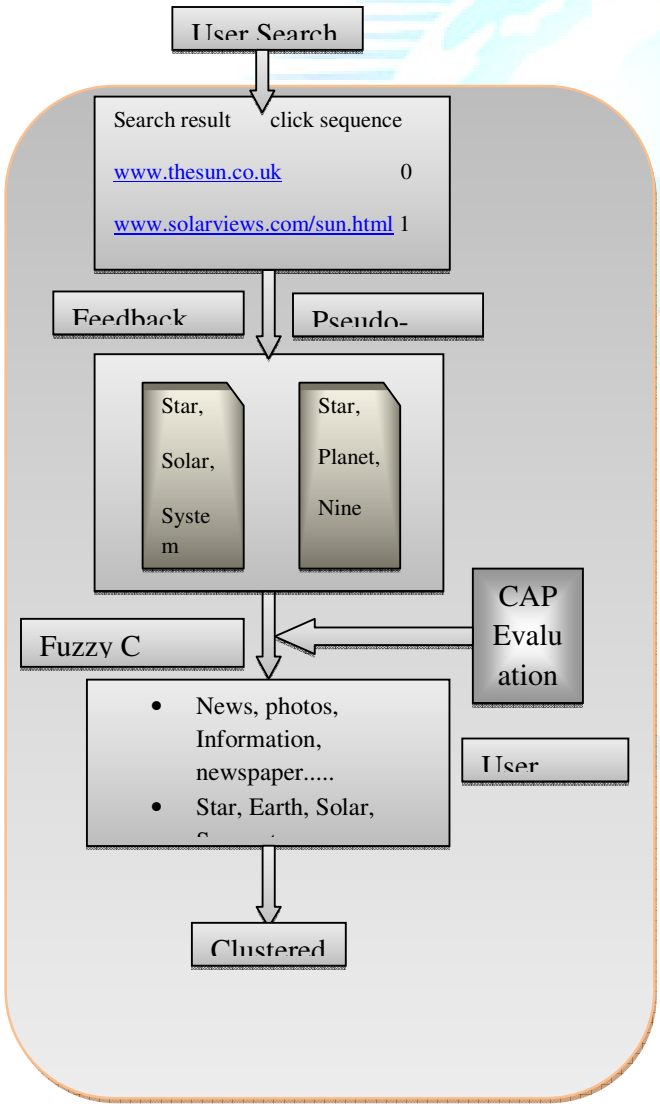
→ I propose a framework to deduce different user search goals for a query by clumping feedback sessions. I demonstrate that clumping feedback sessions is more

efficient than clumping search results or clicked URLs directly. Moreover, the distributions of different user search goals can be obtained conveniently after feedback sessions are clustered.

→ I propose a novel optimization method to combine the enriched URLs in a feedback session to form a pseudo-document, which can effectively reflect the information need of a user. Thus, we can tell what the user search goals are in detail.

→ I propose a new criterion CAP to evaluate the performance of user search goal inference based on restructuring web search results. Thus, we determine the user search goals for a query.

2 ARCHITECTURE FRAMEWORK



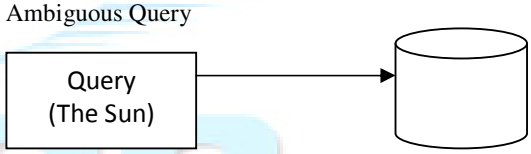
The framework of our approach consists of two parts divided by the dashed line. In the upper part, all the feedback sessions of a query are first extracted from user click-through logs and mapped to pseudo-documents and depicted with some keywords. Since we do not know the exact number of user search goals in advance, several different values are tried and the optimal value will be determined by the feedback from the bottom part.

In the bottom part, the original search results are restructured based on the user search goals deduced from the upper part. Then, I evaluate the performance of restructuring search results by our proposed evaluation criterion CAP. And the evaluation result will be used as the feedback to select the optimal number of user search goals in the upper part.

3 ILLUSTRATION OF FEEDBACK SESSIONS

Ambiguous Query- Queries are submitted to search engines to represent the information needs of users. However, sometimes queries may not exactly represent users specific information needs since many ambiguous queries may cover a broad topic and various users may want to get information on distinct aspects when they submit the same query. For example, when the query "the sun" is submitted to a search engine, some users want to locate the homepage of a United Kingdom newspaper, while some others want to learn the natural knowledge of the sun.

An Ambiguous Query



3.1 Restructure web search results- We need to restructure web search results according to user search goals by grouping the search results with the same search goal users with different search goals can easily find what they want. User search goals represented by some keywords can be utilized in query recommendation. The distributions of user search goals can also be useful in applications such as re-ranking web search results that contain different user search goals. Due to its usefulness, many works about user search goals analysis have been investigated. They can be summarized into three classes: query classification, search result reorganization, and session boundary detection.

3.2 Feedback Sessions- The feedback session consists of both clicked and unclicked URLs and ends with the last URL that was clicked in a single session. It is motivated that before the last click, all the URLs have been scanned and evaluated by users. Therefore, besides the clicked URLs, the unclicked ones before the last click should be a part of the user feedbacks. Feedback session can tell what a user requires and what he/she does not care about. Moreover, there are plenty of diverse feedback sessions in user click-through logs. Therefore, for deducing user search goals, it is more efficient to analyze the feedback sessions than to analyze the search results or clicked URLs directly

.

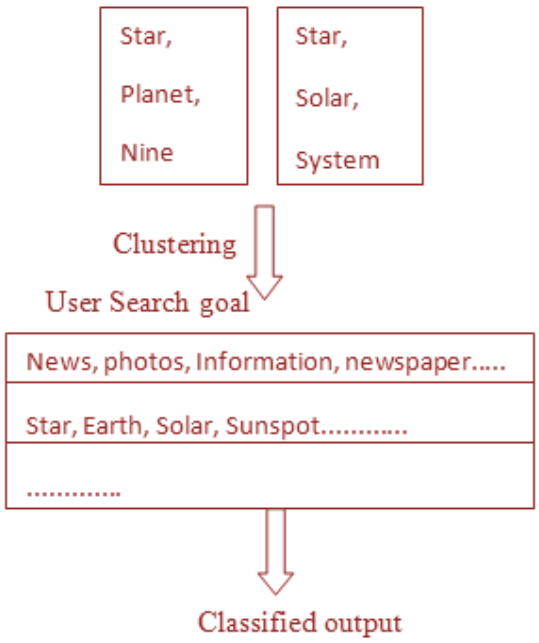
Search result	click sequence
www.thesunco.uk	0
www.solarviews.com/sun.html	1

3.3 Pseudo document- In this paper, I need to map feedback session to pseudo documents User Search goals. The building of a pseudo-document includes two steps. One is representing the URLs in the feedback session. URL in a feedback session is represented by a small text paragraph that consists of its title and snippet. Then, some textual processes are implemented to those text paragraphs, such as transforming all the letters to lowercases, stemming and removing stop words. Another one is Forming pseudo-document based on URL representations. In order to obtain the feature representation of a feedback session, we propose an optimization method to combine both clicked and unclicked URLs in the feedback session.

Search result	click sequence
www.thesunco.uk	0
www.solarviews.com/sun.html	1

Star, Planet, nine	Star, Solar, system

3.4 User Search Goals- I cluster pseudo-documents by FCM clustering which is simple and effective. Since we do not know the exact number of user search goals for each query, we set number of clusters to be five different values and perform clustering based on these five values, respectively. After clustering all the pseudo-documents, each cluster can be considered as one user search goal. The center point of a cluster is computed as the average of the vectors of all the pseudo-documents in the cluster.



4 FUZZY CLUSTERING

4.1 A fuzzy self-constructing algorithm (Data Mining Process)- Feature clustering is a powerful method to reduce the dimensionality of feature vectors for text classification. In this paper, we propose a fuzzy similarity-based self-constructing algorithm for feature clustering. The words in the feature vector of a document set are grouped into clusters, based on similarity test. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature, corresponding to a cluster, is a weighted combination of the words contained in the cluster.

By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided. Experimental results show

that our method can run faster and obtain better extracted features than other methods.

Fuzzy clustering is a class of algorithms for cluster analysis in which the allocation of data points to clusters is not "hard" (all-or-nothing) but "fuzzy" in the same sense as fuzzy logic.

4.2 Explanation of clustering- Data clustering is the process of dividing data elements into classes or clusters so that items in the same class are as similar as possible, and items in different classes are as dissimilar as possible. Depending on the nature of the data and the purpose for which clustering is being used, different measures of similarity may be used to place items into classes, where the similarity measure controls how the clusters are formed. Some examples of measures that can be used as in clustering include distance, connectivity, and intensity.

In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) Algorithm. The FCM algorithm attempts to partition a finite collection of n elements into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centres and a partition matrix, where each element wij tells the degree to which element xi belongs to cluster cj. Like the k-means algorithm, the FCM aims to minimize an objective function. The standard function is:

$$w_k(x) = \frac{1}{\sum_j \left(\frac{d(\text{center}_k, x)}{d(\text{center}_j, x)} \right)^{2/(m-1)}}.$$

Which differs from the k-means objective function by the addition of the membership values uij and the fuzzifier m? The fuzzifier m determines the level of cluster fuzziness. A large m results in smaller memberships wij and hence, fuzzier clusters.

In the limit m = 1, the memberships wij converge to 0 or 1, which implies a crisp partitioning. In the absence of experimentation or domain knowledge, m is commonly set to 2. The basic FCM Algorithm, given n data points (x1, . . ., xn) to be clustered, a number of c clusters with (c1, . ., cc) the center of the clusters, and m the level of cluster fuzziness with,

4.3 Fuzzy c-means clustering- In fuzzy clustering, every point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. An overview and comparison of different fuzzy clustering algorithms is available.

Any point x has a set of coefficients giving the degree of being in the kth cluster wk(x). With fuzzy c-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

The degree of belonging, wk(x), is related inversely to the distance from x to the cluster center as calculated on the previous pass. It also depends on a parameter m that controls how much weight is given to the closest center. The fuzzy c-means algorithm is very similar to the k-means algorithm:

- Choose a number of clusters.
- Assign randomly to each point coefficients for being in the clusters.
- Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than, the given sensitivity threshold):
- Compute the centroid for each cluster, using the formula above.
- For each point, compute its coefficients of being in the clusters, using the formula above.

The algorithm minimizes intra-cluster variance as well, but has the same problems as k-means; the minimum is a local minimum, and the results depend on the initial choice of weights.

Using a mixture of Gaussians along with the expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes.

Another algorithm closely related to Fuzzy C-Means is Soft K-means.

Fuzzy c-means has been a very important tool for image processing in clustering objects in an image. In the 70's, mathematicians introduced the spatial term into the FCM algorithm to improve the accuracy of clustering under noise.

5 ASSOCIATED WORK

In recent years, many works have been done to infer the so called user goals or intents of a query. But in fact, their works belong to query classification. Some works analyze the search results returned by the search engine directly to exploit different query aspects. However, query aspects without user feedback have limitations to improve search engine relevance. Some works take user feedback into account and analyze the different clicked URLs of a query in user click-through logs directly, nevertheless the number of different clicked URLs of a query may be not big enough to get ideal results. However, their method does not work if we try to discover user search goals of one single query in the query cluster rather than a cluster of similar queries. However, their method only identifies whether a pair of queries belong to the same goal or mission and does not care what the goal is in detail. A prior utilization of user click-through logs is to obtain user implicit feedback to enlarge training data when learning ranking functions in information retrieval. In our work, we consider feedback sessions as user implicit feedback and propose a novel optimization method to combine both clicked and unclicked URLs in feedback sessions to find out what users really require and what they do not care. One application of user search goals is restructuring web search results. There are also some related works focusing on organizing the search results. In this paper, we infer user search goals from user click-through logs and restructure the search results according to the inferred user search goals.

6 CONCLUSION

In this paper, a novel approach has been proposed to deduce user search goals for a query by clumping its feedback sessions represented by pseudo documents. First, we introduce feedback sessions to be analyzed to deduce user search goals rather than search results or clicked URLs. Both the clicked URLs and the unclicked ones before the last click are considered as user implicit feedbacks and taken into account to construct feedback sessions therefore, feedback sessions can reflect user information needs more efficiently. Second, we map feedback sessions to pseudo documents to approximate goal texts in user minds. The pseudo documents can enrich the URLs with additional textual contents including the titles and snippets. Based on these pseudo documents, user search goals can then be discovered and depicted with some keywords. Finally, a new criterion CAP is formulated to evaluate the performance of user search goal inference. Experimental results on user click through logs from a commercial search engine demonstrate the effectiveness of our proposed methods.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval.ACM Press, 1999.

[2] R. Baeza-Yates, C. Hurtado, and M. Mendoza, “Query Recommendation Using Query Logs in Search Engines,” Proc. Int’l Conf. Current Trends in Database Technology (EDBT ’04), pp. 588-596, 2004.

[3] D. Beeferman and A. Berger, “Agglomerative Clustering of a Search Engine Query Log,” Proc. Sixth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’00), pp. 407-416, 2000.

[4] S. Beitzel, E. Jensen, A. Chowdhury, and O. Frieder, “Varying Approaches to Topical Web Query Classification,” Proc. 30th Ann. Int’l ACM SIGIR Conf. Research and Development (SIGIR ’07), pp. 783-784, 2007.

[5] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, “Context-Aware Query Suggestion by Mining Click-Through,” Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’08), pp. 875-883, 2008.

[6] H. Chen and S. Dumais, “Bringing Order to the Web: Automatically Categorizing Search Results,” Proc. SIGCHI Conf. Human Factors in Computing Systems (SIGCHI ’00), pp. 145-152, 2000.

[7] C.-K Huang, L.-F Chien, and Y.-J Oyang, “Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs,” J. Am. Soc. for Information Science and Technology, vol. 54, no. 7, pp. 638-649, 2003.

[8] T. Joachims, “Evaluating Retrieval Performance Using Click through Data,” Text Mining, J. Franke, G. Nakhaeizadeh, and I. Renz, eds., pp. 79-96, Physica/Springer Verlag, 2003.

[9] T. Joachims, “Optimizing Search Engines Using Click through Data,” Proc. Eighth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’02), pp. 133-142, 2002.

[10] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, “Accurately Interpreting Click through Data as Implicit Feedback,” Proc. 28th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’05), pp. 154-161, 2005.

[11] R. Jones and K.L. Klinkner, “Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs,” Proc. 17th ACM Conf. Information and Knowledge Management (CIKM ’08), pp. 2008.

[12] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval.ACM Press, 1999.

[13] R. Baeza-Yates, C. Hurtado, and M. Mendoza, “Query Recommendation Using Query Logs in Search Engines,” Proc. Int’l Conf. Current Trends in Database Technology (EDBT ’04), pp. 588-596, 2004.

[14] D. Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log," Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '00), pp. 407-416, 2000.

[15] S. Beitzel, E. Jensen, A. Chowdhury, and O. Frieder, "Varying Approaches to Topical Web Query Classification," Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development (SIGIR '07), pp. 783-784, 2007.

[16] H. Chen and S. Dumais, "Bringing Order to the Web: Automatically Categorizing Search Results," Proc. SIGCHI Conf. Human Factors in Computing Systems (SIGCHI '00), pp. 145-152, 2000.

[17] C.-K Huang, L.-F Chien, and Y.-J Oyang, "Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs," J. Am. Soc. for Information Science and Technology, vol. 54, no. 7, pp. 638-649, 2003.