Semantic Term Matching in Axiomatic Approaches to Information Retrieval

Hui Fang
Department of Computer Science
University of Illinois at Urbana-Champaign

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

ABSTRACT

A common limitation of many retrieval models, including the recently proposed axiomatic approaches, is that retrieval scores are solely based on *exact* (i.e., syntactic) matching of terms in the queries and documents, without allowing distinct but semantically related terms to match each other and contribute to the retrieval score. In this paper, we show that semantic term matching can be naturally incorporated into the axiomatic retrieval model through defining the primitive weighting function based on a semantic similarity function of terms. We define several desirable retrieval constraints for semantic term matching and use such constraints to extend the axiomatic model to directly support semantic term matching based on the mutual information of terms computed on some document set. We show that such extension can be efficiently implemented as query expansion. Experiment results on several representative data sets show that, with mutual information computed over the documents in either the target collection for retrieval or an external collection such as the Web, our semantic expansion consistently and substantially improves retrieval accuracy over the baseline axiomatic retrieval model. As a pseudo feedback method, our method also outperforms a state-of-the-art language modeling feedback method.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models
General Terms: Experimentation
Keywords: Axiomatic model, retrieval heuristics, constraints, query expansion, feedback

1. INTRODUCTION

The axiomatic approach to information retrieval was proposed recently as a new retrieval framework, in which relevance is modeled by term-based retrieval constraints [5, 6]. Several new retrieval functions have been derived by using this approach and shown to be less sensitive to parameter setting than existing retrieval functions with comparable optimal performance; using a fixed parameter value can often achieve near-optimal performance in most test sets [6].

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

However, like most traditional retrieval models, these new axiomatic retrieval functions also have the limitation that retrieval scores are solely based on *exact* (i.e., syntactic) matching of terms in the query and documents, without allowing distinct but semantically related terms to match each other and contribute to the retrieval score. Since it is unlikely that the authors of relevant documents always use exactly the same terms as a user would use in a query, such a limitation makes the retrieval performance of existing models non-optimal. For example, given the query "car", intuitively, a single-term document with the term "vehicle" should have a higher score than a single-term document with the term "fish" because "car" is semantically more related to "vehicle" than "fish". However, existing retrieval models do not *directly* support such semantic matching and would treat both documents equally as not matching the query. Although techniques such as query expansion and pseudo-relevance feedback can support semantic term matching to certain extent, query expansion purely based on semantic relations between terms have so far not been very successful, presumably because of the difficulty in assigning appropriate weights to the new terms [26, 14]. Pseudo feedback is much more successful, but the semantic matching exploited is restricted by the top-ranked documents and it does not allow us to incorporate external resources.

In this paper, we show that there is a natural way to incorporate semantic term matching to the inductively defined axiomatic retrieval functions – all we need to do is to generalize the primitive weighting function to incorporate the semantic similarity of terms. Specifically, we follow the spirit of axiomatic approaches [5, 6] and formally define several constraints on semantic term matching. We then use these constraints to provide guidance on how to compute the term semantic similarity and how to regularize the weights of the original terms and the semantically related terms. We show that our method can be implemented as query expansion in the axiomatic framework, and when term similarity is computed using feedback documents, it can also be regarded as a method for pseudo feedback in the axiomatic approaches. We conduct experiments over several representative data sets. The results show that the proposed semantic expansion works well for all the six inductively defined axiomatic retrieval functions, significantly improving the retrieval accuracy over the original baseline axiomatic retrieval functions on all the data sets we experimented with. Moreover, the analysis of semantic term matching constraints can predict parameter boundaries that are consistent with the empirically discovered optimal ranges of parameters. Fur-

thermore, as a pseudo feedback method, our method outperforms a state-of-the-art language modeling approach for pseudo feedback [30] due to its capability of selecting better terms for query expansion.

The rest of the paper is organized as follows. We discuss related work in Section 2 and briefly review the existing axiomatic approach to IR in Section 3. We then present our work on incorporating semantic term matching to the axiomatic framework in Section 4, and discuss experiment results in Section 5. Finally, we conclude in Section 6.

2. RELATED WORK

Many studies have tried to bridge the vocabulary gap between documents and queries in traditional retrieval models, mostly based on either co-occurrence-based thesaurus [11, 24, 18, 20, 10, 29, 23, 2] or hand-crafted thesaurus [26, 12]. Some researchers used both [14, 4]. Although our general strategy would be applicable to exploit both types of thesauri, in this paper, we focus on the use of co-occurrence-based thesaurus and leave other possibilities as future work.

The earliest study of co-occurrence-based thesaurus can be traced back to the early sixties. Lesk [11] studied term expansion in the vector space model, where term similarity is computed based on the cosine coefficient [22]. Smeaton et al. [24] studied query expansion based on classical probabilistic model. These previous studies suggested that query expansion based on term co-occurrences is unlikely to significantly improve performance [18]. Qiu et al. [20] showed that adding terms that have the greatest similarity to the entire query, rather than individual terms, can obtain more improvement. Xu et al. [29] showed that the analysis of word occurrences and relationships on a local set of documents (i.e. the top ranked documents retrieved by the original query) yields better performance than on the whole corpus. In language modeling approaches, Berger et al. [3] proposed a translation model to incorporate term relationship into language modeling approaches. Cao et al. [4] extended the translation model to integrate both co-occurrence and hand-crafted thesaurus and achieve reasonable performance improvement. Bai et al. [2] showed that query expansion based on co-occurrences can improve the performance in language modeling approaches.

Although the motivation is similar, our work differs from the previous work in that (1) we attempt to integrate term semantic relationship as a component in our retrieval model; (2) we take an axiomatic approach and define constraints to guide us in the incorporation of semantic term matching. Similar to previous work [11, 24, 18, 20, 23, 2], our method can also be implemented as query expansion. Thus, when we compute term similarity based on the documents in the collection, it bears some similarity to traditional feedback methods [21, 30, 19, 16], which also select terms from documents as to expand the query. But our method selects terms that are semantically related to each individual query term and relies on the axiomatic approaches to combine them, while feedback methods select terms that discriminate the feedback documents, which are not necessarily related to any individual query term. Because of this difference, our method is complementary to the traditional feedback method. Indeed, our experiment results show that they can be combined to further improve performance.

3. AXIOMATIC RETRIEVAL MODEL

The basic idea of the axiomatic approach to information retrieval is to search in a space of candidate retrieval functions for one that can satisfy a set of reasonable retrieval constraints; the assumption is that if a retrieval function satisfies all the desirable retrieval constraints, it would likely be effective empirically [5, 6]. Compared with other retrieval models, this approach has the advantage of connecting relevance more directly with terms through formalized retrieval constraints.

In [6], several interesting new retrieval functions have been derived using formalized retrieval constraints and an inductive decomposition of the function space. These new functions are shown to perform as well as traditional retrieval functions but with much more robust parameter setting. The inductive definition decomposes a retrieval function into three component functions: primitive weighting function, document growth function and query growth function.

The primitive weighting function gives the score of a one-term document $\{d\}$ and a one-term query $\{q\}$. It is usually instantiated as

$$S(\{q\},\{d\}) = \begin{cases} \omega(q) & q = d \\ 0 & q \neq d \end{cases} \quad (1)$$

where $\omega(q)$ is an IDF-like function of q [6].

The query growth function describes the score change when we add a term to a query and is instantiated as

$$S(Q \cup \{q\}, D) = S(Q, D) + S(\{q\}, D).$$

The document growth function describes the score change when we add a term to a document, and is instantiated based on some existing retrieval functions. The instantiation corresponding to Okapi [1] is

$$\begin{aligned} S(Q, D \cup \{d\}) \quad = \quad & \sum_{t \in D \cup Q - \{d\}} S(Q,\{t\})\lambda(|D|+1, c(t,D)) \\ & + S(Q,\{d\}) \cdot \lambda(|D|+1, c(d,D)+1). \end{aligned}$$

where $\lambda(x,y) = \frac{y}{\frac{b}{avdl}x + b + y}$, $0 \leq b \leq 1$, $|D|$ is document length, and $c(t,D)$ is the count of term t in D.

In general, a different instantiation of these component functions would result in a different retrieval function. In [6], several such inductively defined axiomatic retrieval functions are derived, and they are all shown to be effective. The following function (F2-EXP) is one of the best performing functions, which will be used in this paper as an example axiomatic retrieval function to illustrate how we can incorporate semantic term matching; however, the proposed method can be easily applied to all the other derived functions.

$$S(Q,D) = \sum_{t \in Q \cap D} c(t,Q) \cdot \left(\frac{N}{df(t)}\right)^{0.35} \cdot \frac{c(t,D)}{c(t,D) + b + \frac{b \cdot |D|}{avdl}}$$

4. INCORPORATING SEMANTIC TERM MATCHING

In this section, we show how we can naturally incorporate semantic term matching into the inductively defined axiomatic retrieval model proposed in [6]. Following the spirit of axiomatic approaches, we first define three constraints on semantic term matching.

[1] The instantiation of the document growth function is more general than the one given in [6], which is $S(Q, D \cup \{d\}) = \sum_{t \in D \cap Q - \{d\}} S(Q,\{t\})\lambda(|D|+1, c(t,D)) + S(Q,\{d\}) \cdot \lambda(|D|+1, c(d,D)+1)$. Given $q \neq d$, $S(\{q\},\{d\}) = 0$, these two instantiations are equivalent.

4.1 Semantic Term Matching Constraints

Let $s(t,u) \in [0,+\infty]$ be any given semantic similarity function between two terms t and u. Without loss of generality, we assume that term t is semantically more similar to term u than to term v if and only if $s(t,u) > s(t,v)$, i.e., a large value of s indicates a high similarity. Since intuitively a term has the highest similarity to itself, we assume $\forall u \neq t, s(t,t) > s(t,u)$. We also assume that s is symmetric, i.e., $\forall t,u, s(t,u) = s(u,t)$. Based on such a semantic similarity function, we now define three constraints that we would like any reasonable retrieval function to satisfy.

STMC1: Let $Q = \{q\}$ be a query with only one term q. Let $D_1 = \{d_1\}$ and $D_2 = \{d_2\}$ be two single-term documents, where $q \neq d_1$ and $q \neq d_2$. If $s(q,d_1) > s(q,d_2)$, then $S(Q,D_1) > S(Q,D_2)$.

STMC1 requires a retrieval function to give a higher score to a document with a term that is more semantically related to a query term. Thus, even though D_1 and D_2 do not match the query Q syntactically, we would like D_1 to have a higher score because term d_1 is more semantically related to query term q than term d_2 is. Clearly, STMC1 directly constrains the primitive weighting function.

STMC2: Let $Q = \{q\}$ be a single term query and d be a non-query term such that $s(q,d) > 0$. If D_1 and D_2 are two documents such that $|D_1| = 1$, $c(q,D_1) = 1$, $|D_2| = k$ and $c(d,D_2) = k$ ($k \geq 1$), where $c(q,D_1)$ and $c(d,D_2)$ are the counts of q and d in D_1 and D_2 respectively, then $S(Q,D_1) \geq S(Q,D_2)$.

STMC2 requires that matching an original query term q exactly should always contribute no less to the relevance score than matching a semantically related term d, no matter how many times term d occurs in the document.

STMC3: Let $Q = \{q_1, q_2\}$ be a query with only two query terms and d be a non-query term such that $s(q_2,d) > 0$. Let D_1 and D_2 be two documents. If $|D_1| = |D_2| > 1$, $S(\{q_1\},\{q_1\}) = S(\{q_2\},\{q_2\})$, $c(q_1,D_1) = |D_1|$, $c(q_1,D_2) = |D_2| - 1$ and $c(d,D_2) = 1$, then $S(Q,D_1) \leq S(Q,D_2)$.

STMC3 intends to capture the following intuition: Suppose we have a query with two *equally* important terms q_1 and q_2. Suppose a document D_1 matches q_1 n (> 1) times, but does not match q_2 or any of its semantically related terms. If we change one of the occurrences of q_1 in D_1 to a term semantically related to q_2 to form a document D_2, D_1 should not have a lower score than D_2, because D_2 covers more distinct query terms than D_1.

4.2 Extension based on STMCs

The constraints defined above provide some guidance on how to extend the inductively defined axiomatic retrieval functions to incorporate semantic term matching.

First, it is clear that these existing axiomatic functions violate all the three constraints we defined, simply because the semantic similarity function s is not part of the retrieval function. For example, based on the primitive weighting function shown in Equation (1), any single-term document will be assigned a zero score if the term in the document is not matching exactly the query term, which clearly violates STMC1.

To make the primitive weighting function satisfy STMC1, a natural solution is to define the following *generalized primitive weighting function* based on a given similarity function s.

$$S(\{q\},\{d\}) = \omega(q) \times f(s(q,d)),$$

where f is a monotonically increasing function. Note that it is reasonable to require $\forall q \in Q, f(s(q,q)) = 1$ for any query Q, because the score of generalized primitive weighting function should be comparable with the score of the original one when the two terms match exactly. One way to ensure such property is to define f in terms of normalized similarity.

$$f(s(q,d)) = \frac{s(q,d)}{s(q,q)} \times \lambda(q,d)$$

where

$$\lambda(q,d) = \begin{cases} 1 & q = d \\ \beta & q \neq d \end{cases} \quad (2)$$

β is used to regulate the weighting of the original query terms and the semantically similar terms. The value of β should satisfy $0 < \beta < \frac{s(q,q)}{s(q,d)}$, because $f(s(q,d)) < f(s(q,q)) = 1$) when $d \neq q$.

The generalized primitive weighting function clearly satisfies STMC1, and if we combine it with any existing instantiations of document growth function and query growth function, the derived retrieval functions would also satisfy STMC1 unconditionally. We further analyze STMC2 and STMC3 on such derived functions and find that these constraints are satisfied when β is within a certain range. Specifically, the analysis of STMC2 provides a tighter upper bound for β, while the analysis of STMC3 provides a tighter lower bound. The actual values of these bounds depend on the instantiation of document growth function. As an example, the lower and upper bounds for F2-EXP is:

$$\frac{b}{2+b} \times \frac{s(q,q)}{s(q,d)} \leq \beta \leq \frac{1}{b+1} \times \frac{s(q,q)}{s(q,d)} \quad (3)$$

We see that the bounds of β depend on both the query and semantic similarity function s. In our experiments, on each data set, the lower bound of β is determined by the lowest value of $\frac{s(q,q)}{s(q,d)}$ for all the query terms while the upper bound of β is determined by the highest value of $\frac{s(q,q)}{s(q,d)}$, which are the minimal requirements of β.

Since a term can potentially have a huge number of semantically related terms, the computation of the generalized retrieval functions can be expensive. To reduce the computation cost, we can reasonably restrict our attention to the most similar terms for each query term. Such simplification is not expected to affect the retrieval score significantly, because the dropped terms would contribute little to the score anyway. Thus we redefine the generalized primitive weighting function as follows:

$$S_{gen}(\{q\},\{d\}) = \begin{cases} \omega(q) \cdot \frac{s(q,d)}{s(q,q)} \cdot \lambda(q,d) & d \in \varepsilon(q) \\ 0 & otherwise \end{cases} \quad (4)$$

where $\varepsilon(q)$ is the set of K most semantically similar terms of q according to the similarity function s, $\omega(q)$ is as in Equation (1) and $\lambda(q,d)$ is defined in Equation(2).

Even with this simplification, the computation can still potentially involve enumerating all the combinations of query terms and document terms. Fortunately, there is an efficient way to compute such a retrieval function based on query expansion as shown in the next section.

4.3 As Query Expansion

Let us first introduce some notations. $S(Q,D)$ is the scoring function of the original inductively defined axiomatic re-

trieval function, where only syntactic term matching is considered. $S_{gen}(Q, D)$ is the generalized inductively defined axiomatic retrieval function obtained by combining the generalized primitive weighting function with the original document growth and query growth function.

The generalized primitive weighting function (i.e., Equation (4)) can be re-written as follows.

$$S_{gen}(\{q\},\{d\}) = \begin{cases} \omega(q) & d = q \\ \omega(q:d) & d \in \varepsilon(q)/\{q\} \\ 0 & otherwise \end{cases}$$

where $\omega(q:d) = \omega(q) \times \beta \times \frac{s(q,d)}{s(q,q)}$.

Let $\varepsilon'(q)$ be the set of K most semantically similar terms of q excluding itself, i.e., $\varepsilon'(q) = \varepsilon(q)/\{q\}$. Let P be the set of the K most similar terms of all query terms, i.e., $P = \bigcup_{q \in Q}(\varepsilon'(q))$. $\forall t \in P$, let $\rho(t)$ be the set of query terms that are semantically similar to t. Define S' such that $\forall t \in P, S'(\{t\},\{t\}) = \omega(\rho(t):t) = \frac{\sum_{u \in \rho(t)} \omega(u:t)}{|Q|}$; otherwise $S'(Q, D) = S(Q, D)$.

Theorem: $\forall Q, D$, $S_{gen}(Q, D) = S'(Q \cup P, D)$.

Proof:

$$\begin{aligned} S_{gen}(Q,D) &= \sum_{q \in Q} S_{gen}(q,D) \\ &= \sum_{q \in Q}(S_{gen}(q,D_q) + \sum_{t \in \varepsilon'(q) \cap D} S_{gen}(t,D_t)) \\ &= \sum_{q \in Q}(S(q,D_q) + \sum_{t \in \varepsilon'(q) \cap D} S'(t,D_t)) \\ &= S(Q,D) + \sum_{q \in Q} \sum_{t \in \varepsilon'(q) \cap D} S'(t,D_t) \\ &= S(Q,D) + \sum_{t \in P} S'(t,D_t) \\ &= S'(Q,D) + \sum_{t \in P} S'(t,D) \\ &= S'(Q \cup P, D) \end{aligned}$$

where D_t is the part of the document D that only contains t. (i.e., $|D_t| = c(t, D) = c(t, D_t)$).

The first step is based on query growth function. The second step assumes that the relevance score of a document can be computed as the sum of the disjoint subsets of the document, which holds for all the inductively defined axiomatic retrieval functions. The third step is based on the fact that S_{gen} and S' use the same document growth function and the fact that $S'(\{t\},\{t\}) = \omega(\rho(t):t)$ is consistent with generalized primitive function when $t \in P$.

The theorem shows that scoring a document using S_{gen} can be reduced to scoring using S' with an expanded query formed by adding, for each query term, K most similar terms to the query. Note that the weight of a similar term t is computed from $\omega(\rho(t) : t)$ instead of $\omega(t)$ as used in the traditional query expansion methods.

4.4 Term Semantic Similarity Function

The remaining challenge is to define $s(t_1, t_2)$ in STMC1. In general, we may exploit any knowledge and resources available to us to compute term similarity and there are many ways to compute it. For example, co-occurrences of terms obtained from the analysis of a document collection usually reflect underlying semantic relationships that exist between terms [23, 3, 4, 2], and we may use measures such as Dice similarity [1] and mutual information [25, 15, 9, 8, 13, 7] to compute term similarity. In this paper, we adopt the mutual information as the basic semantic similarity metric, leaving other choices for future work.

The mutual information (MI) of two terms t and u in a set of documents can be computed as follows [25]:

$$I(X_t, X_u) = \sum_{X_t, X_u \in \{0,1\}} p(X_t, X_u) \log \frac{p(X_t, X_u)}{p(X_t)p(X_u)}$$

X_t and X_u are two binary random variables corresponding to the presence/absence of term t and term u in each document or segment.

Mutual information is a principled way to measure term correlations, and it satisfies our requirements about the similarity function s. The next choice we have to make is which corpus to use when computing the mutual information. A natural choice would be the document collection from which we retrieve documents. However, such a choice may not be ideal because an ambiguous term can have multiple senses in a large corpus. As a result, the semantically related terms found by mutual information could be a mix of terms corresponding to different senses of the original term, introducing noise in query expansion. Thus, it is crucial to compute mutual information over a "clean" corpus, where ideally only one (correct) sense of the query term occurs. How can we find such a "clean" corpus? One possibility is to use the top-M documents returned by the retrieval systems for the query. The rational is that we can reasonably assume there is only one sense of a query term in the set of relevant documents, and the top-M documents are reasonable approximations of the set of relevant documents. This is indeed in line with what previous work in query expansion has found – local document analysis tends to be more effective than global document analysis [29].

However, the top-M documents would clearly be a *biased* corpus, and in this sense, it is not a good corpus for computing mutual information. For example, it is likely that a query term occurs in all the top-M documents. The abundance of a query term would then cause popular terms in the top-M documents to generally have a high mutual information. In particular, a common term (e.g., "can") would have a high mutual information, even if it also occurs in many other documents where the query term does not occur. To solve this problem, we need to supplement the top-M documents with some additional documents that do not necessarily contain any query term. Thus we will randomly choose $r \times M$ documents from the collection and combine them with the top-M documents as a mixed corpus for computing mutual information.

Clearly, the choice of r may also affect the mutual information results. How do we choose a good value for r? Once again, constraint analysis can provide some guidance. The following notations will be used in defining the constraints: N is the total number of documents in the document collection. $df(t)$ is the number of documents that contain t in the collection. W is the working set containing $r \times M$ random documents plus the top M documents returned by the system; since the $r \times M$ documents are chosen from the documents ranked below the top-M documents, we clearly have $M + M \times r \leq N$. $df(t_1, t_2|W)$ is the number of documents that contain both t_1 and t_2 in the working set W. $df(t|W)$ is the number of documents that contain t in the working set W.

Intuitively, the value of r should not be very small, because we need enough number of random documents to penalize the common terms. Consider the scenario in Figure 1(a), where t_1 is a "truly" semantically related term, while t_2 is a common term. t_1 is semantically more similar to q than t_2, although t_2 co-occurs with q in more documents than t_1. This intuition can be captured by the following Term Semantic Similarity Constraint(TSSC).

TSSC1: Let q be a query term and t_1 and t_2 be two non-query terms. If $df(q,t_1|W) = \frac{M}{2}$, $df(t_1|W) = \frac{M}{2}$, $df(q|W) = M$, $df(q,t_2|W) = M$, $df(t_2|W) = M + \frac{r \times M}{2}$, then $s(q,t_1) > s(q,t_2)$.

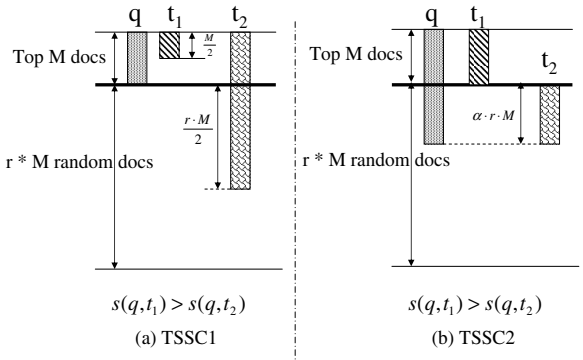

Figure 1: TSSC

On the other hand, the value of r should not be very large because we want to ensure that the dominant sense of a query term is the one determined by the whole query. Consider the scenario in Figure 1(b). Suppose a query term q has two senses. The first sense is the one determined by the whole query (i.e., in the top M documents), and a term t_1 is semantically related to this sense of q (i.e., they co-occur in the top M documents). Now suppose another term t_2 is semantically related to another sense of q (i.e., they co-occur in the random documents). Intuitively, t_1 should have a higher similarity score than t_2. The following constraint captures this intuition.

TSSC2: Let q be a query term and t_1 and t_2 be two non-query terms. If $0 < \alpha < 1$, $df(q,t_1|W) = M$, $df(t_1|W) = M$, $df(q|W) = M + \alpha \times r \times M$ and $df(t_2|W) = df(q,t_2|W) = \alpha \times r \times M$, then $s(q,t_1) > s(q,t_2)$.

α is the percentage of the documents that contain q in a random sample of the whole collection after the top M documents excluded, i.e., $\alpha = \frac{df(q)-M}{N-M}$.

The above two constraints are satisfied only when the value of r is within a certain range. Indeed, TSSCs provide a lower and an upper bounds for r.

$$1 < r < \frac{N}{df(q)} \tag{5}$$

The value of r is collection and query dependent. For each collection, we use the median of the document frequency of all query terms to compute the upper bound of r.

4.5 Summary

We briefly summarize the high-level steps involved in the proposed method for incorporating semantic term matching:

1. Construct a working set where term semantic similarity can be computed.
2. For every query term, find the top L most similar terms based on the working set.
3. Gather the top L similar terms for all the query terms, then select the top K ranked terms based on $\omega(\rho(t) : t)$.
4. Expand the original query with the K terms. Note that the weight of an expansion term is computed based on $\omega(\rho(t) : t)$ instead of $\omega(t)$.

In the first step, the working set can be constructed over any reasonable resources in the following way: Given any collection of documents and a query, we first use the original inductively defined axiomatic retrieval function to rank the documents. We then merge the top M returned documents with $r \times M$ random documents selected from the same collection to form a working set for computing term similarity. The collection to be used can be either the target collection for retrieval (called *internal expansion*) or any other collections (called *external expansion*). To form a large pool of terms, L is usually fixed to 1000. Four parameters need to be tuned: the number of expansion terms (i.e., K), the number of top documents (i.e., M), the number of random documents (i.e., r) and the scaling parameter β. The optimal values of β and r are expected to be within a certain range based on Equation (3) and Equation(5), which is also supported by our experiment results.

5. EXPERIMENTS

5.1 Experiment Design

We conduct three sets of experiments. First, we evaluate the effectiveness of the semantic term matching. Second, we examine the parameter sensitivity of the method. Finally, we compare it with a model-based feedback method in language modeling approaches [30].

All experiments are conducted over two collections used in recent Robust track [28, 27]: (1) TREC Disk 4&5 (minus Congressional Record) with 249 official topics of Robust track in 2004. The document set has 1908MB text and 528,000 documents. This is labeled as "ROBUST04". (2)AQUAINT data with 50 official topics of Robust track in 2005. The document set has 3GB text and 1,033,461 documents. This is labeled as "ROBUST05". Some experiments are also conducted over six other data sets used in the previous studies [5, 6, 31]: news articles (AP88-89), technique reports (DOE), government documents (FR88-89), Web data (WEB2g), and the ad hoc data in TREC (TREC7 and TREC8). In all the experiments, we use the title-only queries, because short keyword query is the most frequently used query type by web users and semantic term matching is necessary for such short queries.

The performance is measured using the official measures in Robust track: MAP (mean average precision) and gMAP (geometric mean average precision). gMAP [27, 28] is a variant of the traditional MAP measure that uses a geometric mean rather than an arithmetic mean. This measure emphasizes the performance of poorly-performing topics.

The preprocessing only involves stemming with Porter's stemmer. As pointed out in the previous work [6], using a fixed parameter value ($b = 0.5$), F2-EXP can often achieve near-optimal performance in many test sets. Thus, we fix b to 0.5 in our experiments. We use the optimal value of b for the other five inductively defined axiomatic retrieval functions. In the first and third sets of experiments, M and K are both fixed to 20 and r is fixed to 29, so that we will get a total of 600 documents in the working set. We

Table 1: Performance of different axiomatic functions.

Method		ROBUST04		ROBUST05	
		MAP	gMAP	MAP	gMAP
F1-LOG	BL	0.241	0.138	0.200	0.131
	docAX	0.261	0.150	0.241	0.126
		8.3%‡	**8.7%**‡	**21%**‡	**-3.8%**‡
	segAX	0.267	0.148	0.256	0.134
		11%‡	**7.3%**‡	**28%**‡	**2.3%**‡
F1-EXP	BL	0.240	0.137	0.199	0.128
	docAX	0.262	0.150	0.246	0.126
		9.2%‡	**9.5%**‡	**24%**‡	**-2.4%**‡
	segAX	0.266	0.148	0.252	0.128
		11%‡	**8.0%**‡	**27%**‡	**0.0%**
F2-LOG	BL	0.251	0.141	0.196	0.125
	docAX	0.278	0.157	0.270	0.131
		11%‡	**11%**‡	**38%**‡	**4.8%**‡
	segAX	0.284	0.156	0.281	0.135
		13%‡	**11%**‡	**43%**‡	**8.0%**‡
F2-EXP	BL	0.248	0.142	0.192	0.122
	docAX	0.285	0.157	0.258	0.136
		15%‡	**11%**‡	**34%**‡	**11%**‡
	segAX	0.288	0.158	0.267	0.137
		16%‡	**11%**‡	**39%**‡	**12%**‡
F3-LOG	BL	0.240	0.138	0.200	0.131
	docAX	0.259	0.146	0.241	0.138
		7.9%‡	**5.8%**‡	**21%**‡	**5.3%**‡
	segAX	0.267	0.149	0.253	0.131
		11%‡	**7.9%**‡	**27%**‡	**0.0%**
F3-EXP	BL	0.239	0.137	0.198	0.127
	docAX	0.261	0.150	0.244	0.125
		9.2%‡	**9.5%**‡	**23%**‡	**-1.6%**‡
	segAX	0.265	0.148	0.254	0.130
		11%‡	**8.0%**‡	**28%**‡	**2.4%**‡

tune the value of β and report the best performance unless otherwise stated. **BL** is the baseline method without expansion (i.e., without semantic term matching). **docAX** and **segAX** are semantic expansion methods with MI computed based on co-occurrences in documents and 100-word segments, respectively. In all the result tables, ‡ and † indicate that the improvement is statistically significant according to Wilcoxin signed rank test at the level of 0.05 and 0.1 respectively.

5.2 Effectiveness of Semantic Term Matching

Table 1 shows the performance of the internal expansion for all six functions. The semantic term matching consistently and significantly outperforms the baseline on both data sets in terms of MAP. But, gMAP decreases in a few cases, which indicates that most of the performance improvement comes from the easy topics. F2-EXP is the best of all the functions. We further test the semantic expansion on top of F2-EXP on six other data sets, and found that semantic expansion outperforms the baseline (i.e., F2-EXP) significantly (Table 2) on all the data sets except FR88-89. Due to the limit of space, we only report the performance of F2-EXP in the remaining experiments.

Table 3 shows the performance when semantic similarity is computed over the internal resource (i.e., collection itself), the external resource (i.e., a pool of Google snippets returned for a query), and both (i.e., first use external expansion, then do another round of internal expansion). We make the following observations. First, the expansion method improves the performance significantly in all cases. Second, the web-based external expansion method is consistently more effective than the internal expansion method in both measures. This indicates that the use of good external resources improves the effectiveness especially over the poorly-performing topics, which is consistent with what others have observed [27]. Finally, combining both internal and external expansion further improves the accuracy.

Table 2: Performance of F2-EXP on more data sets.

Data	MAP			gMAP		
	BL	docAX	segAX	BL	docAX	segAX
TREC7	0.186	0.236	0.247	0.083	0.098	0.098
		27%‡	**33%**‡		**18%**‡	**18%**‡
TREC8	0.250	0.277	0.278	0.147	0.172	0.167
		11%‡	**11%**‡		**17%**‡	**14%**‡
WEB2g	0.282	0.324	0.324	0.188	0.220	0.220
		15%‡	**15%**‡		**17%**‡	**17%**‡
FR88-89	0.217	0.227	0.224	0.058	0.062	0.069
		4.6%	**3.2%**		**6.9%**	**19%**
AP88-89	0.220	0.266	0.267	0.074	0.088	0.086
		21%‡	**21%**‡		**19%**‡	**16%**‡
DOE	0.174	0.186	0.184	0.069	0.078	0.074
		6.9%‡	**5.8%**‡		**13%**‡	**7.3%**‡

Table 3: Performance when using different resources.

Method		ROBUST04		ROBUST05	
		MAP	gMAP	MAP	gMAP
BL		0.248	0.142	0.192	0.122
Internal Expansion	docAX	0.285	0.157	0.258	0.136
		15%‡	**11%**‡	**34%**‡	**11%**‡
	segAX	0.288	0.158	0.267	0.137
		16%	**11%**‡	**39%**‡	**12%**‡
External Expansion		0.300	0.196	0.270	0.196
		21%‡	**38%**‡	**41%**‡	**61%**‡
External + Internal Expansion	docAX	0.300	0.178	0.289	0.203
		21%‡	**25%**‡	**51%**‡	**66%**‡
	segAX	0.302	0.175	0.290	0.198
		22%‡	**23%**‡	**51%**‡	**62%**‡

5.3 Sensitivity Analysis

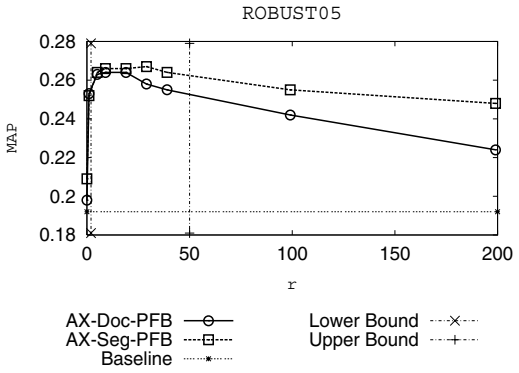

Figure 2: Performance Sensitivity (r)

Next, we study the performance sensitivity for the four parameters in the semantic expansion. Here we only show plots on ROBUST05, but similar trends can be observed for all the other data sets. Figure 2 shows the sensitivity curve for r. Equation (3) gives $1 < r < 50$ for the ROBUST05 data set. The performance is relatively stable when r is within the range, while it decreases when r is out of the range. Figure 3 shows the sensitivity curve for β. Equation (5) gives $0.27 \leq \beta \leq 3.8$ for the ROBUST05 data set. The optimal value is indeed within the predicted range, although docAX and segAX have different optimal value of β. Figure 4 shows the sensitivity curve for K. The performance is near optimal when K is 20. The performance is relatively stable when more terms are added. Figure 5 shows the curve for M.

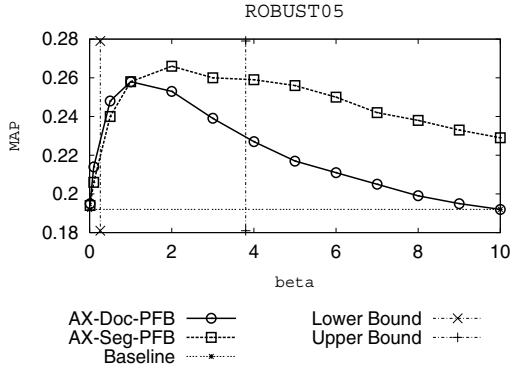

Figure 3: Performance Sensitivity (β)

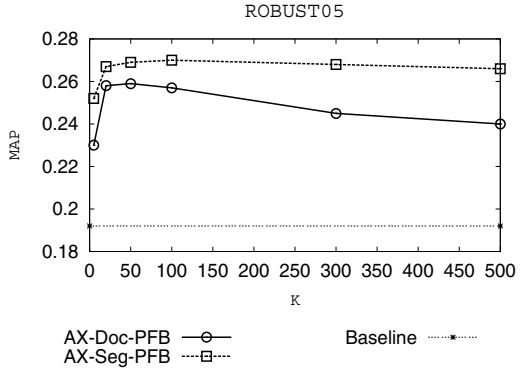

Figure 4: Performance Sensitivity (Num of Terms)

We observe the performance is optimal when M is around 20. The performance decreases when more documents are used, likely because the assumption that top M documents are all relevant is not true for larger values of M.

5.4 Comparison with Feedback Methods

Both our semantic expansion and traditional feedback methods select terms for query expansion. Traditional feedback methods [21, 30] select terms that have higher weight in the feedback documents, while our method selects terms that are semantically related to any query term. It would be interesting to compare their performance. In Table 4, we report the performance of the model-based feedback method in language modeling approaches [30]. Internal PFB (IPFB) is the pseudo feedback method. External PFB (EPFB) is the feedback method where the feedback terms are obtained from the Google snippets. We set μ to the optimal value for each data set, the number of feedback terms to 20 and the number of documents to 20. We tune the value of feedback coefficient and the value of mixture noise [30] and report the best performance. Comparing Tables 3 and 4 shows that

Table 4: LM Feedback & Additive Effect

Method	ROBUST04		ROBUST05	
	MAP	gMAP	MAP	gMAP
BL	0.251	0.140	0.196	0.131
Internal PFB	0.275	0.139	0.254	0.105
IPFB + **docAX**	0.284	0.151	0.269	0.133
	3.3%‡	**8.6%**‡	**5.9%**†	**27%**†
IPFB + **segAX**	0.283	0.144	0.280	0.138
	2.9%†	**3.6%**†	**10%**‡	**31%**‡
External PFB	0.282	0.172	0.226	0.156
EPFB + **docAX**	0.293	0.170	0.278	0.168
	3.9%‡	**-1.2%**‡	**23%**†	**7.7%**†
EPFB + **segAX**	0.293	0.168	0.279	0.166
	3.9%‡	**-2.3%**‡	**24%**‡	**6.4%**‡

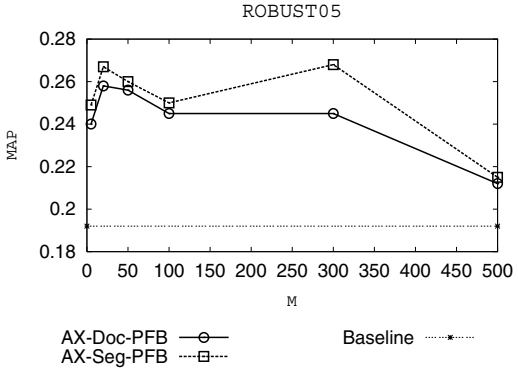

Figure 5: Performance Sensitivity (Num of Docs)

Table 5: Performance (MAP) of term selection (segAX)

Data	Weighting Function	Term Selection	
		KL-Div.	F2-EXP
ROBUST04	KL-Div.	0.275	0.288 (4.72%)‡
	F2-EXP	0.285	0.288 (1.05%)
ROBUST05	KL-Div.	0.254	0.273 (7.48%)‡
	F2-EXP	0.265	0.267 (0.755%)

the expansion method in axiomatic framework outperforms the model-based feedback method for both internal and external feedback. More interestingly, as shown in Table 4, our method can be combined with the traditional feedback methods to further improve performance, which shows that our method is complementary with the traditional feedback method.

Finally, we design experiments to study whether the performance gain of the semantic expansion comes from better term selection or from better term weighting. Assume A and B can be either our expansion method (i.e., F2-EXP) or traditional method (i.e., KL-Div.). We use method A to select terms for the method B, which means that we exclude any terms that are not nominated by A when using B. However, these terms are still weighted using B. This way we have four combinations shown in Table 5. The performance of using the terms selected by F2-EXP is consistently better than that of using the terms selected by KL-divergence method. For example, on ROBUST05 data set, the performance of KL-divergence method can be improved from 0.254 to 0.273 by using the terms selected by our method. The results indicate that the performance improvement of our method clearly comes more from better term selection.

6. CONCLUSION AND FUTURE WORKS

In this paper, we propose a natural way to incorporate semantic term matching into axiomatic retrieval models. Following the previous work in axiomatic retrieval, several retrieval constraints are defined to capture intuitions on semantic term matching. The advantage of this method is that the constraints provide us guidance on the parameter setting and on the choice of term semantic similarity measure. Our method can be efficiently implemented as a query expansion method in the axiomatic framework.

The expansion based on semantic term matching was evaluated on several representative large retrieval collections. The results show that our method is effective for all the six inductively defined axiomatic retrieval functions. Furthermore, our method works for both internal resources (e.g. collection itself) and external resources (e.g. the results re-

turned by Google). The parameter sensitivity confirms the hypothesis that the constraint analysis can provide an upper bound and a lower bound for the optimal values of r and β. The performance is relatively stable when the values of the parameters are set within the range derived from the constraint analysis. As query expansion, our method outperforms the model-based feedback method in language modeling approach and is shown to be complementary to the traditional feedback methods and can be combined with them to further improve performance.

There are many interesting future research directions. First, we can use more resources, such as WordNet [4, 26, 12, 14, 17], to compute term semantic similarity. Second, our method can also be applied to cross-lingual retrieval task. Finally, the term similarity between query terms are ignored in our work. It would be interesting to study the expansion based on query concepts instead of individual query term, which is along the line of [16, 20].

7. ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under award number IIS-0347933. We thank the anonymous SIGIR reviewers for their useful comments.

8. REFERENCES

[1] M. Adriani. Using statistical term similarity for sense disambiguation in cross-language information retrieval. *Information Retrieval*, 2:69–80, 2000.

[2] J. Bai, D. Song, P. Bruza, J.-Y. Nie, and G. Cao. Query expansion using term relationships in language models for information retrieval. In *Fourteenth International Conference on Information and Knowledge Management (CIKM 2005)*, 2005.

[3] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, 1999.

[4] G. Cao, J.-Y. Nie, and J. Bai. Integrating word relationships into language models. In *Proceedings of the 2005 ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[5] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 2004 ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[6] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 2005 ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[7] J. Gao, J.-Y. Nie, H. He, W. Chen, and M. Zhou. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

[8] J. Gao, J.-Y. Nie, E. Xun, J. Zhang, M. Zhou, and C. Huang. Improving query translation for cross-language information retrieval using statistical models. In *Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[9] M.-G. Jang, S. H. Myaeng, and S. Y. Park. Using mutual information to resolve query translation ambiguities and query term weighting. In *Proceedings of the 37th annual meeting of the association for computational linguistics*, 1999.

[10] Y. Jing and W. B. Croft. An association thesaurus for information retreival. In *Proceedings of RIAO*, 1994.

[11] M. Lesk. Word-word associations in document retrieval systems. *American Documentation*, 20:27–38, 1969.

[12] S. Liu, F. Liu, C. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proceedings of the 2004 ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[13] A. Maeda, F. Sadat, M. Yoshikawa, and S. Uemura. Query term disambiguation for web cross-language information retrieval using a search engine. In *Proceedings of the fifth international workshop on information retrieval with Asian languages*, 2000.

[14] R. Mandala, T. Tokunaga, H. Tanaka, A. Okumura, and K. Satoh. Ad hoc retrieval experiments using wordnet and automatically constructed thesauri. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 475–481, 1998.

[15] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7, 1960.

[16] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[17] D. Moldovan and A. Novischi. Lexical chains for question answering. In *Proceedings of the 19th International Conference on Computational linguistics*, 2002.

[18] H. J. Peat and P. Willett. The limitations of term co-occurence data for query expansion in document retrieval systems. *Journal of the american society for information science*, 42(5):378–383, 1991.

[19] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR'98*, pages 275–281, 1998.

[20] Y. Qiu and H. Frei. Concept based query expansion. In *Proceedings of the 1993 ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993.

[21] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall Inc., 1971.

[22] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[23] H. Schutze and J. O. Pedersen. A co-occurrence based thesaurus and two applications to information retrieval. *Information Processing and Management*, 33(3):307–318, 1997.

[24] A. F. Smeaton and C. J. van Rijsbergen. The retrieval effects of query expansion on a feedback document retrieval system. *The Computer Journal*, 26(3):239–246, 1983.

[25] C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[26] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 1994 ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[27] E. M. Voorhees. Overview of the trec 2004 robust retrieval track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC2004)*, 2005.

[28] E. M. Voorhees. Overview of the trec 2005 robust retrieval track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC2005)*, 2006.

[29] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 1996 ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[30] C. Zhai and J. Lafferty. Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 403–410, 2001.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, pages 334–342, Sept 2001.