Generating Relevant and Diverse Query Phrase Suggestions using Topical N-grams

Nguyen Kim Anh
School of Information and Communication Technology
Hanoi University of Science and Technology
anhnk@soict.hust.edu.vn

Hung Pham-Thuc
School of Information and Communication Technology
Hanoi University of Science and Technology
hungpt@gmail.com

ABSTRACT

In order to improve the usability of a search engines, Query Suggestion, a technique for generating alternative queries to Web users, has become an indispensable feature for such systems. All major web-search engines and most existing works on query suggestion utilize query logs to determine possible query suggestions. However, for many search systems, query logs are either unavailable or imperfect to learn appropriate models. In this paper, we propose a new method for generating phrase suggestions by using Topical N-grams to discover a set of meaningful phrases from a document corpus. Furthermore, by ranking suggested phrases with hidden topics, our method is able to effectively generate topically diverse as well as semantically related suggestions. Our proposed approach is tested on a variety of datasets and is compared with the best query suggestion approach without query logs. The experimental results clearly demonstrate the effectiveness of our approach in suggesting queries with higher quality.

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval – Query formulation

General Terms

Algorithms, Experimentation

Keywords

Query suggestions, Topical N-grams, search engine

1. INTRODUCTION

With the advent of search engines, it is increasingly easier for Web users to seek desired information. In order to enhance the effectiveness, Query Suggestion has long been proved indispensable to help users explore and express their search intents. Improving the performance and quality of query suggestion techniques has been extensively studied in the past decades to enhance the entire user search experience within the same search intent. Most existing works on query suggestion utilize query logs to suggest queries by measuring the similarity between queries either based on query terms or click-through data [1, 3, 4, 6, 9, 10, 12]. Such query log based techniques are suitable only for search engines with a large user base that can utilize large amounts of past user's queries to offer possible query suggestions. In [3], from detailed study of log of a popular search engine, Baraglia et al. concluded that an average of 2.84 queries are submitted per user per session and most of them are very short and imperfect, which are more likely to be ambiguous. Furthermore, even in the case of general-purpose web search engines, end-users sometimes submit queries that are not in query logs or are not very frequent. Thus, generating meaningful and relevant query suggestions in such scenarios is an interesting and challenging research problem.

SoICT 2014, December 04–05, 2014, Hanoi, Vietnam.

http://dx.doi.org/10.1145/2676585.2676601

In fact, user queries submitted to the web search engines don't always contain appropriate keywords enough for retrieving the related pages to the user intention. An user may remember only a part of the query phrase or don't know the order of keywords that he/she needs to enter in the query string. Furthermore, users cannot clearly express their purposes in several query words due to the lack of knowledge. In order to find satisfactory answers, the users have to rephrase their queries constantly [11]. In these cases, to help the user, information retrieval system can search the query logs to identify queries similar to the user's query that have been successful in the past and can suggest such queries to the user. The scenario, however, is quite different for information retrieval systems that are not on the web or have a smaller user base, and thus, lack large amounts of query log data [2]. Hence, our goal is to offer query suggestions to the user even in scenarios where query logs are unavailable or imperfect. In this paper, we propose a document-centric approach by utilizing the documents in the corpus itself to create a database of query phrases and when the user starts typing a query, we utilize this database to complete the partial user query. The completed queries are then offered as suggestions to the user. Our approach can be applied for the search systems without query logs or with small query logs. Of course, this approach can generate qualified phrase suggestions for user queries that are not in query logs or are not very frequent.

Indeed, query suggestion mechanisms try to suggest various possible completions of the (incomplete) query the user is still typing so as to save time and cover various possible interpretations of the user's query. By using Topical N-grams to extract a set of meaningful phrases from a document corpus and ranking suggested phrases with hidden topics, our method is able to effectively generate topically diverse as well as semantically related suggestions. Our proposed approach is tested on a variety of datasets and is compared with the best query suggestion approach without query logs. The experimental results clearly demonstrate the effectiveness of our approach in suggesting queries with higher quality.

The rest of the paper is organized as follows. Section 2 provides a review of the previous works on query suggestion – both with and

without query logs. In Section 3, we present the problem formulation and describe our proposed approach in detail. Experiments are described in Section 4 and results in Section 5. Section 6 concludes the paper and offers directions for future work.

2. RELATED WORKS

Query Suggestion with Query Logs

Most existing works on query suggestion utilize query logs to suggest queries by measuring the similarity between queries either based on query terms or click-through data [1, 3, 4, 6, 9, 10, 12]. Specially, in [4],.Ebrahimi et al. have proposed a phrase recommender algorithm for web search queries based on the conceptual frequent phrases extracted from the past user queries. By searching conceptual frequent phrases that contain the last one or more sequence of words in the incomplete query, this algorithm suggests the frequent related phrases to complete user's input query. And recently, in [9], Li et al. presented a novel approach to related query suggestion using hidden topic model. Unlike previous methods, their approach utilizes the hidden topic distributions of past user queries to represent their semantic meanings. Moreover, based on the learned topic model using LDA, they can give accurate and fast suggestion for a short query no matter whether the input query was appeared in the past query archive or not [9].

Query Suggestion without Query Logs

Among the works where query logs are not used as the primary resource to generate query suggestions, Feuer et al. [8] described a proximity search based system that suggests alternate queries by adding most frequent terms appearing in close proximity of the query terms in the corpus to the too broad query or by deleting infrequent query terms to the highly specific query. Their work, however, deals with query refinement problem that is quite different from query suggestion or query completion. Furthermore, their approach is not real time and requires the user to type a complete query first [2].

Bast and Weber developed the CompleteSearch engine [5], which can offer real-time auto-completion of the last query term being typed by the user. These completions are extracted from the search results obtained by using the incomplete query and presented in the order of decreasing frequency of the term's occurrence in the search results. However, CompleteSearch is limited in the sense that it only offers completions of the last query word instead of suggesting complete queries related to the incomplete query. Moreover, using frequency as the only criterion for ranking the completions is not a reasonable solution because the most frequent completions of the last query word are not related to the query terms already typed by the user [2].

In [2], Bhatia et al. proposed a probabilistic mechanism for generating query suggestions from the corpus without using query logs. Their approach is simple and intuitive. They utilize the document corpus to extract a set of all N-grams of order 1, 2 and 3. And, as soon as a user starts typing a query, phrases that are co-occurring frequently with the partial user query are selected as completions of this query and are offered as query phrase suggestions. Experimental results on two different datasets in [2] show that their approach achieved statistically significant improvements over two state-of-the-art baselines Similarity-based phrase search and CompleteSearch [5]. To the best of our knowledge, this paper is the first to study the problem of query suggestions in the absence of query logs. However, the set of phrases in [2] is all N-grams of order 1, 2 and 3 (i.e. unigrams, bigrams and trigrams) extracted automatically from the document corpus. Since a N-grams can be considered as a phrase only because of the frequent use of it as a fixed expression, extracted phrases can have not specialized meaning. Moreover, using frequency of a candidate phrase with the former portion of user query in documents of corpus as the only criterion for estimating correlation between this phrase and the user query to rank the completions is not an reasonable solution because the most frequent candidate phrases with that former portion are not semantically related to the user query.

3. PROPOSED APPROACH

3.1 Problem Formulation

In fact, users often cannot clearly express their information need in several query words due to the lack of domain knowledge. To support the user, information retrieval system can extract phrases/ queries related closely to the user's query from an available database of queries to suggest to the user. In the absence of query logs or their imperfection, we propose a document-centric approach by utilizing a document corpus to extract meaning phrases and utilize these phrases to help the users in completion their partial query.

Normally, users choose only candidate phrases related to their partial queries and drop other phrases. Therefore, ranking candidate phrases should be performed on the basis of semantic relations between candidate phrases and user's partial queries. By ranking suggested phrases with hidden topics and then appending these phrases to the partial query, the completed queries are then offered as suggestions to the user. In next section, we first describe the meaning phrase extraction process and then present a approach to rank semantically phrases that can be used to generate query suggestions. The phrase extraction process is performed offline while ranking phrases is online in our method.

3.2 Phrase Extraction using Topical N-grams

Phrases often have specialized meaning and thus, it is not considered as a phrase only because of the frequent use of it as a fixed expression [13]. By consulting the context where the term is located, we can determine meaning phrases. In many situations, topic is very useful to accurately determine the meaning and can play a important role in meaning phrase extraction. Developed from LDA based topic model, Topical-N grams(TNG) is not only a topic model that uses phrases, but also help linguists discover meaningful phrases in right context and more interpretable topics, in a completely probabilistic manner [13]. For the inference problem of TNG model, we use Gibbs sampling to conduct approximate inference in our proposal. Therefore, in order to create a database of phrases that can be used for completing partial user queries, we first use Topical-N grams to discover all bigrams and then extract the longest N-grams phrases by concatenating consecutive bigrams from the document corpus. Clearly, by forming higher order N-gram phrases, the N-grams list produced by TNG is also cleaner. It is important for the real world query phrase recommendation in web search engines.

Using TNG model, we can take the topic of the first/last word token or the most common topic in the bigrams as the topic of the bigrams. In this paper, we also use the topic of the last term as the topic of the bigrams for simplicity as in [13]. Finally, the average topic assignment of concatenated consecutive bigrams is treated as a topic representation of the corresponding higher order N-grams.

3.3 Ranking suggested phrases

Consider an user u who starts typing a query to express a search intent in the query box of a search engine and Q_u is user's incomplete query at any given instant of time.

Let P = {$p_1, p_2, \ldots, p_n$} is the set of meaning phrases extracted by using Topical-N grams on the document corpus and let V is the vocabulary of the corpus. To select phrases that can be used for generating possible query suggestions, we can rank the phrases by the probability of their being typed after Q_u to express an user's search intent and use the top ranked phrases for offering suggestions to the user.

Assume that Q_u can be decomposed as follows:

$$Q_u = Q_c + Q_t$$

where Q_c denotes the completed former portion and Q_t is the uncompleted latter portion that is the last one or more sequence of words of the user query Q_u . Note that $|Q_c| \geq 0$.

Indeed, given an user's submitted partial query, if we are interested only in the relative ordering of phrases then $P(Q_u)$ and $P(Q_t)$ are constants since Q_u and Q_t remain the same for all the phrases. Hence, using Bayes' theorem, we have:

$$P(p_i/Q_u) \sim P(p_i)\ x\ P(Q_u\ /p_i) \quad (1)$$

Assuming that the query terms are conditionally independent, $P(Q_u\ /pi)$ can be written as:

$$P(Q_u\ /p_i) = P(Q_c\ /p_i)\ \text{x}\ P(Q_t\ /p_i) \quad (2)$$

Using equations 1 and 2, we have:

$P(p_i/\ Q_u) \sim P(p_i)\ x\ P(Q_t\ /p_i)\ \text{x}\ P(Q_c\ /p_i)$. It also mean that:

$$P(p_i/\ Q_u) \sim P(Q_t)\ x\ P(p_i/Q_t)\ \text{x}\ P(Q_c\ /p_i) \quad (3)$$

Hence, we have: $P(p_i/\ Q_u) \sim P(p_i/Q_t)\ x\ P(Q_c\ /p_i)$ (4)

According to equation (4), the first component measures the probability that phrase p_i can be typed by the user given Q_t whereas the second component measures the correlation between p_i and Q_c of the user query.

To rank the phrases by the probability of their being typed after given an user partial query Q_u , we first find a set of candidate phrases that can be used to complete Q_u . Since every candidate phrase will represent the latter half of the complete query with respect to the uncompleted portion Q_t of the user query, the candidate phrase must contain Q_t. Let S = {$s_1, s_2, \ldots, s_m$} be the set of m candidate phrases that contain Q_t. The probability that phrase s_i can be typed by the user given Q_t can be estimated as follows:

$$P(s_i/Q_t) = \frac{freq(s_i)}{\sum_{k=1}^{m} freq(s_k)} \quad (5)$$

where *freq(s_i)* is the frequency of s_i.

It is easy to see that the above formula (5) selects phrases on the basis of the last query words Q_t only and does not take into account the context in which the user has typed Q_t. The second component of equation (4) takes into account such a relationship between a phrase and the user's submitted query. Hence, ranking suggested phrases significantly depends on evaluating this second component.

According to [2], this second component evaluates the frequency of candidate phrase s_i with Q_c in documents of corpus. It mean that selecting s_i to complete the partial query Q_c leads many document retrieved from corpus by using a search engine because the resulting query suggestions will consist of terms that frequently co-occur. In this way, the probability that the user has typed Q_c given the selected phrase s_i can be estimated as follows:

$$.P1(Q_c|s_i) \sim \frac{|D_{Q_c} \cap D_{s_i}|}{|D_{s_i}|} \quad (6)$$

Here, D_{Q_c} and D_{s_i} represent the sets of documents that contain Q_c and phrase s_i respectively. Simply, for particular phrase ***s***, we approximate D_s as the set of documents that contain all the constituent words in phrase ***s*** as in [2].

Although this estimation helps in making sure that the resulting query suggestions have good retrieval capability but the most frequent candidate phrases with Q_c are not semantically related to the user submitted partial query as well as the user's search intent. Normally, users choose only suggested phrases related to their partial queries and drop other phrases because they don't know which unrelated suggestion to lead many relevant documents retrieved from corpus by using a search engine. Furthermore, Q_u often are very short and imperfect, which are more likely to be ambiguous. Ranking candidate phrases on the basis of the frequency of candidate phrase with Q_c in documents of corpus doesn't allow us to select a set of phrases that cover various possible interpretations of the user's query.

Using Gibbs sampling, we can infer the topic distribution for user query Q_u and identify its various possible meanings. Moreover, we assume simply that given candidate phrase in the corpus that reflect the same hidden topic distribution as the user's submitted partial query will have a higher probability of being used by the user for formulating queries. It is because appending this phrase to the partial query strengthens its more possible meanings and lessens its ambiguity. Hence, one way to estimate the correlation between the given phrase and the user's submitted query is by using their topical similarity. We calculate the similarity between given phrase s_i and Q_u in terms of their corresponding topic distributions to estimate phrase-query correlation. It mean that $P(Q_c|s_i)$ can be estimated as follows:

$$P2(Q_c|s_i) \sim \text{Cosine (Topics}(Q_u), \text{Topics}(s_i)) \quad (7)$$

where Topics(s_i) is the topic distribution of s_i.

Clearly, by estimating phrase-query correlation with hidden topics, the obtained query suggestions not only are topically diversified but also semantically related to the user submitted partial query.

4. EXPERIMENTAL DESIGN

4.1. Data Description

We evaluated our proposed query suggestion mechanism using the following two datasets:

1. AP News[1]: This dataset consists of 1189 news articles published in AP News between 1988-1990. The AP News writes about topics such as economics, politics and society.
2. Labour[2]: The dataset is crawled from website labourlist.org, consists 1852 news articles from Oct 2011 to May 2014. These articles are written about

[1]http://ap.org/

[2] http://labourlist.org/

centre-left issues and the future of the Labour movement.

For both the datasets, all the documents were pre-processed to remove punctuations and all the text was converted to lower case.

For extracting N-grams, stop words used came from a general stop word list of 573 words used in the Onix Test Retrieval Toolkit[3]. The number of N-grams of order 1, 2 and 3 and the number of Topical N-grams extracted by using Topical-N grams[4] [13] for each dataset per N-grams order is summarized in Table 1. For Topical N-grams model, the number of topics is set to be 30 with 10.000 Gibbs sampling iterations, and the hyper-parameter setting alpha = 50.0, beta = 0.01, gamma = 0.01, delta = 0.03 is used for both the datasets.

Table 1. Number of N-grams extracted by each method

N	*N-grams*	*TNG-N-grams*
1	24788	15826
2	200624	30244
3	189026	10234
4		3008
5		910
6		252
7		84
8		18
9		9
10		3

(a) AP News Dataset

N	*N-grams*	*TNG-N-grams*
1	20279	13061
2	201799	24441
3	202930	8639
4		2249
5		696
6		205
7		83
8		31
9		16
10		11

(b) Labour Dataset

4.2. Baseline Methods

We compare our proposed approach with the following two baseline methods:

1. Probability based 1-2-3-grams ranking (NgramsProb): In this method, phrases are all N-grams of order 1, 2 and 3 extracted automatically from the document corpus and ranking candidate phrases is performed by using probability based estimation via formula (6) [2].
2. Probability based TNG-N-grams ranking (TNGProb): In this method, phrases are all N-grams discovered automatically from the document corpus using Topical-N-grams and ranking candidate phrases is performed by using probability based estimation via formula (6).

In our method named TNGSim, phrases are all N-grams discovered automatically from the document corpus using Topical-N-grams and ranking candidate phrases is performed by using similarity-based estimation via formula (7).

We do not compare our system with any method that uses query logs because they use totally different techniques from our method.

4.3.Evaluation Measure

Evaluating the quality of query recommendation is difficult, since there is usually no ground truth of recommendations. Besides, there is no evaluation metric that seems to be universally accepted as the best for measuring the performance of recommendation algorithms. In fact, the query-logs based query suggestion methods usually give frequent past user queries as suggestions whereas our method generates suggestions by combining user partial query with meaningful phrases discovered from the corpus and hence, may not always generate relevant queries. Also note that for a given partial query, in general, there will be multiple possible valid completion suggestions. Hence, for a given partial query, a query suggestion method is successful if it is able to consider both diversity and relevance in an unified way. Therefore, to measure the performance of suggestion methods, in this paper, we adopt two metrics (Relevance and Diversity) used in [14] to evaluate the quality of suggested queries, where relevance is used to evaluate the semantic relation between suggested phrases and user's partial query and diversity to evaluate the coverage of the suggestion result.

Relevance. We use the same method used in [14] to evaluate automatically the relevance of recommended query phrases. Concretely, we measure the relevance of suggested phrase and user's partial query based on the similarity between their corresponding categories provided by ODP[5].

Diversity. We measure the diversity of suggested queries based on the differences between their top ranked search results provided by Google. Specially, for computing diversity, we used the **Advanced Search within a site or domain of Google** to get results only from the site corresponding to each of two experimental datasets. Concretely, given two queries q and q', we compute the proportion of different URLs among their top k (k = 10 in our case) search results.

4.4. Test queries

In order to compare our proposed approach with the baselines, we need a set of queries for which the suggestions can be generated. All approaches are evaluated with a set of 20 test queries for each dataset and $|Q_t| = 1$. These query sets came from titles of 20 random articles. Then, we removed stop words and selected randomly a number of beginning words from each title. This is due to the user query usually, is not a title but a part of it. For example, a title "Balls speech signals Labour's shift in focus to EU reform" can generate queries such as: "balls speech", "balls speech signals", etc.

5. RESULTS AND REMARKS

5.1. Quality of Suggestions

[3]http://www.lextek.com/manuals/onix/stopwords1.html

[4] Topical-N grams was retrieved from http://mallet.cs.umass.edu/download.php

[5] http://www.dmoz.org/

Table 2: Examples of suggestions generated by different query suggestion methods

NgramsProb	**TNGProb**	**TNGSim**
headline policies(Labour dataset)		
policies presented picked	headline policies	employment policies
headline policies devolution	specific policies	devastating policies
policies presented employees	policies set	policies set
policies presented favourite	key policies	policies appeared genuine
headline policies housing	government policies	failed policies
headline policies benefit	individual policies	coalition health policies
room fell policies	policies improved	government policies
policies presented training	buying policies	individual policies
policies presented minimum	failed policies	buying policies
policies presented guaranteed	headline policies housing	policies improved
stock exchange (AP News dataset)		
american stock exchange	new york stock exchange	stock exchange
stock exchange	american stock exchange	new york stock exchange
nationwide consolidated exchanges	stock exchange	american stock exchange
financial times exchange	tokyo stock exchange	tokyo stock exchange
declining issues exchange	new york mercantile exchange	stock exchange friday
share prices exchange	exchange commission	exchange trading
losers outnumbered exchange	stock exchange friday	new york stock exchange opening
nikkei stock exchange	exchange dealer	stock exchange thursday
advancing issues exchange	exchange trading	foreign exchange trading
issues exchange	commodity exchange	london stock exchange
agriculture department (AP News dataset)		
facility operated departments	agriculture department	agriculture department
operated departments	state department	agriculture department report
agriculture department rumors	agriculture department report	agriculture department reported tuesday
agriculture department bid	agriculture department reported tuesday	agriculture department failed
agriculture department tons	maryland agriculture department	agriculture department program
robert flentge department	department stores	agriculture department confirmed rumors
agriculture department	agriculture department failed	maryland agriculture department
department percent	agriculture department program	agriculture department confirmed
magistrate janice department	agriculture department confirmed rumors	department allotted
scientist department agriculture	research department	department stores

Our experiments were carried on two datasets using above suggestion methods. Some examples of suggestions generated by the different methods for some of the test queries are given in Table 2.

The baseline TNGProb has ability to show the meaningful suggestions in comparison with NgramsProb but these suggestions itself are not topically diverse and semantically related to the user query. The example in Table 3 can clearly show the different results of these two methods NgramsProb and TNGProb. By ranking suggested phrases with hidden topics, TNGSim can show the topically diverse as well as related suggestions. Specially, suggested phrases that are topically diverse and highly correlated with the partial user query are usually on the top of suggestion result. This can completely solve the semantic problem mentioned in [2]. The example in Table 4 can clearly show the different results of these two methods TNGProb and TNGSim.

Figure 1 shows the average relevance values of query suggestions under three different methods. By nature, the average relevance value gradually decreases when the suggestion size increases. The results are listed by top 2, 4, 6, 8, 10 suggestions returned by each of methods. Not surprisingly, the relevance of NgramsProb is the lowest one in the three approaches, since NgramsProb uses all N-grams of order 1, 2 and 3 extracted automatically from the document corpus and ranks candidate phrases only based on *co-occurrence frequency*. Both TNGProb and TNGSim can receive higher relevance value by using Topical N-grams to extract a set of meaningful phrases from a document corpus. Among these two approaches, TNGSim obtains the highest relevance on average by ranking suggested phrases with hidden topics. It indicated the semantically coherency of suggestions and the user's submitted partial query.

Table 3: An example of suggestions generated by NgramsProb and TNGProb.

Query : headline policies (Labour dataset)	
NgramsProb	**TNGProb**
policies presented picked	headline policies
headline policies devolution	specific policies
policies presented employees	policies set
policies presented favourite	key policies
headline policies housing	government policies
headline policies benefit	individual policies
room fell policies	policies improved
policies presented training	buying policies
policies presented minimum	failed policies
policies presented guaranteed	headline policies housing

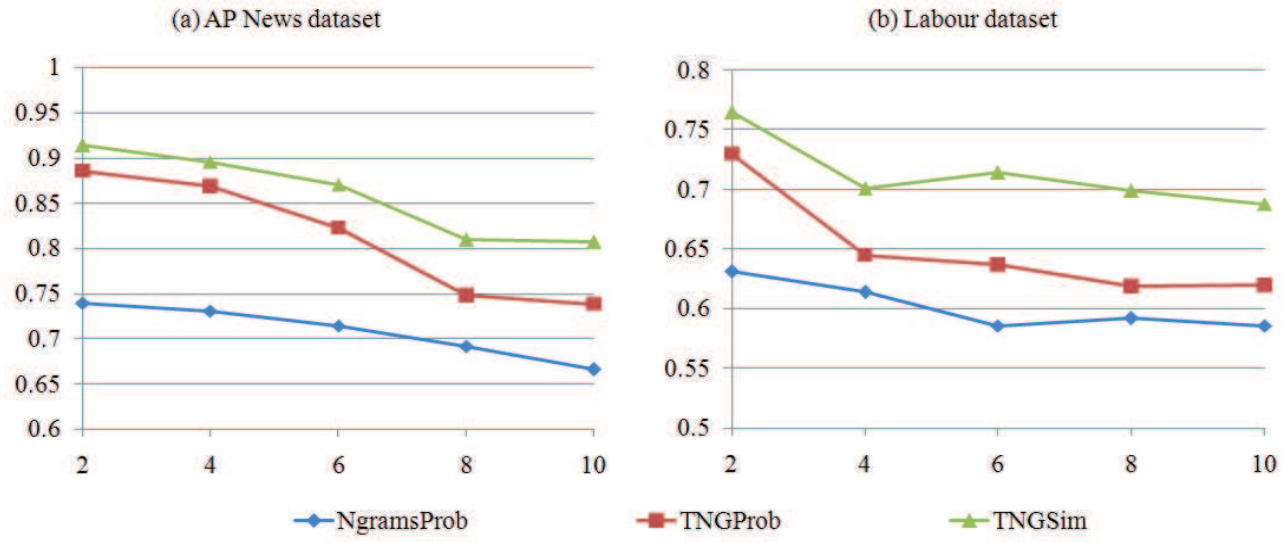

Figure 1: Relevance achieved by different query suggestion methods.

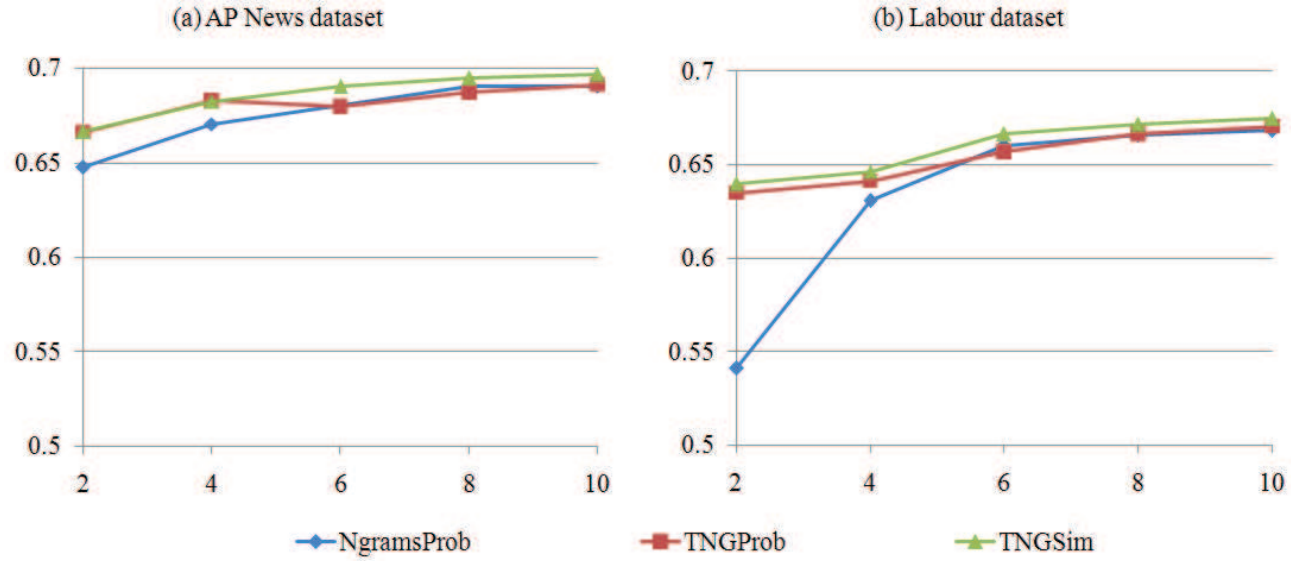

Figure 2:Diversity achieved by different query suggestion methods.

Table 4: An example of suggestions generated by TNGProb and TNGSim.

Query : headline policies (Labour dataset)	
TNGProb	**TNGSim**
headline policies	employment policies
specific policies	devastating policies
policies set	policies set
key policies	policies appeared genuine
government policies	failed policies
individual policies	coalition health policies
policies improved	government policies
buying policies	individual policies
failed policies	buying policies
headline policies housing	policies improved

The diversity is calculated with top 2, 4, 6, 8, 10 suggestions returned by of our method and two baseline methods. The average diversity values of query suggestions under the three different approaches are shown in Figure 2. The results show that TNGSim obtains the highest diversity, specially with top 2, 4 suggestions, but the difference between these methods with top 6, 8, 10 is not much.

In summary. among the three approaches, the proposed TNGSim approach consistently outperforms the other two baselines in terms of relevance and diversity measure. These above results clearly aver that our approach can present with significant improvement in generating diverse and relevant suggestions.

5.2. Retrieval Effectiveness of Suggested Queries

One of the motivations for query suggestion is to present users with queries that can lead to improved retrieval performance [2]. Specially, retrieval effectiveness of the partial query completion can be measured by using a query performance predictor (clarity score) that is used as a measure of ambiguity in a query with respect to a collection of documents [7]. Experimental results on two different datasets in [2] show that their probabilistic-based approach is able to generate suggestions without having an adverse effect on retrieval effectiveness of queries.

Since our goal is generating phrase suggestions to complete the user's partial query, we expect that our method not only is able to select phrases related semantically to the user query but also has a high clarity score. We utilize also query clarity score as proposed by Cronen-Townsend et al. [7] to measure the retrieval performance of suggested queries.

Clarity score for a query is computed as the Kullback-Leibler divergence between the query language model and the collection language model [7]. Mathematically, clarity score for a query q with respect to a collection of documents C is given by

$$Clarity(q, C) = \sum_{v \in V} P(v|q) log_2 \frac{P(v|q)}{P(v|C)}$$

Where V is the vocabulary of the collection [2].

To compute clarity score for each test query, we computed the clarity scores for top 10 suggestions generated by each of methods and computed the average clarity value of resulting suggestions for the query. This computation is repeated for all the test queries and we show the average clarity value achieved for each method. The results are summarized in Table 5. Not surprisingly, the clarity value of NgramsProb is the highest one in the three approaches, since NgramsProb uses all N-grams of order 1, 2 and 3 extracted automatically from the document corpus while TNGProb and TNGSim use only a smaller set of meaningful N-grams from a document corpus. Moreover, NgramsProb uses the frequency of candidate phrase with user's partial query in documents of corpus to select a set of suggested phrases. It leads many document retrieved from corpus by using a search engine. However, since there is many phrases semantically unrelated to the user submitted partial query as well as the user's search intent, users can drop some phrases with high clarity score and choose only a few suggested phrases related to their partial queries. It is because they don't know which unrelated suggestion to lead many relevant documents retrieved from corpus by using a search engine.

Table 5: Mean clarity score achieved by different query phrase suggestion methods.

	AP News dataset	Labour dataset
NgramsProb	4.9	3.5
TNGProb	4.2	2.7
TNGSim	4.23	2.8

Although the clarity value of TNG-based methods is lower than NgramsProb, it is important to note that the difference between TNGProb and TNGSim is not much. Specially, among these two methods, TNGSim obtains the higher clarity value on average than TNGProb. It indicated that our proposed method can have good retrieval capability.

6. CONCLUSIONS AND FUTURE WORK

Query suggestion in the absence of query logs has become a necessary feature for many search systems. Moreover, generating suggestions for rare queries is in fact very difficult and still an open issue due to the lack of information in the query logs and is poorly addressed by state-of-the-art query suggestion techniques. In this paper, we have shown that meaningful query phrase suggestions can be made in the absence of query logs or their imperfection by using Topical N-grams to extract a set of meaningful phrases from a document corpus and ranking suggested phrases with hidden topics. Our method is able to effectively generate topically diverse as well as semantically related suggestions. Our approach can be applied for the search systems without query logs or with small query logs. In particular, this approach can generate highly qualified phrase suggestions for domain-specific search engines. Therefore, our method provides a useful solution to many search systems. It also opens up several development directions for query recommendation without the aid of query logs. Our proposed approach is tested on a variety of datasets and is compared with the best query suggestion approach without query logs. The experimental results clearly demonstrate the effectiveness of our approach in suggesting queries with higher quality.

7. REFERENCES

[1] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 407–416, 2000.

[2] S.Bhatia, D.Majumdar, P.Mitra. Query Suggestions in the Absence of Query Logs, In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pages 795-804, 2011.

[3] R.Baraglia, F.Cacheda., V.Carneiro, D.Fernandez, V.Formoso, R.Perego, F.Silvestri. Search shortcuts: a new approach to the recommendation of queries. In: Proc. RecSys'09. ACM.

[4] M. Barouni-Ebrahimi, Ali A. Ghorbani. A Novel Approach for Frequent Phrase Mining in Web Search Engine Query Streams, *CNSR, page 125-132. IEEE Computer Society, 2007.*

[5] H. Bast and I. Weber. The CompleteSearch Engine: Interactive, Efficient, and Towards IR& DB integration. InCIDR'07, pages 88–95, 2007.

[6] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 875–883, 2008.

[7] S. Cronen-Townsend, Y. Zhou, W. B. Croft. Predicting Query Performance, In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pages 299-306, 2002.

[8] A. Feuer, S. Savev, and J. A. Aslam. Evaluation of phrasal query suggestions. InCIKM'07, pages 841–848, 2007.

[9] L. Li, G. Xu, Z.Yang, P. Dolog, Y. Zhang, M. Kitsuregawa. An efficient approach to suggesting topically related web queries using hidden topic model, World Wide Web – volume 16, issue 3, pp 273-297, 2013.

[10] H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from click-through data for query

suggestion. In Proceeding of the 17th ACM conference on Information and knowledge management, pages 709–718, 2008.

[11] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In Proceeding of the 17th ACM conference on Information and knowledge management, pages 469–477, 2008.

[12] Van Thanh Nguyen and Kim Anh Nguyen, Generating Relevant and Diverse Query Suggestions , In Proceedings of the 5th Asian Conference On Intelligent Information and Database Systems (ACIIDS'13), pp. 176-185, 2013.

[13] X. Wang, A.McCallum, X.Wei. Topical N-grams: Phrase and Topic Discovery,with an Application to Information Retrieval, In Proceeding ICDM '07 Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, pages 697-702, 2007.

[14] X. Zhu, J. Guo, X. Cheng et al. A unified framework for recommending diverse and relevant queries. WWW 2011, India.