Learning to Expand Queries Using Entities

Wladmir C. Brandão* Edleno S. de Moura† Rodrygo L. T. Santos*
Altigran S. da Silva† Nivio Ziviani*

*Computer Science Department
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
{wladmir,rodrygo,nivio}@dcc.ufmg.br

†Computer Science Department
Universidade Federal do Amazonas
Manaus, AM, Brazil
{edleno,alti}@dcc.ufam.edu.br

Abstract

A substantial fraction of web search queries contain references to entities, such as persons, organizations, and locations. Recently, methods that exploit named entities have been shown to be more effective for query expansion than traditional pseudo-relevance feedback methods. In this paper, we introduce a supervised learning approach that exploits named entities for query expansion, using Wikipedia as a repository of high-quality feedback documents. In contrast to existing entity-oriented pseudo-relevance feedback approaches, we tackle query expansion as a learning to rank problem. As a result, not only do we select effective expansion terms, but we also weigh these terms according to their predicted effectiveness. To this end, we exploit the rich structure of Wikipedia articles in order to devise discriminative term features, including each candidate term's proximity to the original query terms, as well as its frequency across multiple article fields and in category and infobox descriptors. Experiments on three TREC web test collections attest the effectiveness of our approach, with gains of up to 23.32% in terms of MAP, 19.49% in terms of P@10, and 7.86% in terms of nDCG compared to a state-of-the-art approach for entity-oriented query expansion.

1 Introduction

Search engines have become the primary gateway for finding information on the Web. The leading web search engine has recently reported to be answering a total of 100 billion queries each month, and to be tracking over 30 trillion unique URLs (Cutts, 2012). Given the size of the Web and the short length of typical web search queries (Gabrilovich et al., 2009; Jansen et al., 2000), there may be billions of pages matching a single query. In this context, an improved understanding of the information need underlying the user's query becomes paramount for an improved search experience. Indeed, misinterpreting this query may result in relevant documents never being retrieved, regardless of how sophisticated the subsequent ranking process is (Li, 2010).

A particularly effective query understanding operation is query expansion (Lavrenko and Croft, 2001; Rocchio, 1971; Zhai and Lafferty, 2001b). Given a set of feedback documents, such an approach appends additional terms to the query in order to close the gap between the vocabulary of the query and that of potentially relevant documents. Automatic query expansion is typically performed in the context of pseudo-relevance feedback (PRF), in which case the feedback set comprises the top retrieved documents for the query, which are assumed to be relevant (Rocchio, 1971). However, this assumption is often invalid, which has prompted the development of improved mechanisms for selecting effective feedback documents (He and Ounis, 2009), and effective expansion terms (Cao et al., 2008; Lee et al., 2009; Udupa et al., 2009).

Recently, methods that exploit named entities have also been shown to be effective for query expansion (Xu et al., 2009). In particular, queries with named entities, such as persons, organizations, and locations, account for over 70% of the web search traffic (Guo et al., 2009). Such queries offer the opportunity to use knowledge bases as repositories of high-quality feedback documents. In this paper, we resort to Wikipedia as a feedback repository, and introduce a novel supervised learning approach to select effective expansion terms. In contrast to related approaches from the literature that leveraged Wikipedia as a repository of feedback documents (e.g., (Milne et al., 2007; Xu et al., 2008, 2009)), we propose to tackle query expansion as a learning to rank problem. As a result, not only do we select effective expansion terms, but we also weigh these terms according to their predicted effectiveness when added to the query.

Other methods that also use external corpus and supervised models to effectively reformulate queries has been proposed (Diaz and Metzler, 2006; Lin et al., 2011). They use information extracted from external sources and supervised learning approaches to select and weight terms for query expansion. Some of them combine information from multiple external sources (Bendersky et al., 2012; Weerkamp et al., 2012). Differently from these methods, we exploit previously ignored semantic evidence from a knowledge base as term features within our supervised learning approach.

Given a query, our approach locates the most appropriate Wikipedia articles that describe the entities referenced in the query. By exploiting the rich structured content of each article, we devise a comprehensive representation of candidate expansion terms, comprising features such as each term's proximity to the original query terms, its frequency

across multiple article fields (e.g., title, summary, body, anchor text from related articles), as well as in category and infobox descriptors. Moreover, to ensure that our learning process is guided towards identifying effective expansion terms, we automatically assign binary performance-oriented labels to training examples in order to weigh candidate terms proportionally to their predicted retrieval performance. Thorough experiments using three standard TREC web test collections attest the effectiveness of our approach, with significant improvements compared to state-of-the-art traditional as well as entity-oriented PRF approaches.

The major contributions of this paper are:

1. We propose a novel learning to rank approach to identify and weight effective expansion terms related to entities in web search queries. In contrast to existing supervised approaches in the literature, our approach weights terms proportionally to their predicted effectiveness.

2. We exploit previously ignored semantic evidence from a knowledge base as term features within our supervised learning approach.

3. We thoroughly evaluate the proposed approach using three standard TREC web test collections in contrast to state-of-the-art PRF and entity-oriented pseudo-relevance feedback (ePRF) approaches.

The remainder of this paper is organized as follows. In Section 2, we review the related literature on query expansion for web search. In Section 3, we describe our learning to rank approach for entity-oriented query expansion. In Sections 4 and 5, we describe the setup and the results of the experimental evaluation of our approach, respectively. Finally, in Section 6, we present our concluding remarks as well as directions for future research.

2 Related Work

The enormous size of the Web (Cutts, 2012) and the short length of typical web search queries (Gabrilovich et al., 2009; Jansen et al., 2000) most often results in an amount of documents matching a user's query that by far exceeds the very few top ranking positions that the user is normally willing to inspect for relevance (Silverstein et al., 1999). Query understanding aims to derive a representation of the user's query better suited for a search engine (Li, 2010). Typical query understanding operations include refinements of the original query, such as spelling correction (Li et al., 2006), acronym expansion (Jain et al., 2007), stemming (Peng et al., 2007; Porter, 1980), term deletion (Kumaran and Allan, 2008), query segmentation (Risvik et al., 2003), and query topic classification (Beitzel et al., 2005).

A query understanding operation of particular interest to the work presented in this paper is query expansion (Rocchio, 1971). In particular, query expansion aims to enhance the representation of the user's initial query by appending useful terms to it. Typically, expansion terms are automatically selected from a set of feedback documents (Lavrenko and

Croft, 2001; Rocchio, 1971; Zhai and Lafferty, 2001b), which are assumed to be relevant. Such a *pseudo*-relevance feedback (PRF) assumption is often invalid, leading to an improper expansion that may cause the subsequent ranking process to drift away from the user's information need (Mitra et al., 1998). In order to overcome this limitation, alternative approaches have been proposed to select more effective terms as well as more effective feedback documents for improving PRF.

Regarding an improved selection of expansion terms, (Cao et al., 2008) found that a non-negligible fraction of expansion terms identified by traditional PRF approaches is either neutral or harmful to the effectiveness of the initial query. As a result, they proposed a supervised classification approach using support vector machines (SVM) to predict the usefulness of expansion terms. In a similar vein, (Udupa et al., 2009) found that the usefulness of a term may vary drastically depending on the already selected terms. Hence, they proposed to take into account term interactions in order to identify a useful *set* of expansion terms. Their approach was based on a spectral partitioning of the weighted term-document matrix using singular value decomposition (SVD). Both approaches showed significant improvements compared to state-of-the-art PRF approaches, such as relevance models (Lavrenko and Croft, 2001) and model-based feedback (Zhai and Lafferty, 2001b). Focusing on difficult queries, Kotov and Zhai (Kotov and Zhai, 2012) conduct a study on methods leveraging the ConceptNet knowledge base to improve of the search results for these poorly performing queries. They proposed a supervised approach using generalized linear regression to select concepts from ConcetpNet and use them to expand difficult queries.

Regarding the selection of feedback documents, various PRF approaches have been proposed to leverage high-quality external resources as feedback, such as a log of the queries submitted by previous users (Cui et al., 2002), a social annotation collection (Lin et al., 2011), a knowledge base such as Wikipedia (He and Ounis, 2007; Li et al., 2007; Milne et al., 2007), and ConceptNet (Kotov and Zhai, 2012), or even a combination of multiple sources (Bendersky et al., 2012; Weerkamp et al., 2012). Another effective approach to query expansion exploits knowledge bases as repositories of feedback *entities*—as opposed to feedback documents. Such an ePRF approach relies on the availability of structured information about named entities identified in the user's initial query (Guo et al., 2009). For instance, (Xu et al., 2008, 2009) proposed an ePRF approach that recognizes the most representative entity in a query, and uses Wikipedia articles related to this entity as feedback documents for query expansion. In their approach, the top terms extracted from the feedback documents are selected according to the terms' likelihood of being effective, as predicted by an SVM classifier, and appended to the initial query. This approach was shown to outperform a state-of-the-art PRF approach based upon relevance models (Lavrenko and Croft, 2001).

In this paper, we introduce a novel supervised approach for the ePRF problem. In contrast to previous approaches, we do not rely on a binary classification of candidate expansion terms as either useful or non-useful. Instead, we tackle query expansion as a learning to rank problem, in order to directly learn an effective ranking of the candidate terms related to an entity in the query. As a result, not only do we choose effective

terms for expansion, but we also learn how to weigh their relative importance in the expanded query. To enable our approach, we exploit term features inspired by previous research (Xu et al., 2008, 2009), as well as novel features derived from the Wikipedia article representing a query entity, such as each term's distribution across multiple article fields, and its proximity to the original query terms.

3 Query Expansion Using Entities

The presence of a named entity in a query provides a unique opportunity for web search engines to improve their understanding of the user's information need, by exploiting the rich evidence about this entity available from a knowledge base. In this section, we introduce a query understanding approach that builds upon this idea. In particular, we propose a supervised entity-oriented query expansion approach called L2EE, an acronym for "Learning To Expand using Entities". The retrieval flow of L2EE is illustrated in Figure 1, covering both its off-line and on-line stages.

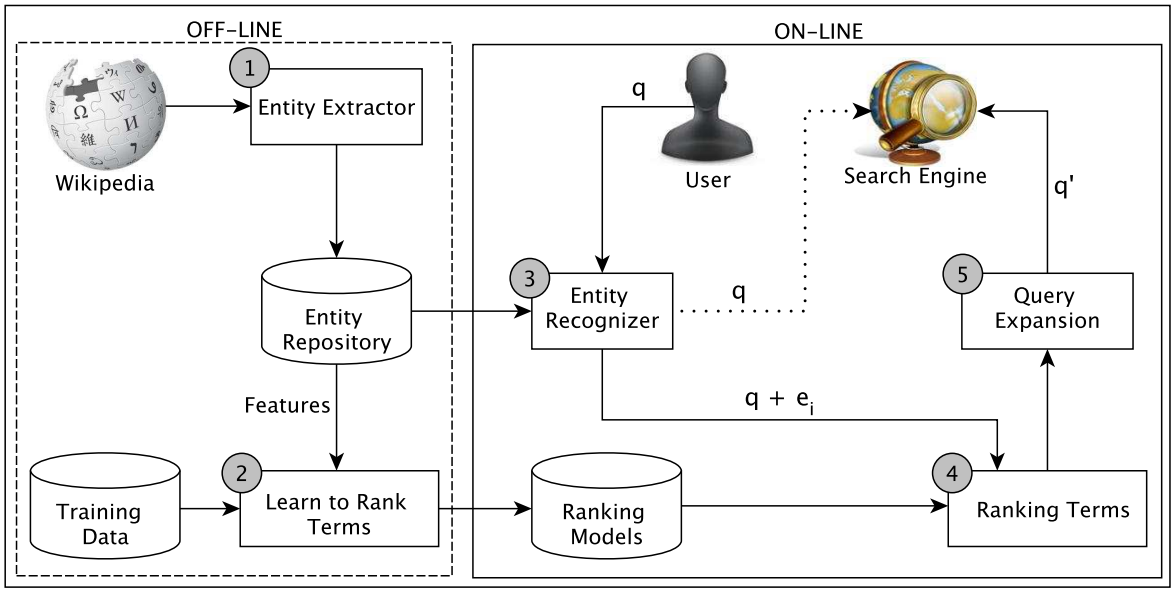

Figure 1: Off-line and on-line query processing with L2EE.

The off-line stage of L2EE is responsible for assembling an entity repository W by processing a knowledge base (step 1 in Figure 1). In addition, given suitable training data, this stage is also responsible for learning the ranking function that will be used for identifying effective expansion terms related to entities in the repository (step 2). In the on-line stage, given a user's query q composed of a sequence of l terms $\{t_1\ t_2\ \cdots\ t_l\}$, L2EE generates a new expanded query q'. To this end, our approach attempts to recognize in the query q a named entity e_i from the repository W (step 3 in Figure 1). If no such entity is found, q is not expanded. Otherwise, candidate terms related to e_i, as recorded

in the repository W, are ranked with respect to their predicted effectiveness given the query q (step 4), using the ranking function learned off-line in step 2. Lastly, the top k ranked terms according to the learned function are appended to q in order to produce the expanded query q' (step 5), which will be then used by the search engine to retrieve the final ranking of results to be presented to the user.

Most of the work of L2EE is done at the off-line stage, and the computational cost of the on-line stage is negligible. Similarly to standard PRF approaches, the query latency of our approach is primarily affected by the number of terms considered for expansion. However, differently from standard PRF approaches, which need to process both the original query, in order to select terms for expansion, and the modified one, L2EE only process the modified query. Thus, the computational overhead of L2EE in query time is lower than standard PRF approaches.

In the remainder of this section, Section 3.1 details our adopted representation for the entities stored in the repository W. Section 3.2 describes our approach for recognizing a named entity in a user's query. Lastly, Section 3.3 formalizes our approach for learning to rank effective expansion terms related to the recognized entity in the query.

3.1 Entity Representation

Our supervised query expansion approach builds an entity repository W using Wikipedia, a free on-line encyclopedia that enables collaborative publication and dissemination of ideas and concepts. Due to its popularity, coverage, accessibility, multilingual support, and high quality content, Wikipedia has rapidly turned into an important lexical semantic resource on the Web. Indeed, it has shown a strong potential to attenuate knowledge acquisition bottlenecks and coverage problems found in current lexical semantic resources (Zesch et al., 2007).

Wikipedia comprises semi-structured documents, known as articles, with each article describing a named entity, such as a person, organization, or location. The textual content of Wikipedia is available for download within periodically released database dumps.[1] We designed a special-purpose parser that processes Wikipedia articles in order to populate W. This process includes lower case conversion, stopword removal, and stemming using Porter's stemmer (Porter, 1980). In addition, during indexing, we discard numeric terms, non-English terms, and terms with special characters.

Consider an entity repository W comprising a set of p entities $E = \{e_1, e_2, \cdots, e_p\}$. Each entity $e_i \in E$ is represented as a tuple $e_i = \langle F_i, A_i, c_i \rangle$, where F_i and A_i are the sets of fields and aliases of e_i, respectively, and c_i is the class to which this entity belongs. In particular, each field in F_i comprises textual content from a specific region in the article that describes the entity e_i. Table 1 presents the article fields considered by our query expansion approach. As an illustrative example, Figure 2 shows an excerpt of the Wikipedia article describing the entity *Barack Obama*, highlighting the fields *title*, *summary*, *category*, and *infobox*.

[1] `http://dumps.wikimedia.org/`

Table 1: Article fields considered by L2EE.

Field	Description
title	the title of the article (unique identifier)
summary	the article's main concepts
infobox	special tabular structure that presents a set of attribute-value pairs describing different aspects of the article
category	categories used by Wikipedia users to classify the article
link	anchor-text from other articles in Wikipedia with a hyperlink to the article
appendix	external sources of information about the article, such as references and further reading
content	textual content of the remaining fields

Figure 2: The Wikipedia article describing the entity *Barack Obama*.

An entity in Wikipedia may be referred to by multiple names. For instance, as illustrated in Figure 2, in Wikipedia, the names "Obama", "Barack Obama", and "44th President of the United States"—to name but a few—are all alternative entry points for the single article representing the entity *Barack Obama*. In order to improve the recognition of named entities in web search queries, we use these multiple names as the set of aliases A_i of the entity e_i.

Lastly, entities in Wikipedia can be classified in different manners. For instance, Wikipedia contributors may assign each article a category, such as those described at the bottom of Figure 2 for the article describing the entity *Barack Obama*. While these categories are leveraged by our query expansion approach as a textual field, as described in Table 1, they are less suitable for assigning each entity a unique class. As we will see in Section 3.3.2, such a unique class is useful for identifying informative terms, i.e., terms that are useful descriptors of a particular entity as opposed to those that describe several entities of the same class. To this end, we exploit infobox templates as a means to identify the single most representative class of an article. In particular, infobox templates are predefined sets of attributes that can be used to build an infobox. For instance, in Figure 2,

the pre-defined attributes *Vice president* and *Preceded by* of the infobox template *president* are used to build the infobox of the entity *Barack Obama*. Accordingly, we choose "president" as the unique class c_i for this entity.

3.2 Entity Resolution

At querying time, we must be able to map the occurrence of a named entity in the query to the corresponding entity in the repository W. However, as discussed in Section 3.1, an entity may be represented by multiple names. Conversely, a single name can be ambiguous, in which case it can refer to multiple entities. As an example of the latter case, consider the entities *Barack Obama* and *Obama, Fukui* (a city in Japan), which can be both referred to by the string "obama" in a query. To overcome these problems, we introduce an entity resolution step ahead of the query expansion.

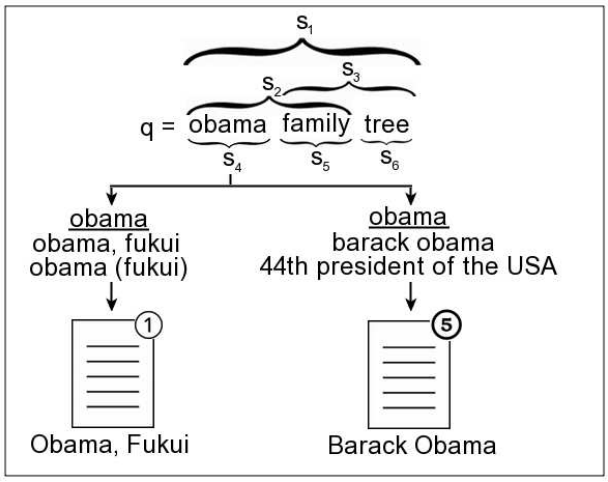

Figure 3: Resolution of named entities in a query.

Given a query q with length l, we build a set $S_q = \{s_1, s_2, \cdots, s_z\}$ of all substrings of consecutive terms from q that have a length b, for all $1 \leq b \leq l$. For instance, in Figure 3, we have six substrings of consecutive terms extracted from the query *"obama family tree"*: s_1 = *"obama family tree"*, s_2 = *"obama family"*, s_3 = *"family tree"*, s_4 = *"obama"*, s_5 = *"family"*, and s_6 = *"tree"*. The query substrings in S_q are then matched against the aliases of all entities in W. If there is no entity $e_i \in E$ such that $|S_q \cap A_i| > 0$, the query is not expanded, as discussed in Section 3. If exactly one entity e_i satisfies this condition, this entity is selected and the resolution process is complete. For instance, the alias *"obama"* for the entity *Barack Obama* exactly matches the string *"obama"* in S_q. Finally, if there are multiple entities whose aliases match a substring in the query, a disambiguation process is triggered. For instance, the string *"obama"* in S_q also matches an alias of the entity *Obama, Fukui*. In this case, we select the entity with the highest indegree in the Wikipedia graph. For instance, in Figure 3, the article *"Barack Obama"* is five times more cited than the article *"Obama, Fukui"*, so the entity represented by the article *"Barack Obama"*

is selected. This simple mechanism based on popularity is an effective solution for the experiments we conducted. In a real life scenario, the user intent should not be aligned with what is well cited in Wikipedia. In this case, other mechanisms should be more effective and can be easily used by our approach.

Since we can identify multiple entities in a single query, we choose the most representative one as the basis for the subsequent query expansion, so as to avoid drifting away from the user's information need. In particular, given a set of entities $\hat{E} \subseteq E$ identified in a query q, we select the most representative entity $e \in \hat{E}$ as the one with the longest title on Wikipedia. In case of a tie, the entity with the highest estimated quality is chosen. Our premise to select the entity with the longest title on Wikipedia is that longer matches tend to be more specific and hence less prone to ambiguity (Pôssas et al., 2005), which could in turn incur topic drift, a classic side-effect of query expansion.

To estimate the quality of Wikipedia articles the obvious choice would be the own Wikipedia quality estimators derived from a manual revision process. However, the manual revision process in Wikipedia is becoming infeasible and has been recently criticized by the scientific community (Hu et al., 2007). The large number of articles, the wide range of subject topics, the evolving content in the articles, the varying contributor background, and abuses contribute to this. Thus, we decided to adopt an automatic machine learning approach to determine the quality of the article that describes an entity in Wikipedia, based upon textual features extracted from this article (Dalip et al., 2009).[2] Specifically, we apply a regression method using the learning algorithm ϵ-Support Vector Regression (SVR) (Vapnik, 1995) to find the best combination of textual features to predict the quality value for any Wikipedia article. Then, we use the predicted quality value of articles to break tied entities. As an example, in Figure 3, five distinct substrings of the query *"obama family tree"* can be mapped to entities in the repository W: *"obama family"*, *"family tree"*, *"obama"*, *"family"*, and *"tree"*. The first two substrings are tied with the longest title. Between them, *"obama family"* has the greater quality estimator, and is hence selected as the most representative entity in the query.

3.3 Ranking Entity Terms

In order to rank effective expansion terms related to the most representative entity identified in the user's query, we introduce a learning to rank approach. In the remainder of this section, we formalize this approach and describe the features that are used to instantiate it in our experiments.

3.3.1 Learning a Ranking Model

In order to tackle query expansion as a ranking problem, we follow the general framework of discriminative learning (Liu, 2009). In particular, our goal is to learn an optimal hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$, mapping the input space $\mathcal{X}$ to the output space $\mathcal{Y}$. To this end, a plethora

[2] Note that the quality of each article is estimated off-line, during the construction of the entity repository W.

of machine learning algorithms could be deployed. In this work, we adopt a pairwise learning to rank formulation, which reduces the ranking problem to a binary classification problem (Liu, 2009), namely, that of choosing the most effective expansion term from a pair of candidate terms. We used the linear RankSVM algorithm (Joachims, 2006) to implement this pairwise learning to rank formulation. As a result, our input space $\mathcal{X}$ comprises pairs of learning instances of the form (x_u, x_v), where each instance x conveys a vector representation $\Phi(q, t)$ of a candidate expansion term t for a given query q, according to the feature extractor Φ. The various features considered in this work are described in Section 3.3.2. In order to guide the learning process towards identifying *effective* expansion terms, we consider an output space $\mathcal{Y}$ comprising binary performance-oriented labels y_{uv}, defined as:

$$y_{uv} = \begin{cases} -1 & \text{if } \delta(t_u) < \delta(t_v), \\ +1 & \text{otherwise,} \end{cases} \tag{1}$$

where $\delta(t)$ measures the gain attained by appending the candidate expansion term t to the query q, according to:

$$\delta(t) = \frac{\epsilon(q \cup \{t\}) - \epsilon(q)}{\epsilon(q)}, \tag{2}$$

where ϵ can be any standard information retrieval evaluation metric, such as mean average precision (MAP) (Baeza-Yates and Ribeiro-Neto, 2011).

Given m training queries $\{q_i\}_{i=1}^{m}$, their associated pairs of candidate expansion terms $(x_u^{(i)}, x_v^{(i)})$, and the label $y_{uv}^{(i)}$ associated with each pair, our goal is to learn a hypothesis h that minimizes the empirical risk $R(h)$, according to:

$$R(h) = \frac{1}{m} \sum_{i=1}^{m} \Delta(y_{uv}^{(i)}, h(x_u^{(i)}, x_v^{(i)})), \tag{3}$$

where the loss function Δ quantifies the penalty incurred by predicting an output $h(x_u^{(i)}, x_v^{(i)})$ when the correct output is $y_{uv}^{(i)}$. In our experiments, the loss function Δ is defined in terms of the total number of swapped pairs $(x_u^{(i)}, x_v^{(i)})$.

Lastly, as candidate hypotheses, we consider linear functions $h(x_u^{(i)}, x_v^{(i)}) = w^T(x_u^{(i)} - x_v^{(i)})$, parametrized by a weight vector w. In particular, our goal is to find a vector w that minimizes the empirical risk in Equation (3). Given a learned weight vector w, we can predict the effectiveness of all candidate expansion terms associated with the most representative entity in an unseen query q. The top k among these terms are then appended to q, with their predicted scores serving as their weight in the expanded query q'.

3.3.2 Ranking Features

To represent candidate terms in a suitable form for our learning to rank approach, we employ a total of five statistical descriptors as term features: Dice's coefficient (DC),

mutual information (MI), term frequency (TF), term spread (TS), and term proximity (TP). Our first two features, DC and MI, are taxonomic features that take into account not only the relevance of a term to an entity, but also to the class to which the entity belongs. These features have been shown to be effective descriptors of terms in Wikipedia articles in our previous analytical study (Brandão et al., 2011). In particular, let t be a candidate term extracted from the article representing the entity e_i, identified from the user's query q. The Dice's coefficient (DC) feature can be defined according to:

$$\mathrm{DC}(t) = 2 \times \frac{|E_t \cap E_i|}{|E_t| + |E_i|}, \tag{4}$$

where E_t is the set of entities that contain the term t and E_i is the set of entities that belong to the same class c_i as the entity e_i. This feature measures the similarity between the sets E_t and E_i. Intuitively, the higher this similarity, the more related to entities in c_i is the term t. Similarly, we can define the mutual information (MI) feature based upon the sets E_t and E_i, according to:

$$\mathrm{MI}(t) = \begin{cases} |E_t \cap E_i| \times \log \frac{|E_t \cap E_i|}{|E_t| \times |E_i|} & \text{if } |E_t \cap E_i| > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

which measures the mutual dependence—i.e., the amount of shared information—between E_t and E_i. As with the DC feature, the higher this measure, the more t is related to c_i.

Our next two features, TF and TS, are statistical measures that take into account the distribution of terms in different fields. In order to formalize these features, let $\mathit{freq}(t, f_j)$ be the frequency of the term t in the field $f_j \in F_i$ of the entity e_i. The term frequency (TF) of term t can be defined as:

$$\mathrm{TF}(t) = \sum_{j=1}^{|F_i|} \mathit{freq}(t, f_j), \tag{6}$$

where $|F_i|$ denotes the total number of available fields for the entity e_i, as described in Table 1. Different from TF, the term spread (TS) feature measures the spread of a term across multiple fields, i.e., the number of different fields in which a term occurs, according to:

$$\mathrm{TS}(t) = \sum_{j=1}^{|F_i|} \mathbf{1}_{f_j}(t), \tag{7}$$

where $\mathbf{1}_{f_j}(t)$ is the indicator function, equal to one if $t \in f_j$, or zero otherwise. Intuitively, the higher the values of the TF and TS features, the more t is related to e_i.

Lastly, we devise a feature to account for the proximity between a candidate term t and the original query terms. In particular, we define the term proximity (TP) feature as:

$$\mathrm{TP}(t) = \sum_{j=1}^{l} \sum_{w=1}^{m} \log \frac{\mathit{freq}(\langle t, t_j \rangle, w)}{2^{w-1}}, \tag{8}$$

where t_j is the j-th term of the query q, l is the total length of q, and $freq(\langle t, t_j \rangle, w)$ is the total number of occurrences of the (unordered) pair $\langle t, t_j \rangle$ within windows of size w sentences across the concatenation of all fields of the entity e_i. Note that $w = 1$ denotes an occurrence of t and t_j within the same sentence. We consider $m = 5$, since preliminary experiments show that for $m > 5$ the value of the feature did not changed significantly.

4 Experimental Setup

In order to validate our entity-oriented query expansion approach, we contrast it to state-of-the-art PRF and ePRF approaches across multiple web test collections. In particular, we aim to answer the following research questions:

1. How effective is our approach for query expansion?
2. How does our approach perform for entity queries?
3. How does our approach perform for difficult queries?
4. How effective is our approach for non-Wikipedia pages[3]?
5. Which features are effective for query expansion?

We address each of these questions in turn in Sections 5.1 through 5.5. In the remainder of this section, we describe the experimental setup that supports these investigations.

4.1 Test Collections

To assess the effectiveness of our learning to rank approach for entity-oriented query expansion, we use three standard TREC web test collections: WT10g (Hawking and Craswell, 2001), GOV2 (Büttcher et al., 2006), and the category B portion of ClueWeb09 (Clarke et al., 2009), or simply CW09B. For each test collection, we generate queries using all the words from the title field of the corresponding search tracks at TREC. Table 2 summarizes salient statistics of these test collections. Besides the TREC tracks that used each collection, we describe the number of documents and queries, as well as the average query length, the number of entity queries (i.e., queries with at least one identified entity), and the number of difficult queries (see Section 5.3 for a precise definition). As our basic retrieval system, we use Indri (Strohman et al., 2005), which provides support for indexing and querying large corpora. The preprocessing of documents and queries included stemming with Porter's stemmer (Porter, 1980) and the removal of standard English stopwords.

As a knowledge base, we used the English Wikipedia. In particular, we built our entity repository W based upon a Wikipedia dump from June 1st, 2012. From this dump, we extracted a total of 2,069,704 unique entities, referred to by a total of 5,521,403 aliases. On average, this amounts to around 2.67 alternative aliases per entity.

[3]We consider non-Wikipedia pages an instance of standard TREC web test collections without Wikipedia pages.

Table 2: Overview of the considered standard TREC web test collections.

	WT10g	GOV2	CW09B
TREC track	Web 00/01	Terabyte 04/05/06	Web 09/10
# documents	1,692,096	25,205,179	50,220,423
# queries	100	149	98
avg. query length	4.18	3.10	2.06
# entity queries	98	148	94
# difficult queries	7	29	11

4.2 Retrieval Baselines

L2EE can be implemented over any standard retrieval model, such as BM25 and language models. In our experiments we implemented it, as well as two state-of-the-art query expansion baselines, on top of the initial ranking produced by the Kullback-Leibler (KL) retrieval model.

In particular, for each input query, we retrieve 1,000 documents using the KL retrieval model with Dirichlet smoothing (Zhai and Lafferty, 2001a). This formulation has been shown to be effective across many retrieval scenarios, and represents the current state-of-the-art in language modeling (Zhai, 2008). In our experiments, the smoothing parameter μ of the Dirichlet prior was empirically set to 2,500, following the training procedure described in Section 4.4.

On top of the initial baseline ranking produced by the KL retrieval model, we compare our learning to rank approach to two state-of-the-art query expansion baselines. Our first query expansion baseline is an implementation of Lavrenko's relevance models (RM1) (Lavrenko and Croft, 2001) provided by Indri, which instantiates the classical PRF approach to query expansion. In addition to RM1, we compare our query expansion approach to the ePRF approach of (Xu et al., 2009), henceforth referred to as QD. As discussed in Section 2, this approach represents the current state-of-the-art in entity-oriented query expansion, and is hence our strongest baseline. In our experiments, RM1 is used to select the top $k = 50$ terms from the top $n = 10$ documents retrieved by the KL retrieval model. As for the QD and L2EE query expansion approaches, both are deployed to select the top $k = 50$ terms related to an entity identified in the query.

In preliminary experiments, we vary k from 10 to 100 and, as expected, we observe that search results performance increases when k increases, i.e., greater k leads to better search results. However, greater k values imply worse time performance. In a search system, from the point of view of time performance, shorter queries are preferable because they take less time to process, i.e., lower k leads to fast query processing. In our experiments we set $k = 50$ to balance time and search result performance. This setting provided the best overall performance during training for the QD and L2EE query expansion approaches, since for $k > 50$ the search result performance gains are negligible and the loss of time performance are significant.

4.3 Oracle

Query expansion approaches usually lead to global improvements compared to a baseline ranking that does not perform any expansion. Nevertheless, query expansion may also be harmful to some queries. This is particularly the case for difficult queries, i.e., queries with a poor first-pass retrieval performance, which end up returning irrelevant documents to be used as feedback (Amati et al., 2004). In order to analyze the performance of our query expansion approach for difficult queries, as well as to assess the room for improvements, we introduce an oracle expansion mechanism, which knows exactly whether or not to expand each individual query.

Given a query q with corresponding relevance assessments, our oracle mechanism begins by selecting $n = 10$ documents that are relevant to this query.[4] Each unique term t_i extracted from the feedback documents is then assessed as to the extent to which it improves the retrieval performance—in terms of MAP—of the query q, when appended to this query. After discarding non-improving terms, the remaining terms are appended to the query q. The improvement in MAP observed for each term t_i in the previous step is used as the weight w_i of the term in the new expanded query.

Our oracle expansion mechanism does not determine the best possible combination of terms and weights to expand queries. It approximates an optimal selection of terms greedily, by selecting one term at a time. While this simplified approach is indeed suboptimal, it provides a reasonably strong lower-bound of the optimal performance. More importantly for the feasibility of our investigations, it avoids the combinatorial selection of the single best set of terms, which may become prohibitive even with a few candidate terms.

4.4 Training and Evaluation Procedures

All retrieval approaches investigated in this paper require some form of supervised training. In order to ensure a fair assessment of these approaches, we perform a 5-fold cross validation for each of the test collections described in Section 4.1. In particular, for each cross-validation round, we train on four folds and test on the remaining fold. Accordingly, we report our results as an average across the test queries in each round, hence ensuring a complete separation between training and test queries at all times.

Regarding the evaluation of the investigated approaches, we report retrieval effectiveness in terms of three evaluation metrics: mean average precision (MAP), normalized discounted cumulative gain (nDCG), and precision at 10 (P@10). In particular, both MAP and P@10 are based on binary assessments of relevance, whereas nDCG can leverage graded relevance assessments. While MAP has been traditionally used for retrieval evaluation (Baeza-Yates and Ribeiro-Neto, 2011), both nDCG and P@10 are typical targets for web search evaluation, by focusing on the retrieval performance at early ranks (Jansen et al., 2000).

[4]For queries with more than $n = 10$ relevant documents, we break ties randomly.

5 Experimental Results

In this section, we describe the experiments we have carried out to evaluate our proposed query expansion approach. In particular, we address the five research questions stated in Section 4, by contrasting the effectiveness of our L2EE approach to state-of-the-art PRF and ePRF baselines, described in Section 4.2, as well as to an oracle expansion mechanism, described in Section 4.3. Significance is verified with a two-tailed paired t-test (Jain, 1991), with the symbol ▲ (▼) denoting a significant increase (decrease) at the $p < 0.05$ level, and the symbol • denoting no significant difference.

5.1 Query Expansion Effectiveness

In this section, we address our first research question, by assessing the effectiveness of our entity-oriented query expansion approach. To this end, Table 3 shows the retrieval performance of L2EE compared to KL, which performs no expansion, RM1, and QD. In order to provide a fair comparison to ePRF approaches, both QD and L2EE fall back to a standard PRF approach for queries without named entities.[5] For all query expansion approaches (i.e., RM1, QD, and L2EE), percentage improvement figures compared to the KL baseline are also shown. In addition, a first instance of the aforementioned significance symbols denotes whether these improvements are statistically significant. For the ePRF approaches (i.e., QD and L2EE), a second such symbol denotes significance with respect to RM1. Finally, for L2EE, a third symbol denotes significance compared to QD. The best value in each row is highlighted in bold.

Table 3: Retrieval performance (all queries).

WT10g				
	KL	+RM1	+QD	+L2EE
MAP	0.1953	0.2030 (+3.94%) $^{\bullet}$	0.2131 (+9.11%) $^{\blacktriangle\bullet}$	**0.2628 (+34.56%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2730	0.2840 (+4.03%) $^{\bullet}$	0.3130 (+14.65%) $^{\blacktriangle\bullet}$	**0.3740 (+37.00%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.4686	0.4693 (+0.14%) $^{\bullet}$	0.4924 (+5.07%) $^{\bullet\bullet}$	**0.5311 (+13.33%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
GOV2				
	KL	+RM1	+QD	+L2EE
MAP	0.2947	0.3185 (+8.07%) $^{\bullet}$	0.3240 (+9.94%) $^{\blacktriangle\bullet}$	**0.3661 (+24.22%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.5416	0.5624 (+3.84%) $^{\bullet}$	0.6047 (+11.65%) $^{\blacktriangle\bullet}$	**0.6866 (+26.77%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.5860	0.6014 (+2.62%) $^{\bullet}$	0.6132 (+4.64%) $^{\blacktriangle\bullet}$	**0.6418 (+9.52%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
CW09B				
	KL	+RM1	+QD	+L2EE
MAP	0.1416	0.1429 (+0.91%) $^{\bullet}$	0.1648 (+16.38%) $^{\blacktriangle\bullet}$	**0.1820 (+28.53%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2408	0.2500 (+3.82%) $^{\bullet}$	0.3082 (+28.00%) $^{\blacktriangle\blacktriangle}$	**0.3663 (+52.12%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.3720	0.3605 (-3.09%) $^{\bullet}$	0.3951 (+6.20%) $^{\bullet\blacktriangle}$	**0.4132 (+11.07%)** $^{\blacktriangle\blacktriangle\bullet}$

From Table 3, we first observe that L2EE significantly improves upon all baselines, and

[5] A complementary evaluation of ePRF approaches focusing on queries with named entities is conducted in Section 5.2.

across all three test collections. In particular, compared to KL, the gains are up to 34.56% in terms of MAP, 52.12% in terms of P@10, and 13.33% in terms of nDCG. Compared to the standard PRF approach implemented by RM1, L2EE brings significant improvements of up to 29.46% in terms of MAP, 46.52% in terms of P@10, and 14.62% in terms of nDCG. Lastly, L2EE also significantly outperforms QD, with gains of up to 23.32% in terms of MAP, 19.49% in terms of P@10, and 7.86% in terms of nDCG. Recalling our first research question, these observations attest the effectiveness of our learning to rank approach for entity-oriented query expansion.

Note that nDCG gain is smaller in all test collections. This occurs because different evaluation metrics have different sensitivity to ranking swaps. MAP and P@10 are based on binary relevance judgments and this might induce a different behavior compared to the nDCG scores, which are based on graded judgments (Radlinski and Craswell, 2010). Note also the relatively large gain in P@10 for CW09. Given its substantially larger size compared to the GOV2 and WT10g collections, as well as the more ambiguous nature of its associated query sets, the CW09 collection represents an arguably more challenging retrieval environment, in which the vocabulary gap between queries and documents is more pronounced. The relatively larger gains observed for CW09 suggest that query expansion can play a more noticeable role in this case, as a technique essentially aimed at improving the representation of the users' queries.

5.2 Effectiveness for Entity Queries

In this section, we address our second research question, by evaluating all ePRF approaches for queries with named entities. In particular, Table 4 shows the retrieval performance of L2EE compared to the KL and QD baselines, considering only queries with entities. As a reference performance, we also include the oracle expansion mechanism, described in Section 4.3. For all ePRF approaches, a first significance symbol denotes a statistically significant difference (or lack thereof) compared to the KL baseline. For L2EE and the oracle, a second symbol denotes significance with respect to QD. Finally, a third symbol for the oracle denotes significance compared to L2EE. The best value between baselines and L2EE is highlighted in bold.

From Table 4, we observe that L2EE significantly improves upon the state-of-the-art QD baseline, with gains of up to 35.26% in terms of MAP, 37.95% in terms of P@10, and 12.90% in terms of nDCG. Recalling our second research question, these observations attest the effectiveness of our supervised approach for exploiting entity-related information for query expansion. Indeed, the improvements compared to QD are larger than those observed in Section 5.1, when queries that did not contain entities were also considered. On the other hand, compared to the performance of the oracle, we observe that there is still a considerable room for further improvements. For instance, considering the CW09B collection, the oracle is ahead by 20.89% in terms of MAP, 19.05% in terms of P@10, and 9.59% in terms of nDCG. In Section 6, we propose further directions to close this gap.

The reason why L2EE outperforms the state-of-the-art QD baseline lies in the fact L2EE is able to select terms that individually contribute more to the effectiveness of search. As

Table 4: Retrieval performance (entity queries).

WT10g				
	KL	+QD	+L2EE	Oracle
MAP	0.1953	0.2320 (+18.79%) ▲	**0.3138 (+60.67%)** ▲▲	0.4295 ▲▲▲
P@10	0.2730	0.3407 (+24.80%) ▲	**0.4444 (+62.78%)** ▲▲	0.5815 ▲▲▲
nDCG	0.4686	0.5153 (+9.96%) ▲	**0.5818 (+24.15%)** ▲▲	0.6835 ▲▲▲
GOV2				
	KL	+QD	+L2EE	Oracle
MAP	0.2947	0.3131 (+6.24%) ▲	**0.3849 (+30.60%)** ▲▲	0.4362 ▲▲▲
P@10	0.5416	0.5839 (+7.81%) ▲	**0.7322 (+35.19%)** ▲▲	0.8218 ▲▲▲
nDCG	0.5860	0.6070 (+3.58%) ▲	**0.6580 (+12.29%)** ▲▲	0.7062 ▲▲▲
CW09B				
	KL	+QD	+L2EE	Oracle
MAP	0.1416	0.2050 (+44.77%) ▲	**0.2518 (+77.82%)** ▲▲	0.3044 ▲▲▲
P@10	0.2408	0.4567 (+89.66%) ▲	**0.6300 (+161.63%)** ▲▲	0.7500 ▲▲▲
nDCG	0.3720	0.4282 (+15.11%) ▲	**0.4778 (+28.44%)** ▲▲	0.5236 ▲▲▲

an example, consider the query "poker tournaments". Table 5 presents the top-5 expansion terms selected by QD and L2EE considering the greater individual contribution, in terms of MAP, and the weights used for query expansion. From Table 5, we observe that L2EE selects terms with greater individual contribution, in terms of MAP, to the effectiveness of search. Furthermore, our approach effectively weigh the expansion terms.

Table 5: Top-5 expansion terms selected by QD and L2EE considering the greater individual contribution, in terms of MAP, and the weights used in expansion for the query "poker tournaments".

QD			L2EE		
Term	MAP	Weight	Term	MAP	Weight
prize	0.0553	-	world	0.0578	0.0668
limit	0.0347	-	season	0.0452	0.0298
player	0.0287	-	player	0.0287	0.0100
rebui	0.0106	-	tour	0.0287	0.0085
tabl	0.0056	-	tabl	0.0056	0.0052

5.3 Effectiveness for Difficult Queries

As discussed in Section 4.3, query expansion can harm the retrieval performance for difficult queries. In this section, we address our third research question, by performing a breakdown analysis of our approach according to query difficulty. To this end, we consider as difficult queries those that cannot be improved by more than 10% (in terms of MAP) using our oracle expansion mechanism. All other queries are regarded as easy. As a result of this simple quantitative criterion, we have 7 difficult queries for WT10g (7.00%), 29 difficult queries for GOV2 (19.46%), and 11 difficult queries for CW09B (11.22%). Tables 6 and 7 show the retrieval performance of our approach compared to KL, RM1, and QD, considering

difficult and easy queries, respectively. For both tables, significance symbols are defined as in Table 3.

Table 6: Retrieval performance (difficult queries).

WT10g				
	KL	+RM1	+QD	+L2EE
MAP	0.4244	0.4113 (-3.08%) $^{\bullet}$	**0.4367 (+2.89%)** $^{\bullet\bullet}$	0.4048 (-4.61%) $^{\bullet\bullet\blacktriangledown}$
P@10	0.5286	0.5000 (-5.41%) $^{\bullet}$	**0.5571 (+5.39%)** $^{\bullet\blacktriangle}$	0.5429 (+2.70%) $^{\bullet\blacktriangle\bullet}$
nDCG	0.7012	0.6914 (-1.39%) $^{\bullet}$	**0.7038 (+0.37%)** $^{\bullet\bullet}$	0.6601 (-5.86%) $^{\bullet\bullet\blacktriangledown}$
GOV2				
	KL	+RM1	+QD	+L2EE
MAP	0.4569	**0.4686 (+2.56%)** $^{\bullet}$	0.4505 (-1.40%) $^{\bullet\bullet}$	0.4476 (-2.03%) $^{\bullet\blacktriangledown\bullet}$
P@10	0.7793	**0.8034 (+3.09%)** $^{\bullet}$	0.7690 (-1.32%) $^{\bullet\bullet}$	0.7862 (+0.88%) $^{\bullet\bullet\bullet}$
nDCG	0.7120	**0.7179 (+0.82%)** $^{\bullet}$	0.7076 (-0.61%) $^{\bullet\bullet}$	0.7070 (-0.70%) $^{\bullet\blacktriangledown\bullet}$
CW09B				
	KL	+RM1	+QD	+L2EE
MAP	0.3221	0.3222 (+0.03%) $^{\bullet}$	0.3192 (-0.90%) $^{\bullet\bullet}$	**0.3255 (+1.05%)** $^{\bullet\bullet\bullet}$
P@10	0.5617	0.5375 (-4.31%) $^{\bullet}$	0.5590 (-0.48%) $^{\bullet\bullet}$	**0.5722 (+1.87%)** $^{\bullet\bullet\bullet}$
nDCG	0.6069	0.6063 (-0.09%) $^{\bullet}$	**0.6104 (+0.57%)** $^{\bullet\bullet}$	0.6103 (+0.56%) $^{\bullet\bullet\bullet}$

Table 7: Retrieval performance (easy queries).

WT10g				
	KL	+RM1	+QD	+L2EE
MAP	0.1780	0.1874 (+5.28%) $^{\bullet}$	0.1963 (+10.28%) $^{\bullet\bullet}$	**0.2521 (+41.62%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2538	0.2677 (+8.00%) $^{\bullet}$	0.2946 (+16.00%) $^{\bullet\bullet}$	**0.3613 (+44.00%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.4511	0.4526 (+0.33%) $^{\bullet}$	0.4765 (+5.63%) $^{\bullet\bullet}$	**0.5214 (+15.58%)** $^{\blacktriangle\blacktriangle\bullet}$
GOV2				
	KL	+RM1	+QD	+L2EE
MAP	0.2554	0.2822 (+10.49%) $^{\bullet}$	0.2934 (+14.87%) $^{\bullet\bullet}$	**0.3464 (+35.63%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.4842	0.5042 (+4.16%) $^{\bullet}$	0.5650 (+18.75%) $^{\bullet\bullet}$	**0.6625 (+37.50%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.5543	0.5733 (+3.42%) $^{\bullet}$	0.5904 (+6.51%) $^{\bullet\bullet}$	**0.6260 (+12.93%)** $^{\blacktriangle\blacktriangle\bullet}$
CW09B				
	KL	+RM1	+QD	+L2EE
MAP	0.1167	0.1181 (+1.20%) $^{\bullet}$	0.1435 (+22.96%) $^{\bullet\bullet}$	**0.1622 (+38.98%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.1966	0.2103 (+5.00%) $^{\bullet}$	0.2736 (+35.00%) $^{\blacktriangle\blacktriangle}$	**0.3379 (+70.00%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.3396	0.3265 (-3.85%) $^{\bullet}$	0.3654 (+7.59%) $^{\bullet\bullet}$	**0.3860 (+13.66%)** $^{\bullet\blacktriangle\bullet}$

From Table 6, we observe that, although the results vary across the three considered test collections, none of the deployed query expansion approaches can significantly outperform the KL baseline. On the other hand, eventual performance drops are not significant either. For easy queries, as shown in Table 7, L2EE outperforms the other baselines in all cases, often significantly, with gains of up to 13.03% in terms of MAP, 15.78% in terms of P@10, and 5.63% in terms of nDCG over the second most effective approach for each of these metrics. Recalling our third research question, the observations in Tables 6 and 7 attest the

robustness of our approach in light of difficult queries, as well as its superior performance for easy queries, which comprise the majority of the query sets considered in our investigation.

5.4 Finding Non-Wikipedia Pages

The previous sections have demonstrated the effectiveness of our approach at exploiting evidence from Wikipedia in order to expand queries with named entities. A natural question that arises in this scenario is whether the observed effectiveness is merely due to an improved ability to rank Wikipedia pages themselves. In order to assess the effectiveness of our approach at ranking non-Wikipedia pages, in this section, we address our fourth research question. To this end, Table 8 shows the retrieval performance of our approach compared to KL, RM1, and QD, considering a modified instance of the CW09B collection without Wikipedia pages, called CW09BNW. Once again, for all query expansion approaches, we present percentage improvements over the KL baseline, with significance symbols defined as in Table 3.

Table 8: Retrieval performance on CW09BNW.

CW09BNW				
	KL	+RM1	+QD	+L2EE
MAP	0.1055	0.1089 (+3.22%) $^{\bullet}$	0.1029 (-2.46%) $^{\bullet\blacktriangledown}$	**0.1210 (+14.69%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2122	0.2184 (+2.92%) $^{\bullet}$	0.1796 (-15.36%) $^{\blacktriangledown\blacktriangledown}$	**0.2562 (+20.73%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.3205	0.3175 (-0.94%) $^{\bullet}$	0.3216 (+0.34%) $^{\bullet\bullet}$	**0.3377 (+5.36%)** $^{\blacktriangle\blacktriangle\blacktriangle}$

Comparing results from Table 3 and Table 8, we can observe that Wikipedia pages play an important role in retrieval performance, even for KL baseline, which performs no expansion. When we consider Wikipedia pages, the retrieval performance in terms of MAP increases 34.21% for KL, 31.22% for RM1, 60.15% for QD and 50.14% for L2EE. Thus, the existence of Wikipedia pages in collection influences retrieval performance for all considered approaches. Moreover, it is critical for the QD baseline, which promotes the Wikipedia page related to the entity to the top of the ranking using it as the only feedback document. Additionally, L2EE is less dependent on the existence of Wikipedia pages than QD baseline. The gain in retrieval performance provided by our method compared to KL baseline is lower (14.69% against 28.53%), but is still significant.

From Table 8, we once again observe that L2EE significantly outperforms all baselines, with gains of 11.11% in terms of MAP, 17.31% in terms of P@10, and 5.01% in terms of nDCG over the second most effective approach for each metric. Recalling our fourth research question, these observations attest the effectiveness of our entity-oriented query expansion approach to expand queries even when no Wikipedia pages are considered in the search results.

5.5 Feature Effectiveness

Throughout Sections 5.1 to 5.4, we have demonstrated the effectiveness of our L2EE query expansion approach in different scenarios, in contrast to state-of-the-art PRF and ePRF

approaches. In order to further our understanding of the reasons behind such an effective performance, in this section, we address our fifth and last research question, by assessing the effectiveness of the features used by our learning to rank approach, as described in Section 3.3.2. To this end, Table 9 shows the retrieval performance of each of these features, when deployed in isolation in order to select the top k expansion terms for each query. As a baseline for this investigation, we include the performance of the KL retrieval model, which performs no expansion. In particular, a significance symbol denotes whether the performance of each of our considered features differs significantly from that of KL. The best value in each column is highlighted in bold.

Table 9: Feature retrieval performance

WT10g			
	MAP	P@10	nDCG
KL	0.1953	0.2730	0.4686
+DC	0.1882 (-3.63%) ▼	0.2870 (+5.13%) ▲	0.4574 (-2.39%) ▼
+MI	0.1875 (-3.99%) ▼	0.2900 (+6.23%) ▲	0.4560 (-2.69%) ▼
+TF	**0.2309 (+18.23%)** ▲	**0.3520 (+28.94%)** ▲	**0.5226 (+11.52%)** ▲
+TS	0.2235 (+14.44%) ▲	0.3380 (+23.81%) ▲	0.5062 (+8.02%) ▲
+TP	0.2082 (+6.60%) ▲	0.3050 (+11.72%) ▲	0.4789 (+2.20%) ▲
GOV2			
	MAP	P@10	nDCG
KL	0.2947	0.5416	0.5860
+DC	0.2898 (-1.66%) •	0.5490 (+1.37%) •	0.5780 (-1.36%) •
+MI	0.2780 (-5.67%) ▼	0.5221 (-3.60%) ▼	0.5623 (-4.04%) ▼
+TF	**0.3210 (+8.92%)** ▲	**0.6013 (+11.02%)** ▲	**0.6100 (+4.10%)** ▲
+TS	0.3098 (+5.12%) ▲	0.5725 (+5.70%) ▲	0.6015 (+2.64%) ▲
+TP	0.3043 (+3.26%) ▲	0.5846 (+7.94%) ▲	0.5936 (+1.30%) •
CW09B			
	MAP	P@10	nDCG
KL	0.1416	0.2408	0.3720
+DC	0.1595 (+12.64%) ▲	0.3714 (+54.24%) ▲	0.3863 (+3.84%) ▲
+MI	0.1578 (+11.44%) ▲	0.3643 (+51.29%) ▲	0.3863 (+3.84%) ▲
+TF	**0.1881 (+32.84%)** ▲	0.4092 (+69.93%) ▲	**0.4168 (+12.04%)** ▲
+TS	0.1822 (+28.67%) ▲	**0.4276 (+77.57%)** ▲	0.4139 (+11.26%) ▲
+TP	0.1691 (+19.42%) ▲	0.3816 (+58.47%) ▲	0.3965 (+6.59%) ▲

From Table 9, we first observe that both statistical features, term frequency (TF) and term spread (TS), as well as our term proximity feature (TP), perform effectively across all three considered collections, with significant improvements compared to the KL baseline in almost all settings (the only exception is for the TP feature on GOV2 in terms of nDCG). In addition, our taxonomic features, dice's coefficient (DC) and mutual information (MI), also show significant improvements on the larger CW09B corpus. Both DC and MI are high-precision features, as opposed to recall-oriented features, as indicated by their consistently positive improvements in terms of P@10 for all considered collections. While recall plays an important role for older ad-hoc test collections such as WT10g and GOV2, its importance is less pronounced for the larger CW09B collection, which comprises a considerable fraction

of navigational (and hence precision-oriented) queries. Contrasting these features to one another, TF and TS are generally the most effective, followed by TP and the taxonomic DC and MI features. Lastly, compared to the results in Table 3, none of these features outperform their combination within our learning to rank approach, further attesting to its effectiveness. Recalling our fifth research question, these observations demonstrate the suitability of our devised features as descriptors of effective expansion terms.

6 Conclusions

We presented a novel learning to rank approach for entity-oriented query expansion. Our supervised learning approach considers semantic evidence encoded in the content of Wikipedia article fields, and automatically labels training examples proportionally to their observed retrieval effectiveness. In contrast to existing supervised approaches in the literature, not only does our approach exploit named entities to select expansion terms, but it also weighs these terms proportionally to their predicted effectiveness.

We thoroughly evaluated our proposed approach using three standard TREC web test collections in contrast to state-of-the-art traditional as well as entity-oriented pseudo-relevance feedback approaches. The results of this evaluation attest the effectiveness of our learning to rank approach for entity-oriented query expansion, with gains of up to 23.32% in terms of MAP, 19.49% in terms of P@10, and 7.86% in terms of nDCG over the most effective baseline for each of these metrics. Moreover, by breaking down our analysis by query difficulty, we demonstrated the robustness of our approach when applied for queries with little room for improvement. In addition, we showed that the observed improvements hold even when no Wikipedia pages are considered in the search results. Lastly, we analysed the performance of each of our ranking features separately, showing that statistical and proximity features are particularly suitable for selecting effective expansion terms.

Contrasting the performance of our approach to that attained by an oracle mechanism, which knows exactly whether to expand each individual query, we showed that there is still room for further improvements. In particular, considering the challenges imposed by query difficulty, we plan to deploy a selective query expansion mechanism. In this vein, our preliminary results using the well-known clarity score metric for query performance prediction (Cronen-Townsend et al., 2002) are promising. Another plan is to assess the effectiveness of alternative learning to rank techniques as well as of additional features, particularly positional and proximity ones.

Acknowledgements

We thank the partial support given by the Brazilian National Institute of Science and Technology for the Web (grant MCT-CNPq 573871/2008-6), Project MinGroup (grant CNPq-CT-Amazônia 575553/2008-1) and authors' individual grants and scholarships from CNPq.

References

Amati, G., Carpineto, C., and Romano, G. (2004). Query difficulty, robustness and selective application of query expansion. In *Proceedings of the 26th European Conference on Information Retrieval*, pages 127–137.

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval*. Addison-Wesley Publishing Company, 2nd edition.

Beitzel, S. M., Jensen, E. C., Frieder, O., Grossman, D., Lewis, D. D., Chowdhury, A., and Kolcz, A. (2005). Automatic web query classification using labeled and unlabeled training data. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 581–582.

Bendersky, M., Metzler, D., and Croft, W. B. (2012). Effective query formulation with multiple information sources. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 443–452.

Brandão, W. C., da Silva, A. S., de Moura, E. S., and Ziviani, N. (2011). Exploiting entity semantics for query expansion. In *Proceedings of the IADIS International Conference WWW/INTERNET*, pages 365–372.

Büttcher, S., Clarke, C. L. A., and Soboroff, I. (2006). The TREC 2006 Terabyte track. In *Proceedings of 15th Text Retrieval Conference*.

Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 243–250.

Clarke, C. L. A., Craswell, N., and Soboroff, I. (2009). Overview of the TREC 2009 Web track. In *Proceedings of 18th Text Retrieval Conference*.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 299–306.

Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2002). Probabilistic query expansion using query logs. In *Proceedings of the 11th International Conference on World Wide Web*, pages 325–332.

Cutts, M. (2012). Spotlight keynote. In *Proceedings of the SES San Franscisco*.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2009). Automatic quality assessment of content created collaboratively by web communities: a case study of Wikipedia. In *Proceedings of the 9th Joint Conference on Digital Libraries*, pages 295–304.

Diaz, F. and Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 154–161.

Gabrilovich, E., Broder, A., Fontoura, M., Joshi, A., Josifovski, V., Riedel, L., and Zhang, T. (2009). Classifying search queries using the Web as a source of knowledge. *ACM Transactions on the Web*, 3(2):1–28.

Guo, J., Xu, G., Cheng, X., and Li, H. (2009). Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 267–274.

Hawking, D. and Craswell, N. (2001). Overview of the TREC 2001 Web track. In *Proceedings of 10th Text Retrieval Conference*.

He, B. and Ounis, I. (2007). Combining fields for query expansion and adaptive query expansion. *Information Processing and Management*, 43(5):1294–1307.

He, B. and Ounis, I. (2009). Finding good feedback documents. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 2011–2014.

Hu, M., Lim, E.-P., Sun, A., Lauw, H. W., and Vuong, B.-Q. (2007). Measuring article quality in wikipedia: models and evaluation. In *CIKM*, pages 243–252.

Jain, A., Cucerzan, S., and Azzam, S. (2007). Acronym-expansion recognition and ranking on the Web. In *Proceedings of the IEEE International Conference on Information Reuse and Integration*, pages 209–214.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, New York.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing and Management*, 36(2):207–227.

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM Conference on Knowledge Discovery and Data Mining*, pages 217–226.

Kotov, A. and Zhai, C. (2012). Tapping into knowledge base for concept feedback: Leveraging ConceptNet to improve search results for difficult queries. In *Proceedings of the 5th ACM International Conference on WebSearch and Data Mining*, pages 403–412.

Kumaran, G. and Allan, J. (2008). Effective and efficient user interaction for long queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 11–18.

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 120–127.

Lee, C.-J., Lin, Y.-C., Chen, R.-C., and Cheng, P.-J. (2009). Selecting effective terms for query formulation. In *Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, pages 168–180.

Li, H. (2010). Query understanding in web search: By large scale log data mining and statistical learning. In *Proceedings of the 2nd Workshop on NLP Challenges in the Information Explosion Era*, page 1.

Li, M., Zhu, M., Zhang, Y., and Zhou, M. (2006). Exploring distributional similarity based models for query spelling correction. In *Proceedings of the 21st International Conference on Computational Linguistics*, pages 1025–1032.

Li, Y., Luk, W. P. R., Ho, K. S. E., and Chung, F. L. K. (2007). Improving weak ad-hoc queries using Wikipedia as external corpus. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 797–798.

Lin, Y., Lin, H., Jin, S., and Ye, Z. (2011). Social annotation in query expansion: a machine learning approach. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405–414.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Milne, D. N., Witten, I. H., and Nichols, D. M. (2007). A knowledge-based search engine powered by wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 445–454.

Mitra, M., Singhal, A., and Buckley, C. (1998). Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 206–214.

Peng, F., Ahmed, N., Li, X., and Lu, Y. (2007). Context sensitive stemming for web search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 639–646.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Pôssas, B., Ziviani, N., Meira, W., and Ribeiro-Neto, B. (2005). Set-based vector model: An efficient approach for correlation-based ranking. *ACM Transactions on Information Systems*, 23(4):397–429.

Radlinski, F. and Craswell, N. (2010). Comparing the sensitivity of information retrieval metrics. In *SIGIR*, pages 667–674.

Risvik, K. M., Mikolajewski, T., and Boros, P. (2003). Query segmentation for web search. In *Proceedings of the 12th International Conference on World Wide Web*.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall.

Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12.

Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: a language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*.

Udupa, R., Bhole, A., and Bhattacharyya, P. (2009). "A term is known by the company it keeps": on selecting a good expansion set in pseudo-relevance feedback. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 104–115.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc.

Weerkamp, W., Balog, K., and de Rijke, M. (2012). Exploiting external collections for query expansion. *ACM Transactions on the Web*, 6(4):18:1–18:29.

Xu, Y., Ding, F., and Wang, B. (2008). Entity-based query reformulation using Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 1441–1442.

Xu, Y., Jones, G. J. F., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on Wikipedia. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 59–66.

Zesch, T., Gurevych, I., and Mühlhäuser, M. (2007). Analyzing and accessing Wikipedia as a lexical semantic resource. In *Proceedings of the Biannual Conference of the Society for Computational Linguistics and Language Technology*, pages 213–221.

Zhai, C. (2008). Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213.

Zhai, C. and Lafferty, J. D. (2001a). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 334–342.

Zhai, C. and Lafferty, J. D. (2001b). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th ACM Conference on Information and Knowledge Management*, pages 403–410.