Artif Intell Rev (2006) 26:11–21
DOI 10.1007/s10462-007-9040-7

An attempt to enhance performance in user session based information retrieval

Michael Smullen · Colm O'Riordan

Published online: 21 September 2007

Abstract The ongoing surge in the amount of online information has made the process of accurate retrieval much more difficult. Providers of information retrieval systems have come under a lot of pressure to improve their techniques to cater for the modern user. Conventional systems are often limited as they fail to understand the true search intent of the user. This is usually a result of both poor query formulation by the user and an inability of the search engine to process the query adequately. In this paper, an approach is presented that attempts to learn a user's short-term interests through the clustering of their search results. A profile is maintained for each user to assist in the process of context resolution for a given query. The details of such an approach and experimental results to evaluate its effectiveness are presented in this paper.

Keywords Context · Query re-formulation · Information retrieval

1 Introduction

User context is one of the key components for accurate retrieval of information. Current systems tend to ignore this factor and the majority of systems return a uniform set of query results irrespective of the context in which the query was submitted (Sugiyama et al. 2004).

Researchers have examined a number of techniques to improve upon standard query performance. Most of this research has focused on relevance feedback techniques and collection analysis approaches (Rocchio 1971; Xu and Croft 1996), whereby the query is modified over a number of iterations to obtain more accurate results. The principal drawback with this approach is the need for the user to supply feedback explicitly to the system. The majority of users are not willing to provide any feedback and of those who are, most will only provide a limited amount (Dumais et al. 2003).

In this paper, a method to learn, in an implicit and efficient manner, a user's interests and preferences is proposed. This is achieved through clustering of the search results over the

M. Smullen · C. O'Riordan (✉)
Department of Information Technology, NUI, Galway, Ireland
e-mail: colm.oriordan@nuigalway.ie

duration of a short-term user session. Once the initial query is submitted, the profile is created which is updated through all subsequent queries. Following the initial query, the system will use this profile in an attempt to provide more relevant results for all future queries.

The following example highlights some of the potential benefits of this approach: a user submits the initial query *computer software* to begin the session. A number of clusters can be created and stored as part of the user's profile. These clusters represent distinct concepts relating to *computer software*. In a subsequent query as part of the same session the user submits the query *Python*. Typically, one would imagine documents pertaining to topics such as the Python programming language and pertaining to snakes as part of the returned set. However, by taking account of the user profile as part of the retrieval process, one can ascertain that this user previously submitted a query related to computer programming. Therefore it would make more sense to score documents related to programming languages higher than those related to snakes. This represents a simple example of how maintaining a profile can assist in the retrieval process.

One of the main thrusts of this research is to generate automatically user sessions in order to undertake empirical analysis of the proposed system and algorithms. This is achieved through the use of available standard test collection which comprise documents, queries and relevance judgements. Similar queries are clustered together to form concepts or interests. By using these clusters of queries to represent user interests, a number of user sessions can be created. These user sessions can be generated in a number of ways and facilitate the modelling of different user types—*easy* user, *moderately difficult* user and *difficult* user. Each user type has its own values for a number of different attributes such as the number of topics in which that type of user is interested in and the probability and frequency that this type of user will change between query topics/interests in a given session.

This paper presents an approach to learn a user profile to enhance the search process. A novel manner to create user sessions from a standard test collection is also presented. The following section presents related work in the area of query enhancement with a particular emphasis on expansion techniques and clustering. In Sect. 3, the system design and experimental setup are discussed. Section 4 discusses an approach to modelling user sessions and profile creation. In Sect. 5, experimental results are presented to show the effectiveness of the technique. Section 6 presents some results to justify the use the clustering method for the creation of search sessions. Conclusions are presented in Sect. 7.

2 Related work

Query enhancement has long been a topic of research in information retrieval. Query expansion (Xu and Croft 2000) has been shown to improve the effectiveness of ranked retrieval by automatically adding terms to a query. The original query is run using conventional methods. Terms are then extracted from the top ranking documents for addition to the original query. Terms chosen for expansion are typically those that occur most frequently.

The following are a number of approaches adopted by researchers in this area.

2.1 Query expansion using previous return sets

In (Fitzpatrick and Dent 1997) past user queries are used to enhance automatic query expansion. Results of past queries are used to create affinity pools, from which expansion terms are chosen. The top ranking documents from up to three previous queries that are similar to the current query are used to form the affinity pool. Using a *tf-idf* (Salton and MacGill 1983)

Table 1 Attributes of the CACM test collection

Collection	Num Docs	Terms	Terms/Docs	Queries	Terms/Query	Relevants/Query
CACM	3204	10,446	40.1	52	11.4	15.3

scoring mechanism, terms are then chosen for expansion of the current query. The end result was an improvement on relative average precision of 15% for the TREC-5 collection.

2.2 Implicit feedback measures

Most research on context-based information retrieval is focused around gathering contextual information through behavioural monitoring, environment detection, gesture recognition and perceptual evidence, such as eye tracking (Ruthven 2004) and other measures (Kelly and Belkin 2001). Such approaches, although potentially very effective, can be quite cumbersome and computationally expensive.

2.3 Concept hierarchies

Sieg et al. (2003) incorporate domain-specific concept hierarchies with an interactive query formulation process to produce a less ambiguous query. They modify the user's initial query based on their interaction with a modular concept hierarchy. Experiments are performed based on ambiguous words and associated documents for each word sense. Their results show higher precision is achieved for these ambiguous queries. The major cost here is the creation of the concept hierarchy as part of the pre-processing phase. This requires a form of document clustering, in this case the k-means clustering technique.

2.4 Clustering techniques

Clustering has been extensively used in information retrieval. Some of the IR domains in which it has proved to be beneficial include: query expansion—both relevance and local feedback techniques, creation of concept hierarchies for test collections and cluster-based browsing of search results.

In work by Lu et al. (1996) the top-ranked documents are clustered and the singleton clusters removed. The intention is to increase the concentration of relevant documents and thus promote more effective query expansion. Buckley et al. present an approach wherein the documents in the return set are clustered, and the cluster that best matches the query is used for query expansion (Buckley et al. 2000).

3 System design and experimental setup

3.1 Document collection

The experiments in this paper are performed using the CACM (articles published in the Communications of the ACM from 1958 to 1979) test collection. The collection contains 52 queries together with their relevance judgements allowing for a formal evaluation of any obtained results. Details pertaining to the collection are presented in Table 1.

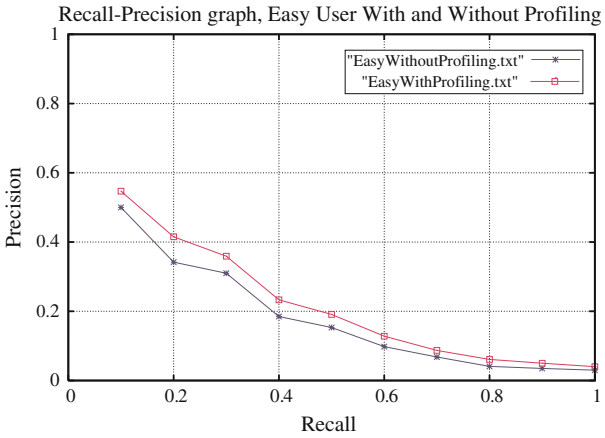

Fig. 1 Indexing strategy overview

3.2 Collection processing

A number of pre-processing steps are needed before any experiments can be undertaken. This involves parsing of the text, performing stop-word removal, stemming and indexing the remaining terms in the collection.

The text parser is implemented by means of a finite state machine (Baeza-Yates and Ribeiro-Neto 1999). Stop-word removal requires indexing of the stop-words and a comparison of each word in the document collection against this index. Any words contained in the index are removed. Stemming is acheived using the Porter algorithm (Porter 1997).

The final step is to index the remaining terms in the collection. This is achieved by using a nested binary search tree as illustrated in Fig. 1. Nodes in the outer tree represent each term in the collection together with global statistics such as the number of occurrences of this term, inverse document frequency, the number of documents containing the term and language modelling probabilities. The inner tree contains document occurrence information. The tree is sorted on the document identifier and each node stores information regarding term frequency (in this particular document), offset of occurrence and probability statistics.

3.3 Retrieval model

The vector space model of retrieval is used. Documents and queries are represented as t-dimensional vectors of the form:

$$\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, w_{3,j}, ..., w_{t,j}) \tag{1}$$

where $\overrightarrow{d_j}$ is a document or query and $w_{i,j}$ is the weight of term i in document j. Weights are assigned to terms in the collection through the following formula:

$$w_{i,j} = f_{i,j} \times idf_i \tag{2}$$

The vector space model calculates the degree of similarity between a document and a query as the correlation between the two vectors $\overrightarrow{d_j}$ and $\overrightarrow{q}$. This can be quantified as the cosine of the angle between the two vectors

$$\text{sim}(d_j, q) = \frac{\overrightarrow{d_j} \bullet \overrightarrow{q}}{|\overrightarrow{d_j}| \times |\overrightarrow{q}|} \tag{3}$$

4 Representing user sessions

The notion of a user session is necessary in order to examine algorithms for implicitly detecting user interests. The CACM collection contains once-off queries together with their relevance judgements. This information alone is inadequate for modelling and experimenting with user sessions. It is also too demanding of resources to perform usability testing on the system.

Other researchers have generally tackled the problem of testing session-based retrieval in one of two ways:

1. Perform the time-consuming task of collecting query logs for individual users.
2. Tailor testing entirely toward query disambiguation using a designed set of queries. This detracts from the credibility of the results with regard to real-world scenarios.

Neither of the above methods are ideal.

When a session begins, the typical user will have one or more information needs. Common practice in these circumstances is for such a user to issue a series of queries to best represent these needs. Once this process begins, a number of factors come into play. The user may decide to shift between needs during the query process. It is also possible that the user will terminate the search at any given time possibly because they have had their information need satisfied. Questions arise regarding the best way to simulate these types of behaviour.

In this work, user sessions are automatically generated. The sessions generated are viewed as the behaviour of an unknown user issuing a series of queries to satisfy their information need. The CACM collection, (which forms the basis of the experiments in this paper) contains 52 natural language queries and their corresponding relevance judgements.

In order to build the notion of user sessions, the queries are clustered into a number of different topics. These clusters depict how the queries relate to one another and represent the building blocks of a user information need. The clustering process produced eight clusters with strong intra-cluster similarities between the queries. These act as eight different user interests.

The basis to forming a user session is to determine its key characteristics. The following variables are used:

- A user has a set number of topics in which he/she is interested
- It may be possible for a user to cycle between interests
- The probability that when a query related to a particular interest is issued, the user will move to another query as part of a new interest
- The level of similarity between interests when a user cycles between interests, and
- The probability that a user will terminate their session following each query issued.

For the purpose of the experiments presented in this paper, three different user types are modelled—*easy, moderately difficult* and *difficult* user. These are differentiated by the different values assigned to the user-session variables.

In order to define user types the following are specified:

1. The number of interests for the user—this will be high for *difficult* and *moderately difficult* users (two to four interests). An *easy* user will have only one interest.

Table 2 User session examples

User type	Num interests	Cycling	Query sequence
Easy	1	No	Q10, Q63, Q18, Q19, Q62
Mod diff	3	Yes	Q18, Q61, Q44, Q32, Q33, Q19, 63, Q40, Q43
Ext diff	4	Yes	Q9, Q8, Q28, Q19, Q4, Q62, Q26, Q63, Q37, Q10, Q7, Q18

2. Whether cycling between interests is active—this is where a user issues a query related to one interest and follows this with a query related to a different interest but will return to the original interest in a future query.
3. The degree of change between interests; when a *difficult* user changes interest then the interest least similar (most contextually different) to their current interest is selected. A *moderately difficult* user moves to a randomly chosen new interest.
4. The probability of a user terminating a session; in order to simulate fully real world scenarios we must cater for the possibility of a user terminating their search at any given moment. There is a 5% chance of a user terminating their session following a query. This is constant for all users.

Examples of sessions related to the three different user types are depicted in Table 2.

4.1 Maintaining the user profile

In general, user profiling may be carried out implicitly or explicitly. In this work, implicit profiling is carried out by means of clustering the top N search results for an individual query. These clusters are then viewed as topics of interest for this user and their centroid term vectors are maintained as part of the profile. This approach works well in most cases and has been adopted in the past, as part of local feedback (Xu and Croft 2000). For difficult queries however, there exists the possibility of topic drift.

The clustering algorithm adopted in this work is a one-pass method whereby documents are added to clusters if their similarity is within a pre-defined threshold (0.10).[1] Once all documents under consideration have been clustered, all or some of the clusters are maintained as part of the profile. Following experimentation and analysis, it was ascertained that maintaining all non-singleton clusters provided the best results. This is in keeping with the cluster hypothesis (Hearst and Pedersen 1996). This states that in general, relevant documents tend to be more similar to each other than to non-relevant documents. Therefore it is reasonable to assume that singleton clusters are less likely to be relevant to the current query. This is especially true where there exist a large number of relevant documents per query, as is the case with the CACM collection.

The user profile can restrict the number of clusters to be maintained (between 4 and 8). This range was chosen to keep a bound on the number of clusters to be maintained and consequentially constraints the associated computational expense. All clusters are time-stamped such that if the upper bound on the number of clusters is reached, the oldest cluster can be dropped from the profile. When a cluster is being added, it is possible that it has a high degree of similarity (≥ 0.4) (again chosen following empirical testing) with another cluster in the profile. In this circumstance, the two clusters are merged and the time-stamp updated.

Once a query is submitted it is compared with all clusters in the user's profile using the cosine similarity measure. If the maximum similarity is within the pre-defined threshold (set

[1] This value was chosen as following empirical analysis of different values, this performed well.

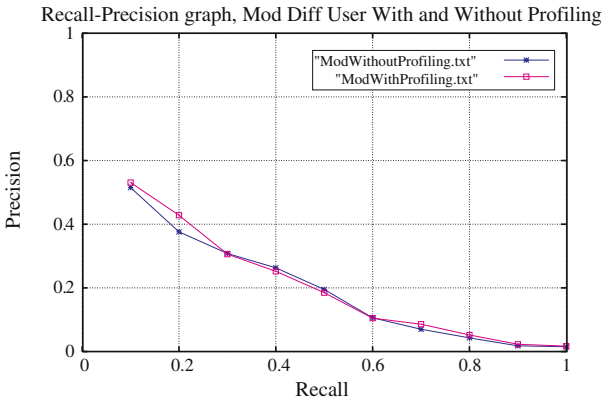

Fig. 2 Results for easy user

to be 0.2, following some experimentation with a range of values) then this cluster is used as part of the document scoring mechanism. It is assumed that a cluster falling within this threshold is somewhat representative of the context of the user's current query. If no cluster falls inside this threshold then the retrieval process proceeds as it would irrespective of user profiling.

When a cluster is identified as being relevant then the scoring mechanism reflects. The scoring mechanism is updated from the basic cosine similarity measure to the following approach:

$$\text{score}(d) = \text{sim}(Q, d) + \beta \times \text{sim}(Q, C) \times \text{sim}(d, C) \tag{4}$$

Where sim (x, y) represents the cosine similarity between x and y, Q represents the current query, C represents the cluster extracted from the user profile, d represents the document being scored, and β is a constant (set to 0.6 (determined empirically to give good performance)) reflecting how much credence is give to the profiling approach.

The above formula has a number of useful properties. It ensures that document d will receive the highest value if it is similar to both the current query and the cluster chosen as representative of the context of the current query. It also guarantees that documents that would not receive a high score using the cosine similarity metric are not exempt from being ranked highly in the final result.

5 Results

A number of experiments were conducted in order to measure the effectiveness of the proposed approach. In these experiments, a number of user sessions are created to simulate user behaviours; these use sessions can be categoried as *easy, moderately difficult* and *difficult*. The experiments compare, for each of these types of user, the performance of the proposed system using profiling and a system with no profiling in use.

The graph in Fig. 2 depicts the improvement garnered through the use of user profiling. The results obtained are from an *easy* user session. The results are as expected given the nature of the users i.e. a number of queries pertaining to a particular interest. By virtue of the fact that all queries relate to the same interest, it allows the profiling approach to be

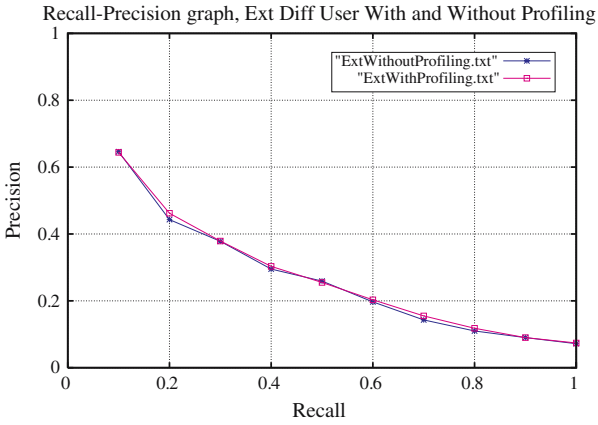

Fig. 3 Results for moderately difficult user

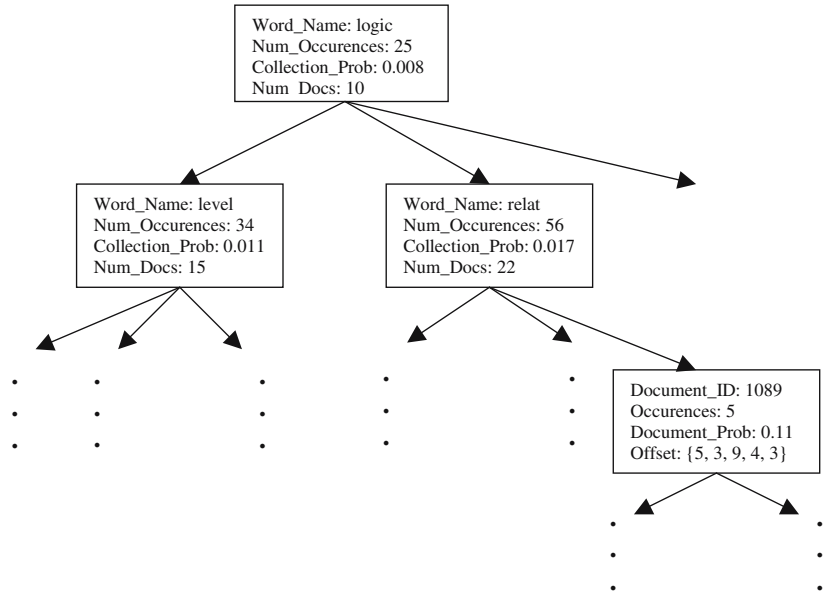

Fig. 4 Results for difficult user

maximally beneficial i.e. almost all queries issued will relate to previously stored concepts of interest for this user. The ratio of queries updated is also quite high for this user at 40%.

Results achieved for the *moderately difficult* user, as shown in Fig. 3, show a modest improvement over the baseline result. The decrease in improvement in comparison to the *easy* user can be attributed to the larger number of topics and associated changing between topics for these users. Furthermore, there will be a smaller ratio of queries updated (25%). Again, this is due to the extra interests displayed by this user type and detracts from the overall improvement.

For *difficult users*, there is an increased prevalence of changing between interests and hence it is more difficult for the system to improve performance for these users; the result

in Fig. 4 reflects this fact (almost identical to baseline result). Again, the ratio of queries updated (20%) through profiling is reduced and this serves to reduce the overall effect of this approach.

The results imply that profiling is most effective when the user has a small number of interests and doesn't tend to shift behaviour to other interests. In other cases, the system is less effective. This can be attributed to both the reduced number of queries being modified and the increased difficulty in context resolution.

6 Evaluating clustering algorithm performance for user session creation

The experiments in this paper are focused on session-based retrieval. As discussed in Sect. 4, sessions are created to depict three different types of system user. The idea is that an *easy* user has a number of queries related to a single concept of interest. At the other end of the spectrum, the *difficult* user would issue queries related to a number of interests with random shifts between topics. Naturally, the *moderately difficult* user falls between these two extremes.

As discussed in Sect. 4, these sessions are fabricated by means of clustering system queries in an effort to establish query topics within the system. The idea is that queries within the same cluster refer to the same concept and as a result, documents relevant to one are also relevant to the other. Consider the case whereby an individual cluster of N queries represents an *easy* user session. If these queries are indeed conceptually similar then any information gleaned from the results of one query should positively transfer to another query within the same cluster. However, it is possible that two queries, which are closely related in query space (textually), refer to entirely different concepts. It is therefore readily apparent that the effectiveness of the approach is dependent on the correctness of the query clustering process. The purpose of this section is to determine the validity of the technique.

6.1 Clustering algorithm

The clustering algorithm used as part of user session creation is a one-pass clustering technique. All queries are represented as t-dimensional vectors and similarity is calculated by means of the cosine similarity metric. A threshold of 0.20 was used to determine whether or not a query is clustered. This technique, although providing the benefit of not having to decide on a fixed number of clusters (as with K-means), does have its shortcomings. Firstly, comparisons between queries are totally word-based and do not take advantage of any other sources of evidence. It can be argued that this could potentially create anomalies within the clusters. Furthermore, because the technique requires only a single pass, the efficiency of the approach is offset by a degree of inaccuracy, less evident in more intricate algorithms. In spite of this, there was enough evidence (there are only 52 queries with relevance judgements used in the CACM collection) to suggest that one-pass clustering would adequately suffice for the required purpose.

6.2 Clustering analysis

In order to evaluate analytically the efficacy of clustering queries a direct comparison between query similarity (word-based using cosine similarity) and their corresponding relevance judgements is required. In order to do so, two $N \times N$ matrices were created where N represents the number of queries under consideration. The first matrix contains similarity scores (cosine similarity) between each pair of queries in the corpus. The second is made up of similarity

scores for the corresponding answer sets. The following metric was used to evaluate the similarity between two sets of relevance judgements.

$$\text{Sim}(RelJ_i, RelJ_k) = \frac{|RelJ_i \cap RelJ_k|}{|RelJ_i| + |RelJ_k|} \tag{5}$$

where $RelJ_i$ is the set of relevance judgements for query i and $|RelJ_i|$ is the number of elements in the set $RelJ_i$.

If the clustering approach adopted is valid, then there should exist a strong similarity between each pair of vectors within the two matrices. To evaluate this, the cosine similarity scores between the 52 query and relevance judgement vectors was examined . This yielded the following results. Note that absolute similarity is represented as 1.0 with total dissimilarity having a score of 0.0. The maximum score obtained for an individual query was a value of 0.986. The minimum score was 0.791 with a mean score of 0.920.

These results illustrate a high correlation and highlight the effectiveness of clustering search results to extract concepts of interest within the CACM collection. They also justify the use of such an approach in order to automatically generate search sessions for the collection and moreover to incorporate the notion of user profiling as outlined in Sect. 4.1.

7 Conclusions

In this paper, we presented an approach to learn the short-term interests of users. We provided a detailed overview of query enhancement and user profiling techniques. Two key parts of our method are (1) clustering search results to ascertain concepts of interest for a user and (2) creation of user sessions to test the effectiveness of our method.

Initial results obtained from testing were promising and support use of our approach. Future work will require a detailed examination and optimisation of the parameters used in the profiling algorithm, use of explicit feedback and a shift towards a larger scale of experiments using larger test collections and query sets.

References

Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. ACM Press/Addison-Wesley.

Buckley C, Mitra M, Walz JA, Cardie C (2000) Using clustering and super Concepts within SMART: TREC 6. Inform Process Manage 36(1):109–131

Dumais S, Joachims T, Bharat K, Weigend A (2003) SIGIR 2003 workshop report: implicit measures of user interests and preferences. SIGIR Forum 37(2):50–54

Fitzpatrick L, Dent M (1997) Automatic feedback using past queries: social searching?. In: SIGIR '97: proceedings of the 20th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA, ACM Press, pp 306–313

Hearst M, Pedersen J (1996) Reexamining the cluster hypothesis: scatter/gather on retrieval results. In: Proceedings of SIGIR-96, 19th ACM international conference on research and development in information retrieval. Zürich, CH, pp 76–84.

Kelly D, Belkin N J (2001) Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevant feedback. In: Procedings fo the 24th annual international ACM conference on research and development in information retrieval. pp 408–409

Lu XA, Ayoub M, Dong J (1996) Ad hoc experiments using EUREKA. In: TREC

Porter MF (1997) An algorithm for suffix stripping. In: Sparck Jones K, Willett P (eds) Readings in Information Retrieval Morgon Kaufmann, Son Franciso pp 313–316

Rocchio J (1971) Relevance feedback in information retrieval, Prentice-Hall, pp 313–323

Ruthven I (2004) 'and this set of words represents the user's context...'. In: Workshop on information retrieval in context at the ACM SIGIR 2004 conference

Salton G, MacGill M (1983) Introduction to modern information retreival. McGraw-Hill
Sieg A, Mobasher B, Lytinen S, Burke R (2003) Concept based query enhancement in the arch search agent. In: Proceedings of the 4th international conference on Internet computing. Las Vegas, NV
Sugiyama K, Hatano K, Yoshikawa M (2004) Adaptive web search based on user profile constructed without any effort from users. In: WWW '04: proceedings of the 13th international conference on World Wide Web. New York, NY, USA, ACM Press, pp 675–684
Xu J, Croft WB (1996) Query expansion using local and global document analysis'. In: SIGIR '96: proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA, ACM Press, pp 4–11
Xu J, Croft W (2000) Improving the effectiveness of information retrieval with local context analysis. ACM Trans Inf Syst 18(1):79–112