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

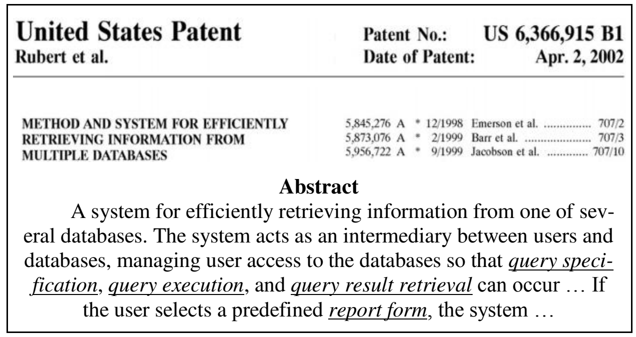
Diversifying query suggestions; Patent retrieval; Citation search

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 is 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").



United States Patent — Rubert et al.
Patent No.: US 6,366,915 B1
Date of Patent: Apr. 2, 2002

METHOD AND SYSTEM FOR EFFICIENTLY RETRIEVING INFORMATION FROM MULTIPLE DATABASES

5,845,276 A * 12/1998 Emerson et al. 707/2
5,873,076 A * 2/1999 Barr et al. 707/3
5,956,722 A * 9/1999 Jacobson et al. 707/10

Abstract

A system for efficiently retrieving information from one of several databases. The system acts as an intermediary between users and databases, managing user access to the databases so that *query specification*, *query execution*, and *query result retrieval* can occur … If the user selects a predefined *report form*, the system …

Figure 1: Query Document Example

Table 1: Relevant Documents for Figure 1.

No	Title	Aspect
A	System for generating structured query language statements and integrating …	*query specification*
B	Combining search criteria to form a single search …	*query specification*
C	Query language execution on heterogeneous database servers using …	*query execution*
D	System and method for generating reports from a computer database …	*report form*

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

SIGIR'14, Month 1–2, 2010, City, State, Country.

Table 2: Features for Similarity Learning.

Category	Features
Topical Relatedness	PMI of $\langle t_i, t_j \rangle$ calculated by 8-word windows recognized in all documents in a corpus PMI of $\langle t_i, t_j \rangle$ measured by titles PMI of $\langle t_i, t_j \rangle$ calculated by 8-word windows identified in query document
Retrieval Effective-ness	Query Clarity (QC) [5] Query Scope (QS) [8] Inverse Document Frequency (IDF) Inverse Collection Term Frequency (ICTF)

query aspects by defining a "query aspect" as a set of related terms from the query document and use term clustering algorithms to identify *n* term sets. Once *n* query aspects (i.e., term clusters) are identified, we generate multiple queries relevant to the identified aspects, and suggest the top k ranked queries. Our experiments show that diversified suggestions are effective for retrieving more relevant documents in comparison to existing suggestion methods.

2. FRAMEWORK

2.1 Query Aspect Identification

The first step is identifying *n* query aspects by representing a query aspect 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 $m \times (m-1)/2$ term pairs (the similarity is undirected). By estimating the similarity for each term pair $\langle t_i, t_j \rangle$, 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 describe how to estimate the similarity for $\langle t_i, t_j \rangle$.

We define similarity between terms by a mixture of topical relatedness (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 effective for retrieving relevant documents. To achieve this, we introduce the similarity function.

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

where t_i and t_j is a term pair from a query document.

In Eq. (1), $T(t_i, t_j)$ measures topical relatedness between t_i and t_j, while $R(t_i, t_j)$ estimates retrieval effectiveness. λ is a controlling parameter. For T, we utilize term statistics obtained from the document 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:

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

where f_k indicates a feature defined in Table 2 and ω_k is a weight of the *k*-th feature. To predict more accurate similarity, we employ a supervised learning approach. Given a term pair $\langle t_i, t_j \rangle$, a supervised learner estimates its similarity score by learning an optimal value of the feature weights ($\omega = \{\omega_1, \ldots, \omega_k\}$).

We now generate training examples as follows. For each query document, *N* different term pairs are extracted, and we label each pair as positive or negative, i.e., $L(\langle t_i, t_j \rangle) \in \{0,1\}$. 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 estimated 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 assume that PMI estimated from a relevant document indicates topical association effective for retrieving relevant documents.

2.2 Query Generation

In this step, based on *n* identified query aspects, we generate queries by exploiting the query generation method proposed in [12]. For each query aspect (i.e., a set of terms), we first retrieve pseudo-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 diversification model proposed in [14] and introduce the following probabilistic 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:

$$(1-\lambda) \cdot \mathrm{P}(q \mid D_Q) + \lambda \cdot \mathrm{P}(q, \bar{S} \mid D_Q) \tag{3}$$

where S is the list of selected queries to be suggested and q is a candidate query from L.

In Eq. (3), $\mathrm{P}(q \mid D_Q)$ denotes the relevance of q to D_Q, while $\mathrm{P}(q, \bar{S} \mid D_Q)$ indicates the novelty of q to S. That is, these two probabilities are optimizing *relevance* and *diversity*, controlled by λ. $\mathrm{P}(q \mid D_Q)$ can be computed by $\prod_{t \in q} \mathrm{P}_{\mathrm{LM}}(t \mid D_Q)$, i.e., the unigram language model estimated from D_Q, and $\mathrm{P}(q, \bar{S} \mid D_Q)$ can be estimated using the identified query aspects.

By the set of query aspects A_Q we can marginalize $\mathrm{P}(q, \bar{S} \mid D_Q)$ as:

$$\mathrm{P}(q, \bar{S} \mid D_Q) = \textstyle\sum_{ap \in A_Q} \mathrm{P}(ap \mid D_Q) \cdot \mathrm{P}(q, \bar{S} \mid ap) \tag{4}$$

where ap is a query aspect in A_Q.

In Eq. (4), we consider $\mathrm{P}(ap \mid D_Q)$ as an *importance* of ap for D_Q, which is estimated by $\prod_{t \in ap} \mathrm{P}_{\mathrm{LM}}(t \mid D_Q)$.

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:

$S \leftarrow \emptyset$

While $|S| \leq k$ do

  $q^* \leftarrow \underset{q \in L \setminus S}{\operatorname{argmax}} (1-\lambda) \cdot \mathrm{P}(q|D_Q) + \lambda \cdot \mathrm{P}(q, \bar{S}|D_Q)$

  $L \leftarrow L \setminus \{q^*\}$

  $S \leftarrow S \cup \{q^*\}$

End While

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, $\mathrm{P}(q, \bar{S}|ap)$ can be derived as:

$$\mathrm{P}(q, \bar{S}|ap) = \mathrm{P}(q|ap) \cdot \mathrm{P}(\bar{S}|ap) \quad (5)$$

$\mathrm{P}(q|ap)$ measures the *coverage* of q with respect to ap, and $\mathrm{P}(\bar{S}|ap)$ 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 q, S, and ap. Specifically, we assume that a query's top 100 retrieved documents can represent underlying topics of the query, and $\mathrm{P}(q|ap)$ can be estimated by how much of topics in ap are covered by q. The equation is given as:

$$\mathrm{P}(q|ap) \approx |Ret_q \cap Ret_{ap}| / |Ret_{ap}| \quad (6)$$

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 $\mathrm{P}(\bar{S}|ap)$, 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.

$$\mathrm{P}(\bar{S}|ap) \approx \mathrm{P}(\overline{qs_1}, \overline{qs_2}, \ldots, \overline{qs_{n-1}}|ap) \approx \textstyle\prod_{qs \in S}(1 - \mathrm{P}(qs|ap)) \quad (7)$$

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

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 query logs. We modify this method to fit in our search environments; we first 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.

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 $p < 0.05$).

Metric \ Method	PAT		ACL	
	BL0	QA	BL0	QA
R100	0.1091	-	0.4452	-
Max. R100	-	0.1491*	-	0.4695*
Agg. R100	-	0.1918*	-	0.6369*

(**Evaluation Measures**) Although there has been considerable research on measuring diversity for search results (e.g., [4]), these previous measures are not appropriate for our search environments; [4] only evaluates the retrieval results for a single query but we suggest multiple queries for a query document and some multi-query session-based metric is required; in addition, there was no emphasis on recall in session search results. Thus, to evaluate "diversity" in multi-query sessions, we propose Session Novelty Recall.

Session Novelty Recall (SNR) is a recall-based metric for multi-query sessions. First, given multiple retrieval results, we ignore relevant documents already found by previous suggestions, i.e., newly retrieved relevant documents are only counted. Second, following the idea in [9], we discount the documents retrieved by later suggestions. The computation is given as follows.

First, we construct a rank list, *L*, by concatenating the top 100 documents from each ranked list in a session. Next, in the list, we discard any retrieved documents which are retrieved by any previous queries, i.e., the rank list contains only distinct retrieval results. In addition, each retrieved result is labeled by the query which first retrieved it.

$$\mathrm{SNR@100} = \textstyle\sum_{i=1}^{|L|} \frac{\mathrm{rel}(d_i^j)}{\log_k(j+k-1)} / |R| \quad (8)$$

where d_i^j is the document placed at the *i*-th rank in *L* and retrieved by the *j*-th suggestion in a session, *R* is the set of relevant documents, *k* is # of queries that the user examines where $k > 1$, $\mathrm{rel}(d)$ returns 1 if *d* is relevant; otherwise, 0. Ideally, if the first query retrieved every relevant document, the value is maximized. In addition to this measure, we employ **normalized Session DCG** (nSDCG) [9] to measure retrieval effectiveness of the top *k* suggested queries.

3.2 Results

(**Query Aspect Identification Performance**) In this experiment, we hypothesize that more relevant documents are retrieved if the identified query aspect is effective. We measure the retrieval effectiveness of each query aspect by formulating a query using the terms in each query aspect. Table 3 shows the retrieval results of query aspects and baseline. For each query document, 10 query aspects are identified and a single baseline query is used. We measure recall (R@100) in two different ways: (1) selecting the best one among *n* different query aspects (Max R@100) and (2) aggregating the retrieved relevant documents (within rank 100) by all query aspects (Agg. R@100). We report an average value of each metric over the query documents in each corpus.

Table 4: Session evaluation using 5 and 10 suggestions. #Q is the number of queries suggested for each query document. In each row, a significant improvement over each baseline is marked by its number, e.g., [12] indicates improvement over 'BL 1&2', and the paired *t*-test is performed with $p < 0.05$.

Metric	#Q	BL1	BL2	DivQS (n = 10)	DivQS (n = 20)
PAT (patent)					
SNR @100	5	0.1560	0.1715[1]	0.1855[1]	**0.1961**[12]
	10	0.1893	0.1989	0.2322[12]	**0.2509**[12]
nSDCG @100	5	0.0812	0.0827	0.1209[12]	**0.1319**[12]
	10	0.0783	0.0959	0.1127[12]	**0.1212**[12]
ACL (academic)					
SNR @100	5	0.5459	0.5731[1]	0.6329[12]	**0.6519**[12]
	10	0.6078	0.6351[1]	0.7192[12]	**0.7392**[12]
nSDCG @100	5	0.3273	0.3116	0.4200[12]	**0.4347**[12]
	10	0.3385[2]	0.3099	0.4357[12]	**0.4457**[12]

First, regarding Max R@100, our method can generate at least one query aspect which can significantly outperform the baseline. Second, from Agg. R@100 we see that significantly more relevant documents are retrieved when using all identified aspects. This is a useful result because query aspects can find relevant documents that are missed by BL0 and the query suggestions generated by these aspects should also perform well.

(**Diverse Query Suggestion Performance**) We now evaluate diverse query suggestion results in terms of retrieval effectiveness and diversity. For each query document, we suggest 5 and 10 queries by identifying 10 or 20 different query aspects in each query document (i.e., n = 10 or 20). The baselines (BL1&2) generate the same number of query suggestions for the same query document. Table 4 reports retrieval performance of each method. First, in both domains, BL2 can outperform BL1 in terms of SNR. Second, the queries suggested by our method (DivQS) can provide significantly more diversified results and retrieve more relevant documents. SNR verifies that our method is more effective at finding new relevant documents missed by previous queries (since SNR ignores the relevant documents retrieved by any previous queries). Third, considering nSDCG, our method is significantly better at placing relevant documents at higher ranks. This is because the queries generated by our method contain more discriminative terms from relevant documents.

4. RELATED WORK

In this paper, we are interested in the diversity between query-suggestion pairs, which has been studied in recent work (e.g., [13][15][16]). Song et al. [16] selected query candidates from query logs by ranking them in the order which maximizes the similarity and diversity between the queries. Santos et al. [15] used the related queries, from query logs, which contain common clicks or common sessions for diversifying suggestions. However, these methods cannot be used for our task because they are based on proprietary training data (to learn ranking functions) and query logs (to generate suggestions), which are not available. Instead, the query suggestion methods proposed in [2][12] are more easily applied in QDS environments but do not consider diversity.

5. CONCLUSION

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

[1] Azzopardi, L., Vanderbauwhede, W., and Joho, H. (2010). Search System Requirements of Patent Analysts. *SIGIR*.

[2] Bhatia, S., Majumdar, D., and Mitra P. (2011). Query Suggestions in the Absence of Query Logs. *SIGIR*.

[3] Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. *LREC*.

[4] Clarke, C. L. A., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Buttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. *SIGIR*.

[5] Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting Query Performance. *SIGIR*.

[6] Fujii, A., Iwayama, M., and Kando, N. (2007). Overview of the patent retrieval task at the NTCIR-6 workshop. *NTCIR-6*.

[7] Ganguly, D., Leveling, J., Magdy, W., and Jones, G. J. F. (2011). Patent query reduction using pseudo-relevance feedback. *CIKM*.

[8] He, B., and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. *18th Symposium on String Processing and Information Retrieval*.

[9] Jarvelin, K., Price, S. L., Delcmbre, L. M. L., and Nielsen, M. L. (2008). Discounted Cumulated Gain based Evaluation of Multiple-query IR Sessions. *ECIR*.

[10] Joho, H., Azzopardi, L., and Vanderbauwhede, W. (2010). A Survey of Patent Users: an analysis of tasks, behavior, search functionality and system requirement. *IIiX*.

[11] Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating query substitutions. *WWW*.

[12] Kim, Y., Seo, J., and Croft, W. B. (2011). Automatic Boolean Query Suggestion for Professional Search. *SIGIR*.

[13] Kharitonov, E., Macdonald, C., Serdyukov, P., and Ounis, I. (2013). Intent models for contextualizing and diversifying query suggestions. *CIKM*.

[14] Santos, R. L. T., Macdonald, C., and Ounis, I. (2010). Exploiting query reformaulations for web search result diversification. *WWW*.

[15] Santos, R. L. T., Macdonald, C., and Ounis, I. (2012). Learning to rank query suggestions for adhoc and diversity search. *Information Retrieval*, 16(4).

[16] Song, Y., Zhou, D., and He, L-w. (2011). Post-Ranking Query Suggestion by Diversifying Search Results. *SIGIR*.

[17] Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: a language-model based search engine for complex queries (extended version). *Technical Report*, UMASS CIIR.