Query Expansion with Freebase

Chenyan Xiong
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
cx@cs.cmu.edu

Jamie Callan
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
callan@cs.cmu.edu

ABSTRACT

Large knowledge bases are being developed to describe entities, their attributes, and their relationships to other entities. Prior research mostly focuses on the construction of knowledge bases, while how to use them in information retrieval is still an open problem. This paper presents a simple and effective method of using one such knowledge base, *Freebase*, to improve *query expansion*, a classic and widely studied information retrieval task. It investigates two methods of identifying the entities associated with a query, and two methods of using those entities to perform query expansion. A supervised model combines information derived from Freebase descriptions and categories to select terms that are effective for query expansion. Experiments on the ClueWeb09 dataset with TREC Web Track queries demonstrate that these methods are almost 30% more effective than strong, state-of-the-art query expansion algorithms. In addition to improving average performance, some of these methods have better win/loss ratios than baseline algorithms, with 50% fewer queries damaged.

Keywords

Knowledge Base; Query Expansion; Freebase; Pseudo Relevance Feedback

1. INTRODUCTION

During the last decade, large, semi-structured *knowledge bases* or *knowledge graphs* have emerged that are less structured than typical relational databases and semantic web resources but more structured than the texts stored in full-text search engines. The weak semantics used in these semi-structured information resources is sufficient to support interesting applications, but is also able to accommodate contradictions, inconsistencies, and mistakes, which makes them easier to scale to large amounts of information. *Freebase*, which contains 2.9 billion 'facts' (relationships and attributes) about 48 million 'topics' (entities) is one well-known example of this class of resources.

ICTIR '15, September 27-30, 2015, Northampton, MA, USA
 ISBN 978-1-4503-3833-2/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2808194.2809446.

Typically the information in a knowledge base is organized around entities and relations. Knowledge base entities correspond to real-world entities and concepts (e.g., `Carnegie Mellon University`). Most entities have brief text descriptions, attributes, and typed relations to other entities. For example, in *Freebase* the basketball player `Michael Jordan` is represented by an object that is linked to a brief text description, attributes such as his career statistics, categories such as `people` and `athlete`, and related entities such as `Chicago Bulls`.

Although Freebase and other knowledge bases contain information that can improve understanding of a topic, how to use it effectively for information retrieval tasks is still an open problem. An intuitive use is query expansion, which generates expansion terms to enhance the original query and better represent user intent. Most recently, Dalton et al. [13] did query expansion using entity names, aliases and categories with several methods of linking entities to the query. They enumerate all possible expansion queries from combinations of linking methods, expansion fields, and hyper-parameters, and treat every expansion query's ranking scores for documents as features of a learning to rank model. However, the effectiveness on the ClueWeb09 and ClueWeb12 web corpora is mixed when compared with strong, state-of-the-art query expansion baselines. Given their thorough exploration of Freebase information and use of supervision, it seems that expansion using Freebase is a rather complicated and challenging task, in which existing techniques can only provide moderate improvements. How to do query expansion using Freebase in a both simple and effective way, especially on web corpora, remains an open problem.

This paper focuses on query expansion and presents several simple yet effective methods of using Freebase to do query expansion for a web corpus. We decompose the problem into two components. The first component identifies query-specific entities to be used for query expansion. We present implementations that retrieve entities directly, or select entities from retrieved documents. The second component uses information about these entities to select potential query expansion terms. We present implementations that use a tf.idf method to select terms, and category information to select terms. Finally, a supervised model is trained to combine information from multiple sources for better expansion.

Our experiments on the TREC Web Track adhoc task demonstrate that all our methods, when used individually, are about 20% more effective than previous state-of-the-art

query expansion methods, including Pseudo Relevance Feedback (PRF) on Wikipedia [27] and supervised query expansion [4]. In addition to these improvements, experimental results show that our methods are more robust and have better *win/loss* ratios than state-of-the-art baseline methods, reducing the number of damaged queries by 50%. This makes query expansion using Freebase more appealing, because it is well-known that most query expansion techniques are 'high risk / high reward' insofar as they often damage as many queries as they improve, which is a huge disadvantage in commercial search systems. The supervised model also successfully combines evidence from multiple methods, leading to 30% gains over the previous state-of-the-art. Besides being the first to improve query expansion this much on the widely used ClueWeb09 web corpus, the methods presented here are also fully automatic.

The next section provides a more in-depth discussion of prior research on query expansion. Section 3 provides a background description of Freebase which is essential to this paper. New methods of using Freebase for query expansion are discussed in Section 4. Experimental methodology and evaluation results are described in Sections 5 and 6 respectively. The last section summarizes the paper's contributions and discusses several interesting open problems suggested by this research.

2. RELATED WORK

Usually queries to web search engines are short and not written carefully, which makes it more difficult to understand the intent behind a query and retrieve relevant documents. A common solution is query expansion, which uses a larger set of related terms to represent the user's intent and improve the document ranking.

Among various query expansion techniques, Pseudo Relevance Feedback (PRF) algorithms are the most successful. PRF assumes that top ranked documents for the original query are relevant and contain good expansion terms. For example, Lavrenko et al.'s RM model selects expansion terms based on their term frequency in top retrieved documents, and weights them by documents' ranking scores:

$$s(t) = \sum_{d \in D} p(t|d) f(q,d)$$

where D is the set of top retrieved documents, $p(t|d)$ is the probability that term t is generated by document d's language model, and $f(q,d)$ is the ranking score of the document provided by the retrieval model [17]. Later, Metzler added inverse document frequency (IDF) to demote very frequent terms:

$$s(t) = \sum_{d \in D} p(t|d) f(q,d) \log \frac{1}{p(t|C)} \quad (1)$$

where $p(t|C)$ is the probability of term t in the corpus language model C [20].

Another famous PRF approach is the Mixture Model by Tao et al. [26]. They assume the terms in top retrieved documents are drawn from a mixture of two language models: query model θ_q and a background model θ_B. The likelihood of a top retrieved document d is defined as:

$$\log p(d|\theta_q, \alpha_d, \theta_B) = \sum_{t \in D} \log(\alpha_d \; p(t|\theta_q) + (1 - \alpha_d) \; p(t|\theta_B)).$$

α_d is a document-specific mixture parameter. Given this equation, the query model θ_q can be learned by maximizing the top retrieved documents' likelihood using EM. The terms that have non-zero probability in θ_q are used for query expansion.

Although these two algorithms have different formulations, they both focus on term frequency information in the top retrieved documents. So do many other query expansion algorithms [11, 18, 22, 28]. For example, Robertson et al.'s BM25 query expansion selects terms based on their appearances in relevant (or pseudo relevant) documents versus in irrelevant documents [24]. Lee et al. cluster PRF documents and pick expansion terms from clusters [18]. Metzler and Croft include multi-term concepts in query expansion and select both single-term concepts and multi-term concepts by a Markov Random Field model [22].

The heavy use of top retrieved documents makes the effectiveness of most expansion methods highly reliant on the quality of the initial retrieval. However, web corpora like ClueWeb09 are often noisy and documents retrieved from them may not generate reasonable expansion terms [3, 14]. Cao et al.'s study shows that top retrieved documents contain as many as 65% harmful terms [4]. They then propose a supervised query expansion model to select good expansion terms. Another way to avoid noisy feedback documents is to use an external high quality dataset. Xu et al. proposed a PRF-like method on top retrieved documents from Wikipedia, whose effectiveness is verified in TREC competitions [14, 27]. Kotov and Zhai demonstrated the potential effectiveness of concepts related to query terms in ConceptNet for query expansion, and developed a supervised method that picks good expansion concepts for difficult queries [15].

Another challenge of query expansion is its 'high risk / high reward' property, that often as many queries are damaged as improved. This makes query expansion risky to use in real online search service because users are more sensitive to failures than successes [8]. Collins-Thompson et al. [11] address this problem by combining the evidences from sampled sub-queries and feedback documents. Collins-Thompson also propose a convex optimization framework to find a robust solution based on previous better-on-average expansion terms [9, 10]. The risk is reduced by improving inner difference between expansion terms, and enforcing several carefully-designed constraints to ensure that expansion terms provide good coverage of query concepts.

The fast development of Freebase has inspired several works that use Freebase in query expansion. Pan et al. use the name of related Freebase objects [23]. The related objects are those whose names exactly or partially match the original query, or have a neighbor whose name matches. Dempster-Shafer theory is used to select expansion terms that have supporting evidence from different objects. More recently, Dalton et al.'s entity query feature expansion (EQFE) method explores many kinds of information in Freebase. [13]. They link query to Freebase objects by query annotation, keyword matching, and annotation of top retrieved documents. The name, alias, Freebase types, and Wikipedia categories of linked entities are considered as possible expansion phrases. The combination of different linking methods, different expansion fields of linked entities, and different values of hyper-parameters are enumerated to generate a vast amount of expansion queries. A document's ranking scores with all expansion queries are treated as fea-

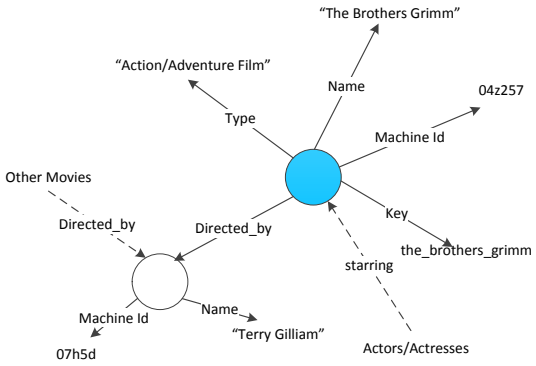

Figure 1: A sub-graph of Freebase.

tures for a learning to rank model. Their methods show great improvements on a cleaner corpus (Robust04), but their effectiveness on noisier web corpora (ClueWeb09 and ClueWeb12) is mixed compared with state-of-the-art expansion baselines. To the best of our knowledge, it remains unclear how to use Freebase for effective query expansion on web corpora.

3. FREEBASE OVERVIEW

Freebase[1] is a large public knowledge base that contains semi-structured information about real world entities and their facts. By July 21st, 2015, Freebase contained 48 million entities and 2.9 billion facts, which makes it the largest among public knowledge bases such as Open Information Extraction [2], NELL [5] and DBPedia [19].

The information in Freebase is organized as a graph, as shown in Figure 1. An entity (object) is expressed as a node in the graph with a unique Machine Id. An edge in the graph links an entity to another entity or its attribute. There are many kinds of edges in Freebase, representing different facts. For example, in Figure 1, `The Brothers Grimm` is connected to `Terry Gilliam` by a `Directed_by` edge, showing that the movie `The Brothers Grimm` is directed by the director `Terry Gilliam`. This research mainly explores some of Freebase's facts, as listed Table 1. For an object's textual fields, such as 'name' and 'description', only the English text is used.

Another resource used in this research is Google's FACC1 [1] annotations, which link Freebase objects to documents in the well-known ClueWeb09 and ClueWeb12 web corpora. The annotation was done automatically by Google. The annotation is of good quality and is tailored to favor Precision over Recall. In a small scale manual examination, the Precision was 80% to 85%; the Recall, although hard to estimate, is believed to be 70% to 85% [1].

4. EXPANSION USING FREEBASE

In this section we introduce our methods of using Freebase for query expansion. We first discuss our unsupervised expansion methods utilizing different information from Freebase. Then we propose a supervised query expansion method to combine evidence from our unsupervised methods.

[1] `http://www.freebase.com/`

Table 1: Freebase facts used.

Fact	Description
mId	Unique Id
key:en	English Key (if any)
rdfs:label	Object's Name
/common/topic/description	Text Description
rdf:type	Category (if any)

4.1 Unsupervised Expansion Using Freebase

We perform unsupervised query expansion using Freebase in two steps: object linking and term selection. In object linking, we develop implementations that retrieve objects directly, or select them from annotations in top ranked documents. In term selection, we also present two implementations: one uses the tf.idf information from object descriptions; the other uses similarity of query and the term's distributions in Freebase's categories.

Formally, given a query q, and a ranked list of documents from initial retrieval $D = \{d_1, ...d_j..., d_N\}$, the goal of the object linking step is to generate a ranked list of Freebase objects $O = \{o_1, ...o_k..., o_K\}$, with ranking scores $r(O) = \{r(o_1), ...r(o_k)..., r(o_K)\}$. The goal of term selection is to find a set of expansion terms $T = \{t_1, ...t_i..., t_M\}$ and their scores $s(T) = \{s(t_1), ...s(t_i)..., s(t_M)\}$ from linked objects using their descriptions $e(O) = \{e(o_1), ...e(o_k)..., e(o_K)\}$ and Freebase categories $C = \{c_1, ...c_u..., c_U\}$.

4.1.1 Linking Freebase Objects to the Query

Our first linking method retrieves objects directly. The query q is issued to the Google Search API[2] to get its ranking of objects O with ranking scores $r_s(O)$. The ranking score ranges from zero to several thousands, with a typical long tailed distribution. We normalize them so that the ranking scores of each query's retrieved objects sum to one.

Our second approach selects related objects from the FACC1 annotations in top retrieved documents. It is a common assumption that top retrieved documents are a good representation of the original query. Intuitively the objects that appear frequently in them shall convey meaningful information as well. We utilize such information by linking the query to objects that are frequently annotated to top retrieved documents.

Specifically, for a query q's top retrieved documents D, we fetch their FACC1 annotations, and calculate the ranking score for object o_k as:

$$r_f(o_k) = \sum_{d_j \in D} tf(d_j, o_k) \log \frac{|F|}{df(o_k)}. \quad (2)$$

In Equation 2, $tf(d_j, o_k)$ is the frequency of object o_k in document d_j's annotations, and $df(o_k)$ is the total number of documents o_k is annotated to in the whole corpus. $|F|$ is the total number of documents in the corpus that have been annotated in the FACC1 annotation. $\frac{|F|}{df(o_k)}$ in Equation 2 serves as inverse document frequency (IDF) to demote objects that are annotated to too many documents. $r_f(o_k)$

[2] `https://developers.google.com/freebase/v1/getting-started`

is normalized so that ranking scores of each query's objects sum to one.

4.1.2 Selecting Expansion Terms from Linked Objects

We develop two methods to select expansion terms from linked objects.

The first method does tf.idf based Pseudo Relevance Feedback (PRF) on linked objects' descriptions. PRF has been successfully used with Wikipedia articles [3, 14, 27]. It is interesting to see how it works with Freebase.

Given the ranked objects O and $r(O)$, a term's score is calculated by:

$$s_p(t_i) = \sum_{o_k \in O} \frac{tf(e(o_k), t_i)}{|e(o_k)|} \times r(o_k) \times \log \frac{|E|}{df(t_i)} \quad (3)$$

where $tf(e(o_k), t_i)$ is the term frequency of t_i in o's description, $|e(o_k)|$ is the length of the description, $df(t_i)$ is the document frequency of t_i in the entire Freebase's description corpus E. $|E|$ is the total number of entities in Freebase that have a description.

Our second term selection method uses Freebase's entity categories. Freebase provides an ontology tree that describes entities at several levels of abstraction. We use the highest level in the ontology tree, such as */people* and */movie*, to make sure sufficient instances exist in each category. There are in total $U = 77$ first level categories in Freebase. The descriptions of entities in these categories are training data to learn the language models used to describe these categories.

Our second approach estimates query and terms distributions on categories, and selects terms that have similar category distributions with the query.

The distribution of a term in Freebase categories is estimated using a Naive Bayesian classifier. We first calculate the probability of a term t_i generated by a category c_u via:

$$p(t_i|c_u) = \frac{\sum_{o_k \in c_u} tf(e(o_k), t_i)}{\sum_{o_k \in c_u} |e(o_k)|}$$

where $o_k \in c_u$ refers to objects in category c_u.

Using Bayes' rule, the probability of term t_i belonging to category c_u under uniform priors is:

$$p(c_u|t_i) = \frac{p(t_i|c_u)}{\sum_{c_u \in C} p(t_i|c_u)}.$$

Similarly, the category distribution of a query q is:

$$p(q|c_u) = \prod_{t_i \in q} p(t_i|c_u),$$

$$p(c_u|q) = \frac{p(q|c_u)}{\sum_{c_u \in C} p(q|c_u)}.$$

The similarity between the two distributions $p(c_u|t_i)$ and $p(c_u|q)$ is evaluated by negative Jensen-Shannon divergence:

$$s_c(t_i) = -\frac{1}{2}\mathrm{KL}(p(C|q)||p(C|q,t_i)) - \frac{1}{2}\mathrm{KL}(p(C|t_i)||p(C|q,t_i))$$

where:

$$p(C|q,t_i) = \frac{1}{2}(p(C|q) + p(C|t_i))$$

and $\mathrm{KL}(\cdot||\cdot)$ is the KL divergence between two distributions. $s_c(t_i)$ is the expansion score for a term t_i. We use a min-max normalization to re-range all $s_c(t_i)$ into $[0, 1]$.

Table 2: Unsupervised Query Expansion Methods Using Freebase.

	Link by Search	Link by FACC1
Select by PRF	`FbSearchPRF`	`FbFaccPRF`
Select by Category	`FbSearchCat`	`FbFaccCat`

As a result, we have two methods that link related Freebase objects to a query, and two methods to select expansion terms from linked objects. They together form four unsupervised expansion methods, as listed in Table 2.

4.2 Supervised Expansion Using Freebase

Different object linking and term selection algorithms have different strengths. Object search links objects that are directly related to the query by keyword matching. FACC1 annotation provides objects that are more related in meanings and does not require exact textual matches. In expansion term selection, PRF picks terms that frequently appear in objects' descriptions. The category similarity method selects terms that have similar distributions with the query in Freebase's categories. They together provide three scores describing the relationship between a query-term pair: tf.idf Pseudo Relevance Feedback score in retrieved objects, tf.idf Pseudo Relevance Feedback score in top retrieved documents' FACC1 annotations, and a negative Jensen-Shannon divergence score between category distributions.

The three scores are used as features for a supervised model that learns how to select better expansion terms. All terms in linked objects' descriptions are used as candidates for query expansion. The ground truth score for a candidate term is generated by its influence on retrieved documents, when used for expansion individually. If a term increases the ranking scores of relevant documents, or decreases the ranking scores of irrelevant documents, it is considered to be a good expansion term, and vice versa.

The influence of a term t_i over retrieved documents is calculated as:

$$y(t_i) = \frac{1}{|R|} \sum_{d_j \in R} (f(q + t_i, d_j) - f(q, d_j)) - \frac{1}{|\bar{R}|} \sum_{d_j \in \bar{R}} (f(q + t_i, d_j) - f(q, d_j))$$

where R and $\bar{R}$ are the sets of relevant and irrelevant documents in relevance judgments. $f(q, d_j)$ is the ranking score for document d_j and query q in the base retrieval model. $f(q + t_i, d_j)$ is the ranking score for d_j when the query is expanded using expansion term t_i individually. Binary labels are constructed using $y(t)$. Terms with $y(t) > 0$ are treated as good expansion terms and the rest as bad expansion terms.

Our ground truth label generation is a little different than Cao et al.'s [4]. Their labels were generated by a term's influence on documents' ranking positions: if relevant documents are moved up or irrelevant document are moved down by a term, it is considered a good expansion term, otherwise a bad one. In comparison, we use influence on ranking scores which reflect an expansion term's effectiveness more directly. Our preliminary experiments also confirm that both their method and our method work better with our ground truth labels.

We used a linear SVM classifier to learn the mapping from the three features of a term t to its binary label. To get the expansion weights, we used the probabilistic version of SVM in the LibSVM [7] toolkit to predict the probability of a term being a good expansion term. The predicted probabilities are used as terms' expansion scores, and those terms with highest scores are selected for query expansion.

4.3 Ranking with Expansion Terms

We use the selected expansion terms and their scores to re-rank the retrieved documents with the RM model [17]:

$$f^*(d_j, q) = w_q f(q, d_j) + (1 - w_q)(\sum_{t_i \in T} s(t_i) f(t_i, d_j))). \quad (4)$$

In Equation 4, $f^*(q, d_j)$ is the final ranking score to re-rank documents. $f(q, d_j)$ and $f(t_i, d_j)$ are the ranking scores generated by the base retrieval model, e,g, BM25 or query likelihood, for query q and the expansion term t_i respectively. w_q is the weight on the original query. T is the set of selected expansion terms and $s(t_i)$ is the expansion score of the term t_i. Expansion scores are normalized so that the scores of a query's expansion terms sum to one.

5. EXPERIMENTAL METHODOLOGY

In this section we introduce our experimental methodology, including dataset, retrieval model, baselines, hyperparameters, and evaluation metrics.

Dataset: Our experiments use ClueWeb09, TREC Web Track 2009-2012 adhoc task queries and the relevance judgments provided by TREC annotators. This dataset models a real web search scenario: queries are selected from the search log from Bing, and ClueWeb09 is a widely used web corpus automatically crawled from the internet by Carnegie Mellon University. ClueWeb09 is known to be a hard dataset for query expansion [3, 13, 14], because it is much noisier than carefully edited corpora like the Wall-street Journal, news and government web sets.

We use Category B of ClueWeb09 and index it using the Indri search engine [25]. Typical INQUERY stopwords are removed before indexing. Documents and queries are stemmed using the Krovetz stemmer [16]. Spam filtering is very important for ClueWeb09 and we filter the 70% most spammy documents using the Waterloo spam score [12].

We retrieved Freebase objects, and fetched their descriptions using the Google Freebase API on July 16th, 2014. Entity linking from documents to entities are found in FACC1 annotation [1], which was published in June 2013. Corpus statistics such as term IDF and categories' language models were calculated from the April 13th, 2014 Freebase RDF dump.

Retrieval Model: We use Indri's language model [20] as our base retrieval model. The ranking score of a document is the probability of its language model generating the query. Dirichlet smoothing is applied to avoid zero probability and incorporate corpus statistics:

$$p(q|d_j) = \frac{1}{|q|} \sum_{t_i \in q} \frac{t f(d_j, t_i) + \mu p(t_i|\mathcal{C})}{|d_j| + \mu}, \quad (5)$$

where $p(t_i|\mathcal{C})$ is the probability of seeing term t_i in the whole corpus, and μ is the parameter controlling the smoothing strength, set to the Indri default: 2500.

Baselines: We compare our four unsupervised expansion methods (as listed in Table 2) and the supervised method described in Section 4.2 (`FbSVM`) with several baselines. The first baseline is the Indri language model (`IndriLm`) as in Equation 5. All relative performances and Win/Loss evaluations of other methods are compared with `IndriLm` if without specific reference. Our second baseline is the Sequential Dependency Model (`SDM`) [21], a strong competitor in TREC Web Tracks.

We also include two well-known state-of-the-art query expansion methods as baselines. The first one is Pseudo Relevance Feedback on Wikipedia (`RmWiki`) [3, 14, 27]. We indexed the Oct 1st 2013 Wikipedia dump using same setting we used for ClueWeb09. Standard Indri PRF with the IDF component [14] was performed to select expansion terms.

The other query expansion baseline is the supervised query expansion (`SVMPRF`) [4]. We extracted the 10 features described in their paper, and trained an SVM classifier to select good expansion terms. We used our term level ground truth labels as discussed in Section 4.2, because their model performs better with our labels. Following their paper, the RBF kernel was used, which we also found necessary for that method to be effective.

For clarity and brevity, we do not show comparison with other methods such as RM3 [17], Mixture Model [26], or EQFE [13] because they all perform worse on ClueWeb09 than `RmWiki` and `SDM` in our experiment, previous TREC competitions [14], or in their published papers.

Parameter Setting: Hyper parameters in our experiment, including the number of expansion terms (M), number of objects (K) in Freebase linked for expansion, and number of PRF documents for `RmWiki` and `SVMPRF`, are selected by maximizing the performance on training folds in a five-fold cross validation. The number of expansion terms is selected from $\{1, 3, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, the number of entities is selected from $\{1, 3, 5, 10, 15, 20, 30, 40, 50\}$ and the number of PRF documents is selected from $\{5, 10, 15, 20, 25, 30\}$.

Parameters of SVM in supervised expansion (`FbSVM` and `SVMPRF`) are selected by another five-fold cross validation. In each of the five folds of the outside cross validation that were used to select expansion parameters, we performed a second level cross validation to select the parameters of SVM. The explored range of cost c of linear kernel and RBF kernel is $\{0.01, 0.1, 1, 10, 100\}$. The range of γ in RBF kernel is $\{0.1, 1, 10, 100, 1000, 10000\}$.

To keep the experiment tractable, other parameters were fixed following conventions in previous work [4, 14, 27]. The weight of the original query w_q is set to 0.5, and the re-rank depth is 1000. We chose re-ranking instead of retrieval again in the whole index because the latter is very expensive with the large set of expansion terms and did not show any significant difference in our experiments. When using FACC1 annotations to link objects, we used the FACC1 annotations in the top 20 retrieved documents provide by `IndriLm`. The candidate terms for `FbSVM` were generated from the top 20 retrieved objects and top 20 linked FACC1 annotations. To reduce noise in object description, we ignored terms that contained less than three characters.

Evaluation Metric. Our methods re-ranked the top retrieved documents, so we mainly focus evaluation on the top 20 documents in the re-ranked list. We chose ERR@20 as

Table 3: Performance of unsupervised expansion using Freebase. Relative gain is calculated using ERR over IndriLm. Win/Loss/Tie is the number of queries helped, hurt, and not changed comparing with IndriLm. †, ‡, § and ¶ mark the statistic significant improvements ($p < 0.05$) over IndriLm, SDM, RmWiki and SVMPRF respectively. The best results in each column are marked **bold**.

Method	MAP@20	NDCG@20	ERR@20	Relative Gain	Win/Loss/Tie
IndriLm	0.357	0.147	0.116	NA	NA
SDM	$0.387^{\dagger}$	$0.166^{\dagger}$	$0.122^{\dagger}$	5.52%	58/27/115
RmWiki	0.362	$0.161^{\dagger}$	0.114	−1.70%	67/72/61
SVMPRF	0.367	$0.158^{\dagger}$	0.125	8.00%	63/72/65
FbSearchPRF	$\mathbf{0.436}^{\dagger,\ddagger,\S,\P}$	$\mathbf{0.186}^{\dagger,\ddagger,\S,\P}$	$\mathbf{0.152}^{\dagger,\ddagger,\S,\P}$	30.80%	84/30/86
FbSearchCat	$0.421^{\dagger,\ddagger,\S,\P}$	$0.182^{\dagger,\ddagger,\S,\P}$	$0.144^{\dagger,\ddagger,\S,\P}$	23.99%	67/43/90
FbFaccPRF	$0.428^{\dagger,\ddagger,\S,\P}$	$0.184^{\dagger,\ddagger,\S,\P}$	$0.145^{\dagger,\ddagger,\S,\P}$	24.71%	97/55/48
FbFaccCat	$0.400^{\dagger,\S,\P}$	$0.173^{\dagger}$	$0.136^{\dagger,\ddagger,\S}$	17.25%	88/67/45

our main evaluation metric, which is the main metric of the TREC Web Track adhoc task. We also show the evaluation results for MAP@20 and NDCG@20.

6. EVALUATION RESULTS

In this section, we first discuss the average performance of our unsupervised expansion methods using Freebase, comparing with current state-of-the-art baselines. Then we evaluate our supervised expansion method. Besides average performances, we also analysis our methods' robustness at the individual query level. We conclude our analysis with case studies and discussions.

6.1 Performance of Unsupervised Expansion

The average performances on MAP, NDCG and ERR are shown in Table 3. The relative gain and Win/Loss ratio are compared with IndriLm on ERR. Statistical significance tests are performed using the Permutation test. Labels †, ‡, § and ¶ indicate statistical significance ($p < 0.05$) over IndriLm, SDM, RmWiki and SVMPRF respectively.

Our unsupervised expansion methods outperform all state-of-the-art baselines by large margins for all evaluation metrics. All the gains over IndriLm are statistically significant, while SVMPRF and RmWiki are only significantly better on NDCG. Three of the methods, FbSearchPRF, FbSearchCat and FbFaccPRF, are significantly better than all baselines. FbFaccCat's improvements do not always pass the statistical significance test, even when the relative gains are almost 10%. This reflects the high variance of query expansion methods, which is addressed in Section 6.3.

Comparing the performances of our methods, linking objects by search works better than by FACC1 annotations, and selecting expansion terms by PRF works better than using category similarity. One possible reason is that objects from FACC1 annotation are noisier because they rely on the quality of top retrieved documents. Also the category similarity suffers because suitable categories for query or terms may not exist.

We further compare our unsupervised expansion methods at the query level. The results are shown in Table 4. Each cell shows the comparison between the method in the row and the method in the column. The three numbers are the number of queries in which the row method performs better (win), worse (loss), and equally (tie) with the column method respectively. The results demonstrate that our methods do perform differently. The two most similar methods are FbSearchPrf and FbSearchCat, doing the same on 80 queries out of 200. But 36 queries have no returned objects from the Google Search API, on which two methods retreat to IndriLm. Otherwise our four unsupervised methods perform the same for at most 49 queries.

These results showed the different strengths of our unsupervised methods. The next experiment investigates whether they can be combined for further improvements by a supervised method.

6.2 Performance of Supervised Expansion

The performance of our supervised method FbSVM, which utilized the evidence from our unsupervised methods, is shown in Table 5. To investigate whether the combination of multiple sources of evidence is useful, we conduct statistical significance tests between FbSVM with our unsupervised methods. †, ‡, § and ¶ indicates statistical significance in the permutation test over FbSearchPRF, FbSearchCat,FbFaccPRF and FbFaccCat correspondingly.

The results demonstrate that evidence from different aspects of Freebase can be combined for further improvements: FbSVM outperforms IndriLm by as much as 42%. Statistical significance is observed over our unsupervised methods on $NDCG$, but not always on MAP and ERR. We have also run statistical significance tests between FbSVM and all other baselines, which are all statistically significant as expected.

FbSVM and SVMPRF differ in their candidate terms and features. FbSVM selects terms from Freebase, while SVMPRF selects from web corpus. FbSVM uses features from Freebase's linked objects' descriptions and categories, while FbSVM uses term distribution and proximity in top retrieved documents from web corpus. Table 6 shows the quality of candidate terms from two sources. Surprisingly, Freebase' candidate terms are slightly weaker in quality (39.4% vs. 41.4%) and there are more of them. However, FbSVM's classification Precision is about 10% relatively better than SVMPRF, as shown in Table 7. The Recall of FbSVM is lower, but FbSVM still picks more good expansion terms given the larger number of good candidate terms in Freebase.

Nevertheless, the marginal gains of FbSVM over our best performing unsupervised method FbSearchPRF are not as high as expected. Our preliminary analysis shows that one possible reason is the features between query and terms are limited, i.e. only three dimensions. Another possible reason is the way of using the supervised information (document relevance judgments). Document relevance judgments are used to generate labels at the term level using heuristics,

Table 4: The query level Win/Loss/Tie comparison between our methods. Each cell shows the number of queries helped (Win), damaged (Loss) and not changed (Tie) by row method over column method.

	FbSearchPRF	FbSearchCat	FbFaccPRF	FbFaccCat
FbSearchPRF	NA/NA/NA	73/47/80	82/74/44	95/65/40
FbSearchCat	47/73/80	NA/NA/NA	72/86/42	87/72/41
FbFaccPRF	74/82/44	86/72/42	NA/NA/NA	84/67/49
FbFaccCat	65/95/40	72/87/41	67/84/49	NA/NA/NA

Table 5: Performance of supervised expansion using Freebase. Relative gain and Win/Loss/Tie are calculated comparing with IndriLm on ERR. †, ‡, § and ¶ mark the statistically significant improvements over FbSearchPRF, FbSearchCat,FbFaccPRF and FbFaccCat respectively. Best results in each column are marked **bold**.

Method	MAP@20	NDCG@20	ERR@20	Relative Gain	Win/Loss/Tie
FbSearchPRF	0.436	0.186	0.152	30.80%	84/30/86
FbSearchCat	0.421	0.182	0.144	23.99%	67/43/90
FbFaccPRF	0.428	0.184	0.145	24.71%	97/55/48
FbFaccCat	0.400	0.173	0.136	17.25%	88/67/45
FbSVM	**0.444**	**0.199**†,‡,§,¶	**0.165**‡,§,¶	42.42%	96/63/41

while the final document ranking is still computed using unsupervised retrieval models. A more powerful machine learning framework seems necessary to better utilize Freebase information. This would be a good topic for further research.

6.3 Query Level Analysis

A common disadvantage of query expansion methods is their high variances: they often hurt as many queries as helped. To evaluate the robustness of our methods, we compare the query level performance of each method versus IndriLm and record the Win/Loss/Tie numbers. The results are listed in the last columns of Tables 3 and 5. Table 3 shows that SDM, which is widely recognized as effective across many datasets, is reasonably robust and hurts only half as many queries as it helps. It also does not change the performance of 115 queries partly because 53 of them only contain one term on which nothing can be done by SDM. In comparison, RmWiki and SVMPRF hurt more queries than they help, which is consistent with observations in prior work [8].

Our methods have much better Win/Loss ratios than baseline query expansion methods. When selecting terms using PRF from linked objects' descriptions, FbSearchPRF and FbFaccPRF improve almost twice as many queries as they hurt. The variance of term selection by category is higher, but FbSearchCat and FbFaccCat still improve at least 30% more queries than they hurt. Linking by object retrieval has slightly better Win/Loss ratios than by FACC1 annotation, but it also helps a smaller number of queries. One reason is that for some long queries, their is no object retrieved by Google API.

More details of query level performance can be found in Figure 2. The x-axis is the bins of relative performances on ERR compared with IndriLm. The y-axis is the number of queries that fall into corresponding bins. If the performance is the same for a query, we put it into 0 bin. If a query is helped by 0 to 20%, we put it into bin 20%, etc. Figure 2 confirms the robustness of our methods. Especially for FbSearchPRF and FbFaccPRF, more queries are helped, less queries are hurt, and much less queries are extremely damaged.

Table 6: Candidate term quality from top retrieved documents (Web Corpus) and linked objects' descriptions (Freebase). Good and bad refer to the number of terms that have positive and negative influences on ranking accuracy respectively.

Source	Good	Bad	Good Fraction
Web corpus	9,263	13,087	41.4%
Freebase	19,247	29,396	39.6%

Table 7: Classification performance of supervised methods.

Method	Precision	Recall
SVMPRF	0.5154	0.0606
FbSVM	0.5609	0.0400

FbSVM's robustness is average among our expansion methods, and is better than RmWiki and SDM. Fewer queries fall into bin 0, as it is rare that none of our evidence affects a query. However the number of damaged queries is not reduced. One possible reason is that the ground truth we used to train the SVM classifier is the individual performance of each candidate term, and only the average performance is considered in model training/testing. As a result, our model might focus more on improving average performance but not on reducing risk.

6.4 Case Study and Discussion

To further understand the properties of our object linking and term selection methods, Table 8 lists the queries that are most helped or hurt by different combinations of methods. The ↑ row shows the most helped queries and ↓ row shows those most hurt[3]. The comparison is done on ERR compared to IndriLm too.

Table 8 shows the different advantages of linking by object search and FACC1 annotation. For example, the query 'fybromyalgia' is damaged by FbFaccPRF, while improved by FbSearchPRF and FbSearchCat. The FACC1 annotation

[3]More details including linked objects and expansion terms are available at http://boston.lti.cs.cmu.edu/appendices/ICTIR2015/.

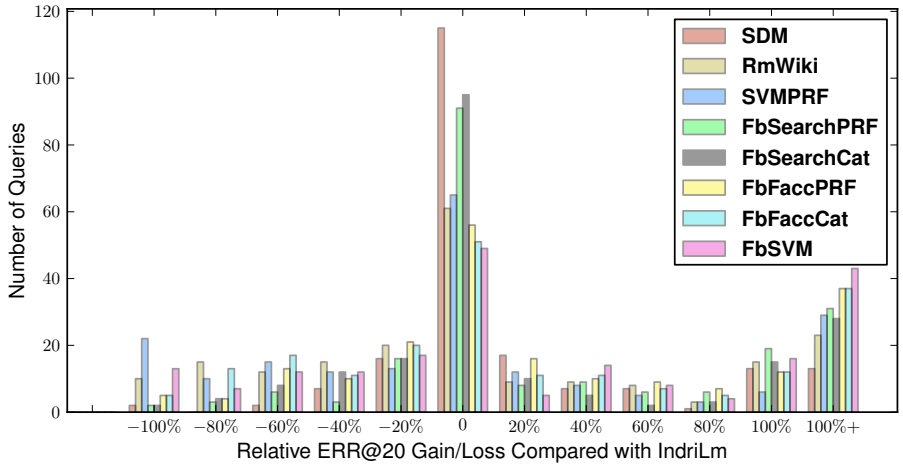

Figure 2: Query level relative performance. X-axis is the bins of relative performance on ERR compared with `IndriLm`. Y-axis is the number of queries that fall into each bin. Bin 0 refers to queries that were not changed, 20% refers to queries that improved between (0%, 20%], etc. The left-to-right ordering of histograms in each cell corresponds to the top-to-bottom ordering of methods shown in the key.

leads to a set of weakly related objects, like doctors and organizations focused on diseases, which generate overly-general expansion terms. Instead, object search is precise and returns the object about 'fybromyalgia'. Sometimes the generality of FACC1 annotations can help instead. For example, for query 'rock art' whose main topic is about rock painting, object search links to objects about rock music, while FACC1 annotation is more general and links to related objects for both rock painting and rock music.

Our two term selection methods also have different behaviors. An exemplary case is the query 'computer programming', on which `FbSearchPRF` and `FbFaccPRF` perform very well, while `FbSearchCat` and `FbFaccCat` do not. The linked objects of two methods are both reasonable: object search links to mostly programming languages, and FACC1 annotation brings in programming languages, textbooks and professors. With the good quality of linked objects, PRF selects good expansion terms from their descriptions. However, category similarity picks terms like: 'analysis', 'science', 'application' and 'artificial', which are too general for this query. The granularity of the Freebase ontology's first level is too coarse for some queries, and lower levels are hard to use due to insufficient instances. Nevertheless, when the linked objects are noisy, like for the query 'sore throat', the category information helps pick more disease related terms using the '/medicine' category and provides better performance.

Some queries difficult for all methods. For example, 'wedding budget calculator' contains the entities 'wedding', 'budget' and 'calculator', but actually refers to the concept 'wedding budget' and how to calculate it. Similar cases are 'tangible personal property tax' and 'income tax return online', whose meanings cannot be represented by a single Freebase object.

There are also queries on which Freebase is very powerful. For example, the query 'UNC' asks for the campuses of the University of North Carolina. Freebase contains multiple objects about UNC campuses, and campuses of other related universities, which generate good expansion terms. Freebase is also very effective for 'Figs', 'Atari', 'Hoboken' and 'Korean Language', whose meanings are described thoroughly by linked Freebase objects.

To sum up, our object linking and term selection methods utilize different parts of Freebase, and thus have different specialties. In object linking, object search is aggressive and can return the exact object for a query, when there are no ambiguities. FACC1 annotation relies on top retrieved documents and usually links to a set of related objects. Thus it is a better choice for queries with ambiguous meanings. In term selection, Pseudo Relevance Feedback via tf.idf directly reflects the quality of linked objects, and is better when the linked objects are reasonable. In contrast, category similarity offers a second chance to pick good expansion terms from noisy linked objects, when proper category definition exists for the query. `FbSVM` offers a preliminary way to combine the strength from different evidence and does provide additional improvements, however, more sophisticate methods that better use supervision and richer evidence from Freebase are possible for for future research.

Our work and EQFE by Dalton et al. [13] both focus on exploring the effectiveness of Freebase in improving information retrieval. One difference is that EQFE uses the entity's name, alias, and category names as possible expansion phrases, while we select terms from the linked entity's descriptions. Our techniques are also different: EQFE enumerates explored evidence to generate many features, and relies on a learning to rank model to handle them; our methods use classic query expansion techniques like pseudo relevance feedback, distributions of category language models, and term level supervised expansion. We also perform differently in different datasets: EQFE works better on a cleaner corpus (Robust04) while ours works better on a noisier corpus (ClueWeb09). Further study of the differences and strengths of these two approaches may lead to more general methods that can work well under multiple scenarios in future research.

Table 8: The queries most helped and hurt by our methods. ↑ row shows the five most-helped queries for each method, and ↓ shows the most-hurt queries.

	FbSearchPRF	FbSearchCat	FbFaccPRF	FbFaccCat
↑	porterville hobby stores fybromyalgia computer programming figs	unc porterville fybromyalgia bellevue figs	signs of a heartattack computer programming figs idaho state flower hip fractures	porterville idaho state flower bellevue flushing atari
↓	von willebrand disease website design hosting 403b ontario california airport rock art	rock art espn sports ontario california airport computer programming bobcat	wedding budget calculator poem in your pocket day fybromyalgia ontario california airport becoming a paralegal	poem in your pocket day ontario california airport computer programming bobcat blue throated hummingbird

7. CONCLUSION AND FUTURE WORK

In this work, we use Freebase, a large public knowledge base, in query expansion, a classic and widely studied information retrieval task. We present several simple yet effective methods to utilize different Freebase information, including annotation, description and category. We decompose the expansion problem into two components. The first component links Freebase objects to queries using keyword based retrieval, or an object's frequency in FACC1 annotations. The second component uses information about linked objects to select expansion terms. We develop two implementations including a tf.idf based Pseudo Relevance Feedback algorithm, and an algorithm that selects terms whose distributions in Freebase categories are similar with the query's. Finally, a supervised model is trained to combine evidence from multiple expansion methods. Our experiments on the ClueWeb09 dataset and TREC Web Track queries demonstrate that our methods are almost 30% more effective than previous state-of-the-art expansion systems. In addition, some of our methods significantly increases the win/loss ratios by reducing the number of damaged queries by 50%. To the best of our knowledge, our work is the first to show the effectiveness of Freebase for query expansion on the widely used web ClueWeb09 corpus.

This work only focused on a single dataset, albeit a dataset has been difficult for other query expansion methods. Previous query expansion methods have been very successful on cleaner corpora, but few of them work well on ClueWeb09. Future research must investigate the generality of our methods on a wider variety of datasets. Analyzing the differences between our methods and previous methods, and perhaps combining them, may lead to a better expansion system.

Entity linking is a widely studied topic, but previous research mostly focused on linking entities to documents, while linking entities to queries is more challenging due to the lower quality of query string. Recently the SIGIR 2014 *Entity Retrieval and Disambiguation Challenge* (*ERD '14*) [6] workshop provided web queries and documents with 'gold standard' Freebase annotations to promote research on identifying Freebase entities in different types of text. It would be beneficial to introduce better query object linking algorithms into our systems.

We have experimented with using a term's tf.idf score and category distribution information to select good expansion terms. One can use additional Freebase resources to derive more features for term selection, for example, relationships, attributes, and the contexts of FACC1 annotations in web corpus.

Last but not least, when combining evidence from different resources, our model FbSVM only considers average performance and ignores reducing the risk. Fancier models can be developed for better average performance and lower risk at the same time. Extracting more evidence from Freebase about controlling the risk is another potential way to better utilize the rich and complex data in Freebase.

8. ACKNOWLEDGMENTS

We thank the anonymous reviewers for comments that improved the paper. This research was supported by National Science Foundation (NSF) grant IIS-1422676 and a Google Research Award. Any opinions, findings, conclusions, and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors.

9. REFERENCES

[1] FACC1 Annotation on ClueWeb09. http://lemurproject.org/clueweb09/FACC1/. Accessed: 2014-06-26.

[2] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction for the web. In *Processding of the Internaltional Joint Conference on Artifical Intelligence, IJCAI(2007)*, volume 7, pages 2670–2676. IJCAI, 2007.

[3] M. Bendersky, D. Fisher, and W. B. Croft. Umass at Trec 2010 Web Track: Term dependence, spam filtering and quality bias. In *Proceedings of The 19th Text REtrieval Conference, (TREC 2010)*. NIST, 2010.

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

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, (AAAI 2010)*, volume 5, page 3. AAAI Press, 2010.

[6] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. ERD'14: Entity recognition and disambiguation challenge. In *SIGIR '14: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2014.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[8] K. Collins-Thompson. *Robust model estimation methods for information retrieval*. PhD thesis, Carnegie Mellon University, December 2008.

[9] K. Collins-Thompson. Estimating robust query models with convex optimization. In *Proceedings of the 21st Advances in Neural Information Processing Systems, (NIPS 2009)*, pages 329–336. NIPS, 2009.

[10] K. Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, (CIKM 2009)*, pages 837–846. ACM, 2009.

[11] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2007)*, pages 303–310. ACM, 2007.

[12] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

[13] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2014)*, pages 365–374. ACM, 2014.

[14] N. Dong and C. Jamie. Combination of evidence for effective web search. In *Proceedings of The 19th Text REtrieval Conference, (TREC 2010)*. NIST, 2010.

[15] A. Kotov and C. Zhai. Tapping into knowledge base for concept feedback: Leveraging ConceptNet to improve search results for difficult queries. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pages 403–412. ACM, 2012.

[16] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 1993)*, pages 191–202. ACM, 1993.

[17] 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, (SIGIR 2001)*, pages 120–127. ACM, 2001.

[18] K. S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2008)*, pages 235–242. ACM, 2008.

[19] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal*, 2014.

[20] D. Metzler. *Beyond bags of words: effectively modeling dependence and features in information retrieval*. PhD thesis, University of Massachusetts Amherst, September 2007.

[21] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2005)*, pages 472–479. ACM, 2005.

[22] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2007)*, pages 311–318. ACM, 2007.

[23] D. Pan, P. Zhang, J. Li, D. Song, J.-R. Wen, Y. Hou, B. Hu, Y. Jia, and A. De Roeck. Using dempster-shafer's evidence theory for query expansion based on freebase knowledge. In *Information Retrieval Technology*, pages 121–132. Springer, 2013.

[24] S. E. Robertson and S. Walker. Okapi/keenbow at TREC-8. In *Proceedings of The 8th Text REtrieval Conference, (TREC 1999)*, pages 151–162. NIST, 1999.

[25] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, volume 2, pages 2–6. Citeseer, 2005.

[26] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2006)*, pages 162–169. ACM, 2006.

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

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