Diversifying Query Suggestions Based on Query Documents

Youngho Kim
University of Massachusetts Amherst
yhkim@cs.umass.edu

W. Bruce Croft
University of Massachusetts Amherst
croft@cs.umass.edu

ABSTRACT
Many domain-specific search tasks are initiated by document-length queries, e.g., patent invalidity search aims to find prior art related to a new (query) patent. We call this type of search Query Document Search. In this type of search, the initial query document is typically long and contains diverse aspects (or sub-topics). Users tend to issue many queries based on the initial document to retrieve relevant documents. To help users in this situation, we propose a method to suggest diverse queries that can cover multiple aspects of the query document. We first identify multiple query aspects and then provide diverse query suggestions that are effective for retrieving relevant documents as well being related to more query aspects. In the experiments, we demonstrate that our approach is effective in comparison to previous query suggestion methods.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – Query Formulation, Search Process.

Keywords
Diversifying query suggestions; Patent retrieval; Citation search

1. INTRODUCTION
Many domain-specific search tasks can start from document-length initial queries. For example, prior-art search aims to find past relevant patents which may conflict with new patents [6][10]; in academic literature search, academic authors need to find relevant papers that should be cited in their writings. One unique characteristic of these search tasks is more emphasis on recall, i.e., not missing relevant documents more important than placing a relevant document at the top rank. In this paper, we call this type of domain-specific search task Query Document Search (QDS). Note that we use the term “query document” to refer to the document-length initial query in domain-specific searches.

Query suggestion (e.g., [11]) can be particularly helpful for QDS. For example, patent examiners use about 15 queries to validate a new patent [10]. In addition, patent engineers have stated that automatic suggestion of search vocabulary is required for patent search systems [1]. Although a number of existing methods (e.g., [2][12]) can be used, these techniques need improvement for QDS and do not consider diversity.

In this paper, to improve query suggestions for QDS, we introduce the concept of diversifying query suggestions based on query documents. Emphasizing diverse query suggestions is important because otherwise the system may suggest multiple similar queries which would produce near-duplicate search results. In addition, diversified suggestions can help to retrieve more relevant documents related to a query document. Typically a query document can be quite long (e.g., a patent document can contain thousands of terms) and would include several aspects (or sub-topics). So, many relevant documents are related to these different aspects, and suggesting queries related to multiple aspects can be effective for retrieving more relevant documents. As an example, Figure 1 shows an example query document. This query document is a United States patent, published in 2002, which describes Information Retrieval (IR) systems using multiple databases. The patent application mentions several components (or aspects) such as “query specification”, “query execution”, “query retrieval result”, etc., and the queries suggested for this patent would be more effective if they can cover such query aspects. In fact, many relevant documents for this patent are related to the aspects. Table 1 lists the relevant documents for the query document in Figure 1. In this example, A and B are related to the aspect “query specification”, whereas C refers to “query execution”. In addition, D describes report systems, which forms another aspect (i.e., “report form”).

Table 1: Relevant Documents for Figure 1.

<table>
<thead>
<tr>
<th>No</th>
<th>Title</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>System for generating structured query language statements and integrating …</td>
<td>query specification</td>
</tr>
<tr>
<td>B</td>
<td>Combining search criteria to form a single search …</td>
<td>query execution</td>
</tr>
<tr>
<td>C</td>
<td>Query language execution on heterogeneous database servers using …</td>
<td>report form</td>
</tr>
<tr>
<td>D</td>
<td>System and method for generating reports from a computer database …</td>
<td></td>
</tr>
</tbody>
</table>

Motivated by these types of examples, we propose a method to suggest diverse queries based on query documents. To solve this, we adopt a three-step process: (Step 1) Query Aspect Identification, (Step 2) Query Generation, and (Step 3) Diversifying Query Suggestions. Given an initial query document, we extract diverse
We now generate training examples as follows. For each query
we extract terms from the query document and use term clustering algorithms
to group the terms if they are topically associated and are also effec-
tively clustered together. In other words, we make clustering algorithms
determine this, we use the following conditions, and an example is
positive if its terms are highly associated and effective for retrieving
relevant documents; otherwise, the term pair is negative. To
determine this, we use the following conditions, and an example is
positive if it satisfies every condition; otherwise the example is
negative.

i) Two terms involve high “retrieval effectiveness” if they have
a high generation probability based on the language model esti-
imated for any relevant document.

ii) Two terms are highly “associated” if their PMI estimated from
any relevant document is greater than a threshold.

For each relevant document, we generate a unigram language model and assume that the top 100 terms ranked by the language
model satisfy the first criteria. For the second constraint, we as-
sume that PMI estimated from a relevant document indicates topi-
cal association effective for retrieving relevant documents.

2. FRAMEWORK

2.1 Query Aspect Identification

The first step is identifying n query aspects by representing a query
as a set of related terms from the query document. We address this by using term clustering methods. Specifically, for a query
document, we extract m distinct terms using their tfidf
weights (stop-words are ignored), and generate m x (m - 1)/2 term pairs (the similarity is undirected). By estimating the similarity for each term pair (ti, tj), we can generate a m-by-m symmetric similarity matrix whose diagonal value is 1. Then, we apply a term clustering algorithm using this matrix for generating n different
term sets. In this paper, we extract 500 terms from each query
document, and use a spectral clustering algorithm. Next, we de-
scribe how to estimate the similarity for (ti, tj).

We define similarity between terms by a mixture of topical relat-
edness (or association) and retrieval effectiveness when terms are clustered together. In other words, we make clustering algorithms
group the terms if they are topically associated and are also effec-
tive for retrieving relevant documents. To achieve this, we intro-
duce the similarity function.

$$\text{Sim}(t_i, t_j) = (1 - \lambda) \cdot T(t_i, t_j) + \lambda \cdot R(t_i, t_j) \tag{1}$$

where ti and tj are a term pair from a query document.

In Eq. (1), T(t_i, t_j) measures topical relatedness between ti and tj, while R(t_i, t_j) estimates retrieval effectiveness. $\lambda$ is a controlling
parameter. For T, we utilize term statistics obtained from the doc-
ument corpus (e.g., Point-wise Mutual Information (PMI)). To estimate R, we leverage the features from query performance
predictors (e.g., query clarity [5], query scope [8], etc.).

Using the features listed in Table 1, we can rewrite Eq. (1) as:

$$\text{Sim}(t_i, t_j) = \sum_k \omega_k \cdot f_k(t_i, t_j) \tag{2}$$

where $f_k$ indicates a feature defined in Table 2 and $\omega_k$ is a weight
of the k-th feature. To predict more accurate similarity, we employ a
supervised learning approach. Given a term pair (ti, tj), a sup-
ervised learner estimates its similarity score by learning an optimal value
of the feature weights ($\omega = (\omega_1, ... , \omega_k)$).

Table 2: Features for Similarity Learning.

<table>
<thead>
<tr>
<th>Category</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topical Relatedness</td>
<td>PMI of (ti, tj) calculated by 8-word windows recognized in all documents in a corpus</td>
</tr>
<tr>
<td></td>
<td>PMI of (ti, tj) measured by titles</td>
</tr>
<tr>
<td></td>
<td>PMI of (ti, tj) calculated by 8-word windows identified in query document</td>
</tr>
<tr>
<td>Retrieval Effectiveness</td>
<td>Query Clarity (QC) [5]</td>
</tr>
<tr>
<td></td>
<td>Query Scope (QS) [8]</td>
</tr>
<tr>
<td></td>
<td>Inverse Document Frequency (IDF)</td>
</tr>
<tr>
<td></td>
<td>Inverse Collection Term Frequency (ICTF)</td>
</tr>
</tbody>
</table>

For each query document, N different term pairs are extracted, and we label each pair as positive or negative, i.e., $L((t_i, t_j)) \in \{0, 1\}$. A term pair is
positive if its terms are highly associated and effective for retrieving
relevant documents; otherwise, the term pair is negative. To
determine this, we use the following conditions, and an example is
positive if it satisfies every condition; otherwise the example is
negative.

i) Two terms involve high “retrieval effectiveness” if they have
a high generation probability based on the language model esti-
imated for any relevant document.

ii) Two terms are highly “associated” if their PMI estimated from
any relevant document is greater than a threshold.

For each relevant document, we generate a unigram language model and assume that the top 100 terms ranked by the language
model satisfy the first criteria. For the second constraint, we as-
sume that PMI estimated from a relevant document indicates topi-
cal association effective for retrieving relevant documents.

2.2 Query Generation

In this step, based on n identified query aspects, we generate que-
ries by exploiting the query generation method proposed in [12].
For each query aspect (i.e., a set of terms), we first retrieve pseu-
do-relevant documents (PRD) obtained by the terms in the aspect;
we use those terms as a query and assume that top k retrieved
documents are pseudo-relevant. In addition, we generate an equal
number of non-relevant documents (NRD) by randomly selecting
another k documents from those ranked below the top k. Then, we
train binary decision trees using PRD and NRD where the terms in
PRD are used as attributes. Once a decision tree is learned, we generate a query by extracting attributes (terms) on a single path
from the root to a positive leaf node (i.e., pseudo-relevance). We
define a query as a list of keywords (e.g.,{battery, charger, cellular,
phone}), and ignore the attributes associated with negation.
See [12] for more details.

2.3 Diverse Query Suggestion

We define diversifying query suggestions as suggesting k queries
that will be effective for finding relevant and novel documents for
a query document. To do this, we exploit the xQuAD diversifica-
tion model proposed in [14] and introduce the following probabilis-
tic query suggestion framework. In this approach, among all
generated queries, we select the queries that are more relevant to
the query document and novel relative to the current suggestion
list. Figure 2 describes this framework.

Given a query document $D_q$ and a list of generated queries $L$, we
iteratively choose the most probable query obtained by:

$$P(q, S|D_q) = \mathbf{P}(q, S|D_q) \tag{3}$$

where $S$ is the list of selected queries to be suggested and $q$ is a

candidate query from $L$.

In Eq. (3), $P(q|D_q)$ denotes the relevance of $q$ to $D_q$, while
$P(q, S|D_q)$ indicates the novelty of $q$ to $S$. That is, these two
probabilities are optimizing relevance and diversity, controlled by
$\lambda$. $P(q|D_q)$ can be computed by $\prod_{t \in q} P_{LM}(t|D_q)$, i.e., the uni-
grain language model estimated from $D_q$, and $P(q, S|D_q)$ can be
estimated using the identified query aspects.

By the set of query aspects $A_q$ we can marginalize $P(q, S|D_q)$ as:

$$P(q, S|D_q) = \sum_{ap \in A_q} P(ap|D_q) \cdot P(q, S|ap) \tag{4}$$

where $ap$ is a query aspect in $A_q$.

In Eq. (4), we consider $P(ap|D_q)$ as an importance of $ap$ for $D_q$, which is estimated by $\prod_{t \in ap} P_{LM}(t|D_q)$.
Following estimation can be given.

We conduct experiments on two domains: the patent and academic domains. For the patent domain, we use the patent corpus provided by [6]. To develop query documents (new patents), we ran a query aspect identification method in [2] which can suggest relevant n-grams without using query logs. We modify this method to fit in our search environments; first we extract all n-grams of order 1, 2, 3, 4, and 5 from pseudo-relevant documents obtained by the BL0, rank them by the correlation between candidate n-grams and the terms in the query document, and suggest the top k ranked n-grams. The other baseline (BL2) is a query suggestion method proposed in [12]. We generate keyword queries by ignoring the terms associated with negation.

**ALGORITHM Diversifying Query Suggestions (DivQS)**

**INPUT:** \( L \) (a list of generated queries), \( k \) (the number of queries to be suggested), \( D_q \) (query document)

**OUTPUT:** \( S \) (a list of query suggestions)

**PROCESS:**
1. \( S \leftarrow \emptyset \)
2. While \( |S| \leq k \) do
3. \( q^* \leftarrow \arg\max (1 - \lambda) \cdot P(q|D_q) + \lambda \cdot P(q,S|D_q) \)
4. \( L \leftarrow L \cup \{q^*\} \)
5. \( S \leftarrow S \cup \{q^*\} \)
6. End While
7. Return \( S \)

**Figure 2:** A framework of Diversifying Query Suggestions.

By assuming that the current candidate query \( q \) is independent of the queries already selected in \( S \), \( P(q,S|ap) \) can be derived as:

\[
P(q,S|ap) = P(q|ap) \cdot P(S|ap)
\]

where \( P(q|ap) \) measures the coverage of \( q \) with respect to \( ap \) and \( P(S|ap) \) provides a measure of novelty to the current suggestion list \( S \) for a given \( ap \). To estimate these probabilities, we utilize retrieval results obtained by \( ap \) and \( S \). Specifically, we assume that a query’s top 100 retrieved documents can represent underlying topics of the query, and \( P(q|ap) \) can be estimated by how much of topics in \( ap \) are covered by \( q \). The equation is given as:

\[
P(q|ap) \approx \frac{|Ret_q \cap Ret_ap|}{|Ret_ap|}
\]

where \( Ret_{ap} \) is the set of the top 100 documents retrieved by \( ap \).

Note that we use the terms in a query aspect as a query. For the estimation of \( P(S|ap) \), we further assume that the queries chosen as suggestions in \( S \) are independent to each other for \( ap \), and the following estimation can be given:

\[
P(S|ap) \approx P(qs_1, qs_2, ..., qs_{n-1}|ap) \approx \prod_{qs \in S} (1 - P(qs|ap))
\]

where \( qs \) is a query in \( S \) and \( P(qs|ap) \approx \frac{|R_{qs} \cap R_{ap}|}{|R_{ap}|} \).

Using the above estimations, we select \( k \) queries as suggestions for each query document.

**3. EXPERIMENTS**

**3.1 Experimental Set-up**

We conduct experiments on two domains: the patent and academic domains. For the patent domain, we use the patent corpus provided by [6]. To develop query documents (new patents), we randomly selected 102 more recent patents, and consider patents cited in each query patent as “relevant”. For the academic domain, we use the ACL Anthology Reference Corpus [3], and randomly select 150 more recent query documents (papers). We regard the articles cited in each query paper as “relevant”. For all query documents, references are hidden, and the sentences containing citations are removed. Queries and documents are stemmed by the Krovetz stemmer. To identify query aspects and generate diverse suggestions, we perform 5-fold cross-validation with random partitioning. For each query suggestion, we use the query likelihood model implemented by Indri [17]. We assume that the searchers only examine the top 100 of every query result since 100 patents are examined on average [10].

**Baselines** For each query document, we generate an initial baseline query (BL0) by the query generation method described in [7]. We use BL0 for evaluating query aspect identification. To evaluate diverse suggestion results, we employ two different baselines for evaluation. The first baseline (BL1) is implemented by the method in [2] which can suggest relevant n-grams without using method in [2] which can suggest relevant n-grams without using

<table>
<thead>
<tr>
<th>Metric \ Method</th>
<th>PAT</th>
<th>ACL</th>
</tr>
</thead>
<tbody>
<tr>
<td>R100</td>
<td>0.1091</td>
<td>-</td>
</tr>
<tr>
<td>Max. R100</td>
<td>-</td>
<td>0.4695*</td>
</tr>
<tr>
<td>Agg. R100</td>
<td>-</td>
<td>0.6369*</td>
</tr>
</tbody>
</table>

Table 3: Query Aspect Evaluation. ‘QA’ is our query aspect identification method (using 10 aspects). A * denotes a significant improvement over ‘BL0’ (the paired t-test with \( p < 0.05 \)).
In this paper, we proposed a framework for diversifying query suggestions to help domain-specific searchers. We identify diverse query aspects, generate many queries related to these, and suggest effective and diverse queries based on the identified aspects. Through experiments, we showed that the suggestions generated by our system produce more diverse and effective search results in comparison to baseline methods. The main contribution of our work is diversifying query suggestions based on query documents, which has not been addressed. In addition, our method is easily reproducible and general; we do not require any manually constructed data or external resources, and effectiveness was verified in two different domains. For future work, we plan to conduct experiments in the legal domain (e.g., finding relevant cases).

### 6. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

### 7. REFERENCES


