Automatic Tag cloud Realization of web search results using Incremental Clustering by Directions Algorithm

Maheswari.R
PG Scholar
Department of Computer Science and Engineering
National Engineering College
Kovilpatti, India

D. VijayaKumar
Assistant Professor,
Department of Computer Science and Engineering,
National Engineering College,
Kovilpatti, India.

ABSTRACT

This paper concerns a subject oriented clustering algorithm for clustering the web search results obtained from web search engines. The algorithm is designed to create a list of words which serve as suggestions for users of search engines to modify their current search query. When a user executes a query, the algorithm shows potential directions in which the search can be continued. In this algorithm, the computational complexity of selecting different subjects is reduced by interpreting the set of all web page representations and their distances between them as a complete weighted graph. An incremental clustering approach has been proposed which avoids the process of reclustering the web pages. A list of suggested words or the clusters is presented to the user in the form of a tag cloud in which terms are arranged in a radial manner to increase the relevancy of search process.

Keywords

Clustering; informatin reterival; interactive query expansion; search methods, Direction

1. INTRODUCTION

With the increase in information on the World Wide Web it has become difficult to find the desired information on search engines. The size of publically indexed World Wide Web has probably surpassed 23.99 billion pages in July 2010 [1] and as yet growth shows no sign of leveling off. As a result prevailing search engines like Yahoo, AltaVista, Google return thousands of matches in response to a user query. Web users have to go through the list and examine the titles and (short) snippets sequentially to identify their required results. Since, it is assumed that users do not formulate search queries using the best terms.

Search engines support users in the process of preparing queries. One of the most common methods is presenting suggestions for finishing queries while they are being typed. In another method, a set of terms and phrases related to the query is presented to the user apart from a list of found Web pages.

Query expansion is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. In the context of web search engines, query expansion involves evaluating a user's input (what words were typed into the search query area and sometimes other types of data) and expanding the search query to match additional documents. Thus, it is useful in reducing query/document mismatch.

This paper focuses on a novel approach to facilitating users in forming queries. The approach is based on clustering-by Directions (CBD) algorithm. This paper presents an enhanced version of this algorithm. The enhancements include an incremental clustering of data and improved version of the interface and modifications in the algorithm, which reduce its computational complexity. This paper also presents the analysis of the computational complexity of the algorithm, new experimental results, and the evaluation of the algorithm.

2. RELATED WORK

A technique which supports users of search engines in narrowing the search is clustering search results. Web pages of similar subject are assigned to the same cluster. Users can view contents of clusters in order to find the group of the most relevant Web pages. In this technique, it is required to assign labels to clusters so that users know what type of Web pages each cluster contains. Ramesh Singh et al[2] introduced a clustering technique based on suffix Tree clustering algorithm. This approach treat the base cluster labels as candidate labels for the final cluster, and use the scoring to select the highest ranked candidate as the final label.

Another method, called Semantic web search results clustering which uses Lingo and Word Net is described by Ahmed Sameh in [3] which finds the synonyms of frequent words in the WordNet database, and adding the synonyms to the pool of frequent terms that comprise the cluster label candidates. Moreover, Pantel and Lin presented clustering-by-committee algorithm which has elements of the *k*-means algorithm [4].

Another technique of clustering web pages is based on fuzzy method [5] where Feature clustering of data is performed to reduce the dimensionality of feature vectors. The words in the feature vector of a document set are grouped into clusters based upon a 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. Elias Iosif [6] proposed a technique in which it clusters the web pages based on computing semantic similarity between words or terms. Relevant Web documents are downloaded via a Web search engine and the contextual information of words of interest is compared. There are also techniques based on algorithms for interactive query expansion. They present to a user a list of concepts related to a given query. Such a method was presented by Fonseca *et al.* [7]. Concepts related to the query were extracted from logs of past queries. Crabtree *et al.* [8] focused on analyzing different aspects of terms used in queries to find meaningful groups of refinements. It enabled users to refine ambiguous queries. White and Marchionini [9] analyzed the effectiveness of query expansion. Another method of finding terms related to the given query is based on thesauri, which are defined dictionaries of synonyms and related words. Such a technique was included in the query expansion method proposed by Li *et al.* [10]. A different approach to query expansion is based on personalization. The expansion algorithm can create user's profile, and it can

generate expansion terms with regard to this profile. Such techniques were proposed by Chirita *et al.* [11]. There are also techniques for improving the search, which take advantage of metadata and association rules [12].

3. MOTIVATION

While keyword search empowers ordinary users to search vast amount of data, delivering relevant results for keyword queries is challenging. Query Expansion is the process to retrieve relevant data. When users of search engines form queries, they put themselves somewhere in the space of knowledge of the Internet. They define to some extent their areas of interest. The CBD algorithm shows possibilities of moving from there. It is designed to make narrowing suggestions. It is developed to inspire users how to add words to their actual query.

4. MAIN IDEA OF CBD ALGORITHM

The aim of CBD algorithm is to divide search results into clusters or to show related terms for the query. In order to achieve this, the CBD algorithm first selects different directions, and afterward, it determines how the user can move in each direction. It is done regardless if there are subsets of Web pages with the similar subject or not.

In the algorithm, the location in the space where the user is placed after executing the query is determined by the response of the query. It is not defined by the query itself. The query is a user's request, whereas Web pages found by a search engine determine where the user is located. The location is interpreted as an average Web page found for a given query. Such a Web page is the one which would appear after adding up all analyzed Web pages and dividing them by their quantity.

The functionality available by the algorithm presented in this paper does not interfere with the process of finding pages on the Internet and it does not change the order in which pages are listed by a search engine. It is assumed that this algorithm is only an extension to an existing search engine.

The algorithm can be recursive. When a query modified once by suggestions presented by the algorithm is still not accurate enough, then a new list of suggestion can be prepared. It can be calculated by the same algorithm.

5. STRUCTURE OF CBD ALGORITHM

The CBD algorithm consists of the following steps:

1) Calculating vectors which represent Web pages and distances between these vectors;

2) Selecting different Subjects;

3) Assigning Web pages to directions and selecting terms which represent Subjects;

4) Showing terms on the user interface.

The initial version of the algorithm is presented in [13]. This section describes the algorithm and clarifies its structure.

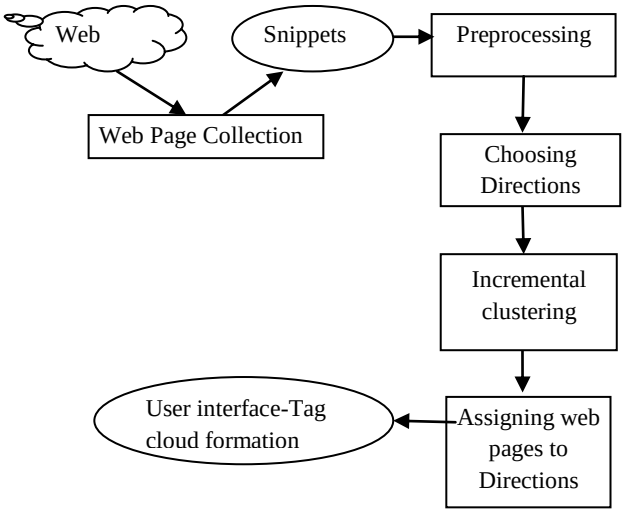

Fig 1. Overview of the System

Furthermore, it presents a new type of interface to the algorithm. The algorithm works as an extension to a search engine. It analyzes Web pages found by a search engine for the query given. If there is large number of Web pages listed, the algorithm analyzes only those which are on the top of the list.

5.1 Web Page Representations

It is necessary to prepare representations of Web pages because Web pages are written in natural language, which is not an appropriate form for the algorithm. These representations would be sets of values describing contents of Web pages.

In order to make representations of Web pages, the CBD algorithm uses a modification of the classic version of vector space model (VSM) [14]. VSM is designed to make representations of documents in a given set of them. Each of the documents is represented by a set of term–weight pairs. Terms are words which occur in documents apart from high-frequency words like conjunctions and pronouns. In the classic version of VSM, weights are calculated on the basis of *tf–idf* weighting. It takes into account two factors. The first is the number of occurrences of the word in the document (*term frequency*). The second concerns the number of documents in which the word occurs (*inverse document frequency*).

The set of documents on which the CBD algorithm operates is in fact the set of all Web pages available on the Internet. The algorithm analyzes only a small part of them; however, to calculate parameters like *idf*, statistics for all Web pages are needed. Because of a large number of Web pages, it is problematic. This parameter is replaced in CBD algorithm with another one, which is equal to the inverse of the logarithm of frequency of word occurrences in language. It emphasizes the significance of rare words. Parameter *tf* is used in the algorithm in the same way as in the classic version of VSM.

In the CBD algorithm, the words used in the query and their plural forms are excluded from representations like high frequency words. Moreover, the weights of terms are normalized. Each weight is divided by the square root of the sum of squares of all others weight referring to the same Web page.

Each Web page is represented by pairs: $\{(w1, \phi1), (w2, \phi2), \ldots, (wn, \phi n)\}$, where wk is a word and ϕk is a weight between

zero and one. For each word *wk*, weight ϕk is equal to the value expressed by

$$\varphi_k = \left(\frac{p_k}{\log(Q_k)}\right) \Big/ \left(\sqrt{\sum_{i=1}^{n}\left(\frac{p_i}{\log(Q_i)}\right)^2}\right)$$

where *pk* is the *term frequency* parameter that is equal to the number of occurrences of the word *wk* on the Web page, and *Qk* is the frequency of occurrences of word *wk* in language. Pairs calculated in this way form a representation of a Web page. They also define a vector. This vector is placed in a space in which every dimension corresponds to a different word. In CBD algorithm, distances between each two vectors are calculated. These distances are defined with the use of cosine similarity [5]. Its formula is presented in

$$sim(v_a, v_b) = \left(\sum_{i=1}^{n}\varphi_{ai}\varphi_{bi}\right) \Big/ \left(\sqrt{\sum_{i=1}^{n}\varphi_{ai}^{\,2}}\sqrt{\sum_{i=1}^{n}\varphi_{bi}^{\,2}}\right) \quad (2)$$

where *va* and *vb* are the representations, function sim expresses the similarity between two representations, ϕax stands for the weight ϕx calculated for the representation *va* with the use of (1), and ϕ_{bx} is the corresponding weight in the representation *vb*. The sim function corresponds to the cosine of the angle between vectors. Interpreting the similarity as the cosine of the angle instead of the angle itself makes calculations easier and it is the most popular method. Furthermore, it is assumed that weights assigned to words which are not included in a representation are equal to zero. It is significant when a word is included in one representation, and it is not included in the other one. In the CBD algorithm, vectors are normalized; thus, the expression in the last brackets is always equal to one.

5.2 Choosing Directions

One of the most important problems in the algorithm is to select different directions in which the search can be continued. In order to achieve this, the algorithm chooses directions in VSM which aim at different groups of Web page representations. Each direction is pointed out by a vector placed in the space.

Let us denote the number of chosen directions as *D* and the set which contains all analyzed representations as *W*. From the set *W*, subsets with *D* elements can be extracted, and in every subset, the sum of distances between each two representations can be calculated. In the initial version of the algorithm, in order to choose representations serving as directions, the subset in which the sum of distances is the greatest is selected. If there are many subsets with the greatest sum, any of them is chosen. Those representations included in the selected subset serve as vectors pointing out directions. This way of selecting representations can be interpreted as finding a subset with the greatest value of function *f* expressed by

$$f(v_{i1},.....v_{iD}) = \sum_{v_{ip}\varepsilon\{(v_{i1},.....v_{iD})\}} \; \sum_{v_{ir}\varepsilon\{(v_{i1},.....v_{iD})\}} sim(v_{ip}, v_{ir})$$

where *vi*1, . . . , *viD* are the representations included in a subset, and function *sim*(*vip, vir*) stands for the distance between representations. In Fig. 1, this approach is presented for 2-D space where four different directions are chosen. Points stand for representations. They are selected in such a way that the sum of distances *a*, *b*, *c*, *d*, *e*, and *f* is the greatest.

It needs to be determined how many directions should be chosen in the algorithm. The algorithm makes it possible to select various numbers of directions which correspond to various areas of interest. However, the higher this number is, the more detailed are the subjects pointed out by the directions. This leads to a problem that, for a high number of directions, it may be difficult to notice the relation between some subjects pointed out by the directions for the given query.

For example, when 16 directions are selected for the query *document*, one of the found directions is concerned with women's rights. In fact, some Web pages listed as a result of the query *document* are really associated with this subject. Thus, the algorithm recognizes it as a possible direction in which the search can be continued. However, its relevance to the query may be unclear for the users who are not interested in this topic.

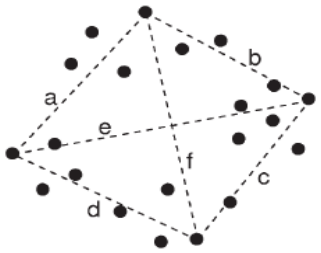

Fig 2. Selecting points for which sum of the distances is greatest

Nevertheless, such cases occur for numbers of directions which are not expected to be chosen in the algorithm due to another factor that affects the number of selected directions. This other factor is the user's interface. It is hard to present a large number of directions to a user. Hence, it is expected that only about two to ten directions are selected in the algorithm.

Unless the number of directions is high, it does not influence the quality of the results. In this paper, a new type of interface is introduced. The number of directions which best fits this interface is equal to six. Moreover, a survey showed that this number is the most preferable to users.

5.*3 Assigning Web Pages to Directions*

After choosing directions, the ICBD algorithm is proposed to assign web pages to directions. It is an incremental clustering approach. Traditionally, clustering took a dataset, processed it and produced a result set. If the dataset was changed, the entire dataset had to be reclustered from scratch. This reclustering could be costly in terms of processing time.

Incremental clustering is one technique developed to avoid this. It is based on the idea that it is possible to process documents one at a time and assign them to a cluster, without significantly affecting the state of the existing clusters. In short, this means documents that are already clustered do not have to be reclustered when a new document is added to the dataset.

Here, instead of assigning the cluster of direction for which the distance to the vector of direction is the smallest, adds it where possible to the cluster that has the most similar centroid vector to the document. The idea with this approach is that the cluster with the highest similarity to the document

will have the greatest number of similar documents in it and would be the best cluster to place the document in.

For this enhanced algorithm, the centroid vector is calculated by determining the average weight of every term that is in at least one document vector present for that particular cluster. Mathematically, the centroid vector (CV) is defined as the following:

$$cv = \frac{1}{|s|}\sum_{d \in s} \overline{d} \qquad (4)$$

where *S* is the set of documents, $\overline{d}$ is the term vector representation for document *d* in the cluster.

After selecting directions and assigning Web pages to clusters of directions, the algorithm selects terms representing directions. In order to choose these terms for each direction, a vector representing direction is created. It is built by adding up all vectors representing Web pages that the cluster contains. The vector resulting from this operation corresponds to a set of Term–weight pairs similar to vectors representing Web pages.

The terms which have the greatest weight assigned in the vector representing directions are presented to the user as suggestions for modification of the query. The terms which occur in less than every fifth document within the cluster are omitted as they are not representative.

5.4 Interface

Terms are presented to the user in a unique version of a tag cloud designed for the needs of the algorithm. A tag cloud is a form of presenting keywords usually manually assigned to pieces of information like pictures, articles, or video clips [17]. In this paper, a new type of tag cloud is introduced. The structure of the tag cloud is such that terms representing the same direction are placed in one line and these lines are put in a radial arrangement. In the horizontal lines, terms are placed alternately in two rows to reduce the width of a tag cloud.

Fig 3. Tag cloud created for the Query "Computer"

Terms within each line are displayed with different font sizes, depending on their importance in the cluster of direction. The higher it is, the greater is the font in which the term is displayed. Terms in smaller font are displayed closer to the center of the tag cloud. A sample tag cloud for the query *car* is shown in Fig. 3.

The CBD algorithm distinguishes between singular and plural forms of words as search engines show different results, depending on the forms of words used in the query. In the presented interface, a user needs to click on the term in order to add it to the current query. However, it does not automatically initialize the search process. It is expected that the user can add many words from the tag cloud before requesting the search engine to perform the search.

6 EXPERIMENTAL RESULTS

Experiments with the CBD algorithm were performed using the Google search engine. For each query executed in the experiment, the algorithm analyzed the first hundred Web pages listed. An example considers searching for the web pages related to the term *Airplane*. When the algorithm was employed, the clusters were made and vectors representing these clusters were calculated. Table 1 presents values from all four vectors representing clusters. The tables contain words with the greatest values assigned in these vectors.

On the list of suggested words which are selected from clusters there are some words which seem to be mismatched, for example *RC*. However, *RC* stands for *radio controlled*. Probably, if users of a search engine looking for such types of airplanes included the acronym RC in their queries instead of the words radio controlled, they would receive better results. The execution of clustering by directions algorithm showed that this acronym is more characteristic than words *radio controlled* or it is more often used on web pages concerning this subject.

TABLE I. **WORDS WITH GREATEST VALUES ASSIGNED TO QUERY AIRPLANE FOR CLUSTERS1 AND 2.**

Cluster 1		**Cluster 2**	
Word	***Value Assigned***	***Word***	***Value Assigned***
All	0.5458563	Boeing	1.3119233
Rc	0.5305351	New	1.2526884
News	0.5216494	Paper	0.6115229
Online	0.4643512	Aircraft	0.6034160
New	0.4521498	All	0.5428369
Brittanica	0.3889384	Employee	0.5410535
Employee	0.3885748	Sale	0.4064602
Aviation	0.3851739	Aviation	0.3885748
Videos	0.3839032	Piper	0.3842978
Dictionary	0.3631988	combat	0.3717858
Store	0.3525352	Movie	0.3695160
Fingerhut	0.3520118	site	0.3410405

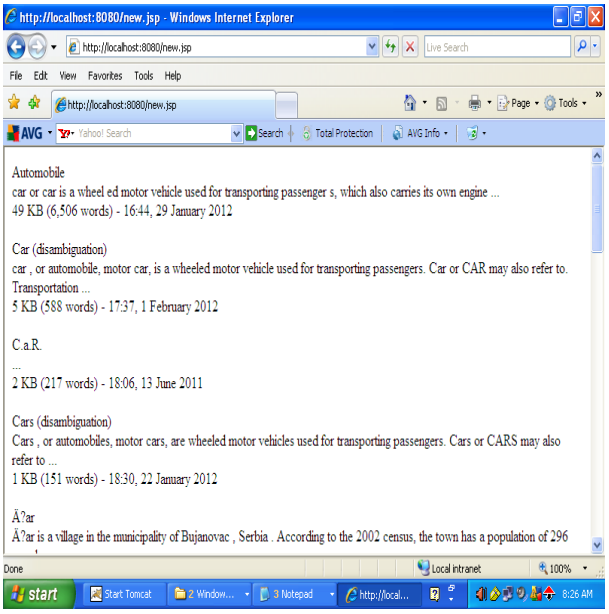

Fig 4. Retriving only text for the search query "Car"

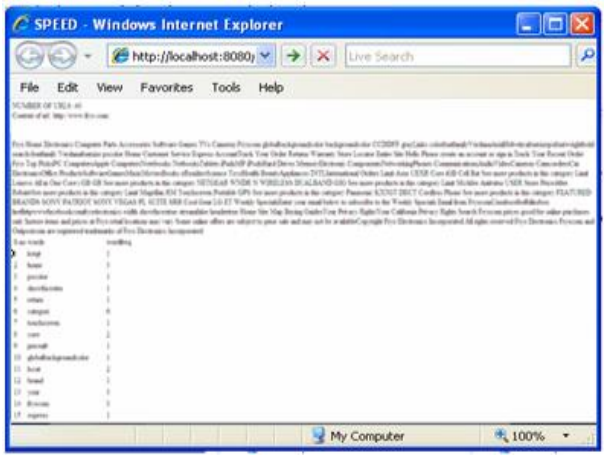

Fig 5. Calculating word Frequency

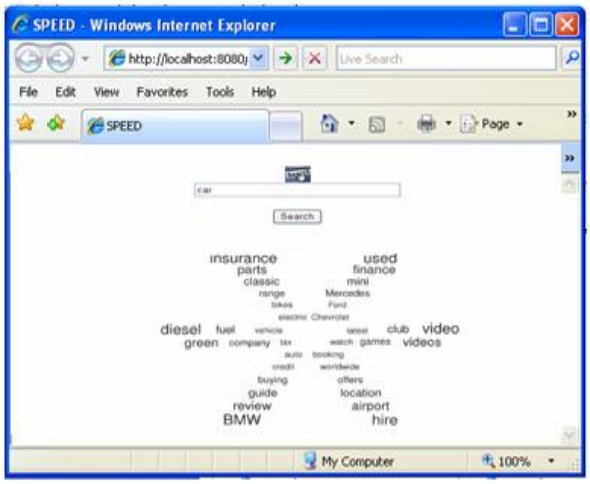

Fig 6. Tag cloud creation for the search query "Car"

In order to evaluate the CBD algorithm, three measures were used: Precision K at N, mean average precision (MAP), and normalized discounted cumulative grain (NDCG) [18]. These measures are designed to evaluate search results. In the evaluation presented in this paper, the CBD algorithm is considered as the one which searches for related terms. In search engines, Web pages found for a given query are ranked. In the CBD algorithm, the rank of a term stands for its importance in a tag cloud.

Measures are calculated for a tag cloud in which six directions and six terms in each direction are presented. As 36 terms are displayed, the precision K at N measure is applied in the form of K at 36 measure. In the tag cloud, terms in each direction are ranked independently. Thus, while calculating MAP and NDCG measures, it is assumed that each direction contains an independent list of found terms. The levels of the term relevance were considered to be equal to either one or zero.

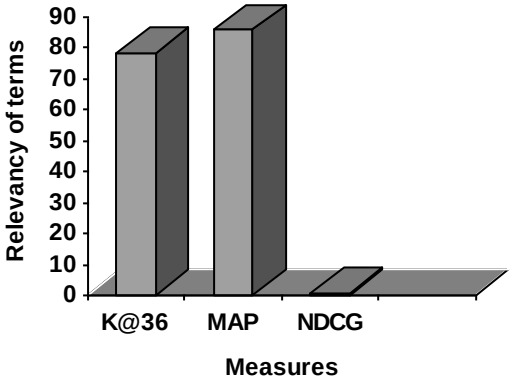

Fig 7. Performance of CBD algorithm for Extensive queries

According to the values of MAP and NDCG, the majority of irrelevant terms had lower ranks. The algorithm calculated results within ca. 20 s on 2.80-GHz Intel Celeron IV. It does not include the time needed to download text of Web pages, which was about 2 s for each Web page. If the algorithm is integrated with a search engine, it does not need to be executed every time when any user executes a query. The results of the algorithm for different queries can be stored, and they can be only updated.

7 CONCLUSION

The tag cloud which is a result of an execution of clustering by directions algorithm shows different possibilities of changing the query. The purpose of the algorithm is achieved. Apart from this, the algorithm can be used for clustering the search results. The range of potential applications of the CBD algorithm is very wide. In particular, it can be used to support finding information in e-learning systems. The CBD algorithm is an attractive alternative to other methods designed to simplify the search process. It can become a commonly used extension to search engines and other applications designed for finding information. By using ICBD, the reclustering of web pages is reduced which leads to increase in performance of the Query results. The time for searching the internet is also reduced.

8 REFERENCES

[1]http://www.worldwidewebsize.com/index.php?lang=N

[2] Ramesh singh , Dhuruv Dhingra, and Aman Arora, "SCHISM-A web search Engine using semantic

taxonomy", IEEE potentials Oct 2010 volume:29, NO:5, pp.36-40.

[3] Ahmed Sameh and Amar Kadray, " Semantic web search results clustering using Lingo and Word Net", International Journal of Research and Reviews in Computer Science Vol. 1, No. 2, June 10, pp.71-76.

[4] P. Pantel and D. Lin, "Document clustering with committees," in Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, Aug. 2002, pp. 199–206.

[5] Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee, "A Fuzzy Self Constructing Feature Clustering Algorithm for Text Classification", IEEE transactions on knowledge and data engineering, Volume. 23, NO:.3, March 2011, pp.335-349.

[6] Unsupervised Semantic Similarity Computation between Terms Using Web Documents, Elias Iosif, IEEE transactions on knowledge and data engineering, vol. 22, no. 11, Nov 2010, pp. 1637 -1647.

[7] B. M. Fonseca, P. Golgher, B. Pôssas, B. Ribeiro-Neto, and N. Ziviani, "Concept-based interactive query expansion," in Proc. 14th ACM Conf. Inf. Knowl. Manag., Bremen, Germany, Oct. 2005, pp. 696–703.

[8] D. Crabtree, P. Andreae, and X. Gao, "Understanding query aspects with applications to interactive query expansion," in Proc. IEEE/WIC/ACMInt. Conf. Web Intell., Silicon Valley, CA, Sep. 2007, pp. 691–695.

[9] R. W. White and G. Marchionini, "Examining the effectiveness of realtime query expansion," Inf. Process. Manage., vol. 43, no. 3, pp. 685–704, ay 2007.

[10] J. Li,M. Guo, and S. Tian, "A new approach to query expansion," in Proc. 4th IEEE Int. Conf.Mach. Learn. Cybern., Guangzhou, China, Aug. 2005, vol. 4, pp. 2302–2306.

[11] P. A. Chirita, C. S. Firan, andW. Nejdl, "Personalized query expansion for the Web," in Proc. 30th ACM SIGIR Conf. Res. Develop. Inf. Retrieval,Jul. 2007, pp. 7–14.

[12] Y. Takama and S. Hattori, "Mining association rules for adaptive search engine based on RDF technology," IEEE Trans. Ind. Electron., vol. 54, no. 2, pp. 790–796, Apr. 2007.

[13] A. L. Kaczmarek, "Clustering by directions algorithm to narrow search queries," in Proc. IEEE Human Syst. Interaction Conf., Krakow, Poland, May 2008, pp. 689–694.

[14] V. Vinay, K. Wood, N. Milic-Frayling, and I. J. Cox, "Comparing relevance feedback algorithms for Web search," in Proc. WWW: Special Interest Tracks Posters 14th ACM Int. Conf. World Wide Web, Chiba, Japan, May 2005, pp. 696–703.

[15] M. Okabe and S. Yamada, "Semisupervised query expansion with minimal feedback," IEEE Trans. Knowl. Data Eng., vol. 19, no. 11, pp. 1585–1589, Nov. 2007.

[16] G. Salton and C. Buckley, "Term-weighting approaches in automatic textretrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.

[17] C. Seifert, B. Kump, W. Kienreich, G. Granitzer, and M. Granitzer, "On the beauty and usability of tag clouds," in Proc. 12th IEEE Int. Conf. Inf. Vis., London, U.K., Jul. 2008, pp. 17–25.

[18] E. Agichtein, E. Brill, and S. Dumais, "Improving Web search ranking byincorporating user behavior information," in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, Aug. 2006, pp. 19–26.

[19] M. Wu, J. She, G. Zeng, and Y. Ohyama, "Internet-based teaching and experiment system for control engineering course," IEEE Trans. Ind. Electron., vol. 55, no. 6, pp. 2386–2396, Jun. 2008.

[20] M. T. Restivo, J. Mendes, A. M. Lopes, C. M. Silva, and F. Chouzal, "A remote laboratory in engineering measurement," IEEE Trans. Ind. Electron., vol. 56, no. 12, pp. 4836–4843, Dec. 2009.