Noname manuscript No.
(will be inserted by the editor)

Query Expansion Using Pseudo Relevance Feedback on Wikipedia

Andisheh Keikha · Faezeh Ensan · Ebrahim Bagheri

the date of receipt and acceptance should be inserted later

Abstract One of the major challenges in Web search pertains to the correct interpretation of users' intent. Query Expansion is one of the well-known approaches for determining the intent of the user by addressing the *vocabulary mismatch problem*. A limitation of the current query expansion approaches is that the relations between the query terms and the expanded terms is limited. In this paper, we capture users' intent through query expansion. We build on earlier work in the area by adopting a pseudo-relevance feedback approach; however, we advance the state of the art by proposing an approach for feature learning within the process of query expansion. In our work, we specifically consider the Wikipedia corpus as the feedback collection space and identify the best features within this context for term selection in two supervised and unsupervised models. We compare our work with state of the art query expansion techniques, the results of which show promising robustness and improved precision.

1 Introduction

The global search space is approaching 10 billion queries per month which shows that users rely heavily on search for retrieving information from the Web [1,2]. One of the challenges that a search engine faces is to find users' intent from simple short keyword-based queries. Studies have already shown that the average length of a search query is 2.4 words [3]. This short length is one of the main reasons why queries can be ambiguous by nature. It has been estimated that 4% of web queries and 16% of the most frequent queries are ambiguous [2]. For instance, a user entering the query Hotel California might want to search for the Eagles album, or be interested in hotels in California or a hotel named California.

Address(es) of author(s) should be given

Other than ambiguity, coverage, also known as recall, is an important concern. Empirical studies have shown that state-of-the-art search engines have high precision but do not necessarily have a high recall [2]. In other words, it is probable that a web page that is related to the users' intent, but does not contain the specific query terms, would not appear in the results. For instance, a user searching for *gain weight* is most likely searching to find information about how to gain muscle as opposed to not gaining fat or even losing fat. However, when such a query is searched for, e.g. in Google, the result set that is retrieved has little, if any, overlap with the result set that is retrieved when queries such as *gain mass* or *gain muscle not fat* are entered. Given that in these three cases, the intent of the user is the same, the expectation is that the retrieved results be at least partially overlapping. Query expansion is one of the approaches for tackling the problem of low coverage and ambiguity. Query reformulation and expansion, in particular, try to tackle the so called *vocabulary mismatch problem*. When indexing a document, the search engine crawler only considers and extracts the syntactical surface form of a term; therefore, if a user searches for another word with even the exact same meaning, the search engine will not be able to retrieve that document even though it might be relevant to the users intent. In other words, a semantically similar document to a query might not be included within the result set due to *vocabulary mismatch*.

One of the traditional approaches in query expansion is the pseudo-relevance feedback technique [4]. In this approach, the query is submitted to the search engine and the top results are extracted and considered as being relevant to the query (called feedback documents). These related documents are then scanned for more keywords related to the query. The extracted keywords are ranked based on a significance measure and are added to the query, resulting in an expanded query. In order to rank and select keywords from feedback documents, a variety of word weighting schemas have been used in the literature such as TF-IDF [4], Rocchios Weight [5], Binary Independence Model [6], Chi-Square [7], Robertson Selection Value [8], and Kullback-Leibur Distance [9], just to name a few.

It has been shown that the traditional pseudo relevance feedback method can harm the results of ad hoc retrieval if the initial top retrieved documents include irrelevant documents [10]. Li et al [11] have shown that in most, if not all, cases the feedback documents do in fact contain irrelevant documents to the query. In this paper, inspired by the idea of pseudo relevance feedback, we consider Wikipedia articles as feedback documents instead of top results of a search engine in order to avoid the inclusion of irrelevant documents in the feedback document collection. In our proposed work, the most related Wikipedia articles to the query are identified and considered as feedback documents, based on which query expansion is performed. We are not the first to propose the use of Wikipedia articles instead of top retrieved documents. The work in [10] uses Wikipedia for query categorization, however the results of the paper does not cover broad queries, whereas in our approach, we evaluate our work on all query types (ambiguous and unambiguous) and the comparative analysis of our work shows improvement even on ambiguous queries. The work in [11] reranks the retrieved documents using Wikipedia categories, however the details of the term selection method is not provided in that article. In our work we propose a novel disambiguation approach to find the best Wikipedia articles relevant to a query. We propose both supervised and unsupervised term selection approaches in the pseudo relevance feedback process and compare our work with the state of the art to show how our proposed approach is more efficient in terms of robustness and performance.

In this paper, we provide the following main contributions:

1. We propose a hybrid approach for the disambiguation of search queries in the context of Wikipedia articles. In our work, we map each query onto a set of coherent Wikipedia articles that collectively represent the underlying semantics of the search query.
2. Given a set of coherent Wikipedia articles for a query, we rank and select a set of terms from those articles for the purpose of query expansion. We employ and empirically compare the performance of various unsupervised schemes for extracting terms from Wikipedia articles.
3. By considering only 20% of the extracted Wikipedia articles for the queries, and the possible candidate terms for query expansion, we propose a supervised term feature selection function that enables us to select appropriate terms to be included in the query expansion process.

The rest of this paper is organized as follows: Section 2 describes the proposed approach. The extensive experimental results consisting of parameter tuning, supervised approaches for term selection, and comparative analysis is covered in Section 3. The related work is reviewed in Section 4, followed by some concluding remarks and areas of future work.

2 The Proposed Approach

The main objective of our approach is to find an accurate representation of the query intent in terms of additional terms that can be effectively used in query expansion. To this end, we use the Wikipedia corpus as the feedback document collection. The primary goals of our work are i) to find a set of Wikipedia articles that can unambiguously represent the underlying semantics of the search query and can be the basis for finding suitable terms for query expansion; and ii) to identify discriminative features that can be used in term selection for query expansion that show improved robustness and performance. Figure 1 shows the overview of the steps in our approach.

In order to be able to identify the intent of the query, we rely on information from Wikipedia articles. The reason we adopt this strategy is because queries are short and lack sufficient *context* to be used for extracting intent. Therefore, we build context for queries by identifying relevant Wikipedia articles that might be relevant to the query at hand. Once the relevant Wikipedia articles are identified, we consider them to be the context for query and use both

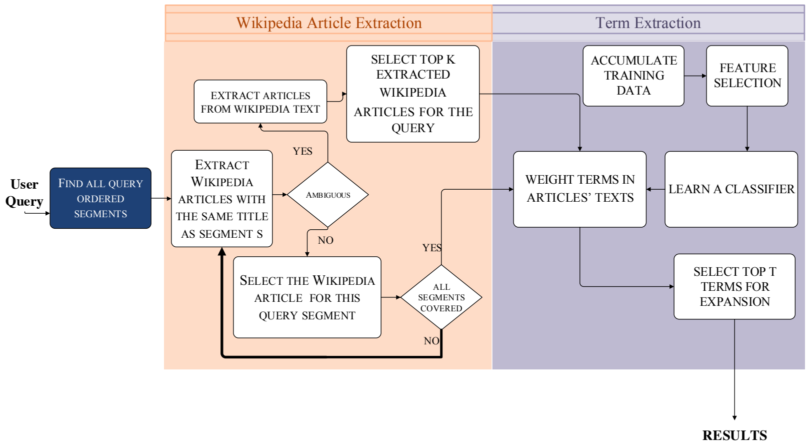
Wikipedia Article Extraction
Term Extraction
User Query
FIND ALL QUERY ORDERED SEGMENTS
EXTRACT WIKIPEDIA ARTICLES WITH THE SAME TITLE AS SEGMENT S
AMBIGUOUS
YES
EXTRACT ARTICLES FROM WIKIPEDIA TEXT
SELECT TOP K EXTRACTED WIKIPEDIA ARTICLES FOR THE QUERY
NO
SELECT THE WIKIPEDIA ARTICLE FOR THIS QUERY SEGMENT
ALL SEGMENTS COVERED
YES
NO
ACCUMULATE TRAINING DATA
FEATURE SELECTION
WEIGHT TERMS IN ARTICLES' TEXTS
LEARN A CLASSIFIER
SELECT TOP T TERMS FOR EXPANSION
RESULTS

Fig. 1: Approach overview

supervised and unsupervised term selection methods for performing query expansion. More concretely and as shown in Figure 1, we first identify a set of candidate Wikipedia articles that can be considered relevant to the query. The extracted articles are evaluated to see whether they are ambiguous or not. We treat ambiguous queries and unambiguous queries differently. Once a set of Wikipedia articles are selected, all the terms in these articles are processed and ranked. For processing the articles to extract terms, we propose two main approaches: unsupervised and supervised term selection. In the unsupervised method, the terms to be included in the expanded query are selected based on the value of a set of predetermined features. In the supervised approach, we first curate a training set, which consists of eight term features. Based on the curated training set, we determine the degree of impact of each feature on the performance and robustness of the query expansion results. To this end, we apply a feature selection method to select the best subset of features, and then employ machine learning techniques to learn the term selection function based on the limited set of selected features. In the supervised term selection method, we select the top terms based on the trained term selection function. We present the details of each step in the following subsections.

2.1 Query Disambiguation and Annotation

In order to build context for a given search query, traditional forms of text annotation [12–14] cannot be directly applied due to the very short length of a query. Therefore, we consider each query to be a collection of words, which can be used to extract n-grams. We refer to each n-gram extracted from a query as a segment. In the rare case, when a user is looking for one self-contained piece of information and her search query is formulated very accurately, then the largest n-gram in the query, i.e., the query itself, might correspond to one Wikipedia article. For instance, for a search query such as Barack Obama, one can easily find a corresponding Wikipedia article. However, in reality, users are not necessarily looking for information that have directly corresponding Wikipedia semantics. Furthermore, they might use different syntactic representations to express the same semantic content. Therefore, we need to look into the various segments of the query to disambiguate the query and relate it to the most suitable Wikipedia articles. For instance, for the query: Obama Family Tree, one cannot find a corresponding Wikipedia article; therefore, the semantics of the query needs to be expressed through a combination of Wikipedia articles. For this reason, we look at all the possible query segments, such as obama family, family tree, obama tree, tree obama, for identifying relevant Wikipedia articles. In other words and more succinctly, our objective is to identify the most relevant Wikipedia articles to the given query such that they can serve as the *context* for the query.

In order to identify the most relevant Wikipedia articles for a query, we differentiate between ambiguous and unambiguous queries. We automatically determine whether a query can have multiple senses and therefore be considered to be ambiguous or not. Depending on this, we adopt a different strategy for determining relevant articles. For instance, we can determine that a query such as Barack Obama is unambiguous but a query like Hotel California is ambiguous.

2.1.1 Unambiguous Queries

We first consider all queries to be unambiguous and try to find relevant Wikipedia articles for them. In order to find related Wikipedia articles, we derive all possible query segments as n-grams. We iteratively find the largest n-grams in the query that have a corresponding Wikipedia article. We repeat the process until we have covered all of the terms in the query in at least one of the selected n-grams. For instance, in the query obama family tree, we first try to identify a Wikipedia article that corresponds to the exact query. Since no such article can be found, we then consider the next possible segments which are obama family, family tree and obama tree. For the first two segments, articles with the same title are found and as all the terms in the query are covered by these two segments, there is no need to consider the next largest n-grams. Therefore, we represent this query through two Wikipedia articles, namely Obama Family and Family Tree[1].

While this process finds very accurate Wikipedia article representations for unambiguous queries, it will not be as effective when faced with ambiguous queries. For instance, when applied to a query such as hotel california,

[1] Wikipedia articles `https://en.wikipedia.org/wiki/Obama_Family` and url-https://en.wikipedia.org/wiki/Family_tree respectively.

it will not be able to correctly disambiguate between the senses of the query. However, the approach based on the segments allows us to automatically determine whether the query is unambiguous and the extracted Wikipedia articles are reliable or the query is ambiguous and further processing is needed. In order to determine the ambiguity of a query, the list of extracted Wikipedia articles are considered. If any of the extracted Wikipedia articles has a redirection from a Wikipedia disambiguation page, then this shows that the specific query segment that was associated with that article could possibly have different senses. Considering the hotel california query as an example, the largest segment would be mapped to the Hotel California article in Wikipedia which is redirected from Hotel_California_(disambiguation)[2]; therefore, pointing to a possible ambiguity in the query. We consider such queries to be ambiguous and further process them as follows.

2.1.2 Ambiguous Queries

For the cases where the query is determined to be ambiguous, we adopt a term frequency search of query terms within relevant Wikipedia articles to determine what is the most likely sense of the query. Given search queries are very short and therefore lack proper context, we adopt a *popularity-based* disambiguation method [15], which assumes that the correct sense of a word, when lacking context, is the one that is the most frequently observed. Therefore, we will assume that the best sense of an ambiguous query is the one that is more frequently observed on Wikipedia. Our ranking strategy allows us to disambiguate search queries based on a popularity-based approach.

To this end, we rank Wikipedia articles based on their relevance to the query terms according to the following equation adopted from [16]:

$$Rank_d(q) = \sum_{t \in q} tf(t_d).idf(t).lenghtNorm(d) \tag{1}$$

where $Rank_d(q)$ provides a rank score for document d with respect to query q, $tf(t_d)$ is term frequency of term t in document d, $idf(t)$ is the inverse document frequency of the term, and $lenghtNorm(d)$ is the normalization value of document text length. TF-IDF is a traditional but very promising approach in ranking the importance of a word. It assumes both frequency and uniqueness of the word at the same time. In Equation 1, we rank the documents based on the importance of the query terms in those documents. A word is important if it has been repeated many times in one document but not in the others. For example when searching for "DBpedia papers", the word "paper" might not become a significant factor of importance if we search in academic papers, however the word "DBpedia" is certainly important since it would not be frequently observed in a uniform way in all papers. We also normalize the length of the documents, so that longer documents are not privileged because of their length.

[2] https://en.wikipedia.org/wiki/Hotel_California_(disambiguation)

Table 1: Examples of extracted Wikipedia articles.

	Query	Identified Wikipedia Articles
Non-Ambiguous Queries	Native American casino	Native_American_gaming
	Mercy killing	Non-voluntary_euthanasia
Ambiguous Queries	Land mine	Land_mine This_Land_Is_Mine Land_mine_situation_in_Chechnya Land_mine_situation_in_Nagorno-Karabakh Land_mine_contamination_in_Bosnia_and_Herzegovina Smart_mine PFM-1
	Poliomyelitis and Post Polio	Poliomyelitis Post-polio_syndrome Post-Polio_Health_International Joseph_Bowler Post-polio_syndrome Ivar_Wickman Ontario_March_of_Dimes

Table 1 shows the Wikipedia articles extracted for four sample queries, two non-ambiguous, and two ambiguous, taken from the TREC 2010 dataset. As seen in both of the unambiguous queries, the extracted Wikipedia articles for none of the queries has the same title as the query segments. We have been able to successfully extract the correct Wikipedia article despite vocabulary mismatch because we have considered all *redirection* links on Wikipedia to denote semantic similarity between the redirected pages. Therefore, if two notions are expressed in different syntactic forms but capture the same semantics, we are able to identify them through the redirects links on Wikipedia. For instance, when searching for the query segment mercy killing, we were able to determine Non-voluntary_euthanasia as the best matching Wikipedia article; therefore, despite the vocabulary mismatch problem, we are able to find the related Wikipedia article to the user query.

Now, for the two ambiguous queries, we first process them as if they were unambiguous; however, in the process if any disambiguation pages are encountered, we note this as an indication of possible ambiguity. For instance, when processing the query *land mine*, we encounter Land_mine_(disambiguation)[3], and we consider the query to be an ambiguous query. Therefore, we employ a popularity based approach for determining the most likely sense of the query. We rank Wikipedia articles based on their relevance to the query and take the top-k to represent the query. We empirically determine the value of k in our experiments.

[3] https://en.wikipedia.org/wiki/Land_mine_(disambiguation)

Table 2: Term weighting schemes.

Function	Formula
TF [17]	$0.5 + \frac{0.5 \times f(t,d)}{max\{f(w,d) : w \in d\}}$
TF-IDF [18]	$tf(t,d) \times \log \frac{N}{\lvert\{d \in D : t \in d\}\rvert}$
BIM [4]	$\log \frac{p(t\lvert R)[1 - p(t\lvert C)]}{p(t\lvert C)[1 - p(t\lvert R)]}$
Chi-Square [4]	$\frac{(p(t\lvert R) - p(t\lvert C))^2}{p(t\lvert C)}$
Weighted Degree (WD)	$\sum_{k=1}^{n} Weight(node_i, node_k)$
Weighted PageRank (WPR)	Calculated using Equation 3
WD in Cluster (WD_c)	WD after WSI graph clustering is applied
WPR in Cluster (WPR_c)	WPR after WSI graph clustering is applied

2.2 Term Extraction

Now for a user query, regardless of its ambiguity, we need to identify and select a set of terms that best describe the users' intent; therefore, we consider the Wikipedia articles identified in the previous phase to be the feedback documents within a pseudo-relevance feedback approach and select the top terms from within these documents based on a ranking scheme. We propose two different approaches for this step: 1) unsupervised term selection, and 2) supervised term selection. The details of these two approaches are described in the following subsections.

2.2.1 Unsupervised Term Selection

Within the unsupervised approach, the main ideas is that important words that can be used for expanding the query can be identified based on some metrics that can measure importance. Based on such metrics, we measure the importance of each word that is present in the pseudo-relevant document set retrieved from Wikipedia. We rank the words based on their importance and set the most important words to be considered in query expansion.

More succinctly, we exploit eight different term weighting schemes for selecting the most relevant terms to be included in the query expansion process. The terms in the retrieved Wikipedia articles are ranked based on these term weighting schemes and those terms that have the highest value are selected to be included in the query expansion process. These eight weighting schemes are listed and described in Table 2. Term Frequency (TF) is a normalized way of calculating the frequency of a term in a given set of documents. In our work and in order to calculate this scheme, all the extracted Wikipedia articles for the query are considered as one document and the TF of each word is calculated. The reason for this is because the different Wikipedia articles that are extracted for a given query are in fact the representatives of the various aspects of the query. Term Frequency-Inverse Document Frequency (TF-IDF) is an extension of the TF scheme which measures how important a word is for a given document within the context of the whole corpus. The IDF scheme offsets frequency when a word is generally very frequent in the corpus. Binary Independence Model (BIM) assumes that words in both the document and query spaces are completely independent (similar to the assumption of the naive bayes classifier). Furthermore, the Chi-Square scheme works on a similar basis to BIM and measures the importance of a word within the context of the relevant documents. Both of these schemes rely on $p(t|R)$ and $p(t|C)$, which are the probability of term t occurring in relevant documents (R) and the probability of term t occurring in the corpus in general (C), respectively as shown in Table 2.

Other than the mentioned features, we introduce four additional features that are calculated based on a graph representation of the terms. In order to calculate the graph-based schemes, an undirected graph is constructed over all the terms in the feedback document collection in such a way that the nodes are the terms and the edges are the similarity between the terms calculated though "Resnik Similarity" [19]. Based on this graph structure, we calculate the weighted degree and weighted PageRank value for each node. These two schemes are calculated as shown in Equations 2, and 3.

$$WD(node_i) = \sum_{k=1}^{n} Weight(node_i, node_k) \tag{2}$$

where n is the number of nodes that has an edge to $node_i$, and $Weight(node_i, node_k)$ is the weight of the edge connecting $node_i$ and $node_j$.

$$\begin{aligned} PageRank(node_i) = & \\ & \alpha \times PageRank(node_i) \\ & +(1-\alpha)\sum_{k=1}^{n} \frac{Weight(node_i, node_k)}{\sum_{k=1}^{n} Weight(node_i, node_k)} \\ & \times PageRank(node_k) \end{aligned} \tag{3}$$

Equation 3 will iterate over all nodes until the PageRank value converges with an error threshold below β.

These schemes help to extract terms that are more strongly connected in the graph. The nodes with high Weighted Degrees represent those terms that are highly similar to the other terms in the document; therefore, they have a high chance of being central words that could very well represent the topical content of the feedback documents. Furthermore, weighted PageRank shows the probability that a word would be selected based on the connections that it has and its weight with the neighboring nodes. Therefore, a high Weighted PageRank value shows that the term has a high number of strong connections with other terms.

These two schemes are very helpful when the Wikipedia article focuses mainly on one aspect of a concept, however when there are more aspects discussed in one Wikipeia article, there might be some terms that are related to one of the aspects, which might be unrelated to the query. Such terms can be strongly connected to each other, and as a result have a high Weighted Degree and weighted PageRank values, but at the same time harm the results if selected to be included in query expansion. For example for the query "mercy killing", the concept "Non-voluntary_euthanasia" is extracted. In one part of this Wikipedia article, the issue of killing babies being born with a health problem is discussed, and as a result terms like "baby", "child", "parent", and "doctor" are strongly connected, and have high weighted degree, and Weighted PageRank in this context; however, such terms could harm the results if applied in the context of query expansion for the "mercy killing" query.

To enrich our features with some additional features that can overcome this problem, we consider using a graph partitioning algorithm that can group the graph into different partitions. We use the Word Sense Induction (WSI) algorithm [2] to partition the graph. Using such algorithms, the graph will be grouped to strongly connected components in which each component of the graph consists of a set of nodes (terms) that are semantically close to each other. Each component is considered as one semantic aspect of the query, so the terms in each component of the graph are related to one aspect of the query. Applying the algorithm, the graph partitions that the query terms appear in are considered as new subgraphs themselves, and Weighted Degree and weighted PageRank are calculated inside those subgraphs instead of the complete graph. We call these schemes WD in Cluster, and Weighted PageRank in Cluster, respectively. Table 2 summarizes the eight schemes used in this step.

2.2.2 Supervised Term Selection

Our hypothesis in the supervised term selection method is that there might be a more discriminative combination of the weighting schemes that can more effectively determine better terms for query expansion. For instance, in the unsupervised method, we only consider the weighting schemes separately; however, it is possible that better results would be obtained if these schemes were combined. Hence in the supervised term selection approach, we would like to build a term selection function using a subset of the eight weighting schemes.

To do so, we adopt a machine learning-based method to learn a term weighting function to optimize the effectiveness of query expansion. As the first step of this method, we curate a training dataset based on a subset of the queries in our query collection (introduced in the evaluation section). The queries are then manually labeled with appropriate Wikipedia articles and best terms to be included in query expansion are determined by an expert. For each of the selected terms, the eight weighting schemes are calculated and used as features. Having in mind that reducing the number of features can defy curse of dimensionality and improve prediction performance [20], we apply a feature selection method to select a subset of the features based on their effectiveness on query expansion. The selected features are then exploited within a machine learning technique to learn an appropriate classifier that would determine whether a term would be included in query expansion or not. The classifier can predict how each candidate term can improve the results of the search engine, and the best terms are selected for query expansion. The details of these steps are described in the following.

Step 1: Training Data Preparation. The training data is manually curated based on queries from the Robust04 dataset, for each of the queries of which the terms in the most relevant Wikipedia articles are selected and the values of the eight weighting schemes are calculated. These eight values as well as a label showing how much the selected term would improve the performance of query expansion form the feature space.

In order to prepare the training data, 20 queries were selected from each topic set of the Robust04 dataset (totally 60 topics). The queries used in training were not used in the testing process. The candidate terms for all of the queries were extracted and then the query and the expanded query with each term was submitted to a base search engine, i.e. Google. The MAP (Mean Average Precision) was calculated for both cases, and the difference between the expanded query and the original query was stored as the degree of improvement. Therefore, a negative value means that the term degrades the result, and a positive one shows improvement. The greater the improvement value is, the more that term contributes to improved results when used for query expansion.

Step 2: Feature Selection. Feature selection can be applied using 1) Feature Ranking (FR), or 2) Feature Subset Selection (FSS) [20]. In the first approach, each feature is evaluated individually, after which they are collectively ranked, and the top k features are selected as the final feature set, while in the latter approach, in each step of the algorithm a subset of features are selected and evaluated. We use the latter approach, since the features are not independent of each other, and the best practice would be not to assume such independence.

An FSS algorithm consists of two steps [21]: 1) finding a subset of features, and 2) evaluating the selected subset. For the first step, many strategies have been introduced in the literature such as exhaustive, heuristic and random search [20]. These search methods are often combined with evaluation measures to produce variants for FSS. In our feature selection algorithm, we use the Best First Search (BFS) which is a heuristic algorithm [22]. In this approach, once the best subset of features is found, a new feature is defined based on this subset of features and added to the feature set as a new feature and its individual constituting features are removed. This process is repeated until all features are exhausted.

For the evaluation step of FSS, two strategies can be adopted: 1) filter or 2) wrapper. The filter model evaluates features based on a heuristic over the general characteristics of the data and not the schemes that are expected to be learned, while the wrapper will apply a classifier over the data to evaluate the features [23]. The problem with the second approach is its performance on very large datasets, but in our case since our training data includes only 60 samples, the wrapper approach would be quite feasible; hence, we use this approach which is more thorough. Also the wrapper approach evaluates and improves the feature set based on the same scheme that will be optimized by the learner and thus could be more effective than the filter model [20].

Step 3: Classifier Training. Once the best subset of features is selected, the features that are selected for each term will be considered to represent that term, and the degree of improvement achieved as a result of including that term in query expansion will be considered to be the target label that needs to be predicted. We employ various machine learning methods such as linear regression, multilayer perceptron, pace regression, RBF networks and additive regression to train a classifier that would produce the degree of improvement for each input term. Each classifier will take as input the term's features and will predict the degree of improvement that is likely to be achieved if this term is included in the query expansion process. Once all the terms are inputted into the classifier, they will be ranked based on the classifier's output and the top t terms are selected for query expansion.

3 Empirical Evaluations

In order to empirically evaluate our work, we used the NIST Special Database 23/NTREC Disk 5 database (the query set and judgments). The contents of the databases are as follows:

1. The Los Angeles Times dataset: This dataset consists of 131,050 documents, which accounts for 475 MB of data. It includes 40% of the articles published by the Los Angeles Times newspaper in the two year period from Jan 1, 1989 to December 31, 1990.
2. The second dataset is based on the Foreign Broadcast Information Service (FBIS) data. This dataset consists of approximately 130,000 documents which is about 470 MB.

Each of the two datasets consists of three collections: i) a set of documents, ii) a set of queries (called topics in TREC) that can be answered by retrieving a subset of the documents, and iii) the expected result sets for the queries, known as the relevance judgments.

For the purpose of comparative analysis, we compared our work with Relevance Model (RMC), as well as a Relevance Model based on Wikipedia (RMW) as two baselines [10]. These two methods propose state of the art query expansion methods that are vastly used for comparative analysis in this domain [24, 10, 25, 26]. For the RMC method, we use the implementation provided through the Indri framework, which is an adaptation of Lavrenko's relevance model [24]. For the RMW expansion method, we use the Lucene search engine to index and retrieve Wikipedia articles for determining the feedback document set for expansion. We employ two commonly used evaluation measures for evaluating our work, namely: i) Mean Average Precision (MAP), and ii) Normalized Discounted Cumulative Gain (nDCG) [27, 28]. MAP considers the performance of the search engine in different recall levels formalized as follows:

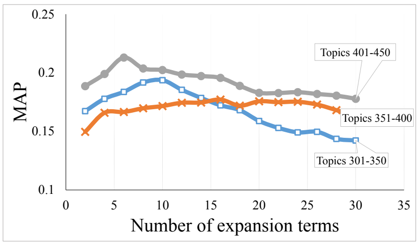

Fig. 2: Tuning the number of expansion terms: MAP over number of terms.

$$Average\_P(q) = \frac{\sum_{k=1}^{n} P(k,q) \times rel(k,q)}{|R|} \tag{4}$$

$$MAP = \frac{\sum_{q \in Q} Average\_P(q)}{|Q|} \tag{5}$$

where $P(k,q)$ is the precision at position k in the result list retrieved for query q and $rel(k,q)$ is an indicator function equaling 1 if the item at rank k is a relevant document to query q, zero otherwise. Q is the set of all queries that are submitted to the search engine, n is the number of documents in the result list and R is the set of relevant documents.

Furthermore, nDCG is a measure that is designed based on two important principles: 1) highly relevant documents are more important than slightly relevant ones, and 2) The higher the rank of a relevant document is, the less desirable it is, because the users are less likely to check them. nDCG is formalized as follows:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{6}$$

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{7}$$

where p is the highest position in the list, rel_i indicates 1 or 0 according to its relevance to the query and $IDCG_p$ represents the ideal DCG_p which means the most relevant documents have the highest ranks.

In the following subsections, we first explain how the model parameters are set and then the comparative analysis of our results and the two baselines are provided.

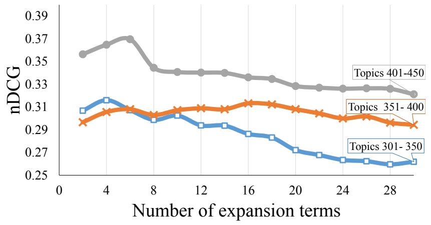

Fig. 3: Tuning the number of expansion terms: nDCG over number of terms.

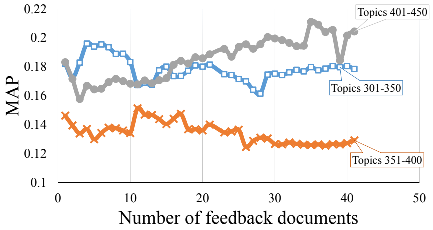

Fig. 4: Tuning the number of extracted articles: MAP over number of extracted articles.

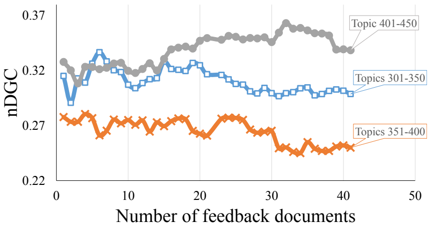

Fig. 5: Tuning the number of extracted articles: nDCG over number of extracted articles.

3.1 Parameter Tuning

In the first step, we are interested in evaluating the impact of the number of terms used in query expansion on the performance of our work. In other words, we would like to determine the best number of terms to be used in query expansion that would yield the highest performance since including too many terms could negatively impact the results, and too few terms would not address the *vocabulary mismatch problem*. To this end, we evaluated different number of terms, and the results are shown in Figures 2 and 3. In this evaluation, we set the size of the feedback document set to seven and used the TF weighting schema to weigh the terms in the feedback document set. Both of these two parameters will be evaluated in the next steps. Figures 2 and 3 show the results. In Figure 2, the horizontal axis represents the number of terms, and the vertical axis shows the MAP of the results over all the queries from one of the topic sets. Topics 301-350 are shown in a blue line and topics 401-450 in gray. These two topic sets have almost the same behaviour with the same intensity, while the orange line, which shows topics 351-400, has almost a uniform behaviour. In Figure 3, the horizontal axis shows the number of terms and the vertical axis shows the values for nDGC, in which Topics 301-350 and 401-450 have very similar behaviour again and different from topics 351-400. Although topics 351-400 shows almost uniform behavior but it is not totally uniform, with more careful observation, it can be seen that its behavior is similar to the other topics with much less intensity. We can derive from these two figures that very small and very large number of terms for expansion negatively impact the performance of the query expansion method. As shown in Figures 2 and 3, very large number of terms can demean the results more than a small number of terms. Even though nDCG shows approximately the same behaviour as MAP towards the number of expansion terms, its change is slight, while the change is more recognizable in MAP values. Based on this analysis, it seems reasonable to choose the expansion word set size from the range of 5 to 15 terms.

Furthermore, the size of the feedback document set could potentially impact the performance of the query expansion method. For this purpose, we examined various feedback document sizes and evaluated its impact on MAP and nDCG. Based on the above experiments, we selected the number of expansion terms to be 10 and employed the TF term weighting method in this analysis. Figures 4 shows the performance for different feedback document sizes. In Figures 4 and 5 where the horizontal axis shows the number of feedback articles, and the vertical axis shows MAP and nDGC respectively. As seen, the performance of the query expansion method seems to be neither predictable nor impacted by the feedback document size, showing a maximum of 2% difference on MAP and less than 2% on nDCG. Therefore, we conclude that the expansion method is not too sensitive to the size of the feedback document set.

3.2 Comparative Analysis

Based on the parameter setting analysis in Section 3.1, we compare our work with the baselines introduced earlier and represent a complete analysis comparing: i) Relevance Model (RMC), ii) Relevance Model on Wikipedia (RMW), iii) our proposed unsupervised model, and iv) our proposed supervised model. We perform our experiments on Topics 301-350, 351-400, and 401-450 of the TREC 2010 dataset.

Table 3: Results of the unsupervised method

Topics	Measure	TF	TF-IDF	BIM	Chi^2	WD	WPR	WD_C	WPR_C
301-350	MAP	0.174	0.154	.181	**0.185**	0.170	0.178	0.178	0.165
	nDCG	0.270	0.268	0.297	**0.306**	0.281	0.291	0.287	0.270
351-400	MAP	0.149	**0.161**	0.129	0.125	0.140	0.152	0.148	0.141
	nDCG	0.274	**0.303**	0.264	0.258	0.280	0.283	0.273	0.282
401-450	MAP	0.208	**0.214**	0.193	0.193	0.208	**0.214**	0.194	0.197
	nDCG	0.344	0.356	0.334	0.333	0.353	**0.363**	0.324	0.327
Overall Average	MAP	0.177	0.179	0.168	0.168	0.172	**0.181**	0.173	0.167
	nDCG	0.296	0.309	0.298	0.299	0.304	**0.311**	0.297	0.290

3.2.1 Unsupervised Term Selection

As the first step, we evaluate the impact of different term weighting schemes in the unsupervised method on the performance of the query expansion method. Based on the outcomes of the earlier parameter setting studies, we set the number of expansion terms to 10 and the feedback document set size to 15. The results of the performance of the unsupervised query expansion method based on different term weighting schemes are shown in Table 3. In each row of the table, the MAP and nDGC is shown for one of the topic sets. Each column shows one of the schemes. The meaning of this evaluation for each column is that the expansion terms are extracted based on the mentioned schemes in the column topic and then the expansion is evaluated based on MAP and nDGC. In topics 301-350, all of the weighting schemes show reasonable results except TF-IDF, while BIM and Chi-Square show the best improvement among all. In Topics 351-400, BIM and Chi-Square do not perform as expected and the results are not acceptable while TF-IDF shows the best result. In Topics 401-450, all the results are in the reasonable range but still BIM, Chi-square, WD_C, and WPR_C are worse than the others whereas WPR shows the best performance.

Looking into the overall average, it can be seen that TF, TF-IDF, WD and WPR show a slightly better performance over BIM, Chi-square, WD_C, and WPR_C. One reason for this could be the fact that in pseudo-relevance feedback each relevant document is expected to be *independently* relevant to the query, while in our method, each extracted Wikipedia article might not be considered independent of the others. For example, for the query *Alzheimers Drug Treatment*, the extracted Wikipedia articles for the feedback document set are: Drug_treatment and Alzheimer's_disease, each of which covers one aspect of the query, and therefore they are not independent. That could be why the schemes that assume such independency such as BIM, WPR in Cluster, and WD in Cluster do not necessarily result in better performance than those that present an overall measure.

3.2.2 Supervised Term Selection

In this section we compare the effect of applying different feature selection approaches and various learning methods on the results of query expansion. Also, we investigate whether the application of feature selection positively affects our results or not. Therefore, as the first comparison, we compare training a fixed classifier method, with and without feature selection. We apply different feature selection approaches and for each of them we show which features are selected. As mentioned in Section 2.2.2, for the feature subset selection methods, we need to select a classifier that evaluates each feature set. For this purpose we adopt the multilayer perceptron as the classifier in all the cases, so that we can only evaluate the effect of feature selection without changing the classifier.

Table 4 summarizes the results of using different feature selection methods in combination with the multilayer perceptron. As seen in the table, the best results, highlighted in bold, are observed when either of the following two feature selection methods are employed: i) Genetic Search or Scatter Search, and ii) Latent Semantic Analysis + Ranker. The first feature selection method (i) has selected WD, and BIM as the best set of features for the multilayer perceptron classifier, and the second one (ii) has only selected BIM. The latter feature selection method is a feature ranking method while the former is a feature subset selection approach. While the other approaches have occasionally shown comparable results on smaller portions of the topics, these two approaches outperform the other methods on all topics and both evaluation metrics.

In the second set of experiments, we evaluate the impact of the classifier on the results. A consideration that needs to be addressed is that the features selected in the previous stage are the best features based on multilayer perceptron, but we need to apply them on other learning methods. It is important to know that the subset eval feature selection method provides a generic selection of variables, not tuned for/by a given learning machine, which in this case is a multilayer perceptron. Therefore, it is reasonable to use this feature selection method on one predictor as a filter and then train another predictor on the resulting variables as discussed in [20]. As a result, we select the WD, and BIM features that showed promising performance in the previous evaluation as the selected features. The outcome of employing different classifiers is reported in Table 5. As seen in the table, the multilayer perceptron achieves the best performance on both of the evaluation metrics and on all three topics.

3.2.3 Overall Comparison

In this section, we report on the overall comparison of both the supervised and the unsupervised term selection methods compared to the state of the art. Based on the results reported in the previous section, from among the unsupervised query expansion methods, Chi-Square, TF-IDF and WPR show the best performance on Topics 301-350, 351-400 and 401-450, repectively.

Table 4: Feature Selection Evaluation

Feature Selection Method	Selected Features	Topics	MAP	nDCG
No feature selection	All of the 8 schemes	301-350	0.16	0.27
		351-400	0.15	0.27
		401-450	0.20	0.32
Classifiersubseteval + (BestFirst Search/Greedy stepwise /Linear forward selection)	TF-IDF	301-350	0.18	0.30
		351-400	**0.16**	0.29
		401-450	0.20	0.34
Classifiersubseteval + Exhaustive Search	WD, tf BIM, Chi^2	301-350	0.18	0.29
		351-400	0.15	0.25
		401-450	0.20	0.33
Classifiersubseteval + (Genetic Search/Scatter Search)	WD, BIM	301-350	**0.19**	**0.31**
		351-400	**0.16**	**0.30**
		401-450	**0.22**	**0.36**
Classifiersubseteval + Race Search	WD, WPR, tf TF-IDF, BIM, Chi^2	301-350	0.18	0.30
		351-400	0.15	0.29
		401-450	**0.22**	**0.36**
Classifiersubseteval + Random Search	TF-IDF, BIM Chi^2	301-350	**0.19**	**0.31**
		351-400	**0.16**	**0.30**
		401-450	0.21	0.35
Latent Semantic Analysis + Ranker	BIM	301-350	**0.19**	**0.31**
		351-400	**0.16**	0.30
		401-450	**0.22**	**0.36**
Wrapper subset eval + Genetic Search	WPR, WPR_c tf, BIM, Chi^2	301-350	0.18	0.29
		351-400	0.15	0.28
		401-450	0.20	0.35

Table 5: Learning method Evaluation

Learning Method	Topics	MAP	nDCG
Linear Regression	301-350	0.17	0.28
	351-400	**0.16**	0.29
	401-450	0.20	0.34
Multi layer perceptron	301-350	**0.19**	**0.31**
	351-400	**0.16**	**0.30**
	401-450	**0.22**	**0.36**
Pace regression	301-350	0.18	0.29
	351-400	**0.16**	**0.30**
	401-450	0.20	0.34
RBF Network	301-350	0.18	0.29
	351-400	0.15	0.28
	401-450	0.20	0.33
Additive Regression	301-350	0.18	0.29
	351-400	0.16	0.28
	401-450	**0.22**	**0.36**

Furthermore, the subset eval method with Genetic Search and multilayer perceptron as the classifier showed to be the best method among the supervised query expansion techniques. We compare these methods with the state of the art baseline methods, namely Relevance Model on Wikipedia (RMW) [10] and

Table 6: Comparison on all queries. * determines statistical significance over RMC and RMW assuming $\alpha = 0.05$

Topics	scheme	RMC	RMW	Unsupervised method	Supervised method
301-350	MAP	0.174	0.184	0.185	0.194*
	nDCG	0.302	0.300	0.306	0.313*
351-400	MAP	0.149	0.157	0.161*	0.163*
	nDCG	0.274	0.296	0.303*	0.301*
401-450	MAP	0.208	0.206	0.214*	0.222*
	nDCG	0.344	0.349	0.363*	0.364*

Relevance Model (RMC) expansion [10] methods. The two baselines, namely RMW and RMC are relevance models for query expansion based on the language modeling framework proposed in [24]. The relevance model is essentially a multinomial distribution which estimates the likelihood of word w given a query Q. In this model both query words and w are sampled independently and identically from a distribution R. The probability of a word in R can be computed as:

$$P(w|R) = \sum_{D \in F} P(D)P(w|D)P(Q|D) \tag{8}$$

where F is the set of documents that are pseudo-relevant to query Q. The relevance model use the top-retrieved documents from an initial search to serve as the set F. This is the model that is referred to as RMC. The work in [10] proposed that instead of looking at an initial search of general documents to instead use the top most related Wikipedia articles for this purpose. The relevance model based on Wikipedia articles is known as RMW.

Both of the proposed unsupervised and supervised methods perform significantly better across the three topics and on both of the evaluation metrics. This is shown in Table 6. The important advantage of the proposed supervised method is that it shows statistically significant improvement over RMC and RMW over all topics and in both metrics.

Our analysis of the observed results based on Relevance Model on Wikipedia (RMW) and Relevance Model (RMC) expansion methods provide some insight as to why both the proposed methods show a better performance. In many of the queries, the RMW expansion method ends up including irrelevant Wikipedia articles in the feedback document set for unambiguous queries that degrade the MAP for the query. For example for the query world bank criticism, the two articles that are extracted in our approach cover the whole intent of the query and do not include any irrelevant Wikipedia articles, while many of the extracted articles in the RMW method are unrelated, although they collectively cover the whole intent. This results in a high recall for the RMW method but at the cost of a lower precision. Table 7 shows a comparison of some example queries and the feedback documents for the RMW method and our method. It is important to note that both of our supervised and unsupervised methods use the same set of Wikipedia articles.

Table 7: Examples of feedback documents and their MAP values.

Query	Feedback documents from RMW	MAP	Feedback documents from our approach	MAP
world bank criticism	Imperial_Bank_of_Persia Dai-Ichi_Kangyo_Ban Bangladesh_climate_Multi_-Donor_Trust_Fund World_Bank Bad_bank Arun_III World_Bank_Group Australia_and_New_Zealand_-Banking_Group Monetary_reform Bank_of_America	0.0522	World_Bank_Group Criticism World_Bank	0.0899
endangered species mammals	United_States_Fish_and_-Wildlife_Service_list_of_endangered_species_of_mammals_-and_birds Lists_of_extinct_species List_of_mammals_of_Australia Fauna_of_Connecticut Lists_of_animals Corynorhinus Longbeaked_echidna Rhinoceros_(genus) Fauna_of_Borneo Canavalia_pubescens	0.0631	Endangered_species Mammal	0.1044
magnetic levitation maglev	Levitation Bangalore_Monorail SCMaglev National_Maglev_Initiative Electromagnetic_suspension Shanghai_Maglev_Train Inductrack William_J._Beaty Maglev	0.4862	Magnetic_levitation Maglev	0.5051

We further investigate the performance of RMC compared to our proposed methods. When looking at specific queries where RMC and our approaches have a difference in performance, we were able identify two sets of queries, the first set of which include queries on which our approach performs better than RMC shown in Table 8 and the second set where RMC has a better performance shown in Table 9. When looking at the differences between the queries, we notice that the queries where RMC shows a better performance seem to be very specific and on topics that do not necessarily have relevant content on Wikipedia. For instance, for the query "unsolicited faxes" there does not seem to be sufficient relevant articles on Wikipedia to warrant the extraction of a specific feedback document set and therefore the expanded word set is not as accurate in our approach compared to RMC. On the other hand, as shown in Table 8, more general queries, such as airport security that are on

Table 8: Sample queries where WikiRelevance outperform RMC

Query	Expansion terms in RMC not WikiRelevance	Expansion terms in WikiRelevancce not RMC
airport security	passeng, thei, aviat	attack, flight, country
land mine ban	state, feder, year, million	action, clearance, international, personnel
pope beatifications	saint, martyr, pius, rome, cardinal	church, pope, cathol, runci, protest

Table 9: Sample queries where RMC outperform WikiRelevance

Query	Expansion terms in RMC not WikiRelevance	Expansion terms in WikiRelevancce not RMC
Most Dangerous Vehicles	thei, accid, time, year, militari	sign, road, image, good, traffic
Unsolicited Faxes	machin, legisl, junk, ad, bill	call, marketing, direct, acma, act

topics that have sufficient coverage on Wikipedia, result in superior feedback document set in our approach and therefore would include more relevant terms in the expansion word set.

3.3 Robustness

A robust query expansion method will improve many and hurt only a small number of queries. The higher the number of improved queries and lower the number of hurt queries are, the more robust the query expansion method is. Robustness is defined as the number of queries that are negatively impacted by the query expansion methods [10]. An ideal query expansion method would improve robustness on any given query. However, in practice, query expansion methods do not necessarily improve the results on all queries; therefore, those methods that improve the results on a higher number of queries are preferred. Figure 6 shows the comparison of the robustness for the four methods. For the Supervised approach, the classifier subset eval feature selection with greedy search is applied to select the best features, and multilayer perceptron is used as the learner. In the unsupervised method, WPR is used as the scheme to evaluate the terms. As seen in the Figure, the supervised approach is more robust than the other approaches. The number of queries that their MAP results are improved is significantly higher in the supervised method compared to the other three. The supervised approach makes 68.6% of the queries better, in comparison to unsupervised method, RMW, and RMC that improve only 54.6%, 52%, and 50% of the queries, respectively.

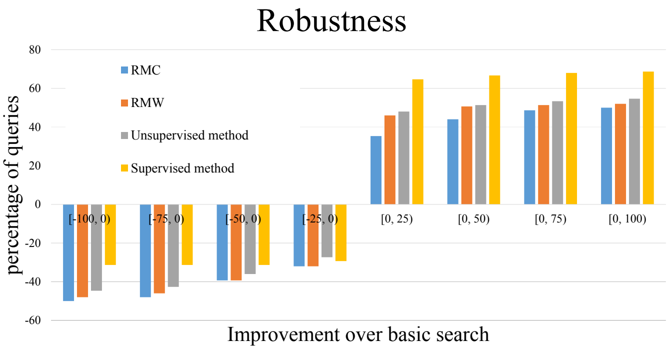

Fig. 6: Comparative analysis of the robustness results (diagram shows the accumulative values).

4 Related Works

Natural language and structure-free queries are prone to error due to the short length of the query and their ambiguity [4]. The most common problem is the vocabulary mismatch problem which refers to how the users may use different terms (synonymy and polysemy, word inflections) when referring to the same concept. Such problems may result in an inability of a search engine to retrieve the desired documents, hence decreasing the recall and precision of the system [4].

To overcome this problem several solutions have been proposed, such as query refinement, pseudo relevance feedback, and result diversification. One of the popular solutions is to expand the query with other terms to best capture the actual intent of the user. Query reformulation and expansion techniques try to tackle the vocabulary mismatch problem, which is primarily concerned with finding semantically similar documents to queries that are not necessarily syntactically similar. Here we review the techniques and approaches that have been proposed to address this problem. Query expansion can be interactive or automatic. In the interactive approach, expansion terms are suggested to the user, and the user can interactively choose or ignore them. In the automatic approach, the terms are automatically added to the query without the user noticing. Here we focus on Automatic Query Expansion (AQE). Different approaches used in AQE can be categorized as follows:

- Thesaurus
  WordNet [29] is a lexical database for the English language. It groups terms into synonym sets called synsets and presents a definition for each term along with some examples. Other than synonyms, definitions, and examples, WordNet provides hypernyms, hyponyms, meronym, and many other relations between terms. Using this resource for query expansion

enable us to use for any word in the query, its synonym, hyponyms, or even some terms from term definitions or glosses as the expansion terms[30]. In other words instead of only searching for the query terms in the documents, the search engine would search for all the terms that are in the synset or hyponym set or definition of the query terms.

- Relevance Feedback
  In Relevance Feedback (RF) methods, relevant documents to the query are considered, and some related words are extracted from those documents for the purpose of query expansion. However in reality, relevant documents are not available, therefore Pseudo Relevance Feedback (PRF) is introduced [4]. PRF takes the initial result set of the specified query and assumes that those results are related to the query. Therefore it uses them to extract terms related to the query for query expansion.
  This approach essentially reinforces the system's original decision, by making the expanded query more similar to the retrieved relevant documents, whereas AQE tries to form a better match with the users' underlying intentions. Although this approach is very promising, it hurts the results in case the initial set of results have non-relevant documents among them [31].
- Query Logs
  Query logs contain information about the interaction of the user with the search engine while formulating a query and browsing the results. They are called query logs or click through logs, because they show which documents the user had in fact clicked on after searching for a query [32]. The documents that a user clicks on can be assumed to be related to the intent of the user.
  The use of click through logs is one of the ways that can be used to find relevant documents and terms to a query [33]. Random walk techniques can be applied on the query-document graph in order to retrieve relevant documents [33]. Also it is possible to extract similar queries from the graph after applying random walk by clustering the queries [34,35].
- Linked Open Data
  Recently researchers have considered the semantic analysis of search queries in the context of the Linked Open Data (LOD) [35–38]. They explore DBpedia, Wikipedia, or Freebase [39,40] to this end. The main difference between such databases and WordNet is that in WordNet, only a limited set of relationships are defined, however in knowledge bases such as DBpedia one can find many different types of relations that are defined as entity properties.

In our work, we have opted to use Wikipedia as the source of information for query expansion. In other work, it has already been shown that using Wikipedia for query expansion can be more effective than WordNet since Wikipedia is a large, dynamic, and objective knowledge base, which rests on articles that are focused on a single concept [37,41]. Also Wikipedia senses cover more search results than WordNet [42]. As such, many recent works are defined around Wikipedia and similar knowledge bases instead of WordNet.

Some of the work that use Wikipedia or similar knowledge bases are reviewed in the following.

Bruce et al. [37] extract the aspects of a query using Wikipedia through title matching between Wikipedia articles and query aspects. To find the best aspects, they use a linked probability measure and apply their detected under-represented aspects in the AbraQ query expansion framework [43]. Similarly, Liu et al [44] represent each aspect of the query as a vector. Query expansion is performed as an iterative method in which in each step a term is added to the expansion set from one of the aspects of the query. Also in their work, aspects can carry different weights. This means that some aspects are more probable to be understood from the query compared to other ones.

The work in [45] finds the DBpedia concepts related to a unambiguous query. In their first step they extract all the concepts that contain one of the segments of the query in either its label, or in Wikipedia text or text of the link to that Wikipedia article, and in the second step they apply a supervised machine learning method to rank their list of extracted concepts. They evaluate their approach by testing how the extracted concepts are related to the query, hence their approach is not concerned with the term selection part which is one of the important contributions of our work. Moreover, for the training purposes of the paper, the features are extracted from manually annotated documents.

In another work, [10] proposes a similar idea to our work, which we compare to as the baseline. For entity article selection, they group queries into three classes (EQ: specific entity, AQ: BQ: broad), the first two groups are queries for which a Wikipedia article with the exact same title can be found. For AQ they apply a heuristic disambiguation method and at the end they select one entity for the query to select the terms from. For term selection, they use a parametrized formula to weigh terms and for finding those parameters, they apply a supervised learning method on a training set. The authors only report their results for the EQ and AQ queries. In our work, we propose a novel method for term extraction from Wikipedia article (which can be more than one) for a query. Also we evaluate the proposed method on all the queries even if a Wikipedia article with the same title cannot be found. For such queries, we propose a method to extract entities related to the query. Such queries are actually the most challenging ones.

Another interesting research is the work of Bendersky et. al [46] which is a relevance model over any unstructured data source. To weight the terms for expansion, they use a parametrized approach, and for parameter tuning they use a supervised learning algorithm over a training set. In our work we specifically use Wikipedia instead of different sources and we believe this choice makes the articles to be more uniform and less prone to error, since our concept extraction is specifically designed for Wikipedia.

5 Concluding Remarks

In this paper we propose two supervised and unsupervised query expansion methods which are inspired by the pseudo relevance feedback query expansion approach, we first find related Wikipedia articles to the user queries. Considering the extracted Wikipedia articles as feedback documents, our approaches weigh terms in those articles and select the top terms for the purpose of expansion. While in the pseudo-relevance feedback method, there is the possibility that the top results, which are considered to be relevant to the query and helpful for query expansion, contain irrelevant documents that can negatively impact the expansion results; in our approach, we extract Wikipedia articles that are very highly likely to be related to the query and therefore decrease the probability of irrelevant documents being included as a part of the feedback document collection. We make use of the redirect and disambiguation articles of Wikipedia to help overcome the vocabulary mismatch problem. Another challenging yet effective step in our work is the selection of the best set of terms for query expansion from the extracted documents. Unlike most approaches that use one or two important features of the terms in a document for expansion, we have used both supervised and unsupervised methods for selecting the terms. Based on the empirical evaluations that we have reported in this paper, we believe that the proposed approaches (supervised and unsupervised) are most effective when used for expanding *information seeking* and *general queries*.

As future work, we are interested in extending this work in two main directions:

- First, based on the findings of this paper, it seems appropriate to distinguish between queries that are general information seeking compared and specific queries. We would like to explore whether the adoption of non-Wikipedia feedback documents for specific queries and supervised query expansion proposed in this paper for generic information seeking queries would result in more accurate query expansion performance. It should be noted that in order to be able to achieve this objective, we need to first develop models that can automatically distinguish between the two query types.
- The second area that we would like to explore is the consideration of structured knowledge bases such as Dbpedia and Freebase in conjunction with the statistical models that have been proposed in this paper. The information presented in knowledge bases such as Dbpedia are expressed in RDF format that are in essence shaped as graphs. The use of such knowledge bases in addition to the consideration of Wikipedia in the form of terms and documents could provide the added advantage of allowing us to develop metrics that take the graph structure of the knowledge bases into account when deciding about relevant articles and terms. In addition, such semantic approach can also allow us to use reasoning techniques to determine the relationship between the query words and the concepts of the knowledge base.

References

1. Hu, J., Wang, G., Lochovsky, F., Sun, J.t., Chen, Z.: Understanding user's query intent with wikipedia. In: Proceedings of the 18th international conference on World wide web, ACM (2009) 471–480
2. Di Marco, A., Navigli, R.: Clustering and diversifying web search results with graph-based word sense induction. Computational Linguistics **39**(3) (2013) 709–754
3. Spink, A., Wolfram, D., Jansen, M.B., Saracevic, T.: Searching the web: The public and their queries. Journal of the American society for information science and technology **52**(3) (2001) 226–234
4. Carpineto, C., Romano, G.: A survey of automatic query expansion in information retrieval. ACM Computing Surveys (CSUR) **44**(1) (2012) 1
5. Rocchio, J.J.: Relevance feedback in information retrieval. (1971)
6. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Journal of the American Society for Information science **27**(3) (1976) 129–146
7. Doszkocs, T.E.: Aid, an associative interactive dictionary for online searching. Online Review **2**(2) (1978) 163–173
8. Robertson, S.E., Walker, S., Beaulieu, M., Willett, P.: Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive track. Nist Special Publication SP (1999) 253–264
9. Carpineto, C., De Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Transactions on Information Systems (TOIS) **19**(1) (2001) 1–27
10. Xu, Y., Jones, G.J., Wang, B.: Query dependent pseudo-relevance feedback based on wikipedia. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM (2009) 59–66
11. Li, Y., Luk, W.P.R., Ho, K.S.E., Chung, F.L.K.: Improving weak ad-hoc queries using wikipedia asexternal corpus. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2007) 797–798
12. Chakaravarthy, V.T., Gupta, H., Roy, P., Mohania, M.: Efficiently linking text documents with relevant structured information. In: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment (2006) 667–678
13. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, ACM (2011) 1–8
14. Ferragina, P., Scaiella, U.: Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM (2010) 1625–1628
15. Jovanovic, J., Bagheri, E., Cuzzola, J., Gasevic, D., Jeremic, Z., Bashash, R.: Automated semantic tagging of textual content. IT Professional **16**(6) (2014) 38–46
16. Hatcher, E., Gospodnetic, O., McCandless, M.: Lucene in action (2004)
17. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Readings in information retrieval **24**(5) (1997) 355–363
18. Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. (2003)
19. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
20. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research **3** (2003) 1157–1182
21. Aha, D.W., Bankert, R.L.: A comparative evaluation of sequential feature selection algorithms. In: Learning from Data. Springer (1996) 199–206
22. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. Pattern Analysis and Machine Intelligence, IEEE Transactions on **19**(2) (1997) 153–158
23. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Best agglomerative ranked subset for feature selection. In: FSDM. (2008) 148–162
24. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2001) 120–127

25. Dalton, J., Dietz, L., Allan, J.: Entity query feature expansion using knowledge base links. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, ACM (2014) 365–374
26. Al-Shboul, B., Myaeng, S.H.: Query phrase expansion using wikipedia in patent class search. In: Information Retrieval Technology. Springer (2011) 115–126
27. Järvelin, K., Kekäläinen, J.: Ir evaluation methods for retrieving highly relevant documents. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2000) 41–48
28. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2004) 25–32
29. Fellbaum, C.: WordNet. Wiley Online Library (1998)
30. Liu, S., Liu, F., Yu, C., Meng, W.: An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2004) 266–272
31. Xu, J., Croft, W.B.: Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems (TOIS) **18**(1) (2000) 79–112
32. Croft, W.B., Metzler, D., Strohman, T.: Search engines: Information retrieval in practice. Addison-Wesley Reading (2010)
33. Craswell, N., Szummer, M.: Random walks on the click graph. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2007) 239–246
34. Radlinski, F., Szummer, M., Craswell, N.: Inferring query intent from reformulations and clicks. In: Proceedings of the 19th international conference on World wide web, ACM (2010) 1171–1172
35. Dang, V., Croft, B.W.: Query reformulation using anchor text. In: Proceedings of the third ACM international conference on Web search and data mining, ACM (2010) 41–50
36. Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: InfoScale. Volume 152. (2006) 1
37. Bruce, C., Gao, X., Andreae, P., Jabeen, S.: Query expansion powered by wikipedia hyperlinks. In: AI 2012: Advances in Artificial Intelligence. Springer (2012) 421–432
38. Crabtree, D.W., Andreae, P., Gao, X.: Exploiting underrepresented query aspects for automatic query expansion. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2007) 191–200
39. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. Springer (2007)
40. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM (2008) 1247–1250
41. Cheung, J.C.K., Li, X.: Sequence clustering and labeling for unsupervised query intent discovery. In: Proceedings of the fifth ACM international conference on Web search and data mining, ACM (2012) 383–392
42. Santamaría, C., Gonzalo, J., Artiles, J.: Wikipedia as sense inventory to improve diversity in web search results. In: Proceedings of the 48th annual meeting of the association for computational Linguistics, Association for Computational Linguistics (2010) 1357–1366
43. Crabtree, D.W., Andreae, P., Gao, X.: Exploiting underrepresented query aspects for automatic query expansion. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2007) 191–200
44. Liu, X., Bouchoucha, A., Sordoni, A., Nie, J.Y.: Compact aspect embedding for diversified query expansions. In: Proc. of AAAI. Volume 14. (2014) 115–121
45. Meij, E., Bron, M., Hollink, L., Huurnink, B., De Rijke, M.: Learning semantic query suggestions. The Semantic Web-ISWC 2009 (2009) 424–440
46. Bendersky, M., Metzler, D., Croft, W.B.: Effective query formulation with multiple information sources. In: Proceedings of the fifth ACM international conference on Web search and data mining, ACM (2012) 443–452