Available online at www.sciencedirect.com
SCIENCE DIRECT®
Applied Soft Computing 6 (2006) 372–383
Applied Soft Computing
www.elsevier.com/locate/asoc

Internet search: Subdivision-based interactive query expansion and the soft semantic web

Maria Chli, Philippe De Wilde *

Department of Electrical and Electronic Engineering, Imperial College, London, UK

Abstract

The expansion of the Internet has made the task of searching a crucial one. Internet users, however, have to make a great effort in order to formulate a search query that returns the required results. Many methods have been devised to assist in this task by helping the users modify their query to give better results. In this paper we propose an interactive method for query expansion. It is based on the observation that documents are often found to contain terms with high information content, which can summarise their subject matter. We present experimental results, which demonstrate that our approach significantly shortens the time required in order to accomplish a certain task by performing web searches.

Keywords: Internet search; Query formulation; Search process; Clustering; Soft semantic web

1. Introduction

The growth of the Internet and the increasing availability of online resources have stimulated interest in the field of information retrieval. Information retrieval concentrates on developing algorithms to locate and select documents from a corpus of material that are relevant to a given query. The development of online information retrieval tools, such as search engines many of which utilise hyperlink analysis [13], has been greatly beneficial to Internet users. However, many of the users find the current process of searching the web unsatisfactory. This dissatisfaction is not necessarily attributed to the search engine program as much as to the inability of the user to formulate the appropriate query. A user has often only a vague idea of what the relevant query terms may be and has to rely on an iterative process in which the retrieved query results are used to formulate the next query. Interactive query expansion, to which this paper contributes, is an approach that attempts to assist in this iterative process.

We have implemented a system, which proposes to the users of a search engine terms that could potentially enhance the search results and lead them more quickly to the target documents. This is done by post-processing the results the search engine produces. Terms from the text of the results are identified based on their information content resulting to reduction of uncertainty about the type of document required. We

* Corresponding author.
E-mail addresses: m.chli@imperial.ac.uk (M. Chli), p.dewilde@imperial.ac.uk (P. De Wilde).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2005.11.003

propose a method that utilises fuzzy logic to present the terms suggested for query expansion on a semantic web.

Experiments carried out with human users have shown that the terms proposed for expansion by our system, can significantly shorten the session of a web search. These experiments together with the results they produced are presented in Section 4.

1.1. Query expansion

Research by Spink et al. [29] has shown that most search engine users typically formulate very short queries of two to three words. Such short queries lack many useful words and do not sufficiently describe the subject that the user wants to search on. In the same work it is suggested that web users tend to go more often from broad to narrow formulation in queries since the most common query modification is to add terms.

The aim of query expansion is to propose possible terms to add to the user's initial query so that the quality of the retrieved results is improved.

A number of methods for performing query expansion have been developed. An extensive review on the subject has been produced by Efthimiadis [8].

Early methods involved extracting terms from thesauri [10,31] but as these proved to be labour-intensive, researchers turned to methods like lexical co-occurrence [30] and clustering [14,17,7,15]. Lexical co-occurrence is the process of developing relationships between words based upon their co-occurrence in documents. In clustering, documents that share a significant number of terms are grouped together and representative words from each cluster are used for the expansion of the original query. Most systems, however, that used clustering for query expansion reported rather pessimistic conclusions on their performance [8]. The similarity of the method proposed here with the methods of lexical co-occurrence and clustering is that the source, which provides the candidate terms for the expansion is the set of the retrieved documents as opposed to some knowledge structure, as is the case with the thesaurus-based approaches. As a consequence, if the user chooses terms that do not yield results from the expected domain, the terms a query expansion algorithm will suggest are not likely to be helpful to the user. Coping with this situation without employing strong prior information is too difficult and will not be dealt with in this work. See Ref. [3] for a review of knowledge-based techniques for query expansion.

In this paper it is assumed that the user provides an initial search string, which is fairly general and yields results that contain the required class of documents as a subset.

Another method to perform query expansion, perhaps the most effective of all, is that of relevance feedback [25,11,21]. In this method the user submits a query, which yields an initial set of results. From this set she selects a number of documents believed to be relevant. The system expands the query based upon the terms in the selected documents. Despite the significant improvement in the quality of results this method produces, the research carried out by Spink et al. [29] shows low use of the relevance feedback facilities provided in search engines. The low use should not necessarily be attributed to the interactive nature of this method. Sometimes when a user has already found a set of relevant documents they may not wish to expand the query further. Also, relevance feedback algorithms are only useful when relevant documents are returned within the top ranked documents of the results. The method we propose does not have this drawback as it examines a large number of the documents retrieved, much larger than what a human user would realistically be able to examine, to propose the discriminatory terms that will lead to the relevant documents.

More recent methods to perform query expansion involve mining user logs [6] and constructing user profiles [23]. Preliminary results produced by these methods have been promising. The major disadvantage, however, of the methods that rely on implicit user cooperation is the issue of privacy [5].

Fuzzy query [4], fuzzy grammar [33] and the soft semantic web [19] have recently been used in information retrieval. Additionally, there has been work that utilises fuzzy association rules [22] to perform query expansion. Soft computing seems a promising alternative; however the contributions cited above have not yet presented results that compare their performance to those of crisp methods. In this work, we construct a soft semantic web in which the proposed terms for query expansion are associated with each other.

Query expansion can be performed manually, interactively or automatically. In interactive and automatic query expansion, the candidate additional terms are provided by the system. The difference between the two methods is that for the interactive method the selection of the terms that are actually added to the query is a decision taken by the user as opposed to the system as is the case for the automatic method. Magennis and van Rijsbergen [18] and Ruthven [26] as well as others, have tried to evaluate and compare the efficiency of the two methods. However, in most of the cases, their experiments were based on simulations and not on real human users, with the exception of the 'inexperienced' users experiment in Ref. [18]. Nevertheless the results of the experiments showed that interactive query expansion has a potential to be an effective technique. Even if these results are to be disregarded, the interactive method gives more control to the searcher who knows her utility better than any automated system.

In this paper we present an interactive method for query expansion. It is founded on the fact that documents contain terms with high information content that can summarise them and can be used to complement a general query string, to optimally reduce the search space for subsequent queries. We demonstrate query expansion results that testify to the validity of our approach.

The rest of the paper is laid out as follows: In Section 1.2 the rationale behind the proposed method is laid out. Section 2 gives details of the metrics used to assess the information content of each word. Subsequently, in Section 3 the implementation of the system is described. In Section 4 the experiments conducted for the evaluation of the proposed method are reported.

1.2. Discriminative document terms

Our fundamental observation is that documents contain *discriminative words* that can summarise the document content. This is particularly true for the majority of commercial web sites whose authors try to condense the content of the page in small, effective messages. The common characteristic of these discriminative words is the rarity of their occurrence in the corpus of all documents in the database. Borrowing from information theoretic concepts, we can say that an infrequent word is normally associated with high information content. This implies that such words are normally very good candidates for expanding a query with the exception of two situations. Namely, when the discriminative word is common within the set of the retrieved documents, although it is rare in the corpus, or it is extremely rare in the set of retrieved documents as well as in the corpus. In the first case, that word obviously does not carry any discriminatory power and in the second case the word is likely to have appeared by chance.

A word, which falls in one of these two situations is probably not a good candidate for query expansion and should therefore be somehow disqualified. If this intuition is used for expanding the query then these two extremes should be disregarded. In the next section we formalise these intuitions and motivate the term selection model used.

2. Term value

The method proposed is to assign a weight w to each word that appears in the retrieved documents.

Define for a set of documents S and a term t, the set $S(t) \subset S$ consisting of the documents that contain t. Assume we are working within a corpus of documents C. After the user submits the initial query, a set of documents $R \subset C$ is retrieved. Our basic assumption (justified experimentally in Ref. [29]) is that the initial query will be general enough to contain the required document in its results. The goal for the system is then to produce a candidate term for expanding the query. The user is then asked about the relevance of the candidate term t, say, and gives back a "yes"/"no" answer, essentially selecting between the set of documents which contain t, $R[t]$, and the set of documents which do not contain t, $R - R[t]$. We require a good candidate term to have the following two properties:

1. The expected reduction in uncertainty as a result of the user's choice should be maximised;
2. It must be clear to the user, whether this term is relevant or not with the target document.

The first property can be quantified using entropy. Before the inclusion/exclusion of term t by the user,

the required document is one of $|R|$ equiprobable documents. So the associated entropy of this is

$$H_{\text{init}} = |R| \times \left(-\frac{1}{|R|} \times \log \frac{1}{|R|} \right) = \log |R|. \quad (1)$$

When the user is asked to include or exclude documents containing the term t, entropy will be reduced to either $\log |R[t]|$ or $\log |R - R[t]|$. If the two possible replies are considered equiprobable, the mean reduction in entropy caused by t will be

$$\Delta H_t = \log \frac{|R|}{|R[t]|^{1/2} \times |R - R[t]|^{1/2}}. \quad (2)$$

The second property is harder to measure theoretically and yet it is a crucial part of the system. Used on its own, the first property will produce a multitude of candidate terms which may provide good entropy reduction but which may not be helpful to the user. The heuristic we apply here is the specificity part of the traditional TF × IDF scheme (IDF term), which is given by

$$\text{IDF}_t = \log \frac{|C|}{|C[t]|}. \quad (3)$$

This measure is known in work [27] using TF. IDF to favour rare, information-rich terms. In our case it acts as an artificial negative bias that will eliminate possible candidate terms that happen to satisfy the first requirement purely by chance. A potential problem of this approach is that it may also bias candidates towards rare, over-technical and potentially unfamiliar terms for the user. In practice, for the document sets tested in this work this effect is minimal.

The two terms described above are combined into a single term as

$$w(t) = \Delta H_t \times \text{IDF}_t. \quad (4)$$

3. Implementation

In summary the algorithm proposed proceeds as follows:

1. The user submits a query to a search engine;
2. The first K results the search engine returned are collected and parsed;
3. Stemming is performed on every term that appears in the collection;
4. The value of every term is calculated, as discussed in Section 2;
5. The terms are sorted according to their cost and a list of the first N of them is presented to the user, as candidate terms for expansion.

In the rest of this section we discuss the algorithm in more detail.

3.1. Initial phase

The procedure is started by the user typing in a query. This query is then executed by a search engine. Since our method is applied during a post-processing phase, it can be used with any information retrieval system, which returns a list of documents. Google [1] was used during the experiments we carried out. Stemming and stop word removal on the query were left to the search engine to do. Subsequently, the pages returned by the search engine are read in and parsed, in order to extract from them their actual textual content lemma by lemma. For the experiments described in Section 4, 200 result pages were read in each time. Multiple occurrences of a lemma within the same document are not taken into account.

3.2. Stemming

The next step is to perform stemming on the terms of the retrieved documents. A stemming algorithm tries to reduce a word variant to its root form. The root is the form of a word after all affixes have been removed. We have implemented a form of the KSTEM stemmer proposed by Krovetz [16] which works by removing suffixes from a word variant piece by piece and after each removal looks up the reduced word in the dictionary. In our version we use WordNet, a lexical database for the English language [20]. Krovetz originally used LDOCE, the Longman's Dictionary of Contemporary English [2]. The basic idea is that, if a word is in the dictionary, this means that it has a different meaning from its root and it should not be stemmed further. For example the verb 'fought' will be reduced to 'fight' and the noun 'fairies' to 'fairy', however it will not be further reduced to the word 'fair'. Additionally, the word

'cooked' will not be reduced to 'cook' as WordNet recognises it in its initial form as an adjective. KSTEM has been found to improve retrieval effectiveness over the well-known Porter stemmer [24], which does not stop stemming until it reaches the root of a word [28].

This practice of reducing all the encountered lemmas to their root form ensures that words with the same stem, meaning and information content like 'memorise', 'memorised' and 'memorising' will not be counted twice in the same document.

3.2.1. Common word filtering

As laid out in Section 2, where the term value calculation is described, an important parameter is the information quantity of a word; a measure of its rarity in the corpus of all documents. An alternative, simpler approach would be to employ a stop list of very common words. While this was also seen to produce good results, using a database of term frequencies in the corpus and calculating the information quantity of each encountered term is a more elegant approach that allows for smooth filtering. A term that is quite commonly used in the corpus, but is used even more frequently in the collection of retrieved documents is potentially one of high significance in the domain that is being searched. It is also likely to be beneficial if used for the expansion of the initial query. A stop list would eliminate such a word, whereas the use of a database allows it to proceed.

In the initial stages of the implementation, the frequency statistics database used was the Brown Corpora list of 2000 most common English words [9]. This list has been compiled from a large number of English literary texts. Although it did filter out numerous of the most common words of the English language like 'the' and 'a', it allowed words like 'link' and 'email' to occupy the top of the lists of words suggested for query expansion for most of the experiments carried out. As one would probably expect, the vocabulary used in English texts found on the web is somewhat different to what is used in the literary texts. In order to overcome this problem a list of common words encountered on the web, similar to Brown's had to be specially compiled from English web pages. The employment of this list as the frequency statistics database of the system greatly improved the results of the experiments performed.

3.3. Term selection

As soon as the user has entered their query, the results have been returned from the search engine and all the encountered lemmas have been read in, the score is calculated for each word in the retrieved documents, as described in Section 2.

The user is subsequently presented with the candidate words clustered according to the documents they appear in. The user selects the lemmas she sees fit, which are then added to the original query with the connectives AND, or AND NOT. This process is repeatedly applied to any subsequent set of results.

4. Evaluation

In order to evaluate the performance of the system, a set of experiments has been carried out. Human users have been asked to perform a series of tasks by performing web searches. The Google [1] search engine has been compared to the system we propose and a significant decrease in the length of the web sessions has been recorded. This section describes the experiments in detail and lays out the results they produced.

4.1. Experiments

As stated in Section 1.1, the algorithm proposed operates under the assumption that the initial search string, which is provided by the user of the system, is fairly general and yields results that contain the required class of documents as a subset. This assumption is supported by research carried out by Spink et al. [29], which suggests that web users tend to go more often from broad to narrow formulation in queries since the most common query modification is to add terms. The system we propose will process the retrieved documents and will propose to the user a set of words to expand the initial query, narrow down the retrieved documents and eventually, after a number of iterations, isolate the required subset of results.

One way to evaluate the system is to test the ability of the proposed algorithm to identify discriminative words, which reflect the existence of discrete subsets of documents encapsulated in the collection of retrieved results. A method to do this is to form

queries, which can be semantically interpreted in more than one way. Such queries are expected to yield several groups of documents, each group corresponding to a different sense of the query. It will be interesting to see whether the implemented system actually suggests words from the retrieved documents that reflect those distinct meanings.

For example when our users were given the task ''Find two web sites where several British customs are described'' many of them formed the query 'British customs'. This query can be interpreted in at least two ways. The word 'customs' may mean a 'specific practice of long standing', or it may mean 'money collected under a tariff' [20]. Performing a trial search on Google one may see results related to HM Customs and Excise web page, news articles that report recent activities HM Customs have been involved in, web pages that describe British traditions, etc. When the above query is given as input to the implemented system and the first 200 results that Google returns are read in and processed, the first 30 suggestions the user is presented with are the following: tradition, excise, include, carry, issue, create, HM, comment, party, condition, medium, regard, cigarette, department, duty, thousand, organisation, follow, source, parliament, soldier, country, investigation, serve, traveller, citizen, law, drug, nation, and vehicle.

The word tradition stands on the top of the list representing one of the interpretations of the query and it is closely followed by the word excise, and a little further down the word HM, which are associated the second interpretation. The user selects the word 'tradition'. This time the user does not need to incorporate any more terms to the query as the search engine returns documents that are of interest to the user on the top of the first page of results.

In the experiment described above, only two iterations were enough to prune the enormous amount of retrieved documents and to lead the user to the documents she was looking for. Spink et al. [29] in the research they carried out suggest that the mean and median number of queries per user session were 4.86 and 8, respectively. If we assume that during a session a user enters an initial query, observes the results, then modifies the original query and tries again, this statistic shows that the user has to do quite a few query modifications before they are presented with the required documents.

4.1.1. Experimental setting

The evaluation of the system was not based solely on ambiguous queries. For a wider assessment we used the tasks defined for the Interactive Track of TREC-10 [12,32] plus some others defined by us. The searches were run on the WWW instead of a precompiled collection, to provide for a more realistic setting.

In total, 24 subjects participated in our experiments. All subjects were educated to graduate level and were recruited from various departments of the University of Cambridge, including Engineering, Classics, Computer Science, Physics, Biological Sciences and Medicine. The average age of the subjects was 27.06 with a standard deviation of 5.27 years. All users used computers and the Internet frequently as part of their work. The average experience of online searching among the subjects was 6.38 years. All users cited Google as their favourite search engine. Native English speakers were the 40% of the users. Examples of the tasks used in the experiments are shown below. The full list of the 20 tasks is given in Appendix A.

1. Name three features to consider in buying a new yacht;
2. Find a book with lots of information for a high school report on the history of London;
3. Find two web sites where several British customs are described;
4. Identify two rules of American and two of European football;
5. Name one of the dangers jaguars face;
6. Name a bond considered among the strongest in 2002.

Each user was asked to complete a set of 10 tasks selected randomly from the list above. These were in turn randomly divided in two groups of five tasks, one group to be completed using Google and the other using Google and query expansion terms suggested by the proposed system. A researcher observed each user while carrying out their assigned tasks and noted the queries they used, the number of iterations it took them to complete each task and the number of documents they viewed. The users were allowed to read the summaries of

the documents Google provides, but only the sites they clicked on counted towards the documents viewed.

4.2. Evaluation results

As can be seen in Table 1 the proposed system diminishes significantly the average number of iterations and documents per user session in comparison to Google.

The fact that Spink et al. [29] recorded much longer user sessions in their work (the mean and median number of queries per user session were 4.86 and 8, respectively) for sessions carried out using Excite than we recorded for tasks carried out on Google may be due to several reasons. Apart from the search engine used being different, the users in our experiments were all of a graduate level of education and had quite some experience in web searching. In addition, the tasks the users were given were fairly well specified. In many cases users search without having in mind a complete specification of the problem they are trying to solve.

In Table 2 some of the queries used by users for the execution of the given tasks are presented as well as the first 30 words that the system suggested for expansion of each one of the queries. These words give an overview of the set of retrieved documents without the user having to look at the documents one by one and make for shorter user sessions.

We use a variant of *k*-means clustering in order to group the retrieved words according to the documents they appear in together. In this technique, instead of a mean of a cluster of terms, we used the term in the cluster with the minimum average distance to other terms in the cluster. The distance between two terms was given by the number of documents containing both terms. In the two tables the clusters are shown in different rows.

Below, a few interesting aspects about some of the results are pointed out.

Table 1
Average iterations and documents viewed per user session

	Google	Proposed system
Average iterations	1.9375	1.425
Average documents viewed	4.2	1.8625

Table 2
Tested queries and suggested expansion terms

Initial query	Suggested terms for expansion
British customs	Excise, carry, issue, create, HM, comment, condition, media, cigarette, department, duty, thousand, organisation, source, soldier, investigation, traveller, citizen, law, drug, vehicle
	Tradition, include, party, follow, serve, parliament, country, regard, nation
Book London	Booking, offer, reservation, attraction, discount, airport, deal, travel, budget, rooms, facility, rooms, facility, luxury, tour, ticket, park, secure, star, city, night, accommodation, garden, shop, stay, apartment, restaurant, street, map, location
	ISBN, publisher
Football rules	Winner, overtime, scores, tie, played, interception, playoff, playing, guard, kickoff, offence, consist, pick, scoring, touch, completion, quarterback, tied, punt, roster, touchdown, defensive, injury, fumble, sport, defence, foot, coin, regulation
	Soccer
Jaguar	Car, feature, picture, photo, club, logo, engine, frame, racing, enthusiast, join, drive, tip, sport, event, classic, rally, driver, jag, model, tail, forum, item, driving, preview, video, speed
	Cat, tail, panther
Bond 2002	TV, villain, actor, gadget, EON, Thanksgiving, soundtrack, upload, autograph, spy, poster, Jinx, trailer, boxed, install, DVD, marathon, baddie, stunt, convertible
	Investor, coupon, portfolio, treasuries, headline, yield, upgrade, swap, dividend
	Dalton
Yacht	ft, meters, frame, schooner, specification wooden, swan, sabre, boatbuilder, keelboat, creek, craft, motor, sailing
	Club, cruise, sail, racing, regatta, bay, event, marina, fleet, sailor, ocean charter, marine, crew, sea, championship

'*Book London*' *query*. The results are dominated by sites which advertise their services on hotel and flight bookings and reservations. The terms 'ISBN' and 'publisher' though, are an indication that a few sites on books on London have been retrieved as well.

'*Football rules*' *query*. The list of results is flooded with football terminology and this was to be expected, however it is interesting to see the word 'soccer' included.

'*Jaguar*' *query*. The results are mostly populated by the most commercial of the meanings, that of the car. Nevertheless the presence of the terms 'cat, 'tail' and 'panther' depicts the existence of a number of web sites that refer to the animal.

'*Bond 2002*' *query*. The query yields an interesting mix of results. This is due to the multiple meanings of the word bond. The majority of the keywords in the list is related to the James Bond film: villain, actor, gadget, EON (the producing company), Jinx (a character in the film), convertible (referring to a car in the film), etc. The words Thanksgiving and marathon, although looking unrelated to the subject, refer to a James Bond film broadcasting marathon to be screened by a US television network on Thanksgiving day, an event which as is apparent from the retrieved documents, had been heavily advertised in sites related to the film. One of the suggested words is associated with the chemical bond; the name Dalton refers either to John Dalton the English chemist and physicist who formulated atomic theory or Timothy Dalton, an actor in the film. The rest of the results reflect the financial meaning of the term bond: investor, coupon (as in coupon rate), treasuries, yield, swap and dividend. The word headline is very common in sites, which refer to financial bonds and this is the reason it appears in the top 30 suggestions.

'*Yacht*' *query*. The terms that the system proposes to the users for expanding the query, in a way answer the question that was given in the task: ''Name three features to consider in buying a new yacht''. One of the clusters is populated by terms, which are either features of yachts or different types and brands of them. As it is apparent from the contents of the other cluster, there is a group of websites retrieved that concern regattas and racing.

5. Using the soft semantic web to present expansion terms

The algorithm, set out in Section 3, gives as output a list of terms that the user can use to expand their query. In Section 4, a clustering technique similar to *k*-means was used in order to group this list of words and make the task of selecting the appropriate term for expansion easier for the user. In this technique, the mean of a set of terms is the term with minimum average distance of all the rest. The distance between two terms was given by the number of documents containing both terms.

While this method produces intuitive results, it relies upon the user specifying the number of clusters, which is inconvenient in practice. We now show an alternative way to present the proposed expansion terms, based on the soft semantic web described in Martin and Azvine [19]. User-friendliness can be improved by building a network of nodes that contain information. The nodes are connected by links that indicate similarity. The similarity measure is based on co-occurrence of terms in documents. The network constructed is called a semantic web, because its links give meaning to the nodes. Attaching meaning to words is something hard for a computer, especially when it is done without the use of a thesaurus. We construct such a semantic web here, using fuzzy logic to deal with uncertainty.

5.1. Weights and similarity

First, weights are attached to terms in a document. One possible formula for the weighting would be the TF × IDF weighting. This weight consists of two factors. The first, term frequency (TF), determines the normalised frequency of the term in a document. The second part is the inverse document frequency (IDF), that measures how important the term is in the whole set of documents. It acts as a negative bias that decreases the weighting of terms that happen to be very frequent in the corpus.

Putting these two together we get the following:

$$\alpha_{i,j} = f_{i,j} \times \log_2 \frac{|C|}{|C_i|} \tag{5}$$

where $\alpha_{i,j}$ weighting of term i in document j, $f_{i,j}$ normalised frequency of term i in document j, $|C|$ number of documents in the corpus, and $|C_i|$ number of documents in the corpus that contain term i.

The normalised frequency $f_{i,j}$ will allow us to obtain weightings that consider the size of the document.

To normalise the frequency we divide it by the frequency of the most common word in the document.[1]

$$f_{i,j} = \frac{\text{freq}_{i,j}}{\max(\text{freq}_{x,j})}, \tag{6}$$

where x represents the most frequent term in the document.

The match between the term vector and the document can now be established in the following way. Define the number of unique words in the document collection as n. Define the term vector $\mathbf{t} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, each term with a weighting α_i and the document vector as $\mathbf{d}_j = (\alpha_{1,j}, \alpha_{2,j}, \ldots, \alpha_{n,j})$ where j is the index of the document and $\alpha_{i,j}$ is the TF–IDF weighting of unique term i in document j. Having obtained the two vectors the match between them is defined as the *similarity*:

$$\text{sim}(\mathbf{t}, \mathbf{d}_j) = \frac{\mathbf{t} \cdot \mathbf{d}}{|\mathbf{t}| \times |\mathbf{d}_j|}, \tag{7}$$

i.e. the cosine of the angle between the two vectors.

Using the definition of the document and term vectors as well as the definition of the dot product and magnitude we can obtain a formula in terms of the weightings:

$$\text{sim}(\mathbf{t}, \mathbf{d}_j) = \frac{\sum_{i=1}^{n} (\alpha_{i,j} \times \alpha_i)}{\sqrt{\left(\sum_{i=1}^{n} \alpha_{i,j}^2\right)} \times \sqrt{\left(\sum_{i=1}^{n} \alpha_i^2\right)}}, \tag{8}$$

where $\alpha_{i,j}$ and α_i are the elements of the document and term vector, respectively and n is the number of terms in the term vector.

5.2. Computational procedure to calculate degrees of specialisation, generalisation and equivalence

Considering the previous two definitions and the customary use of answer sets in terms of fuzzy sets, the following procedure is derived:

1. Define the term vector ($\mathbf{t}$) and the document vector ($\mathbf{d}_j$) as previously described.
2. Compute the similarity based on the degree of match formula that uses the term and document vectors, $\text{sim}(\mathbf{t}, \mathbf{d}_j)$, for all n documents with $j = 1, 2, \ldots, n$.
3. Define the set T_1 of n similarity values as the membership values for a term.
4. Obtain a second set T_2 of membership values of the documents for another term.
5. Define the intercept set $(T_1 \cap T_2)$ of the two sets as $\min(\mu_{T_1}^j, \mu_{T_2}^j)$ and the union set $(T_1 \cup T_2)$ as $\max(\mu_{T_1}^j, \mu_{T_2}^j)$, where μ_A^j indicates the jth membership value of set A.
6. Define $\text{card}(A) = \sum_j \mu_A^j$.
7. If the number of non-zero membership documents (N_{T_1}) of term T_1 is greater than the number of non-zero membership documents (N_{T_2}) of term T_2, then T_1 is defined as being more general than T_2 In the case of $N_{T_2} = N_{T_1}$ we use other rules such as which T_i has the largest $\text{card}(T_i)$. In the case where $\text{card}(T_1) = \text{card}(T_2)$ generalisation and specialisation will have the same values for both sets in both directions and thus choosing which is more general/specialised is arbitrary.
8. We finally define:

   $$\text{equivalence of } T_1 \text{ and } T_2 \text{ as } \frac{\text{card}(T_1 \mathbf{I} T_2)}{\text{card}(T_1 \mathbf{Y} T_2)}.$$

   If $N_{T_1} > N_{T_2}$ then,

   $$T_1 \text{ is a generalisation of } T_2 \text{ to the degree } \frac{\text{card}(T_1 \mathbf{I} T_2)}{\text{card}(T_2)},$$

   and

   $$T_2 \text{ specialises } T_1 \text{ to the degree } \frac{\text{card}(T_1 \mathbf{I} T_2)}{\text{card}(T_1)}.$$

The definition for $N_{T_2} > N_{T_1}$ is the same but with T_1 and T_2 swapped in the formulae.

Using these definitions we can build a soft semantic web of the terms that the system proposes to the user to use for expansion with degrees of specialisation, generalisation and equivalence between the nodes.

5.3. Constructing the soft semantic web

The following example is based on the query "British customs". A set of 200 documents has been retrieved and stored for analysis. The algorithm described in Section 3 was used to extract from the retrieved documents 30 terms that are suitable for

[1] Another way to normalise the term frequency is to divide it by the total number of terms in the document.

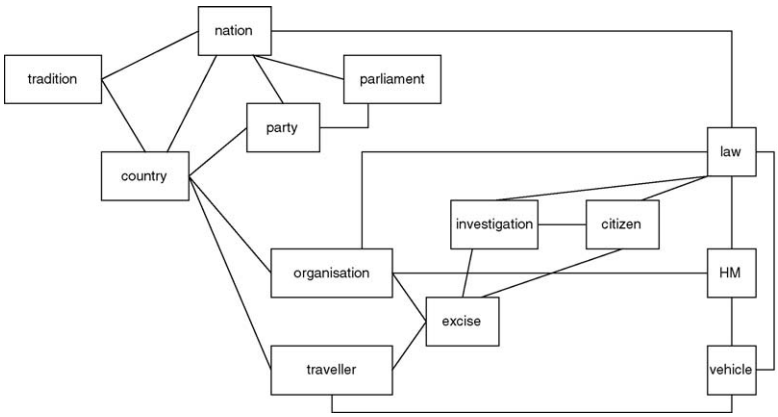

Fig. 1. Soft semantic web showing *equivalence* relations. The nodes are the suggested terms for expansion of the query ‘‘British customs’’. The clusters formed around the term ‘‘tradition’’ and around the terms ‘‘HM’’ and ‘‘excise’’ reflect the dual meaning of the term ‘‘customs’’ of the initial query.

expansion of the initial query. A soft semantic web can be created using the retrieved documents, the proposed terms and the definitions of generalisation, specialisation and equivalence of the previous section.

Below, in Fig. 1, we show such a semantic web, which depicts the *equivalence* relations among the proposed terms. To construct the semantic web the values of equivalence have been calculated for each possible pair of the proposed terms. The user may define the accepted levels of generalisation, specialisation and equivalence. The semantic web shown associates nodes whose value for equivalence was greater than 0.5.

The dual meaning of the term ‘‘customs’’ in the initial query is reflected in the semantic web. The nodes are arranged in two fairly distinct clusters, one around the term ‘‘tradition’’ and another one around the terms ‘‘HM’’ and ‘‘excise’’.’

In Fig. 2, a soft semantic web, which illustrates the *specialisation* relations among the proposed terms, is

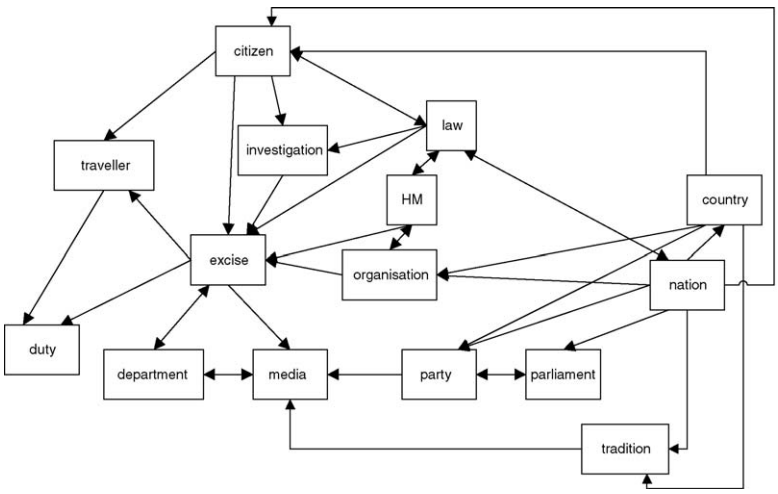

Fig. 2. Soft semantic web showing *specialisation* relations. The nodes are the suggested terms for expansion of the query ‘‘British customs’’. An arrow leads from a more general to a more specialised term. The existence of double arrows shows possible equivalence between the linked nodes.

shown. An arrow leads from a more general to a more specialised term. The existence of double arrows shows possible equivalence between the linked nodes. Filtering of the most important relations was performed in this instance of the semantic web as well.

6. Potential limitations

As with any query expansion method that uses the set of retrieved documents as their source from which to extract candidate terms for expansion, this method requires that the user inputs an initial query which is reasonably broad. If the set of retrieved results does not contain as a subset the documents the user is interested in, the words that will be suggested will not be useful for query expansion. However, they may still serve as a summary of the retrieved results and help the user to reformulate the original query.

The method proposed can be further refined by including a counter-bias against over-technical and potentially unfamiliar terms to the users that might be favoured by the IDF component of the term value, as discussed in Section 2.

7. Conclusions

In this paper we have presented an interactive method for query expansion based on subdivision. The method is founded on the fact that documents contain some terms with high information content, which can summarise their subject matter. These terms are extracted from the collection of the retrieved results. The information quantity they carry as well as their ability to prune down the search space is then evaluated. The top ranking words are presented to the user as a list of candidate terms for expansion, or on a semantic web constructed using term collocation in the retrieved documents and fuzzy logic.

Evaluation of the proposed system with human users who were asked to complete a set of tasks utilising a search engine produced promising results. Tasks carried out on our system were compared with others carried out on Google alone. In terms of both the number of iterations and the number of documents viewed until each task was completed, the user sessions recorded on the proposed system were significantly shorter.

Acknowledgements

We would like to thank the people who participated in our experiments. Especially we would like to thank M. Dimitriadi for her invaluable help. The second author thanks Amir Eftekar for work on the fuzzy semantic net.

Appendix A

The following are the tasks the users where asked to complete for the experiments:

1. Find a web site likely to contain reliable information on the effect of second-hand smoke.
2. Tell me three categories of people who should or should not get a flu shot and why.
3. List two of the generally recommended treatments for stomach ulcers.
4. Identify two pros or cons of taking large doses of Vitamin A.
5. Get two price quotes for a new digital camera (3 or more megapixels and 2× or more zoom).
6. Find two web sites that allow people to buy soy milk online.
7. Name three features to consider in buying a new yacht.
8. Find two web sites that will let me buy a personal CD player online.
9. I want to visit Antarctica. Find a web site with information on organized tours/trips there.
10. Identify three interesting things to do during a weekend in Kyoto, Japan.
11. Identify three interesting places to visit in Turkey.
12. I would like to go on a sailing vacation in Australia, but I do not know how to sail. Tell me where I can get some information about organized sailing cruises in that area.
13. Find three articles that a high school student could use in writing a report on the Titanic.
14. Tell me the name of a web site where I can find material on global warming.
15. Find three different information sources that may be useful to a high school student in writing a biography of M. Jordan.
16. Find a book with lots of information for a high school report on the history of London.

17. Find two web sites where several British customs are described.
18. Identify two rules of American and two of European football.
19. Name one of the dangers jaguars face.
20. Name a bond considered among the strongest in 2002.

References

[1] The Google Search Engine, 1978, http://www.google.com.
[2] Longman's Dictionary of Contemporary English, New ed., Longman.
[3] R.C. Bodner, F. Song, Knowledge-based approaches to query expansion in information retrieval, in: Proceedings of the Canadian Conference on AI, 1996, pp. 146–158.
[4] D.Y. Choi, Enhancing the power of web search engines by means of fuzzy query, Decis. Support Syst. 35 (1) (2003) 31–44.
[5] R. Cooley, B. Mobasher, J. Srivastava, Data preparation for mining world wide web browsing patterns, Knowl. Inform. Syst. 1 (1) (1999) 5–32.
[6] H. Cui, J.-R. Wen, W.-Y. Ma, Query expansion by mining user logs, IEEE Trans. Knowl. Data Eng. 15 (4) (2003) 829–839.
[7] C. de Loupy, P. Bellot, M. El-Beze, P.F. Marteau, Query expansion and classification of retrieved documents, in: Proceedings of the Seventh Text REtrieval Conference (TREC-7), 1998, pp. 382–389.
[8] E.N. Efthimiadis, Query expansion, Annu. Rev. Inform. Syst. Technol. 31 (1996) 121–187.
[9] W.N. Francis, H. Kucera, Brown Corpus Manual, in: http://helmer.aksis.uib.no/icame/brown/bcm.html1979.
[10] S. Gauch, J.B. Smith, Search improvement via automatic query reformulation, ACM Trans. Inform. Syst. 9 (3) (1991) 249–280.
[11] D. Harman, Towards interactive query expansion, in: Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 1988, pp. 321–331.
[12] W.R. Hersh, P. Over, The TREC-2001 interactive track report, in: Proceedings of the 10th Text Retrieval Conference (TREC-10), Gaithersburg, November, (2001).
[13] J. Hou, Y. Zhang, Effectively finding relevant web pages from linkage information, IEEE Trans. Knowl. Data Eng. 15 (4) (2003) 940–951.
[14] K.S. Jones, Automatic Keyword Classification for Information Retrieval, Butterworth, London, UK, 1971.
[15] M. Shamim Khan, S.W. Khor, Web document clustering using a hybrid neural network, Appl. Soft Comput. 4 (4) (2004) 423–432.
[16] R. Krovetz, Viewing morphology as an inference process, in: Proceedings of ACM SIGIR Conference, 1993, pp. 191–202.
[17] A. Leuski, Evaluating document clustering for interactive information retrieval, in: Proceedings of the 10th International Conference on Information and Knowledge Management, ACM Press, 2001, pp. 33–40.
[18] M. Magennis, C.J. van Rijsbergen, The potential and actual effectiveness of interactive query expansion, in: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 1997, pp. 324–332.
[19] T.P. Martin, B. Azvine, Acquisition of soft taxonomies for intelligent personal hierarchies and the soft semantic web, BT Technol. J. 21 (4) (2003) 113–122.
[20] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[21] M. Mitra, A. Singhal, C. Buckley, Improving automatic query expansion, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 1998, pp. 206–214.
[22] M.J. Marín-Bautista, D. Sánchez, J. Chamorro-Martínez, J.M. Serrano, M.A. Vila, Mining web documents to find additional query terms using fuzzy association rules, Fuzzy Set Syst. 148 (1) (2004) 85–104.
[23] M. Nikravesh, V. Loia, B. Azvine, Fuzzy logic and the Internet (FLINT): Internet, World Wide Web and search engines, Soft Comput. 6 (2002) 287–299.
[24] M.F. Porter, Viewing morphology as an inference process, Prog. Autom. Libr. Inform. Syst. 14 (3) (1980) 130–137.
[25] S.E. Robertson, C.L. Thompson, M.J. Macaskill, J.D. Bovey, Weighting, ranking and relevance feedback in a front-end system, J. Inform. Sci. 12 (1–2) (1986) 71–75.
[26] I. Ruthven, Re-examining the potential effectiveness of interactive query expansion, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 2003, pp. 213–220.
[27] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inform. Process. Manage. 24 (5) (1988) 513–523.
[28] M. Sanderson, Retrieving with good sense, Inform. Retriev. 2 (1999) 47–67.
[29] A. Spink, D. Wolfram, B.J. Jansen, T. Saracevic, Searching the web: the public and their queries, J. Am. Soc. Inform. Sci. Technol. 52 (3) (2001) 226–234.
[30] O. Vechtomova, S. Robertson, S. Jones, Query expansion with long-span collocates, Inform. Retriev. 6 (2) (2003) 251–273.
[31] E.M. Voorhees, Using WordNet to disambiguate word senses for text retrieval, in: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 1993, pp. 171–180.
[32] R. White, J. Jose, I. Ruthven, Comparing explicit and implicit feedback techniques for web retrieval: TREC-10 interactive track report, in: Proceedings of the 10th Text Retrieval Conference (TREC-10), Gaithersburg, November, (2001.
[33] F. Wang, A fuzzy grammar and possibility theory-based natural language user interface for spatial queries, Fuzzy Set Syst. 113 (1) (2000) 147–159.