Inf Retrieval (2006) 9: 231–248
DOI 10.1007/s10791-006-0837-9

Construction of query concepts based on feature clustering of documents

Youjin Chang · Minkoo Kim · Vijay V. Raghavan

Received: 10 April 2004 / Revised: 3 March 2005 / Accepted: 7 March 2005

Abstract In Information Retrieval, since it is hard to identify users' information needs, many approaches have been tried to solve this problem by expanding initial queries and reweighting the terms in the expanded queries using users' relevance judgments. Although relevance feedback is most effective when relevance information about retrieved documents is provided by users, it is not always available. Another solution is to use correlated terms for query expansion. The main problem with this approach is how to construct the term-term correlations that can be used effectively to improve retrieval performance. In this study, we try to construct *query concepts* that denote users' information needs from a document space, rather than to reformulate initial queries using the term correlations and/or users' relevance feedback. To form *query concepts*, we extract features from each document, and then cluster the features into primitive concepts that are then used to form *query concepts*. Experiments are performed on the Associated Press (AP) dataset taken from the TREC collection. The experimental evaluation shows that our proposed framework called QCM (Query Concept Method) outperforms baseline probabilistic retrieval model on TREC retrieval.

Keywords concept-based information retrieval · query reformulation · query concepts

1. Introduction

An information retrieval (IR) system returns a set of documents satisfying the information need expressed by a user's question. The purpose of information retrieval is to retrieve all

Y. Chang (✉)
Graduate School of Information and Communication, Ajou University, Suwon, Korea
e-mail: xaritas@ ajou.ac.kr

M. Kim
Department of Information and Computer Engineering, Ajou University, Suwon, Korea
e-mail: minkoo@ ajou.ac.kr

V. V. Raghavan
The Center for Advanced Computer Studies, University of Louisiana, Lafayette, USA
e-mail: raghavan@cacs.louisiana.edu

the relevant documents, while filtering out non-relevant document (van Rijsbergen, 1979). Nowadays, Web is staged in the center of the information technology. In search engines, the users might feel difficulty to precisely formulate their queries. The problem is illustrated through the commonly observed phenomenon, during information search on the Web, where the queries are usually of a few words long and a large number of hit documents are returned to the user. Most people are not good at making effective queries right away. Therefore, they spend large amount of time in reformulating their queries to accomplish effective retrieval.

Many researchers have tried to find appropriate solutions for representing users' interest correctly (Han et al., 1994; Koenemann, 1996; Qiu and Frei, 1993; Salton and Buckley, 1990; Xu and Croft, 1996). They have studied the query reformulation method for improving the initial query through query expansion and term reweighting. The query reformulation involves two basic steps: expanding the initial query with new terms and reweighting the terms in the expanded query. These query expansion approaches are grouped in three categories (Han et al., 1994): manual query expansion, automatic query expansion and semi-automatic query expansion. Manual or semi-automatic query expansion needs a third-party to accomplish an expansion. However, automatic query expansion expands initial query without other's intervention.

In this paper, we propose an automatic query expansion method that constructs *query concepts* that denote users' information needs from a document space. Our approach to generate *query concepts* is novel in the sense that we do not employ either Relevance Feedback (RF) (Salton and Buckely, 1990; Baeza-Yates and Ribeiro-Neto, 1999) or term-term correlations derived from a predefined knowledge base. Relevance feedback is a well-known method in query reformulation. Relevance feedback chooses important terms from previously retrieved documents that have been identified as relevant by the users or system, and enhances the importance of selected terms in a modified query.

In our method, we assume that there exist primitive concepts (basis concepts) in a document space, and they can be used to form high-level concepts that can be employed in the field of information retrieval. Hence, we first extract features of given documents and cluster them into primitive concepts, and then, based on these concepts, we form *query concepts*.

In Section 2, we describe related works. In Sections 3 and 4, we outline how primitive concepts and *query concepts* would be constructed. In Section 5, we present the experimental results that are obtained on a part of TREC collection. In the experimental tests, we evaluate our Query Concept Method (QCM) with previous approaches such as Pseudo Relevance Feedback (PRF) (Rocchio, 1971; Baeza-Yates and Ribeiro-Neto, 1999). PRF is a sort of blind relevance feedback generally used as a fully automatic query expansion. The results show that our proposed method outperforms previous approaches.

2. Related work

There are many works related to automatic query reformulation that improves initial queries through query expansion and term reweighting (Bodner and Song, 1996; Chang and Hsu, 1998; Han et al., 1994; Klink, 2001; Koenemann, 1996; Qiu and Frei, 1993; Rocchio, 1971; Salton and Buckley, 1990; Xu and Croft, 1996). The fully automatic methods for query reformulation do not rely on users to make relevance judgments. They are often based on language analysis (Bodner and Song, 1996; Bookman et al., 1999; Spark Jones and Tait, 1984), term co-occurrences, PRF (Rocchio, 1971; Baeza-Yates and Ribeiro-Neto, 1999), or concept-based retrieval (Kim et al., 2000; Klink, 2001; Nakata et al., 1998; Qiu and Frei, 1993).

According to Bodner and Song's research (1996), language analysis approaches require a deep understanding of queries and documents at higher computational costs. The requirements to achieve a deep understanding are still an open problem in the field of artificial intelligence and this query reformulation technique has also been shown to have only small improvements in retrieval performance.

Without users' relevance feedback, there are other strategies to reformulate queries. The idea involves identifying terms that are related to the query terms. Those terms might be synonyms, stemming variations, or terms that are close to the query terms in the text. Two basic types of strategies are global analysis and local analysis. In automatic global analysis, the similarity thesaurus obtained is based on term-term relationships. Unfortunately, this approach does not work well in general because the relationships captured in a thesaurus frequently are not valid in the local context of a given user query (Baeza-Yates and Ribeiro-Neto, 1999). Automatic local analysis adopts clustering techniques for query expansion. The local clustering techniques are based on the set of documents retrieved for the original query and use the top ranked documents for clustering neighbor terms. Such a clustering is based on term co-occurrence inside documents. The idea of applying a global analysis technique to a local set of documents retrieved is called local context analysis. An earlier work done by Xu and Croft (1996) illustrated the advantage of combining techniques form both local and global analysis.

The Pseudo Relevance Feedback (PRF) is a representative method of automatic query expansion. Since precise user's feedback is difficult to obtain, in PRF, multiple documents at the top of the ranked list are assumed to be relevant. This procedure has been found to be highly effective in some cases, most likely those in which the original query statement are long and precise (Baeza-Yates and Ribeiro-Neto, 1999). This approach may impose some problems on selecting terms that are unrelated to relevance and happen to appear in documents that meet the selection criteria. Unreliable terms will be added to the query with subsequent adverse effects on retrieval behavior.

Another related area of our research is concept-based retrieval (Kim et al., 2000; Klink, 2001; Nakata et al., 1998; Qiu and Frei, 1993). It starts from the considerable interest in bridging the gap between the terminology used in defining queries and the terminology used in representing documents (Kim et al., 2000). It treats those query words not as literal strings of letters, but as representing concepts, therefore using concept-based retrieval can retrieve relevant documents even if we do not contain the specific words used in the query. Concept-based retrieval experiments often tested the effects of thesaurus-based query expansion on Boolean retrieval performance. Some of them use a thesaurus such as a WordNet (Fellbaum, 1998) or a rule-base tree, such as in RUBRIC (McCune et al., 1985), to expand query terms. But, this approach requires a lot of time with the processing of individual queries, and does not work well in general because the relationship captured in a thesaurus are not always valid in the context of given user query. Even if the initial query is successfully expanded by chance, we can not guarantee extensible uses of a thesaurus. Besides, it is impossible to make an optimal thesaurus in every field of study and the problem caused by 'polysemy' which has more than one distinct meaning (e.g. chip, model) can be another problem.

While previous query reformulation methods focus on reformulating initial queries by expanding and reweighting the terms in the queries by depending on users' relevance judgment and/or predefined knowledge base such as a thesaurus, in our approach, we try to create a set of concepts from a document space appropriately, and then reformulate initial queries with *query concepts*. To construct *query concepts*, we extract features from each document, and cluster them into primitive concepts that can be used to form *query concepts*. There are

similar studies on extracting *query concepts* from a document space (Kim et al., 2000; Nakata et al., 1998; Qiu and Frei, 1993; Wong and Fu, 2000).

Nakata et al. (1998) introduced a notion of the Concept Index, which aims to index important concepts described in a collection of documents belonging to a group, and provides user-friendly cross-references among them to aid concept-oriented document space navigation. The Concept Index relied on users to identify important concepts by marking keywords and phrases that interest them. Nakata's work addressed a group of individuals who shared the same interest or a task and would profit from making use of the knowledge possessed by the group. This approach supports the hypothesis that documents have a concept that users aim for and want to retrieve. However, it is different from ours since they use collaboration among the members of a group for extracting concepts. In our approach, we try to automatically construct *query concepts* from a document space.

Kim and his colleagues (2000) proposed a method to automatically construct *query concepts* from typical thesauri. Their approach uses production rules to capture *query concepts* (or topics). Although their experiment successfully constructs concepts from thesauri based on the semantics and showed that the automatically constructed rules are more effective than hand-made rules in terms of precision, their experiments were performed on small collections with a domain-specific thesaurus. In order to generate rules, they used a method to pre-specify weight values for the relationships NT (Narrow Term), BT (Broad Term) and RT (Related Term). Since thesauri usually do not provide degrees of relatedness between terms, whenever they use a different thesaurus, they need to adjust the weighting values for the relationships. It means the lack of expansibility of system. Nevertheless, their philosophy to capture user's *query concept* has a connection with basic concept of our research.

We also consider that the Latent Semantic Indexing (LSI) and Probabilistic Latent Semantic Indexing (PLSI) could have similar research interests to our research. LSI (Deerwester et al., 1990) is similar to ours in that they had tried to identify basis concepts for use in retrieval. The techniques for identifying the concept vectors have represented documents, terms and queries directly in the concept space. Our research has investigated that we extract primitive (basic) concepts from a document space directly and then construct *query concepts* by combining those concepts. Indeed, there is the association of ideas between their approaches and ours in that both start from an interest of concept-based representation.

For the methodology, the following is an approach similar to ours for generating primitive concepts through extracting information from document and clustering document features. Wong and Fu (2000) tried to construct concepts through the incremental document clustering technique for extracting features, which is more suitable to Web document classification. The main difference between their approach and ours is that their construction of concepts is for Web classification; not for constructing queries. Even if the procedure to make concepts from documents is similar, they didn't offer a method to form *query concepts* that can be directly applied to a user's query.

The goal of our research is to make *query concepts* that are close to users' information needs from a document space. In the next section, we describe how to construct primitive concepts that can be used to form initial concepts from a document space.

3. Extracting concepts from a document space

We assume that there are primitive concepts (basis concepts) in a document space, and they can be used to form any concepts used in the field of information retrieval. In order to form primitive concepts, we assume that documents contain features that characterize primitive

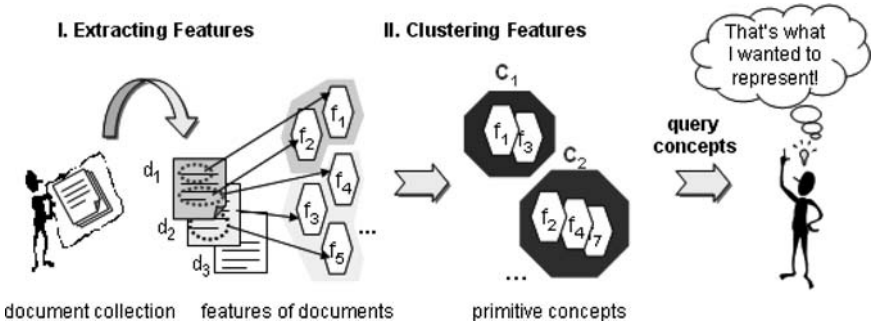

Fig. 1 The procedure of constructing primitive concepts

concepts. We have two steps to form primitive concepts: (1) to extract features from each document, and (2) to cluster the features into primitive concepts. The feature extracting process has two sub-steps: (1) to select significant sentences, (2) to partition these sentences into feature vectors. Figure 1 shows the main steps of constructing primitive concepts. In the next section, we describe each step in detail.

3.1. Extracting features from a document space

In this step, we aim to extract a set of features that are especially unique elements of each document. Let us suppose that there is an apple and someone ate it. We can explain about the apple like this: "This apple tastes sweet. It's red. It's smaller than my fist. I think it's very delicious!" The terms, 'sweet', 'red', 'small' and 'delicious' are attributes of the apple. While 'sweet' and 'delicious' represent a similar sense, 'sweet', 'red', 'small' are totally different properties and can not be mixed-up to form a single concept. Nevertheless, those adjectives represent an apple. In the same manner, we assume that a document consists of a set of orthogonal components. They can be composed of unit, such as a paragraph and/or sentence. The main point is that document can be represented by these orthogonal components (features). Now, we have a problem how to extract such features. To exact the features, we will perform the following steps: (1) extracting features that describe a document, (2) merging similar features among them into one feature.

In order to extract features (in the form of a vector) from a document, we adopt well-known and simple summarizing techniques (Edmundson, 1969; Lam-Adesina and Jones, 2001; Tombros and Sanderson, 1998). The earlier research of Lam-Adesina and Jones (2001) stated that summary generation methods seek to identify document contents that convey the most "important" information within a document. They applied a very robust summarizer that can handle different text types likely to be encountered within a retrieval system. Their sentence extraction method for summary generation was formed by scoring the sentences in a document using some criteria, ranking the sentences, and then taking a number of the top ranking sentences as the summary. In our research, we also select significant sentences using Luhn's keyword cluster method (1958), the title term frequency method and the location method suggested by Edmundson (1969) then generate summary for extracting document features.

After finding significant components of a document, we integrate them into orthogonal components within a document and name them as 'features' or 'feature vectors' that characterize the document. In order to construct feature vectors, we simply partition the significant sentences. Generated feature vectors do not contain the terms in other feature vectors of the

same document. So, this processing makes feature vectors to be orthogonal to each other. This process might give rise to generate a single feature per document if the document contains only one topic. However, we observed that many documents also had several features, which means there were documents which had multiple topics within a document.

3.1.1. Selecting significant sentences (SSS)

To select significant sentences from a document, we use Luhn's keyword cluster method (1958), title term frequency method and the location method (Edmundson, 1969; Tombros and Sanderson, 1998; Lam-Adesina and Jones, 2001). Luhn's keyword cluster technique, though simple, is one of famous methods to produce summaries used alone or in combination with other methods. The technique of frequency analysis of words is used to determine the significance in a document. In that method, they used the most significant cluster in a sentence to measure the significance of the sentence. Luhn (1958) suggested that sentences in which the greatest number of frequently occurring distinct words are found in greatest physical proximity to each other, are likely to be important in describing the content of the document in which they occur. The significance score factor for a sentence is given by SW^2/TW where SW is the number of significant words and TW is the total number of words. To decide significant words in a document, we follow the work of Tombros and Sanderson (1997) which conclude that a reasonable term occurrence value for establishing the significance of a term, we called it significant term occurrence(STO), was 7; where a medium sized documents (between 25–40 sentences). For documents beyond the scope of medium size, reasonable term occurrence values was defined as me $STO = 7 + [0.1 * (25 - \mathrm{NS})]$ for documents with $\mathrm{NS} < 25$ where NS was the number of sentences in the document and $STO = 7 + [0.1 * (\mathrm{NS} - 40)]$ for documents with $\mathrm{NS} > 40$ (Tombros and Sanderson, 1997). Therefore, if one term occurs over STO times, the term is considered as a significant term. Secondly, we score each sentence by a title term frequency. The title of an article often reveals the major subject of that document (Lam-Adesina and Jones, 2001). This hypothesis was examined in TREC documents where the title of each article was found to convey the general idea of its contents. In order to utilize this attribute in scoring sentences, each constituent term in the title section is looked up in the body of the text. Thirdly, we give a location score to the first two sentences of a document for applying the location method. Edmundson (1969) stated that the location of a sentence within a document is often useful in determining its importance to the document. We assign a location score as $1/NS$ where NS is the number of sentences in the document. The final score for each sentence is calculated by summing the individual score factors obtained for the above three methods.

In order to generate an appropriate length of summary, Lam-Adesina and Jones (2001) stated that it was essential to place a limit on the number of sentences to be used as summary contents. Following their suggestion, we set the lower bound of summary length to 15% of original document length and the maximum value up to six sentences, which is the reasonable number of significant sentences selected for a given document (Lam-Adesina and Jones, 2001; Tombros and Sanderson, 1998). Finally we choose the highly ranked k sentences as significant sentences.

3.1.2. Partitioning selected significant sentences into feature vectors

As mentioned before, our goal is to construct *query concepts* that denote users' information needs. For this purpose, we try to find out primitive concepts (basis concepts) that can be used to form the *query concepts*. We treat distinct features of documents as good candidates

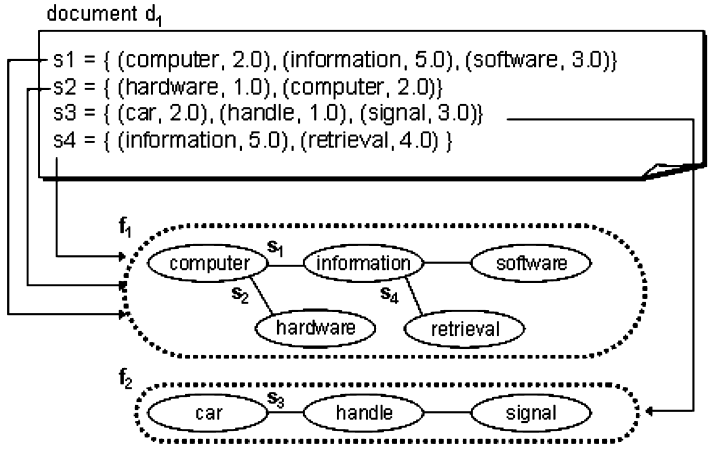

Fig. 2 The procedure of partitioning significant sentences to feature vectors

of primitive concept. In this research, we assume that a document can have several distinct features. Extracting significant sentences only is insufficient to ensure that we obtain discriminating contents from a document, because it is possible that documents can contain several distinct topics as the form of cluster. To find out the features from a document, we group the significant sentences selected from a document (described in Section 3.1.1) into partitions such that they have distinct meanings.

Suppose that we represent a sentence as a vector of terms in the sentence with their *tf* weight values, and there are no stopwords in the sentence. We can consider a set of vectors $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots, \boldsymbol{s_k}\}$, where k is the number of selected significant sentences for a document. To make feature vectors for the document, we partition $\boldsymbol{S}$ as follows. Let us consider each vector $\boldsymbol{s_i}$ as a subgraph such that the vertices of the subgraph are terms of the vector and they are connected. We then consider a feature vector as a connected component (Cormen et al., 2001) of the graph consisting of subgraphs $\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots, \boldsymbol{s_k}$. Since feature vectors are connected components of the graph, feature vector for the document is orthogonal to other feature vectors. For example, there are four significant sentences for document d_1 and let us assume that a sentence is represented as a set of term and its weight value pairs. See Fig. 2.

$\boldsymbol{s_1} = \{$*(computer, 2.0), (information, 5.0), (software, 3.0)*$\}$
$\boldsymbol{s_2} = \{$*(hardware, 1.0), (computer, 2.0)*$\}$
$\boldsymbol{s_3} = \{$*(car, 2.0), (handle, 1.0), (signal, 3.0)*$\}$
$\boldsymbol{s_4} = \{$*(information, 5.0), (retrieval, 4.0)*$\}$

In the above example, for document d_1, we can construct two feature sets $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ using maximally connected component theory:

$\boldsymbol{f_1} = \{$*(computer, 2.0), (information, 5.0), (software, 3.0), (hardware, 1.0), (retrieval, 4.0)*$\}$
$\boldsymbol{f_2} = \{$*(car, 2.0), (handle, 1.0), (signal, 3.0)*$\}$

We can construct feature vectors $\boldsymbol{vf_1}$ and $\boldsymbol{vf_2}$ corresponding to $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$, respectively, as follows. We can see that $\boldsymbol{vf_1}$ and $\boldsymbol{vf_2}$ are orthogonal.

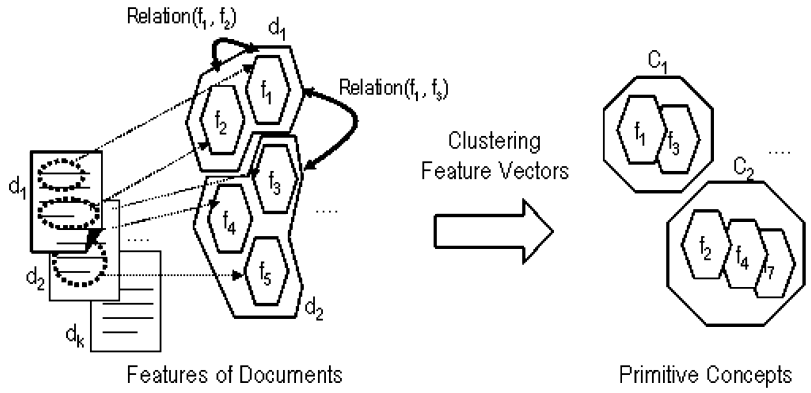

Fig. 3 Clustering feature vectors into primitive concept vectors (centroid vectors)

	car	*computer*	*handle*	*hardware*	*information*	*retrieval*	*signal*	*software*
vf1	*(0.0*	*2.0*	*0.0*	*1.0*	*5.0*	*4.0*	*0.0*	*3.0)*
vf2	*(2.0*	*0.0*	*1.0*	*0.0*	*0.0*	*0.0*	*3.0*	*0.0)*

3.2. Clustering feature vectors into primitive concepts

In the previous section, we construct the feature vectors for each document. The feature vectors in a document are orthogonal to each other, but they might not be orthogonal to the feature vectors from other documents. See Fig. 3. The Relation (f_1, f_2) which means the relation between f_1 and f_2 is a definitely orthogonal. However, we can not guarantee that the Relation (f_1, f_3) is always orthogonal. For example, we assume that both of document d_1 and d_2 are about 'The future of Computer Industry' and the feature f_1 in document d_1 consists of a set of terms such as 'computer', 'information', 'software', 'hardware', and 'retrieval' as described in Section 3.1. If the feature f_3 in document d_2 consists of a set of terms such as 'computer', 'hardware', 'communication' and 'information', the feature f_1 and f_3 are quite similar to each other. Therefore, we could not consider all the feature vectors as good candidates of primitive concepts.

To alleviate this problem, we cluster the feature vectors such that the centroid vectors of clusters are approximately orthogonal to each other. We call the constructed centroid vectors 'primitive concepts'. The clustering method is generally used for descriptive modeling in data mining. Mannila (2002) stated that data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data analyst. If we assume that document collection is a large observational data set which has latent meanings, we can find those concepts through global modeling and local pattern discovery. In previous section, we have already identified features of each document using summarization locally. In this step, we are going to try to cluster features of the whole collection of documents globally.

Many effective clustering algorithms are available (Pelleg and Moore, 2000; Quaresma and Rodrigues, 2000; Willett, 1988; Wong and Fu, 2000; Zhang et al., 1996). In order to cluster feature vectors into primitive concepts, we first selected two popular methods from already-developed methods; for instance, *K*-means and *X*-means clustering methods (Pelleg et al., 2000). However, they did not work well in our case. We conjectured that the reason why the previous clustering methods did not work well is that our feature vectors are extremely

Table 1 Developed algorithm for Clustering feature vectors into primitive concepts

```
Let f1, ..., fn be feature vectors,
m the number of generated clusters,
and u and v are controlled parameters.

m = 1;
Cm = f1;
for i = 2 to n
{
  done = FALSE;
  j = 1;
  while(j < = m and not done) {
    if fi is more than u% overlapped with Cj then {
      Cj = (Cj + fi)/2;
      done = TRUE; /* assign fi to Cj */
    }
    else if fi is more than v% overlapped with Cj then
      done = TRUE; /* ignore fi */
    Else
      j = j + 1;
  }
    if (not done) then {
      m = m + 1;
      Cm = fi; /* Create a new cluster */
    }
  }
```

sparse. Therefore, we develop a simple clustering method to establish the primitive concepts as described in Table 1. In our clustering method, if more than $u\%$ (e.g., 80%) terms in a feature vector $\boldsymbol{f_i}$ are contained in the centroid vector of a cluster $\boldsymbol{C_j}$, we put $\boldsymbol{f_i}$ into $\boldsymbol{C_j}$ and recompute the centroid vector of $\boldsymbol{C_j}$ such that $\boldsymbol{C_j} = (\boldsymbol{C_j} + \boldsymbol{f_i})/2$. If between $u\%$ (e.g., 80%) and $v\%$ (e.g., 20%) of terms in $\boldsymbol{f_i}$ are contained in $\boldsymbol{C_j}$, then we ignore $\boldsymbol{f_i}$. If less than $v\%$ of terms in $\boldsymbol{f_i}$ are contained in $\boldsymbol{C_j}$, we keep trying to put $\boldsymbol{f_i}$ into other clusters. If $\boldsymbol{f_i}$ is not assigned to any clusters, we create a new cluster that contains $\boldsymbol{f_i}$. This algorithm is similar to the single pass and reallocation method which were used in early work in cluster analysis in IR (Frakes and Baeza-Yates, 1992).

After the first step, we get m clusters. The centroids of m clusters are to be the vectors for the newly generated primitive concepts. Since the proposed algorithm ignores some feature vectors which did not belong to any clusters, we need to consider these missing features to be assigned to the generated clusters. Moreover, we should evaluate the quality of the generated clusters by analyzing whether features could belong to other clusters. For this purpose, we apply the above algorithm to the feature vectors once again. In this step, we compare the feature vectors $(\boldsymbol{f_1}, \ldots, \boldsymbol{f_n})$ with the generated centroid vectors $(\boldsymbol{C_1}, \ldots, \boldsymbol{C_m})$. This reallocation process is operated by selecting some initial partition of the feature vectors and then moving the features from cluster to cluster to obtain an improved partition. Before we reassign features to the generated centroid vectors, we sort the generated centroid vectors by the number of terms in each cluster. See Fig. 4. Since the generated concept vectors are not exactly but approximately orthogonal, the feature vectors could be assigned to a large cluster in the reallocation step. Hence, we sort them out by the number of terms in each cluster, and then apply the above algorithm to clustering the feature vectors again (reallocation).

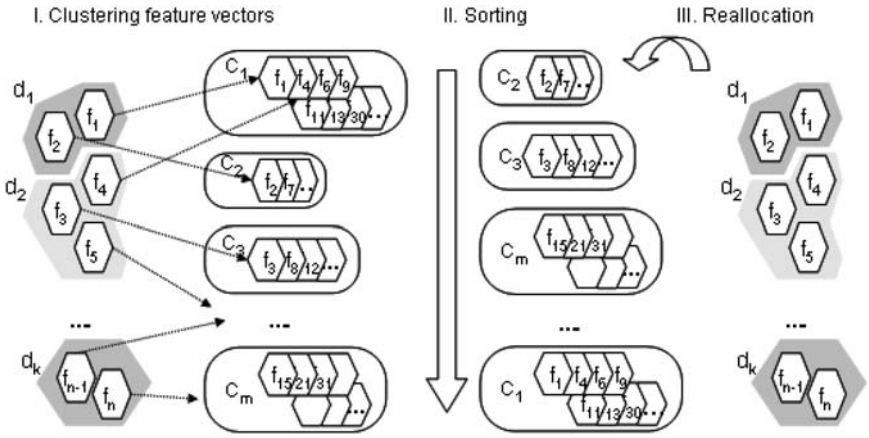

Fig. 4 Example of the clustering process for refining the centroid vectors

4. Information retrieval using primitive concepts

For a query reformulation based on *query concepts*, we select its most associated primitive concepts with the initial query and generate all possible interpretations of the query. The most probable interpretations are chosen as *query concepts* and are added to the initial query during the reformulation process. These *query concepts* are constructed by combining primitive concept vectors. For this purpose, we consider the rule to formulate query concept vector (in the form of a vector). The construction method is as follows (see Fig. 5).

1. Select first top N primitive concept vectors those are similar to the initial query $\boldsymbol{q_0}$, using cosine similarity.

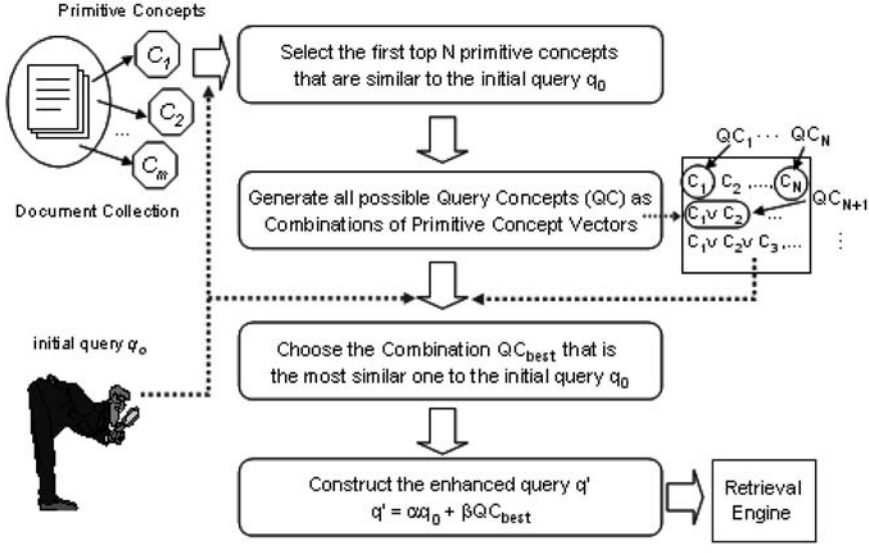

Fig. 5 The process of generating query concept and enhancing an initial query

2. Generate all possible combination of primitive concepts under a DNF (Disjunctive Normal Form) with at most three primitive concept vectors among the ten selected primitive concept vectors. These are called candidates *query concepts*.
3. Choose the DNF that is most similar to the initial query $\boldsymbol{q_0}$, using the cosine similarity. The selected DNF is called $\boldsymbol{QC_{best}}$.
4. Construct the enhanced query $\boldsymbol{q'} = \alpha \boldsymbol{q_0} + \beta \boldsymbol{QC_{best}}$, where $0 \leq \alpha \leq 1$ and $\beta = 1 - \alpha$ are the weighting constants.

We use MAX operation for OR ($\vee$) operation in the DNFs that are generated in step (2) above. For example, let the selected primitive concepts be $\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_{10}}$ where $N = 10$ and the initial query be $\boldsymbol{q_0}$. We can generate all possible DNFs as follows:

$$\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_{10}},$$
$$\boldsymbol{C_1} \vee \boldsymbol{C_2}, \boldsymbol{C_1} \vee \boldsymbol{C_3}, \ldots, \boldsymbol{C_9} \vee \boldsymbol{C_{10}},$$
$$\boldsymbol{C_1} \vee \boldsymbol{C_2} \vee \boldsymbol{C_3}, \boldsymbol{C_1} \vee \boldsymbol{C_2} \vee \boldsymbol{C_4}, \ldots, \boldsymbol{C_8} \vee \boldsymbol{C_9} \vee \boldsymbol{C_{10}}$$

From all possible DNFs, we select one that is most similar to $\boldsymbol{q_0}$ as the query concept. Suppose that $\boldsymbol{C_1} \vee \boldsymbol{C_3}$ is the $\boldsymbol{QC_{best}}$ that is the most similar to the initial query $\boldsymbol{q_0}$. Then, we construct the enhanced query $\boldsymbol{q'} = \alpha \boldsymbol{q_0} + \beta(\boldsymbol{C_1} \vee \boldsymbol{C_3})$.

5. Experiments

For the evaluation of proposed methods, we conducted experiments on the Associated Press (AP) subset of TREC collection; disk 1, 2 and 3 (Harman 1995, Hawking et al. 1999). The AP dataset in directories '88', '89' and '90' of TREC collection totally contains about 240,000 documents. 50 topics (topic 101–150) were chosen to evaluate the performance. For our investigation, title and description fields of the topics were chosen since we assumed that documents and queries have multiple concepts. When we used only a few words in the title field of topics, it was not enough to represent concepts in query topics.

We removed stopwords and used Porter's algorithm (Porter, 1980) for stemming. Now, we describe our experimental results in a series of comparisons as follows. Basically, we conduct baseline retrieval, PRF and our Query Concept Method (QCM) respectively.

5.1. Baseline and PRF methods

We conducted a baseline based on BM25 Probabilistic Model for term weighting (Robertson et al., 1994). We calculated the weight of terms in document d_k and query q_k as follows,

$$\begin{aligned}
&\textit{document_term_weight} : d_k = \frac{(k_1+1)tf}{K+tf} \\
&\textit{query_term_weight} : q_k = \frac{(k_3+1)\cdot tf}{k_3+tf} w^{(1)} \\
&\textit{R/S_weight} : w^{(1)} = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)(N-n-R+r+0.5)} \\
&\textit{length_normalism_ion} : K = k_1 \left((1-b) + b \frac{\text{document_length}}{\text{avg_doc.length}} \right)
\end{aligned} \tag{1}$$

where *tf* is term frequency, $w^{(1)}$ is the weight based on the basic probabilistic model, N is the total number of documents, n is the total number of documents containing term k, R is the total number of relevant documents for the query, r is the number of relevant documents containing term k and K is the normalized document length. We also set the default parameters k_1, b, k_3 used in BM functions as follows. $k_1 = 1.2$, $b = 0.75$, $k_3 = 1000$ (Robertson et al., 1994).

Secondly, we chose the Rochhio's Relevance Feedback methods (1971) to examine PRF approach. We used a Standard_Rocchio formulation to calculate the modified query as follows,

$$\vec{q}_{new} = \alpha\vec{q}_{old} + \frac{\beta}{|D_{rel}|}\sum_{\forall\vec{d}_j\in D_{rel}}\vec{d}_j - \frac{\gamma}{|D_{nrel}|}\sum_{\forall\vec{d}_j\in D_{nrel}}\vec{d}_j \tag{2}$$

Notice that D_r and D_n stand for the sets of relevant and non-relevant documents (among the retrieved ones) according to the user judgment, respectively. α, β and γ are tuning constants. We set $\alpha = \beta = 1$, $\gamma = 0$ which yields a positive feedback strategy (Baeza-Yates and Ribeiro-Neto, 1999). Since PRF is a kind of PRF, we assumed that highly ranked p retrieved documents were relevant and took q terms among term pool set for term ranking in the query expansion process. We assumed that $p = 15$ and $q = 20$. The expansion term ranking criteria was the Robertson selection value (*rsv*) (Robertson, 1990). The *rsv* is defined as,

$$rsv(i) = r(i) \times rw(i) \tag{3}$$

where $r(i)$ is again the number of relevance documents containing term i, and $rw(i)$ is the standard Robertson/Spark Jones relevance weight. $rw(i)$ is defined as,

$$rw(i) = \log\frac{(r(i)+0.5)(N-n(i)-R+r(i)+0.5)}{(n(i)-r(i)+0.5)(R-r(i)+0.5)} \tag{4}$$

where $n(i)$ is the total number of document containing term i, R is the total number of relevant documents for the query, and N is the total number of documents.

Thirdly, we experimented with variation of baseline and PRF by summary information. This summary file was generated during the extraction of feature vectors (see Sections 3.1 and 3.2). When we selected the significant sentences from each document and created new summary files. We labeled these approaches as 'Summary Baseline' and 'Summary PRF' respectively. We conducted these alternative experiments to examine the effectiveness of summary information. Unfortunately, when we retrieved documents using summary information only, the results of 'Summary Baseline' were not successful. 'Summary RRF' was equally hard to achieve good results since 'Summary Baseline' has already missed much information of documents during summarization process. However, Lam-Adesina and Jones (2001) showed that query expansion using document summaries could be considerably more effective than using full-document expansion. Their research has reported an investigation into the use of document summarization for term-selection in pseudo relevance feedback.

Table 2 shows the retrieval performances of baseline, PRF, 'Summary Baseline', 'Summary PRF'. We evaluate the results based on precisions at 5, 10, 15, 20, 30 and 100 documents and Mean Average Precision (MAP). As we mentioned in Section 2, PRF approach might have some problems on selecting terms that are unrelated to relevance and happen to

Table 2 Retrieval results of the baseline, PRF, summary baseline, summary PRF

	BASELINE	PRF	SUMMARY BASELINE	SUMMARY PRF
5 docs	0.1918	0.2041	0.1551	0.2042
10 docs	0.1898	0.1918	0.1429	0.1521
15 docs	0.1741	0.1578	0.1510	0.1375
20 docs	0.1786	0.1398	0.1592	0.1271
30 docs	0.1769	0.1197	0.1565	0.1097
100 docs	0.1571	0.0757	0.1312	0.0744
MAP	0.1387	0.0692	0.1073	0.0583

appear in documents that meet the selection criteria (Baeza-Yates and Ribeiro-Neto, 1999). We surmise that an initial query could not provide appropriate terms to expand. As a result, the direction of blind relevance feedback performed unsuccessfully.

5.2. Query concept methods (QCM)

Finally, we conducted Query Concept Method (QCM). Generally, our methods showed better results than those of previous methods. We used the controlled parameter $u = 0.8$ and $v = 0.2$ for clustering and five different sets of α and β. Since the modified query is generated by $\boldsymbol{q}' = \alpha \boldsymbol{q_0} + \beta \boldsymbol{QC_{best}}$, the result of QCM for $\beta = 0.0$ is same to that of baseline.

Table 3 shows the comparison between baseline and our methods on the retrieval performances and statistical improvement (percentage change). *Rel* is the total number of relevant documents over all queries and *Rret* is the total number of relevant documents retrieved over all queries. The evaluation includes recall/precision averages at 0.0 to 1.0, MAP, precision at 5 to 1000 documents cutoff and R-precision (R-Pr) which denote a precision after *Rel* documents. We observe that the QCM runs consistently outperform baseline run for all queries. The result of QCM3 ($\alpha = 0.5$, $\beta = 0.5$) shows 13% improvement on MAP (0.1387 vs. 0.1568). The result of QCM2 ($\alpha = 0.6$, $\beta = 0.4$) shows about 13.3% improvement on R-precision (0.1833 vs. 0.2077).

Figure 6 shows the retrieval precision curves of Baseline, PRF, Summary variations and QCM methods (from $\alpha = 0.8$, $\beta = 0.2$ to $\alpha = 0.2$, $\beta = 0.8$). It can be seen that the QCM

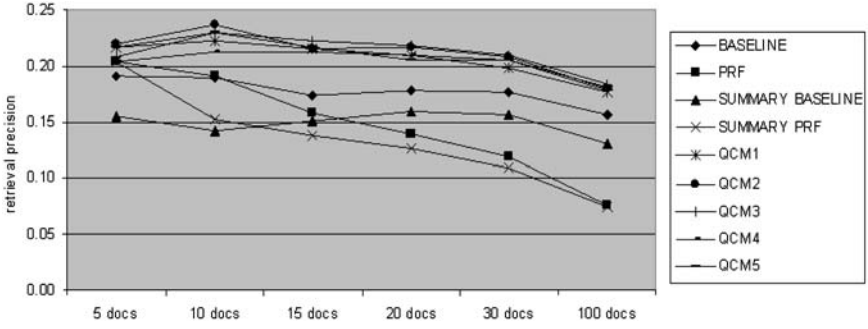

Fig. 6 The retrieval precision of the baseline, PRF, Summary variations and QCM methods

Table 3 Retrieval results of Query Concept Method (QCM) followed by different sets of α and β

	BASELINE	QCM1 ($\alpha = 0.8, \beta = 0.2$)	%chg	QCM2 ($\alpha = 0.6, \beta = 0.4$)	%chg	QCM3 ($\alpha = 0.5, \beta = 0.5$)	%chg	QCM4 ($\alpha = 0.4, \beta = 0.6$)	%chg	QCM5 ($\alpha = 0.2, \beta = 0.8$)	%chg
Rel	4802	4802		4802		4802		4802		4802	
Rret	2847	2891	+1.5	2942	+3.3	2920	+2.6	2877	+1.1	2780	−2.4
0.0	0.4254	0.4418	+3.9	0.4521	+6.3	0.4501	+5.8	0.4352	+2.3	0.4247	−0.2
0.1	0.2936	0.3213	+9.4	0.3268	+11.3	0.3279	+11.7	0.3240	+10.4	0.3277	+11.6
0.2	0.2513	0.2664	+6.0	0.2718	+8.2	0.2755	+9.6	0.2650	+5.5	0.2705	+7.6
0.3	0.2049	0.2262	+10.4	0.2280	+11.3	0.2310	+12.7	0.2233	+9.0	0.2218	+8.2
0.4	0.1645	0.1771	+7.7	0.1878	+14.2	0.1898	+15.4	0.1889	+14.8	0.1866	+13.4
0.5	0.1384	0.1516	+9.5	0.1578	+14.0	0.1603	+15.8	0.1591	+15.0	0.1564	+13.0
0.6	0.1065	0.1090	+2.3	0.1204	+13.1	0.1241	+16.5	0.1166	+9.5	0.1189	+11.6
0.7	0.0813	0.0866	+6.5	0.0931	+14.5	0.0900	+10.7	0.0831	+2.2	0.0826	+1.6
0.8	0.0401	0.0439	+9.5	0.0452	+12.7	0.0470	+17.2	0.0473	+18.0	0.0378	−5.7
0.9	0.0201	0.0184	−8.5	0.0201	+0.0	0.0201	+0.0	0.0192	−4.5	0.0181	−10.0
1.0	0.0009	0.0011	+22.2	0.0012	+33.3	0.0013	+44.4	0.0013	+44.4	0.0014	+55.6
MAP	0.1387	0.1503	+8.4	0.1558	+12.3	0.1568	+13	0.1532	+10.5	0.1509	+8.8
5	0.1918	0.2163	+12.8	0.2204	+14.9	0.2082	+8.6	0.2041	+6.4	0.2163	+12.8
10	0.1898	0.2224	+17.2	0.2367	+24.7	0.2306	+21.5	0.2122	+11.8	0.2306	+21.5
15	0.1741	0.2150	+23.5	0.2150	+23.5	0.2231	+28.1	0.2122	+21.9	0.2163	+24.2
20	0.1786	0.2102	+17.7	0.2173	+21.7	0.2184	+22.3	0.2092	+17.1	0.2051	+14.8
30	0.1769	0.1980	+11.9	0.2082	+17.7	0.2102	+18.8	0.2054	+16.1	0.2048	+15.8
100	0.1571	0.1763	+12.2	0.1796	+14.3	0.1839	+17.1	0.1804	+14.8	0.1780	+13.3
200	0.1313	0.1455	+10.8	0.1473	+12.2	0.1507	+14.8	0.1481	+12.8	0.1459	+11.1
500	0.0908	0.0940	+3.5	0.0959	+5.6	0.0956	+5.3	0.0942	+3.7	0.0921	+1.4
1000	0.0581	0.0590	+1.5	0.0600	+3.3	0.0596	+2.6	0.0587	+1.0	0.0567	−2.4
R-Pr	0.1833	0.2037	+11.1	0.2077	+13.3	0.2070	+12.9	0.2043	+11.5	0.1996	+8.9

methods perform consistently well and all of the average precision of QCM methods are better than those of other methods.

5.3. Query Concept Method (QCM) on TREC 8

We experimented on TREC 8 collection for evaluating our proposed method in a large document collection. TREC 8 collection is a relatively large data set and contains over about 520,000 documents distributed on two CD-ROM disks (TREC disks 4 and 5) taken from the following sources: Federal Register (FR), Financial Times (FT), Foreign Broadcast Information Service (FBIS) and LA Times (LAT).

50 topics (topic 410–450) were chosen to evaluate the performance. The title and description fields of the topics were selected for constructing initial queries. Around 500 stopwords were removed and Porter's algorithm (Porter, 1980) was used for stemming. We conducted baseline, PRF and QCM runs based on a vector space model. Luhn's keyword cluster method

Table 4 Retrieval results of baseline, PRF and Query Concept Method (QCM) on TREC 8

	BASELINE	PRF	%chg	QCM ($\alpha = 0.8, \beta = 0.2$)	%chg
Rel	4707	4707		4707	
Rret	2181	2295	+5.2	2081	−4.6
0.0	0.5217	0.4445	−14.8	0.5478	+5.0
0.1	0.2809	0.2543	−9.5	0.2808	0.0
0.2	0.2189	0.1992	−9.0	0.2186	−0.1
0.3	0.1681	0.1787	+6.3	0.1739	+3.5
0.4	0.1305	0.1539	+17.9	0.1263	−3.2
0.5	0.0892	0.1103	+23.7	0.0807	−9.5
0.6	0.0593	0.0805	+35.8	0.0511	−13.8
0.7	0.0380	0.0553	+45.5	0.0318	−16.3
0.8	0.0198	0.0308	+55.6	0.0180	−9.1
0.9	0.0110	0.0243	+120.9	0.0109	−0.9
1.0	0.0102	0.0161	+57.8	0.0102	0.0
MAP	0.1185	0.1230	+3.8	0.1172	−1.1
5	0.3102	0.2367	−23.7	0.3102	0.0
10	0.2714	0.2163	−20.3	0.2653	−2.2
15	0.2503	0.2109	−15.7	0.2476	−1.1
20	0.2286	0.1969	−13.9	0.2265	−0.9
30	0.2170	0.1857	−14.4	0.2177	+0.3
100	0.1449	0.1451	+0.1	0.1473	+1.7
200	0.1071	0.1099	+2.6	0.1085	+1.3
500	0.0675	0.0690	+2.2	0.0668	−1.0
1000	0.0445	0.0468	+5.2	0.0425	−4.5
R-Pr	0.1720	0.1736	+0.9	0.1732	+0.7

(1958), the title term frequency method (Tombros, 1998) and the location method suggested by Edmundson (1969) were used in the process of extracting features from each document. The controlled parameter $u = 0.8$ and $v = 0.2$ were used in the clustering procedure. Table 4 shows recall/precision at 0.0 to 1.0, MAP, precision at 5 to 1000 documents cutoff and R-precision of the baseline, PRF and QCM method on TREC 8 collection. In this case, we only show the best results of QCM at $\alpha = 0.8$ and $\beta = 2$. We observe that the result of RRF is slightly improved on MAP and R-precision. Unfortunately, the results of QCM can not outperform baseline but are just as much as baseline. Although QCM could not achieve better results on TREC 8 collection, our proposed method was valuable for a query reformulation in case of poorly performing queries, which means that initially retrieved documents could not satisfy users (Chang et al., 2004).

6. Conclusion

This paper has proposed a new paradigm for automatically enhancing initial queries. In the proposed approach, we constructed *query concepts* that denote users' information needs. We suggested a framework to construct the *query concepts, which* extracted features from a document space and clustered them into primitive concepts that are basis elements of the *query concepts*. With the constructed concepts, we have shown promising experimental results. For the improvement of performance, we think that we could adopt more robust summarization methods and/or clustering methods proper to construct primitive concepts. However, since data mining technique is not the main focus to generate *query concepts*, we leave it for another research topic. Our future work is also to consider many other directions not only for extracting features for a document, but also for clustering features and constructing primitive concepts, since our ultimate object is to make *query concepts* suitable for user's information need.

Acknowledgments This work was partially supported by National Research Laboratory on Korea Institute of Science and Technology Evaluation and Planning (KISTEP) (M10302000087-03J0000-04400) and by the Brain Korea 21 Project in 2004.

References

Baeza-Yates R and Ribeiro-Neto B (1999) Modern information retrieval. Addison Wesley, pp. 131, 308

Bodner R and Song F (1996) Knowledge-based approaches to query expansion in information retrieval. In McCalla G (Ed.), Advances in Artificial Intelligence, Springer, New York, pp. 146–158

Bookman L, Houston A, Kuhns RJ, Martin P, Green S, and Woods W (2000) Linguistic knowledge can improve information retrieval. In: Proceedings of the sixth conference on Applied natural language processing. Morgan Kaufmann Publishers Inc., Seattle, Washington, USA, pp. 262–267

Chang C and Hsu C (1998) Integrating query expansion and conceptual relevance feedback for personalized Web information retrieval. In: Proceedings of the seventh international conference on World Wide Web 7, Elsevier Science Publishers B. V., Brisbane, Australia, pp. 151–173

Chang Y, Choi I, Choi J, Kim M, and Raghavan VV (2002) Conceptual Retrieval Based on Feature Clustering of Documents, Workshop on Mathematical/Formal Methods in Information Retrieval at the 25th Annual International ACM SIGIR Conference on Research and Development in IR, in Tampere, Finland, August 15, pp. 89–104

Chang Y, Kim M, and Ounis I (2004) Construction of query concepts in a document space based on data mining techniques. In: Proceedings of the 6th International Conference On Flexible Query Answering Systems (FQAS, 2004), Lecture Notes in Artificial Intelligence, Lyon, France, June 24–26, pp. 137–149

Cormen TH, Leiserson CE, Rivest RL, and Stein C (2001) Introduction to algorithm. Second Edition. MIT Press, McGraw-Hill, New York, NY

Deerwester S, Furnas G, Landauer T, and Harshman R (1990) Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science 41(6):391–407
Edmundson HP (1969) New Methods in Automatic Abstracting. Journal of the ACM 16(2):264–285
Fellbaum C (1998) WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, Mass; London
Frakes WB and Baeza-Yates R (1992) Information Retrieval: Data Structures and Algorithms. Prentice Hall, Englewood Cli#s, NJ
Gersho A and Gray R (1992) Vector quantization and signal compression. Kluwer Academic Publishers, Dordrecht, Netherlands
Han C, Fujii H, and Croft WB (1994) Automatic Query Expansion for Japanese Text Retrieval. UMass Technical Report
Harman D (1995) Overview of the Third Text REtrieval Conference. In: Proceedings of Third Text REtrieval Conference, pp. 1–19
Hawking D, Thistlewaite PB, and Harman D (1999) Scaling Up the TREC Collection. Information Retrieval 1(1–2):115–137
Kim M, Alsaffar AH, Deogun JS, and Raghavan VV (2000) On Modeling of Concept Based Retrieval in Generalized Vector Spaces. International Symposium on Methodologies for Intelligent Systems, pp. 453–462
Kim M, Lu F, and Raghavan VV (2000) Automatic Construction of Rule-based Trees for Conceptual Retrieval. In: Proceedings of SPIRE2000, A Coruna, Spain, IEEE Computer Society Press, pp. 153–161
Klink S (2001) Query reformulation with collaborative concept-based expansion. In: Proceedings of the First International Workshop on Web Document Analysis (WDA2001), Presentation I: Content Extraction and Web Mining. Seattle, WA, USA, pp. 19–22
Koenemann J (1996) Supporting interactive information retrieval through relevance feedback. SIGCHI: ACM Special Interest Group on Computer-Human Interaction. ACM Press, New York, NY, USA, pp. 49–50
Lam-Adesina AM and Jones FJG (2001) Applying summarization techniques for term selection in relevance feedback. In: Proceedings of the 24th Annual International ACM SIGIR Conference. ACM press, New Orleans, Louisiana, USA, pp. 1–9
Leuski A (2001) Evaluating Document Clustering for Interactive Information Retrieval. In: Proceedings of 10th International conference on Information and Knowledge Management (CIKM'01), ACM Press, Atlanta, Georgia, USA, pp. 33–40
Luhn HP (1958) The automatic creation of literature abstracts. IBM journal of research & development 2(2):159–165
Mannila H (2002) Global and local methods in data mining: basic techniques and open problems. In: Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP 2002), Springer-Verlag, Malaga, Spain, pp. 57–68
McCune BP, Tong RM, Dean JS, and Shapiro DG (1985) RUBRIC: A System for Rule-Based Information Retrieval. IEEE Transaction on Software Engineering 11(9):939–945
Nakata K, Voss A, Juhnke M, and Kreifelts T (1998) Collaborative concept extraction from documents. In: Proceedings of the 2nd International Conference on Practical Aspects of Knowledge management (PAKM 98), Basel, Switzerland, pp. 29–30
Pelleg D and Moore A (2000) X-means: extending K-means with efficient estimation of the number of clusters. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000), Morgan Kaufmann, Stanford, CA, USA, pp. 727–734
Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137
Qiu Y and Frei HP (1993) Concept based query expansion. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and Development in Information Retrieval. ACM Press, Pittsburgh, Pennsylvania, USA, pp. 160–169
Quaresma P and Rodrigues IP (2000) Automatic Classification and Intelligent Clustering for WWWeb Information Retrieval Systems. The Journal of Information, Law and Technology (JILT). http://elj.warwick.ac.uk/jilt/00-2/quaresma.html (visited April 7th, 2004)
Robertson S (1990) On term selection for query expansion. Journal of Documentation 46:359–364
Robertson S, Walker S, Jones S, Hancock-Beaulieu M, and Gatford M (1994) Okapi at TREC3. In: Proceedings of the overview of the Third Text REtrieval Conference, pp. 109–125
Rocchio JJ (1971) Relevance feedback in information retrieval in the SMART system. Prentice Hall, Englewood Cliffs, NJ, pp. 313–323
Salton G and Buckley C (1990) Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41(4):288–297
Sparck Jones K and Tait JI (1984) Automatic search term variant generation. Journal of Documentation 40:50–66

Tombros A and Sanderson M (1998) Advantages of Query Biased Summaries in Information Retrieval. In: Proceedings of Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. Melbourne, Australia, pp. 2–10

van Rijsbergen CJ (1979) INFORMATION RETRIEVAL: 2nd Edition. Butterworths, London. http://www.dcs.gla.ac.uk/Keith/Preface.html (visited April 7th, 2004)

Willett P (1988) Recent trends in hierarchic document clustering: a critical review. Information Processing and Management: An International Journal 24(5):577–597

Wong W and Fu A (2000) Incremental document clustering for web page classification. International Conference on Information Society in the 21st century: emerging technologies and new challenges (IS2000), Fukushima, Japan, 2000. pp. 5–8

Xu J and Croft WB (1996) Query expansion using local and global document analysis. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press, Zurich, Switzerland, pp. 4–11

Zhang T, Ramakrishnan R, and Livny M (1996) BIRCH: An efficient data clustering method for very large databases. ACM SIGMOD Record 25(2):103–114