Query Expansion Using Pseudo Relevance Feedback on Wikipedia

Andisheh Keikha
Laboratory for Systems, Software and Semantics (LS3),
Ryerson University
andisheh.keikha@ryerson.ca

Faezeh Ensan
Ferdowsi University of Mashhad
ensan@um.ac.ir

Ebrahim Bagheri
Laboratory for Systems, Software and Semantics (LS3),
Ryerson University
bagheri@ryerson.ca

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.

Keywords

Query Expansion; Wikipedia; ad hoc Information Retrieval

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 [11, 16]. 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 [32]. 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 [11]. For instance, a user entering the query "Hotel California" might want to search for the Eagle's album, or be interested in hotels in California or a hotel named California.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PLDI '13 June 16–19, 2013, Seattle, WA, USA

© 2015 ACM. ISBN 123-4567-24-567/08/06.

DOI: 10.475/123_4

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 [11]. 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 user's 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 [7]. 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 [7], Rocchio's Weight [29], Binary Independence Model [27], Chi-Square [12], Robertson Selection Value [28], and Kullback-Leibur Distance [6], 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 [33]. Li et al [21] 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 [33] 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 [21] 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 (only unigrams) 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.

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 identify the most relevant Wikipedia articles to a given search query, traditional forms of text annotation [8, 13, 24] cannot be directly applied due to the very short length of a query and hence, lack of context. 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 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.

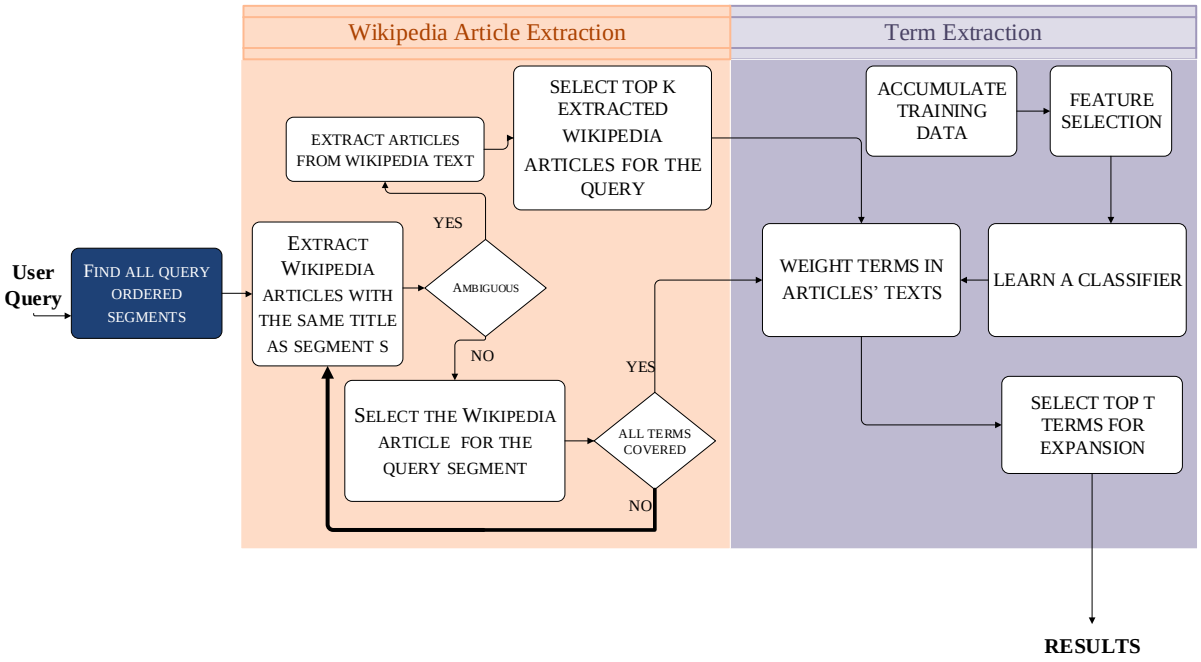

Figure 1: Approach overview

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", 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. Also if no Wikipedia article is found for a query segment, the approach for ambiguous queries is considered.

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 [19], which assumes that the correct sense of a word, when lacking context, is the one that is the most frequently observed.

To this end, we rank Wikipedia articles based on their relevance to the query terms according to the following equation

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

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

adopted from [15]:

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

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. This norm value is multiplied because it is more probable that a small document that has specific terms is more related to the query than a larger document that has those terms. This normalization value is basically the reverse of the square root of number ot terms.

This value ($Rank_d(q)$) is calculated for each document and documents are ranked based on their rank. The higher ranked documents are assumed to be more relevant to the query than the lower ranked ones.

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

In the unsupervised approach, 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 1. 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 1. It is important to mention that these probabilities are not going to be equal to zero ever, because the terms are extracted from Wikipedia articles that are considered as relevant documents.

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" [26]. This similarity scheme is a fast approach to calculate similarities between two terms using WordNet. The performance and accuracy level of this scheme makes it a good match for our approach. 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.

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

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 $node_j$.

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

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.

Table 1: Term weighting schemes.

Function	Formula
TF [31]	$0.5 + \frac{0.5 \times f(t,d)}{max\{f(w,d) : w \in d\}}$
TF-IDF [25]	$tf(t,d) \times \log \frac{N}{\lvert\{d \in D : t \in d\}\rvert}$
BIM [7]	$\log \frac{p(t\lvert R)[1-p(t\lvert C)]}{p(t\lvert C)[1-p(t\lvert R)]}$
Chi-Square [7]	$\frac{(p(t\lvert R)-p(t\lvert C))^2}{p(t\lvert C)}$
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

To enrich our features with some more 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 [11] 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 1 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 as a linear or non-linear model. 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. The overview of the steps of this supervised approach is shown in Figure 2. As the first step, 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 [14], 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 from the three set of topics that are for ad hoc retrieval evaluation 301-450). The queries used in training were not used in the testing process. The candidate terms (unigrams) 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) [14]. 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 [1]: 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 [14]. These search methods are often combined with evaluation measures to produce variants for FSS. In our fea-

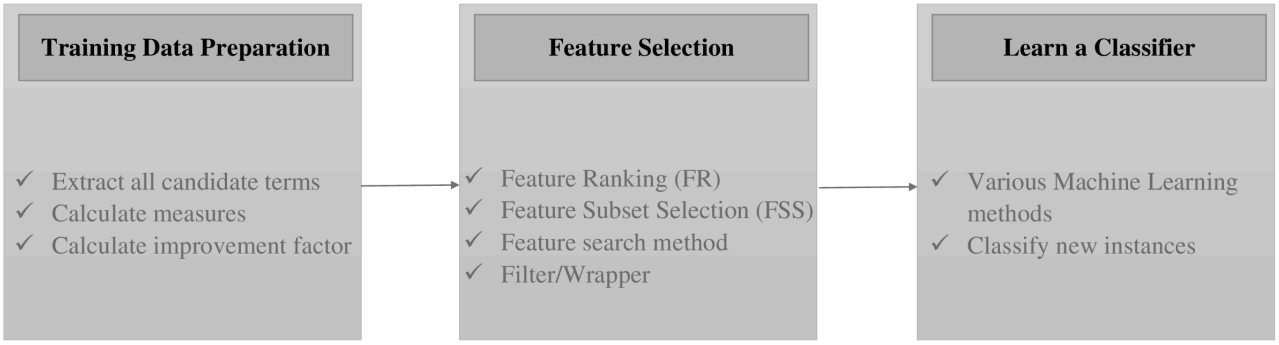

Figure 2: Overview of the supervised term selection process.

ture selection algorithm, we use the Best First Search (BFS) which is a heuristic algorithm [17]. 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 [30]. 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 [14].

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, Radial Basis Function (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). 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 [33]. These two methods propose state of the art query expansion methods that are vastly used for comparative analysis in this domain [2,10,20,33]. We employ two commonly used evaluation measures for evaluating our work, namely: i) Mean Average Precision (MAP), and ii) Normalized Discounted Cumulative Gain (nDCG) [5,18].

3.1 Comparative Analysis

We perform our experiments on Topics 301-350, 351-400, and 401-450 of the TREC 2010 dataset.

3.1.1 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. The results of the performance of the unsupervised query expansion method based on different term weighting schemes are shown in Table 2. 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. 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.

3.1.2 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

Table 2: 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	**0.181**	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	0.148	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	**0.216**	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

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 3 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 feature selection method is employed: Genetic Search or Scatter Search. The feature selection method has selected WD, and BIM as the best set of features.

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 [14]. 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 4. 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 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, the Weighted Page Rank (WPR) scheme is the better term weighting schemes from among the unsupervised query expansion methods. 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 two methods with the state of the art baseline method, namely Relevance Model on Wikipedia (RMW) [33] and Relevance Model (RMC) expansion [33] methods. 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 5. 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.

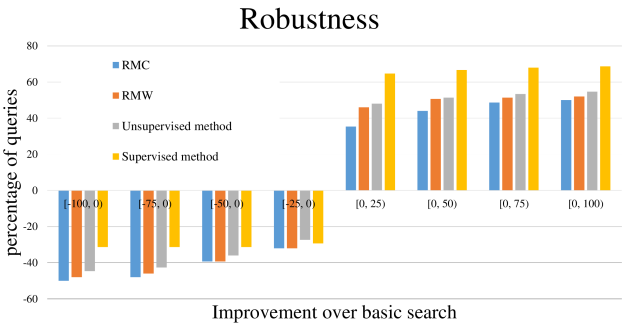

Figure 3: Comparative analysis of the robustness results (diagram shows the accumulative values).

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 [33]. 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 3 shows the comparison of the robustness for the four methods. The figure shows the difference between the MAP results of the expanded query and the original query accumulatively. 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.

Table 3: Feature Selection Evaluation

Feature Selection Method	Selected Features	Topics	MAP	nDCG
No feature selection	All of the 8 schemes	301-350	0.164	0.271
		351-400	0.153	0.268
		401-450	0.202	0.320
Classifiersubseteval + (BestFirst Search/Greedy stepwise /Linear forward selection)	TF-IDF	301-350	0.176	0.297
		351-400	0.158	0.287
		401-450	0.203	0.341
Classifiersubseteval + Exhaustive Search	WD, tf BIM, Chi^2	301-350	0.182	0.290
		351-400	0.148	0.249
		401-450	0.201	0.332
Classifiersubseteval + (Genetic Search/Scatter Search)	WD, BIM	301-350	**0.194**	**0.312**
		351-400	**0.163**	**0.302**
		401-450	**0.220**	**0.359**
Classifiersubseteval + Race Search	WD, WPR, tf TF-IDF, BIM, Chi^2	301-350	0.178	0.297
		351-400	0.148	0.287
		401-450	0.221	0.362
Classifiersubseteval + Random Search	TF-IDF, BIM Chi^2	301-350	0.188	0.309
		351-400	0.158	0.297
		401-450	0.211	0.350
Latent Semantic Analysis + Ranker	BIM	301-350	0.188	0.306
		351-400	0.157	0.291
		401-450	0.216	0.358
Wrapper subset eval + Genetic Search	WPR, WPR_c tf, BIM, Chi^2	301-350	0.179	0.290
		351-400	0.148	0.278
		401-450	0.201	0.349

4. RELATED WORK

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.

Bruce et al. [4] 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 underrepresented aspects in the AbraQ query expansion framework [9]. Similarly, Liu et al [22] 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 [23] 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, [33] 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 [3] 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. CONCLUSION

In this paper we propose two supervised and unsupervised query expansion methods which are inspired by the pseudo relevance feedback query expansion approach considering the extracted Wikipedia articles as feedback documents. Our approaches weigh terms in the specified articles and select the top terms for the purpose of expansion.

Table 4: Learning method Evaluation

Learning Method	Topics	MAP	nDCG
Linear Regression	301-350	0.173	0.282
	351-400	0.160	0.289
	401-450	0.201	0.340
Multi layer perceptron	301-350	0.188	0.311
	351-400	0.162	0.301
	401-450	0.219	0.361
Pace regression	301-350	0.181	0.288
	351-400	0.159	0.301
	401-450	0.196	0.340
RBF Network	301-350	0.179	0.290
	351-400	0.148	0.279
	401-450	0.203	0.331
Additive Regression	301-350	0.183	0.289
	351-400	0.163	0.281
	401-450	0.220	0.359

Table 5: 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	0.193*	0.194*
	nDCG	0.302	0.300	0.314*	0.313*
351-400	MAP	0.149	0.157	0.171*	0.163*
	nDCG	0.274	0.296	0.307*	0.301*
401-450	MAP	0.208	0.206	0.215	0.222*
	nDCG	0.344	0.349	0.358	0.364*

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.

Finally we compared the best results that was obtained from the supervised method that used multi layer perceptron as the machine learning method, with state of the art methods in query expansion. Our results shows significant improvement over traditional approaches.

6. REFERENCES

[1] D. W. Aha and R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Learning from Data*, pages 199–206. Springer, 1996.

[2] B. Al-Shboul and S.-H. Myaeng. Query phrase expansion using wikipedia in patent class search. In *Information Retrieval Technology*, pages 115–126. Springer, 2011.

[3] M. Bendersky, D. Metzler, and W. B. Croft. Effective query formulation with multiple information sources. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 443–452. ACM, 2012.

[4] C. Bruce, X. Gao, P. Andreae, and S. Jabeen. Query expansion powered by wikipedia hyperlinks. In *AI 2012: Advances in Artificial Intelligence*, pages 421–432. Springer, 2012.

[5] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32. ACM, 2004.

[6] C. Carpineto, R. De Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, 19(1):1–27, 2001.

[7] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1, 2012.

[8] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. Mohania. Efficiently linking text documents with relevant structured information. In *Proceedings of the 32nd international conference on Very large data bases*, pages 667–678. VLDB Endowment, 2006.

[9] D. W. Crabtree, P. Andreae, and X. Gao. Exploiting underrepresented query aspects for automatic query expansion. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 191–200. ACM, 2007.

[10] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM, 2014.

[11] A. Di Marco and R. Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754, 2013.

[12] T. E. Doszkocs. Aid, an associative interactive dictionary for online searching. *Online Review*, 2(2):163–173, 1978.

[13] P. Ferragina and U. Scaiella. 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*, pages 1625–1628. ACM, 2010.

[14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] E. Hatcher, O. Gospodnetic, and M. McCandless. Lucene in action, 2004.

[16] J. Hu, G. Wang, F. Lochovsky, J.-t. Sun, and Z. Chen. Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web*, pages 471–480. ACM, 2009.

[17] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(2):153–158, 1997.

[18] K. Järvelin and J. Kekäläinen. 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*, pages 41–48. ACM, 2000.

[19] J. Jovanovic, E. Bagheri, J. Cuzzola, D. Gasevic, Z. Jeremic, and R. Bashash. Automated semantic tagging of textual content. *IT Professional*, 16(6):38–46, 2014.

[20] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[21] Y. Li, W. P. R. Luk, K. S. E. Ho, and F. L. K. Chung. 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*, pages 797–798. ACM, 2007.

[22] X. Liu, A. Bouchoucha, A. Sordoni, and J.-Y. Nie. Compact aspect embedding for diversified query expansions. In *Proc. of AAAI*, volume 14, pages 115–121, 2014.

[23] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. De Rijke. Learning semantic query suggestions. *The Semantic Web-ISWC 2009*, pages 424–440, 2009.

[24] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[25] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[26] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[27] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

[28] S. E. Robertson, S. Walker, M. Beaulieu, and P. Willett. Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive track. *Nist Special Publication SP*, pages 253–264, 1999.

[29] J. J. Rocchio. Relevance feedback in information retrieval. 1971.

[30] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz. Best agglomerative ranked subset for feature selection. In *FSDM*, pages 148–162, 2008.

[31] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24(5):355–363, 1997.

[32] A. Spink, D. Wolfram, M. B. Jansen, and T. Saracevic. Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3):226–234, 2001.

[33] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66. ACM, 2009.