Semantic Method for Query Expansion in an Intelligent Search System

1 Ejiofor C. I., 2 Williams E. E., 3 Nwachukwu E.O., 4 Theo van der Weide

1 University of Port-Harcourt, River State, Nigeria,
2 University of Calabar, Cross River State, Nigeria,
3 Radboud University, Nijmegen, the Netherlands

1 ejiofor1@ yahoo.com, 2 edemwilliam@yahoo.com, 3 eables@sciences@yahoo.com, 4 tvd@cs.ni.nl

ABSTRACT

Hundreds of millions of users each day use web search engines to meet their information needs. Advances in web search effectiveness are therefore perhaps the most significant public outcomes of IR research. Query expansion methods have been extensively studied in information retrieval despite the recent advances in search quality; the fast increase in the size of the Web collection has introduced new challenges for Web ranking algorithms. In fact, there are still many situations in which the users are presented with imprecise or very poor results. One of the key difficulties is the fact that users usually submit very short and ambiguous queries, and they do not fully specify their information needs. Query expansion is one such method for enhancing user query to improve search engine performance and satisfy the user need. Adaptive query expansion (QE) allows users to better define their search domain by supplementing the original query with additional terms related to their preferences and information needs. In this work we propose a novel semantic method based query expansion technique using WordNet, which allows disambiguating queries submitted to search engines. This technique can be seen to significantly improve search engine performance, particularly recall.

Keywords: Query Expansion, WordNet, Information Retrieval, synonyms, Ontology, Semantic, and Search Engine.

1. INTRODUCTION

Information retrieval aims to find documents that are relevant to a user’s information need. In web retrieval, the need is typically expressed as a query consisting of a small number of words [22], and answer documents are chosen based on the statistical similarity of the query to the individual documents in the collection. The amount of information published on the World Wide Web is growing at an astonishing rate, thus making it necessary to devise effective methods for helping users find what they are looking for [16]. Query expansion (QE) allows users to expand their search domain by supplementing their original query with additional terms and phrases [2], [10].

Much research over several decades has led to development of statistical similarity measures that are reasonably effective at finding answers for even the shortest queries [24]. However, enriching a user's query with synonyms or addition of good additional query terms can improve search performance in a text retrieval system and lead to significant improvements in effectiveness.

Search engines require an input from the user in form of a query consisting of keywords, which have a number of disadvantages: researchers normally are not trained for really comprehensive searching. They may not know all the tricks required to locate the right sources of information to review their literature. They may not know how to formulate a query describing all that they want, thus, researchers pay a high price due to ineffective information discovery [25].

In turn, the idea of taking advantage of additional knowledge, by expanding the original query with other topic-related terms, to retrieve relevant documents has been largely discussed in the literature, where manual, interactive and automatic techniques have been proposed [4]. The idea behind these techniques is that, in order to avoid ambiguity, it may be sufficient to better specify: the meaning of what the user has in mind when performing a search, or in other words “the main concept” (or a set of concepts) of the preferred topic in which the user is interested. A better specialization of the query can be obtained with additional knowledge, that can be extracted from exogenous (e.g. ontology, WordNet, data mining) or endogenous knowledge (i.e. extracted only from the documents contained in the repository) [3], [4].

In this paper, we propose discovering semantically similar terms using WordNet and we perform query expansion by generating lexical paraphrases of queries. These paraphrases replace content words in the queries with their synonyms. The following information sources are used in this process: Semantic information obtained from WordNet [17] and statistical information obtained from our document collection were used as the information source. The statistical information is used to moderate the alternatives obtained from the semantic resources, by preferring query paraphrases that contain frequent word combinations. A probabilistic formulation of the query paraphrases is then incorporated into the vector space document-retrieval model [23].

2. RELATED WORKS

Query expansion is a technique that has been proven to be helpful and is implemented by many search engines [11]. A good example is the Yahoo! Web search engine in 2010. The expanded query suggestion appears just below the search bar [26]. In the Yahoo! Web search engine the query expansion terms are suggestions that they can use or ignore. Many search engines use query expansion automatically, and the users are never aware that query expansion terms are being use to refine their query [9]. These search engine use query expansion
automatically to reduce overhead time that is wasted when users are trying to decide which query expansion terms to use. Search engines that use query expansion automatically are very confident in the terms they choose and usually have complicated algorithms in order to choose their terms wisely. Automatically query expansion tends to improve the average overall retrieval performance by improving certain queries and making it worse on others. Quite often automatic query expansion ends up hurting the overall retrieval performance because of semantic noise. Semantic noise that is added to results leads to query drift and low precision. To make sure the benefits outweigh the cost, search engines that use automatic query expansion do a lot of testing to make sure query drift is small enough so that query expansion can still be helpful. A common test done to evaluate automatic query expansion is to look at web pages that are already classified and run searches twice. One test will be done with query expansion and the other test will be done without. Because the web pages are already classified search engines can easily calculate precision and recall to compare the two searches [7].

When query expansion is done manually especially in explicit relevance feedback it can be very time consuming. There is a lot of overhead when users have to scan through pages and decide which pages are relevant and irrelevant, they also need to select terms they want to use to expand their query. Users can select multiple terms by looking through the suggestion and select the ones that they like and dislike. This is done at run-time and this iterative process requires clicking on web pages and terms evaluating their relevance. This overhead and extra time discourages users from query expansion and keeps them from using it. Manual query expansion needs to be done efficiently and intuitively in order for it to be utilized.

3. AUTOMATIC QUERY EXPANSION

The automatic query expansion or modification based on term co-occurrence data has been studied for nearly three decades. The various methods proposed in the literature can be classified into the following four groups:

i. Simple use of co-occurrence data: The similarities between terms are first calculated based on the association hypothesis and then used to classify terms by setting a similarity threshold value [20]. In this way, the set of index terms is subdivided into classes of similar terms. A query is then expanded by adding all the terms of the classes that contain query terms. It turns out that the idea of classifying terms into classes and treating the members of the same class as equivalent is too naive an approach to be useful [14].

ii. Use of document classification: Documents are first classified using a document classification algorithm. Infrequent terms found in a document class are considered similar and clustered in the same term class (thesaurus class) [5]. The indexing of documents and queries is enhanced either by replacing a term by a thesaurus class or by adding a thesaurus class to the index data. However, the retrieval effectiveness depends strongly on some parameters that are hard to determine [6]. Furthermore, commercial databases contain millions of documents and are highly dynamic.

The number of documents is much larger than the number of terms in the database. Consequently, document classification is much more expensive and has to be done more often than the simple term classification).

iii. Use of syntactic context: The term relations are generated on the basis of linguistic knowledge and Co-occurrence statistics [12],[15]. The method uses a grammar and a dictionary to extract for each term t a list of terms. This list consists of all the terms that modify t. The similarities between terms are then calculated by using these modifiers from the list. Subsequently, a query is expanded by adding those terms most similar to any of the query terms. This produces only slightly better results than using the original queries [12].

iv. Use of relevance information: Relevance information is used to construct a global information structure, such as a pseudo thesaurus [18]. A query is expanded by means of this global information structure. The retrieval effectiveness of this method depends heavily on the user relevance information. Moreover, the experiments in (Smeaton, 1983) did not yield a consistent performance improvement. On the other hand, the direct use of relevance information, by simply extracting terms from relevant documents, is proved to be effective in interactive information retrieval [19]. However, this approach does not provide any help for queries without relevance information. In addition to automatic query expansion, semi-automatic query expansion has also been studied [8]. In contrast to the fully automated methods, the user is involved in the selection of additional search terms during the semi-automatic expansion process. In other words, a list of candidate terms is computed by means of one of the methods mentioned above and presented to the user who makes the final decision. Experiments with semi-automatic query expansion, however, do not result in significant improvement of the retrieval effectiveness [8]. Among the various approaches, automatic query expansion by using plain co-occurrence data is the simplest method. In contrast to the approaches presented, we use a similarity thesaurus as the basis of our query expansion. First we show how such a similarity thesaurus is
constructed and then we present a query expansion model in order to overcome the drawbacks of using plain co-occurrence data.

3.1 Query Expansion with WordNet

WordNet ontology is one of the most important resources available to researchers in the field of text analysis, computational linguistics, and many other related areas. WordNet [1] is ontology of lexical references whose design was inspired by the current theories of human linguistic memory. Nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (synsets), each representing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations such as hypernym/hyponym (is-a), and meronym/holonym (part. whole).

The WordNet purpose is to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. WordNet is used in many text classification methods as well as in Information Retrieval (IR) because of its broad scale and free availability.

4. OUR APPROACH OF THE QUERY MODIFIER DESIGN FEATURE

In this section, we described our approach to query expansion and, in particular, focus on the novel use of query associations in the expansion process. Our generalized method for query expansion proceeds as follows: first, a query is submitted to our search system and the keywords as extracted from the query string by the removal of stop words

Let word (Qtn) be the total keywords which consists of keyword1 (k1), keyword2 (k2), keyword3 (k3) and the rest keyword (n) (kn) in a user search query be represented as shown in equation 1

\[ Qtn = \sum_{k=1}^{n} k \]  

(1)

Then, every keyword (kn) has its own semantic words (sn) as many as possible which are available in the semantic data or dictionary (d) as shown in equation (2). S represents the total semantic keywords derived from a single query keyword term which consists of a few keywords (K) in the search fields.

\[ Qtn = \sum_{k=1}^{K} \sum_{s=1}^{n} c \]  

(2)

where k1 is the first query term keyword and kn is the last number of keywords that exit in the user query term. Therefore the query term keywords can be as many as possible as long as the total of retrieval results from the input words is not influenced. In addition, those words can contribute their own semantic words where one word can yield few words which depend on the words provided in the semantic dictionary.

The Figure 1 below shows the expanded query (Q) for a single query term. It starts with the user single query string (Q1) to be translated into synonyms keyword term query (s1, s2, s3, … sn) of words. Each keyword (k1, k2, k3 … kn) has a few semantic or synonym words (s1, s2, s3 … sn) according to the related keywords in the semantic data or dictionary. The collective words which consist of the original user query term and the semantic words of each of the query term extracted from the user query are used to query search engine as shown in equation (3)

\[ Q = \sum_{s=1}^{n} k_{s1} + \sum_{s=1}^{n} k_{s2} + \ldots + \sum_{s=1}^{n} k_{sm} \]  

(3)

The sum of the entire query terms keywords and there synonyms are used to extract relevant result from a search engine and let D be the total retrieved documents that are related to each keywords and there synonyms. A keyword or synonym is related to a document if in each document the first keyword (k1 ∩ d1, k1 ∩ d2, k1 ∩ d3,…k1 ∩ dn), second keyword (k2 ∩ d1, k2 ∩ d2, k2 ∩ d3,… k2 ∩ dn) and the last keyword in the user query term should be (kn ∩ d1, kn ∩ d2, kn ∩ d3,…kn ∩ dn) with the additional results of first synonym or semantic word (s1 ∩ d1, s1 ∩ d2, s1 ∩ d3,…s1 ∩ dn), second synonym (s2 ∩ d1, s2 ∩ d2, s2 ∩ d3,…s2 ∩ dn) and the last synonym word is in document as in (sn ∩ d1, sn ∩ d2, sn ∩ d3,…sn ∩ dn). The relevant documents that can satisfy the users query are counted as shown in equation 4

\[ D = \sum_{d=1}^{n} k \cap d + \sum_{d=1}^{n} s \cap d \]  

(4)

Where n is the number of documents retrieved for a particular search query. The Semantic (S) of the documents are the results to be derived from multiple D’s for the retrieved results which depends on each keyword (k) that has a related synonym words (S) which comes from semantic dictionary (d) like wordNet.
Fig 1 A: More detailed Design of the Expanding Query Term
4.1 Re-Writing Query by Augmenting query Term Synonym

Recurring terms and phrases are extracted according to certain criteria, which include a term frequency threshold to indicate the minimum number of times a term should appear in the input documents to be considered a frequent term and the synonyms of the frequent terms and phrases were extracted using WordNet for query expansion. This is shown in the Algorithm 1 below:

1: //* start adding synonyms */
2: SF ← list of synonym Features, empty
3: STD ← list of synonym Term-Document Arrays, empty
4: for each document d in D
5: for each term t in F
6: find word for term in WordNet
7: if word is found
8: for each syn set syn for word
9: for each synonym syn
10: if synonym is the same as t // avoid adding the original word from the synset
11: continue to next synonym
12: end if
13: if synonym is found in F
14: td ← Term-Document Array for synonym
15: increase term frequency in td for document d by 1
16: f ← Feature that contains term synonym
17: increase total term frequency for term synonym by 1
18: else // synonym not found in F */*
19: if synonym is found in SF // *check for same condition above in previous synonyms */*
20: std ← Term-Document Array for synonym
21: increase term frequency in std for document d by 1
22: sf ← Feature that contains term synonym
23: increase total term frequency for term synonym by 1
24: else // synonym not found in F or SF */*
25: td ← new Term-Document Array
26: increase term frequency in td for document d by 1
27: add td to STD
28: f ← new Feature
29: increase total term frequency for term synonym by 1
30: add synonym, td to f
31: add f to SF
32: end if
33: end if
34: end for
35: end for
36: end if
37: end for
38: end for
39: append SF to F
40: append STD to TD

Algorithm above shows the modified Extract Single Terms algorithm including the changes carried out for adding the synonyms using WordNet.

5. LEARNING TERM-BASED CONCEPTS

A problem of the standard Vector space model used by the conventional search engines is that a query is often too short to rank documents appropriately because conventional search engines use keyword based matching techniques. To cope with this problem, our approach is to enrich the original query by expanding it with terms occurring in the documents of the collection and there synonyms. But in contrast to traditional pseudo relevance feedback methods, where the top | ranked documents are assumed to be relevant and then all their terms are incorporated into the expanded query, a different technique is used to compute the relevant documents as follows:

Let $q = t_1, \ldots, t_n$ be the user query containing the terms $t_1, \ldots, t_n$.

And $q = (w_1, \ldots, w_m)^T$ be the vector representation of this query.

Let $Q = \{q_1, \ldots, q_m\}$ be the set of all synonyms of initial query terms $q_1, \ldots, q_m$.

And $D_k^*$ be the set of relevant documents of the query $q_k$.

The goal is now to learn for each term $t_i$ a concept $c_i(1 \leq i \leq n)$ with the help of the synonyms in query terms and their appropriate relevant documents. For this, the term $t_i$ is searched for its synonyms and if it is found, the relevant synonyms of the query terms are used to learn and expand the concept of a user query for term searching using search engine. Due to the VSM, a concept is also a weighted vector of terms and calculated with:

$$c_i = \sum_{j=1}^{n} \tau_j \delta \Sigma_k \delta_k^*$$

where $0 \leq \tau_i, \delta_i \leq 1$ are weights for the original term and the additional synonym terms, respectively.

The expanded query vector is obtained by the sum of all term-based concepts:

$$q'' = \sum_{i=1}^{n} C_i$$

Before applying the expanded query, it is normalized by

$$q'' = \frac{q''}{\|q''\|}$$

For this approach, the concepts are learned by adding terms from synonyms of the query terms. The complete documents (e.g. all term weights of the document vector) are summed up and added to the query to add more meaning to the query.

6. CONCLUSIONS & FUTURE WORK

In conclusion, the technology of search engines is a very dynamic field, always looking for improvements and new ideas in order to satisfy user needs. The ability of the system to find relevant information based on the user’s search query to a successful system is based on how well a user query is formulated and the terms that
makes up the user query that can satisfy user information needs. This ability can be significantly enhanced by employing an approximate query expansion technique with related terms to improve performance, particularly recall. In this paper, we describe an approach for performing conceptual query expansion based on WordNet's synonym, which produces a diversified set of document search results that is used to increase the search horizon of search engines in order to improve user query need.

Future work includes adding features to support complex multi-concept queries, adding additional features that support interactive query refinement loops and query by-example, and evaluating the approach through user studies. We are also examining the benefit of including conceptual information within the information organization process.

REFERENCES


