Sequential Query Expansion using Concept Graph

Saeid Balaneshin-kordan
Department of Computer Science
Wayne State University
Detroit, Michigan 48202
saeid.balaneshinkordan@wayne.edu

Alexander Kotov
Department of Computer Science
Wayne State University
Detroit, Michigan 48202
kotov@wayne.edu

ABSTRACT

Manually and automatically constructed concept graphs (or semantic networks), in which the nodes correspond to words or phrases and the typed edges designate semantic relationships between words and phrases, have been previously shown to be rich sources of effective latent concepts for query expansion. However, finding good expansion concepts for a given query in large and dense concept graphs is a challenging problem, since the number of candidate concepts that are related to query terms and phrases and need to be examined increases exponentially with the distance from the original query concepts. In this paper, we propose a two-stage feature-based method for sequential selection of the most effective concepts for query expansion from a concept graph. In the first stage, the proposed method weighs the concepts according to different types of computationally inexpensive features, including collection and concept graph statistics. In the second stage, a sequential concept selection algorithm utilizing more expensive features is applied to find the most effective expansion concepts at different distances from the original query concepts. Experiments on TREC datasets of different type indicate that the proposed method achieves significant improvement in retrieval accuracy over state-of-the-art methods for query expansion using concept graphs.

CCS Concepts

•**Information systems** → **Query reformulation;**

Keywords

Query Analysis; Query Expansion; Semantic Networks; Feature-based IR Models

1. INTRODUCTION

Vocabulary mismatch and underspecified queries, which contain only a fraction of concepts that represent the information need (henceforth referred to as explicit concepts), are the two most common reasons for inaccurate and incomplete search results. Synonyms of explicit concepts, as well as other concepts that are relevant to the information need, but are not mentioned in the query (henceforth referred to as latent concepts), can be extracted either from the top retrieved documents [4, 7, 20, 14] or from external knowledge repositories [8, 12, 28, 29, 30], such as knowledge bases and semantic networks, and added to a query through the process known as query expansion. Knowledge bases and knowledge graphs can be very effective for entity-bearing queries and are primarily utilized by first linking queries to entities in a knowledge graph [10, 25] and then enriching the query with elements of textual entity representations, including entity names, the names of related entities, categories and structured attributes [8, 29]. Leveraging general-purpose or domain-specific semantic networks or concept graphs, in which the nodes correspond to words or phrases and the typed edges designate semantic relationships between them, is an alternative approach to query expansion that we focus on in this work. Such approach is applicable to any query, since it does not require a query to contain entities that can be linked to a knowledge base.

Concept graphs can be constructed manually (e.g. ConceptNet [17]), or automatically from a given collection [1, 2, 11, 12] by considering any pair of terms or phrases that frequently co-occur in the same context (e.g. document) as semantically related. Concept graphs are utilized for query expansion by selecting the concepts related to the ones occurring in the query. However, since concept graphs are typically dense [17], there can be a large number of concepts that are immediately related to the query concepts. Although it has been previously shown that there exist very effective expansion concepts in remote layers of concepts related to the original query concepts (i.e. concepts with one or more intermediate concepts between them and the query concepts) [12], the number of candidate concepts that need to be evaluated increases exponentially with the number of layers to consider. However, only a small fraction of hundreds or potentially thousands of concepts that can be discovered in all layers of related concepts in the concept graph can improve retrieval results, while others need to be discarded to avoid noise and concept drift [13, 22, 23]. Figure 1 illustrates this problem for the query *"poach preserve wildlife"*, which we will use as an example throughout this work. According to ConceptNet 5, there are 374 concepts in the first layer of related concepts (that are directly related to the query concepts). Some of these concepts, such as "hunt" and "nature preserve", are relevant to the information need behind this

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983857

query and are useful expansion concepts. However, other related concepts, such as "boil", "injure", "keep", "album" are not relevant to the information need behind this query and should be discarded. The concepts in the third layer, such as "capture" and "wildlife sanctuary" that are also related to the information need behind this query should be separated from many other non-relevant concepts in this layer, even though some of these non-relevant concepts are related to the useful concepts in the second layer.

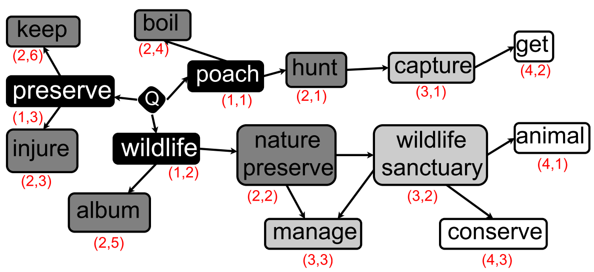


Figure 1: Fragment of the concept graph of ConceptNet 5 showing the concepts related to the concepts in the query "poach wildlife preserve". The first number in parenthesis indicates concept layer, the second number is the index of a concept in the concept layer.

Therefore, accurate evaluation and effective pruning of noisy concepts to find a small number of highly effective concepts for query expansion are the two fundamental challenges in effective utilization of concept graphs for query expansion. In this paper, we propose a two-stage method that addresses these challenges. The proposed method is illustrated for the case of our example query in Figure 2.

In the first stage of the proposed method, all concepts in each concept layer are first sorted according to a quality measure calculated using a number of computationally inexpensive features, such as TF-IDF. Then, in the second stage of the method, a concept selection method that relies on more computationally expensive features is applied to *sequentially* select a set of expansion concepts from the concepts in each layer that are sorted in the first stage. This method selects the concepts from each layer in a one-by-one manner while *maintaining the desired level of precision and minimizing the number of concepts that need to be examined*. Therefore, a limited number of concepts are examined in each layer using computationally expensive features and a limited number of them are selected as expansion concepts. To improve the efficiency and avoid topic drift, only the concepts that are related to the concept selected in layer i are considered in layer $i+1$. As a result, the proposed method avoids calculating computationally expensive features, such as average mutual information, for a large number of concepts in concept layers that are further away from the original query concepts.

The remainder of this paper is structured as follows. First, in Section 2 we discuss the previous work related to this study. Existing concept-based query expansion approaches and the proposed two-stage method for sequential query expansion are presented in detail in Section 3. The features used in both stages of the proposed method, a method to optimize their weights with respect to the target retrieval metric and the results of an experimental evaluation of the proposed method are presented in Section 4, while Section 5 concludes the paper with a summary of the key results and contributions.

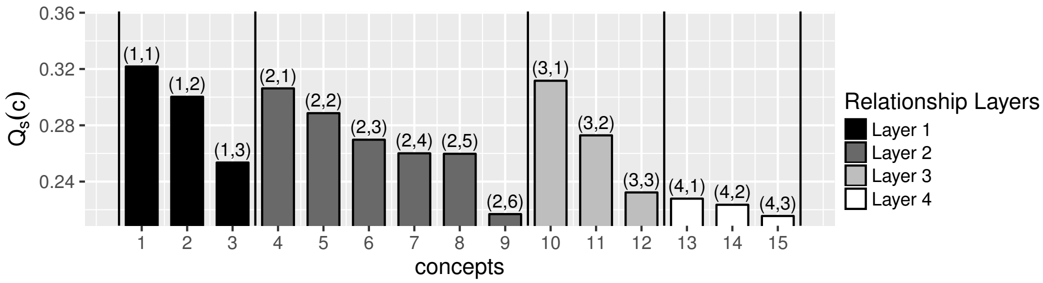
2. RELATED WORK

Concept graphs are widely used in domain-specific [3] and general-purpose [8, 12] information retrieval (IR) systems. They provide structured knowledge that is necessary to fill in the gap between the information provided by a user in the form of a query and the information required by a retrieval system in order to return complete and accurate results. Concept graphs can be constructed from a document collection as in [2, 11, 12]; semantic network, such as ConceptNet [2, 12]; or from an entity-centric knowledge graph, such as DBpedia [2] or Freebase [2]. Since there can be a very large number of concepts in a concept graph that are related to a query, traditional methods for concept selection from the top retrieved documents, such as the one proposed in [7] and [30], that exhaustively evaluate all candidate concepts can be quite inefficient.

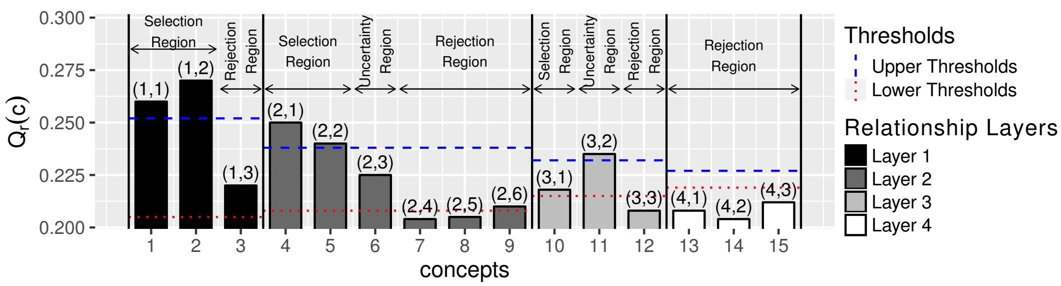
To tackle the difficulty of examining a large number of concepts, simple approaches [16, 26] utilizing external information to prune useless expansion concepts have been previously proposed for domain-specific IR. Experimental evaluation of these methods have shown that it is possible to achieve a significant improvement in retrieval accuracy by pruning the candidate concepts with certain properties, such as semantic types. In particular, a medical IR system proposed in [26] discards candidate expansion concepts from the top retrieved documents that are determined to be unrelated to healthcare based on a simple Wikipedia-based heuristic. The method proposed in [16] does not consider the candidate concepts from the Unified Medical Language System concept graph, the semantic type of which does not belong to a pre-determined list of semantic types known to be effective for specific medical tasks associated with medical record search queries. Since general-purpose retrieval systems operate with a larger and more diverse set of concept and query types than domain-specific ones, they cannot effectively prune candidate expansion concepts based on simple heuristics.

Query expansion methods utilizing general-purpose entity-centric knowledge graphs, such as DBpedia and Freebase, have been extensively investigated in recent years [8, 28, 29, 30]. These methods require annotations of the queries (and, in some cases, also of the documents) with links to Freebase entities, which makes them ineffective for the queries that are ambiguous, broad or do not contain proper nouns designating named entities that can be linked to a knowledge graph.

Kotov and Zhai [12] studied the retrieval effectiveness of expansion concepts from ConceptNet that are related to the query concepts thorough one or several intermediate concepts. In particular, their method first sorts all ConceptNet concepts, which are related to the query concepts through at most 2 intermediate concepts, according to predicted average precision (AP) of retrieval results after adding each concept, and uses the top 100 concepts with the highest predicted AP to create a query expansion language model. They found out that, although the majority of the concepts in the second and third concept layers do not improve the



(a) Initial Concept Sorting (Stage I).



(b) Sequential Concept Selection (Stage II)

Figure 2: Illustration of the proposed two-step concept selection method for a set of related concepts in Figure 1.

accuracy of retrieval results, there are several highly effective concepts in these layers. However, finding them requires evaluation of a large number of concepts.

Sequential analysis (and active learning, its closely related area) have been adopted by many methods to deal with very large datasets. These methods aim to minimize the cost (or time) spent on obtaining reasonably accurate results. In IR, these methods have been applied to minimize (or reduce) the relevance feedback effort (i.e. the number of relevance judgments of retrieved documents), while maintaining an acceptable level of retrieval accuracy [5, 15, 27, 32]. Diaz [9] proposed a method that sequentially selects query expansion terms from the top retrieved documents and achieves a significant improvement over standard pseudo-relevance feedback (PRF) approaches.

3. METHOD

Due to a large number of candidate concepts that are related to the original query concepts, finding effective expansion concepts in a concept graph is a challenging problem, particularly since most of the candidate concepts have zero or negative effect on the accuracy of retrieval results, when they are used for query expansion. The proposed query expansion method is based on the idea of sequential examination of concepts in different layers of a concept graph with respect to the original query concepts. It first evaluates the related concepts at each relationship layer by using a number of inexpensive features and then chooses subsets of related concepts to be evaluated carefully by using more expensive features. The method aims to minimize the total number of concepts evaluated in each layer, while maintaining the precision of retrieval results above a given threshold. This way, selection of effective expansion concepts can be formulated as an optimization problem, in which the objective is to minimize the total number of evaluated concepts subject to precision of retrieval results being above a given threshold.

In this section, we present the details of our proposed method to address the problem of selection of effective expansion concepts from dense, large and noisy concept graphs. First, we discuss the details of the adopted query expansion model and then present the methods to construct concept graphs and use them for sequential selection of query expansion concepts.

3.1 Query Expansion

The proposed method is based on the Latent Concept Expansion (`LCE`) [20] framework. LCE was designed to incorporate the query expansion terms from the top retrieved documents into Markov Random Fields-based retrieval models [19], which allow to account for term dependencies. The proposed method uses the following scoring function of document D with respect to query Q:

$$s(Q, D) = \sum_{i=0}^{k} \alpha_i \sum_{j=1}^{M_i} f_i(D, C_{(i,j)}) \qquad (1)$$

where α_i is the weight of the concepts in the i-th concept layer, k is the number of concept layers that are involved in the concept selection process, and M_i is the number of concepts in the i-th concept layer. $C_{(i,j)}$ in the above equation is the j-th concept in the i-th concept layer. Let us define $\mathbb{C}_i = \{C_{(i,j)}\}_{j=0}^{M_i}$ as the set of concepts in the i-th concept layer. In this case, $\mathbb{C}_0$ contains all the unigrams in a given query. Retrieval models using unigrams only utilize $\mathbb{C}_0$. $\mathbb{C}_1$ includes the query concepts that can be found in the concept graph.

A query is expanded with a limited number of concepts selected in each concept layer $1 \leq i \leq k$. In the above formula, $f_i(D, C_{(i,j)})$ is the matching score of concept $C_{(i,j)}$ in document D. Let us define

$$g(\kappa, D) = \log\left(\frac{tf_{\kappa,D} + \mu\frac{cf_\kappa}{|C|}}{|D| + \mu}\right) \quad (2)$$

as the matching score of concept κ with respect to document D. In the above equation, $g(\kappa, D)$ is the log-likelihood of κ in the language model of D smoothed using Dirichlet prior smoothing, μ is the Dirichlet prior, $|D|$ is the length of document D and $|C|$ is the number of documents in a collection. κ can be a unigram w, ordered $\#uw(b)$ or unordered $\#od(b)$ bigram b. Any other n-gram concepts are represented in terms of these three concept types. For example, the concept "*wild life preserve*" is decomposed into a set of unigrams ("*wild*", "*life*", "*preserve*") and a set of bigrams ("*wild life*", "*life preserve*"). Therefore, the matching score of document D with respect to concept $C_{(i,j)}$ is defined as:

$$\begin{aligned} f_i(D, C_{(i,j)}) = \quad & \gamma_T \sum_{w \in C_{(i,j)}} g(w, D) + \\ & + \gamma_U \sum_{b \in C_{(i,j)}} g\Big(\#uw(b), D\Big) \\ & + \gamma_O \sum_{b \in C_{(i,j)}} g\Big(\#od(b), D\Big) \end{aligned} \quad (3)$$

where γ_T, γ_O and γ_U are the weights of unigrams, ordered and unordered bigrams, respectively. By replacing Dirichlet smoothing in (2) with Jelinek-Mercer smoothing and considering only the concepts from the top retrieved documents as expansion concepts, we obtain the same retrieval function as used in the original `LCE` model [20].

The proposed method for query expansion consists of two stages. In the first stage, candidate expansion concepts are ordered with respect to a quality measure (defined below), while a sequential selection method to find the expansion concepts is applied in the second stage. As a result, only the concepts that are likely to be useful expansion concepts are evaluated in detail. Therefore, the key idea behind the proposed method is to use computationally inexpensive features to initially sort all related concepts and a combination of computationally expensive and inexpensive features to sequentially evaluate them and select the final set of concepts for query expansion. Sorting of the concepts in Stage I of the proposed method provides an initial understanding of concept usefulness, which is utilized in Stage II to minimize the number of evaluated concepts. These two stages as well as different methods to construct the concept graph are explained in more detail below.

3.2 Concept Graphs

Concept graphs used in experiments were constructed in two different ways. One way is to use a manually created semantic network, such as ConceptNet [17]. In this case, we only considered English concepts. If there is a link between the two concepts in ConceptNet, they are considered as related concepts in the concept graph.

The other way to construct a concept graph is to use a collection itself [11]. Only unigram concepts are used in the concept graph in this case. We used Hyper-space Analogue to Language (HAL) similarity measure [6] as a measure of semantic relatedness between the concepts. HAL considers two concepts as highly related if they frequently appear together within a sliding window of certain size (typically, 20 words) throughout a given document collection.

3.3 Sequential Concept Expansion

When concept graphs are large and dense, a very large number of concepts needs to be evaluated to select the useful expansion concepts. If we define $\mathbb{C}^u$ as the set of useful concepts (i.e., those that increase the precision of retrieval results, if added to a query) and $\mathbb{C}$ as the set of all concepts in a concept graph, then the optimal solution to the concept selection problem is obtained by examining all possible subsets of expansion concepts with size 0 to $|\mathbb{C}|$. To obtain this optimal solution, $2^{|\mathbb{C}|}$ subsets of concepts should be evaluated, which is clearly infeasible for any meaningful number of concepts.

A simplified suboptimal solution for the concept selection problem is to evaluate only the concepts that are directly related to the query concepts via a number of intermediate concepts. To further simplify the concept selection process, instead of exhaustively examining all related concepts, we propose to evaluate them *sequentially* (i.e., one after the other). In this approach, starting from the query concepts, the concepts in closer concept layers (i.e., the ones that are semantically closer to the query concepts) are evaluated first. Although the concepts that are semantically closer to the query concepts are not necessarily more useful concepts, they are less affected by the noise propagated from the other concept layers.

Let us define $\mathbb{C}^r_{(i,j)}$ and $\mathbb{C}^u_{(i,j)}$ as the sets of *related* and *useful* concepts, respectively, when examining $C_{(i,j)}$, the j-th concept at relationship level i. Selection of the concept $C_{(i,j)}$ for query expansion can be formulated as a binary hypothesis testing problem with the null hypothesis H_0 and an alternative hypothesis H_1 defined as follows:

$$\begin{aligned} & \mathrm{H}_0: \quad C_{(i,j)} \in \mathbb{C}^r_{(i,j)} - \mathbb{C}^u_{(i,j)} \\ \text{v.s.} \quad & \mathrm{H}_1: \quad C_{(i,j)} \in \mathbb{C}^u_{(i,j)} \end{aligned} \quad (4)$$

After a concept is selected from $\mathbb{C}^r_{(i,j)}$, it is removed from this set. Selecting a concept and adding it to the query changes the usefulness of other concepts; therefore $\mathbb{C}^u_{(i,j)}$ should also be modified after a concept is selected for query expansion.

3.3.1 Stage I: Initial Sorting of Concepts

The concepts are first sorted according to a linear combination of computationally inexpensive features:

$$\tilde{Q}_s(c) = \sum_{j=1}^{m_s} \bar{\lambda}_{s,j} f_j(c) \ , \quad (5)$$

where $\tilde{Q}_s(c)$ is a quality measure of concept c, $f_j(c)$ is a feature function, $\bar{\lambda}_{s,j}$ is a feature weight, and m_s is the number of inexpensive features.

3.3.2 Stage II: Sequential Selection of Concepts

Let us define $\tilde{\mathbb{C}}^u_i$ as the set of concepts selected in the concept layer $i \in \{1, 2, \ldots, k\}$. It is preferable for the set $\tilde{\mathbb{C}}^u_i$ to be as close as possible to the set of useful concepts in the concept layer i (i.e., $\mathbb{C}^u_i$). In each concept layer starting from the first (i.e., $C_{(i,1)}$), the concepts are evaluated sequentially.

After examining the k-th concept layer, the total set of selected concepts is the union the concepts selected in each of the $\{1, 2, \ldots, k\}$ concept layers:

$$\tilde{\mathbb{C}}_k^{ut} = \bigcup_{i=1}^{k} \tilde{\mathbb{C}}_i^{u} \tag{6}$$

An entire set of selected concepts can be obtained by solving the following optimization problem:

$$\min_{\tilde{\mathbb{C}}_k^{ut}} \quad \left\{ \sum_{i=1}^{k} N_i \right\}$$
$$\text{such that} \quad E(\tilde{\mathbb{R}}_\Lambda; \mathbb{T}) > \theta_Q \ , \tag{7}$$

In the above equation, N_i is the number of concepts evaluated in the i-th concept layer. N_i is less than or equal to the number of concepts in the i-th concept layer (i.e., $N_i \leq M_i$). $E(\tilde{\mathbb{R}}_\Lambda; \mathbb{T})$ is a retrieval quality evaluation metric for a set of document rankings, $\tilde{\mathbb{R}}_\Lambda$, based on the training data $\mathbb{T}$. Document rankings $\tilde{\mathbb{R}}_\Lambda$ are those that correspond to the expanded query, which contains the selected concepts $\tilde{\mathbb{C}}_k^{ut}$. In (7), θ_Q is a pre-specified lower threshold for $E(\tilde{\mathbb{R}}_\Lambda; \mathbb{T})$.

The goal of the above optimization procedure is to address the problem of dealing with a large number of related concepts that need to be evaluated in each concept layer. This goal is accomplished by *selecting* concepts in such a way that the least number of concepts is evaluated, while maintaining an acceptable value for the target retrieval metric (e.g. MAP). The set $\tilde{\mathbb{C}}_k^{ut}$ can be approximated by Algorithm 1. In this algorithm, $\tilde{Q}_r(C_{(i,j)})$ is a measure of retrieval effectiveness of the candidate concept $C_{(i,j)}$ that can be calculated using expensive and inexpensive features as a weighted linear combination of feature functions as follows:

$$\tilde{Q}_r(C_{(i,j)}) = \sum_{j=1}^{m_r} \hat{\lambda}_{r,j} f_j(C_{(i,j)}) \ , \tag{8}$$

where $\hat{\lambda}_{r,j}$ is the weight of a feature function $f_j(C_{(i,j)})$, and m_r is the number of expensive and inexpensive features. $\tilde{Q}_r(C_{(i,j)})$ is applied to the concepts that are already sorted using $\tilde{Q}_s(c)$. Different decisions can be made by comparing $\tilde{Q}_r(C_{(i,j)})$ with the upper and lower thresholds (denoted by β_U and β_L). One of the decisions that can be made as a result of such comparisons is whether to select $C_{(i,j)}$ as an expansion concept or to discard it. The other decision is whether to continue examining and evaluating the concepts in the same concept layer or to switch to the next concept layer and start examining its concepts. These decisions are formalized in Table 1.

Computational complexity of this algorithm can be reduced further by discarding the concepts that have $Q_s(c)$ below a threshold $\beta_{s,L}$ in stage I of the algorithm (i.e., those that have $Q_s(c) < \beta_{s,L}$). In this case, the number of concepts that are evaluated in the Stage II of the algorithm can be decreased at the expense of retrieval performance degradation, the degree of which is controlled by the value of β_L.

4. EXPERIMENTS

Statistics of the collections used for experimental evaluation of the proposed method are shown in Table 2. Parameters and hyper-parameters of the proposed method and the baselines were optimized with respect to the Mean Average

Algorithm 1 The proposed two-stage algorithm to obtain a set of expansion concepts.

$i = 1$
$\tilde{\mathbb{C}}_k^{ut} = \{\}$
do
  $\tilde{\mathbb{C}}_i^{u} = \{\}$
  for $c \in \mathbb{C}_i$ **do**
    compute $\tilde{Q}_s(c)$
  end for
  sort $\mathbb{C}_i$ according to $\tilde{Q}_s(c)$
  for $j = \{1, \ldots, M_i\}$ **do**
    compute $\tilde{Q}_r(C_{(i,j)})$
    if $\tilde{Q}_r(C_{(i,j)}) > \beta_U$ **then**
      add $C_{(i,j)}$ to $\tilde{\mathbb{C}}_i^{u}$
    end if
    if $\tilde{Q}_r(C_{(i,j)}) < \beta_L$ **then**
      $i = i + 1$
    end if
  end for
  $\tilde{\mathbb{C}}_k^{ut} = \tilde{\mathbb{C}}_k^{ut} \bigcup \tilde{\mathbb{C}}_i^{u}$
while $\tilde{\mathbb{C}}_i^{u} \neq \{\}$

Table 1: Three possible decisions that can made by evaluating concept c using the proposed method.

Decision	Criterion
Select concept $C_{(i,j)}$ & continue with the same concept layer	If $\tilde{Q}_r(C_{(i,j)}) \geq \beta_U$
Discard concept $C_{(i,j)}$ & continue with the same concept layer	If $\beta_L \leq \tilde{Q}_r(C_{(i,j)}) < \beta_U$
Discard concept $C_{(i,j)}$ & move to the next concept layer	If $\tilde{Q}_r(C_{(i,j)}) < \beta_L$

Table 2: Statistics of experimental collections.

Collection	# of documents	# of terms
TREC7-8	472, 526	2.16×10^8
ROBUST04	528, 155	2.53×10^8
GOV	1, 247, 753	1.37×10^9

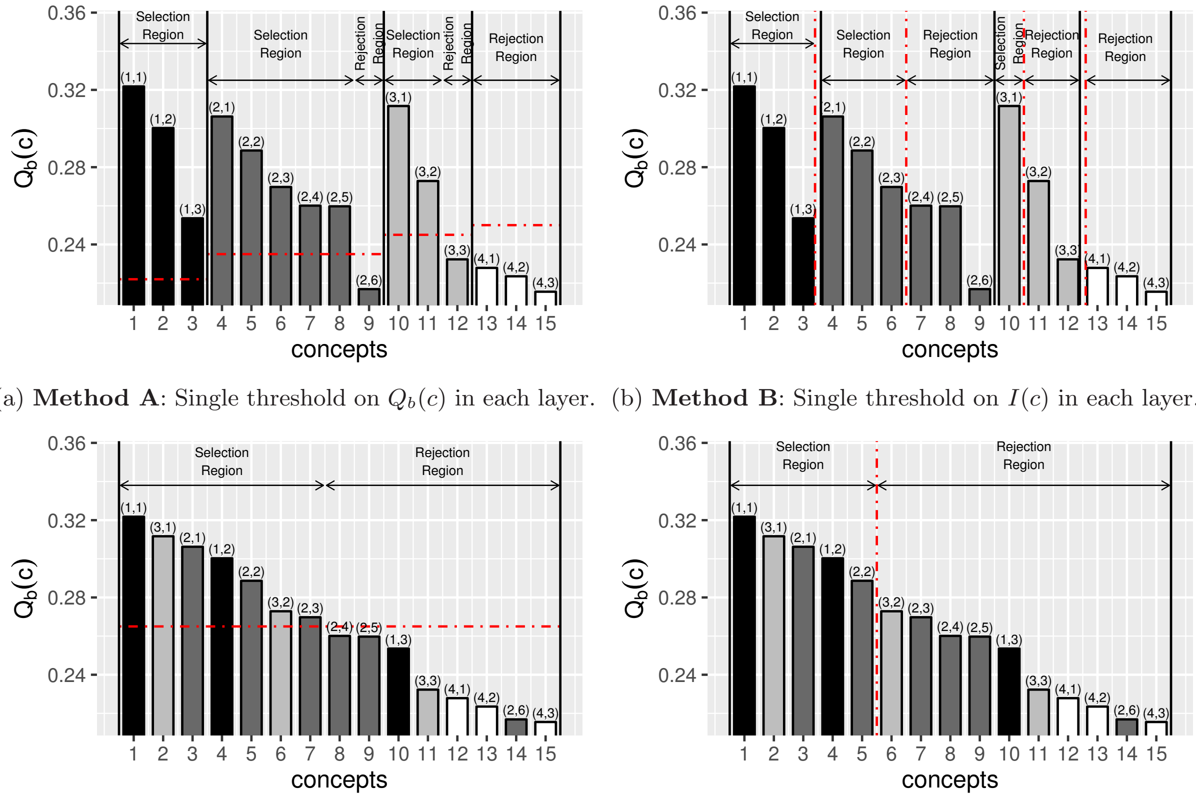
Precision (MAP) on the training set. The concepts in the first concept layer are obtained by using different methods depending on how the concept graph was constructed. If the concept graph is constructed from the collection, this set of concepts consists of all unigrams in the query. If the concept graph is obtained from ConceptNet, this set of concepts consist of the longest query n-grams that correspond to ConceptNet concepts. The concepts in other concept layers were selected by using the links between the concepts in the constructed concept graph, and they can be n-gram concepts with $n \geq 1$.

4.1 Baselines

The primary goal of the sequential concept selection method presented in Section 3.3 is to minimize the number of evaluated candidate expansion concepts from the concept graph. Considering the trade-off between the precision and the computation time, four variations of the proposed method, which

Table 3: **Summary of the proposed method and the baselines**

Method	Optimization Problem		Criteria in the Approximate Solution		
	Objective	Constraint	Selecting	Rejecting	Stopping
Method A	$\min\{\sum_{i=0}^{k} L_i\}$	$E(\tilde{\mathbb{R}}_\Lambda^k;\mathbb{T}) > \theta$	$Q_b(c) > \beta_Q$	$Q_b(c) < \beta_Q$	$i > k$
Method B	$\max\{E(\tilde{\mathbb{R}}_\Lambda^k;\mathbb{T})\}$	$\sum_{i=0}^{k} L_i < \theta$	$I_i(c) < \beta_I$	$I_i(c) > \beta_I$	$i > k$
Method C	$\min\{\sum_{i=0}^{k} L_i\}$	$E(\tilde{\mathbb{R}}_\Lambda^k;\mathbb{T}) > \theta$	$Q_b(c) > \beta_Q$	$Q_b(c) < \beta_Q$	$i > k$
Method D [12]	$\max\{E(\tilde{\mathbb{R}}_\Lambda^k;\mathbb{T})\}$	$\sum_{i=0}^{k} L_i < \theta$	$I(c) < \beta_I$	$I(c) > \beta_I$	$i > k$
Proposed	$\min\{\sum_{i=0}^{k} N_i\}$	$E(\tilde{\mathbb{R}}_\Lambda^k;\mathbb{T}) > \theta$	$Q_r(c) > \beta_U$	$Q_r(c) < \beta_L$	$L_i = 0$



(a) **Method A**: Single threshold on $Q_b(c)$ in each layer. (b) **Method B**: Single threshold on $I(c)$ in each layer.

(c) **Method C**: Single threshold on $Q_b(c)$ in all layers. (d) **Method D**: Single threshold on $I(c)$ in all layers [12].

Figure 3: Graphical summary of the baselines A-D. The thresholds placed on the quality of concepts ($Q_b(c)$) or the number of selected concepts ($I(c)$) in each or all of the concept layers are shown by the red lines.

are summarized in Table 3 and Figure 3, are considered as baselines in experiments. In Table 3:

$$Q_b(C_{(i,j)}) = \sum_{j=1}^{m_s} \hat{\lambda}_{b,j} f_j(C_{(i,j)}) \qquad (9)$$

is a concept quality measure computed as a linear weighted combination of the feature functions. The set of features used to calculate the quality measure $Q_b(c)$ for the baselines is the same as the set of features used to calculate $Q_s(c)$ in (5) for our proposed method. In Table 3, $I(c)$ is the index of a concept in the sorted set of concepts and L_i is the number of selected concepts from the i-th concept layer.

As follows from Table 3, when expansion concept selection problem is formulated as minimization of the number of concepts subject to keeping the evaluation metric above a desired level (i.e., methods A and C), the approximate solution is to select the concepts if their quality measure $Q_b(c)$ is above a threshold and reject otherwise. But, in the case of maximization of retrieval precision by putting a constraint on the number of selected concepts (i.e., methods B and D), the approximate solution is to select a limited number of concepts that result in the biggest improvement in AP.

As can be seen from Table 3, methods A and B, similar to our proposed method, but unlike methods C and D, select query expansion concepts from different concept layers sequentially. In other words, in methods A and B and in our proposed method, the concepts in concept layer i are examined, if their ancestor concept nodes in concept layer $i-1$ are selected. However, methods C and D first find a set of

Table 4: Features used in stages I and II of the proposed method. All of the listed features are considered in stage II of the proposed method, but only the features without asterisks are considered in Step I of the proposed method.

Feature	Description
hgstDocScore	Retrieval score of the highest ranked document containing $C_{(i,j)}$
avgDocScore	Average retrieval score of all documents containing $C_{(i,j)}$
varDocScore	Variance of retrieval score of all documents containing $C_{(i,j)}$
avgTDocScore	Average retrieval scores of the top documents containing $C_{(i,j)}$
termFreqTpDoc	Number of occurrences of $C_{(i,j)}$ in the top documents
docFreqTpDoc	Number of top documents containing $C_{(i,j)}$
nodeDegree	Node degree of $C_{(i,j)}$ in the concept graph
avgNumLinks	Average number of paths between $C_{(i,j)}$ and query concepts
maxNumLinks	Maximum number of paths between $C_{(i,j)}$ and query concepts
avgCooccur*	Average co-occurrence of $C_{(i,j)}$ with query concepts
maxCooccur*	Maximum co-occurrence of $C_{(i,j)}$ with query concepts
avgTCooccur	Average co-occurrence of $C_{(i,j)}$ with query concepts in top retrieved documents
maxTCooccur	Maximum co-occurrence of $C_{(i,j)}$ with query concepts in top retrieved documents
avgTCooccurP*	Average co-occurrence of $C_{(i,j)}$ with at least a pair of query concepts in top retrieved documents
maxTCooccurP*	Maximum co-occurrence of $C_{(i,j)}$ with at least a pair of query concepts in top retrieved documents
avgTCooccur*	Average co-occurrence of $C_{(i,j)}$ with all previously selected concepts in top retrieved documents
maxTCooccur*	Maximum co-occurrence of $C_{(i,j)}$ with all previously selected concepts in top retrieved documents
avgCooccurL*	Average co-occurrence of $C_{(i,j)}$ with selected concepts in concept layer $i-1$
maxCooccurL*	Maximum co-occurrence of $C_{(i,j)}$ with selected concepts in concept layer $i-1$
avgTCooccurL*	Average co-occurrence of $C_{(i,j)}$ with selected concepts in concept layer $i-1$ in top retrieved documents
maxTCooccurL*	Maximum co-occurrence of $C_{(i,j)}$ with selected concepts in concept layer $i-1$ in top retrieved documents
avgTMiP*	Average mutual information of $C_{(i,j)}$ with at least a pair of query concepts in top retrieved documents
maxTMiP*	Maximum mutual information of $C_{(i,j)}$ with at least a pair of query concepts in top retrieved documents
avgTMiL*	Average mutual information of $C_{(i,j)}$ with selected concepts in concept layer $i-1$ in top retrieved documents
maxTMiL*	Maximum mutual information of $C_{(i,j)}$ with selected concepts in concept layer $i-1$ in top retrieved documents

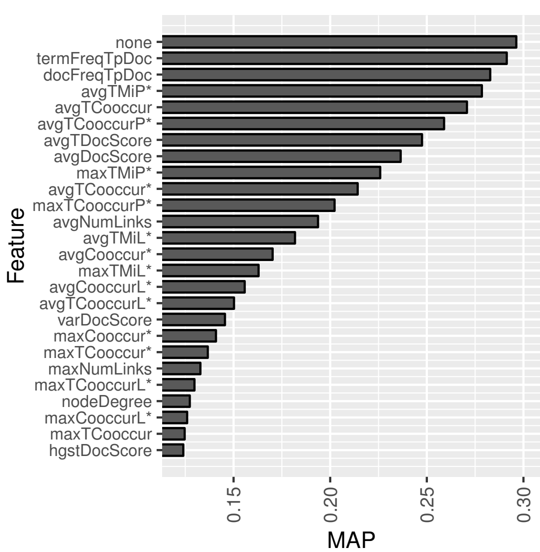


Figure 4: MAP after removing one feature from the list of features in Table 4 that results in the highest decrease of MAP at a time.

all concepts in the layers $1 \leq i \leq k$ and examine all of them at once. Since these methods do not prune the concepts, noise can get propagated from layer to layer.

In methods B and D, the threshold (indicated by β_I in Table 3) is on the number of selected concepts, but, in methods A and C and the proposed method, the thresholds (shown by β_Q, β_L and β_U in Table 3) are on the concept quality measure. Therefore, unlike methods B and D, the thresholds in methods A and C and the proposed method do not limit the number of expansion concepts, and depending on the query, the collection and the required level of retrieval accuracy, the optimal number of expansion concepts is determined by the method. Although methods A and C and the proposed method do not have a predefined threshold on the number of expansion concepts, they have a predefined threshold on concept quality measures ($Q_b(c)$ or $Q_r(c)$). In methods A and B and the proposed method, there are distinct thresholds for each concept layer, while in methods C and D, there is only one threshold for all concept layers. As described in more detail later, β_Q and β_I as well as β_L and β_U are optimized with respect to their objective functions and constraints by using coordinate ascent.

Our proposed method stops at the concept layer i, if no concept is identified at this layer (i.e., if $L_i = 0$), but the methods A-D have predefined limits on the total number of examined concept layers (i.e., k). In other words, the proposed method stops when there is not enough evidence that there are useful concepts in other concept layers, while methods A-D stop when they examine a given number of concept layers. Therefore, unlike the baselines A-D, the number of concept layers examined by the proposed method differ from query to query.

Finally, none of the baselines A-D consider minimizing the number of *evaluated concepts*. The constraints used by methods B and D are on the number of *selected concepts*, while the objective functions of methods A and C are minimizing the total number of *selected concepts*.

The other baselines that are considered in our experimental evaluation are Query Likelihood retrieval model [24] with Dirichlet prior smoothing (`QL`) [31], Relevance Model (`RM`) [14], Sequential Dependence Model (`SDM`) [19] and Latent Concept Expansion (`LCE`) [20].

4.2 Features

Two sets of features are used in the proposed two-stage method. The first set consists of only computationally inexpensive features that are used to initially sort the concepts in the first stage of the proposed method. The second set consists of mostly computationally expensive features that are used to select the concepts in the second stage of the proposed method. Computationally expensive features include the ones that are based on co-occurrence and mutual information [18]. Specifically, the first set of features is used to calculate $Q_s(C_{(i,j)})$ in (5) and the second set is used to calculate $Q_r(C_{(i,j)})$ in (8).

According to Table 4, the number of inexpensive features (designated by m_s in (5)) is 11, and the total number of expensive and inexpensive features (designated by m_r in (8)) is 25. In this table, 16 features depend on the top retrieved documents, 6 on the collection and 3 on the concept graph. The top retrieved documents are obtained only once using `SDM` retrieval model with the original query. The number of top retrieved documents is a hyper-parameter of the proposed method that is estimated via cross-validation.

To determine the relative importance of features, we conducted a study, the results of which for the ROBUST04 collection are reported in Figure 4. In this study, we started with a full feature set and removed one feature, which results in the highest reduction of MAP after being removed from the feature set, at a time. The weights of other features have been updated to satisfy the conditions of the optimization problem each time a feature was removed. As follows from Figure 4, the features that are utilized in both stages of the proposed method have the highest impact on its retrieval accuracy. It can be also concluded that the features that are dependent on the collection tend to have a stronger effect on retrieval performance than other features. Finally, when all the features are removed, retrieval results are obtained using only the concepts in the original query, which have a higher importance weight relative to expansion concepts.

Different combinations of the features listed in Table 4 can be utilized for query expansion, depending on the collection and query set. In particular, from an entire set of features listed in Table 4 we obtained smaller sets of highly effective features for each experimental collection via a backward feature elimination process, in which the features that have negative effect on retrieval accuracy are eliminated one at a time.

4.3 Parameter optimization

Three-fold cross validation was used to evaluate the performance of the proposed method and the baselines. At each cross validation fold, the thresholds β_U and β_L for each concept layer as well as the weights of the features in stages I and II of the proposed method (i.e., $\lambda_{s,j}$ and $\lambda_{r,j}$ in (5) and (8)) were optimized in such a way that the MAP of the top retrieved documents stays above the threshold θ, while the number of concepts examined in stage II of the proposed method is minimized. Coordinate ascent [21] was used to optimize the values of these parameters. Starting from an initial random point, the parameter space was examined in uniform steps (step size was 0.01), one parameter at a time. This process was repeated for all parameters until convergence (if the change in the target retrieval metric from one iteration to another is less than 0.05) or until the number of iterations exceeds 100. The values of θ were chosen based on the MAP of retrieval results of the `QL` method. The values of θ for TREC 7-8, ROBUST04 and GOV collections were set to 0.28, 0.32, and 0.30, respectively, all of which are greater

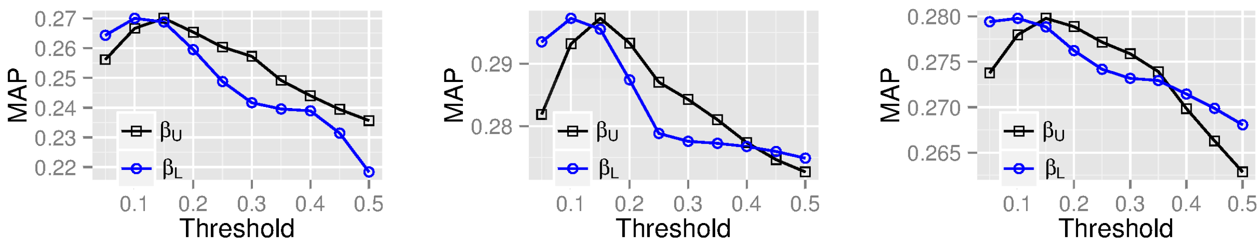
Table 5: Comparison of retrieval performance of the proposed method with the baselines in terms of MAP for different number of examined concept layers.

Col.	Method	Concept Layer 1st	2nd	3rd	4th
TREC7-8	Method D-HAL	0.2220	0.2239	0.2155	0.2120
	Method D-CNet [12]	0.2205	0.2245	0.2214	0.2183
	Method C-HAL	0.2152	0.2227	0.2185	0.2133
	Method C-CNet	0.2182	0.2265	0.2225	0.2218
	Method B-HAL	0.2207	0.2171	0.2266	0.2236
	Method B-CNet	0.2188	0.2294	0.2255	0.2294
	Method A-HAL	0.2172	0.2251	0.2290	0.2282
	Method A-CNet	0.2183	0.2290	0.2329	**0.2335**
	Proposed-HAL	0.2249	0.2348	0.2418	**0.2457**
	Proposed-CNet	0.2222	0.2377	0.2449	**0.2484**
	`SDM`	0.2124	——	——	——
ROBUST04	Method D-HAL	0.2660	0.2644	0.2569	0.2554
	Method D-CNet [12]	0.2640	0.2651	0.2568	0.2555
	Method C-HAL	0.2675	0.2655	0.2608	0.2516
	Method C-CNet	0.2637	0.2628	0.2683	0.2695
	Method B-HAL	0.2684	0.2718	0.2598	0.2535
	Method B-CNet	0.2616	0.2710	0.2665	0.2675
	Method A-HAL	0.2614	0.2758	0.2757	0.2764
	Method A-CNet	0.2689	0.2732	**0.2851**	0.2793
	Proposed-HAL	0.2721	0.2786	0.2865	**0.2898**
	Proposed-CNet	0.2748	0.2814	0.2889	**0.2963**
	`SDM`	0.2359	——	——	——
GOV	Method D-HAL	0.2337	0.2428	0.2355	0.2319
	Method D-CNet [12]	0.2348	0.2396	0.2355	0.2382
	Method C-HAL	0.2404	0.2406	0.2459	0.2322
	Method C-CNet	0.2416	0.2451	0.2378	0.2379
	Method B-HAL	0.2359	0.2466	0.2418	0.2397
	Method B-CNet	0.2420	0.2452	0.2484	0.2421
	Method A-HAL	0.2434	0.2442	0.2491	0.2420
	Method A-CNet	0.2365	0.2455	**0.2524**	0.2422
	Proposed-HAL	0.2455	0.2429	0.2570	**0.2578**
	Proposed-CNet	0.2449	0.2514	0.2575	**0.2591**
	`SDM`	0.2184	——	——	——

than the MAP of the `QL` method on the same collection by 0.08 (see Table 6).

The same training procedure with the same θ as above was used to optimize the parameters of the baseline methods, such as $\hat{\lambda}_{b,j}$ in (9), $\lambda_{s,j}$, $\lambda_{r,j}$ and the thresholds β_Q, β_U and β_L.

Figure 5 illustrates the impact of the upper and lower thresholds on MAP (i.e., β_U and β_L) for different collections at the second concept layer. Because of the dependency between β_U and β_L in the approximate solution to the optimization problems, β_U and β_L are obtained iteratively one after the other by holding the other parameter fixed to a value obtained in the previous iteration. When the value of the upper threshold is less than the optimum, more non-useful concepts are added to the candidate list of expansion concepts. When the value of the upper threshold is greater than the optimum, some useful concepts may not be selected as expansion concepts. When the value of the lower threshold is less than the optimum, the proposed method will evaluate more concepts in total, which is against its main objective. When the value of the lower threshold is greater than the optimum, the selection process may terminate earlier and a number of useful concepts may not be examined at all. In general, although the upper and lower thresholds are dependent on each other, the upper threshold has the main effect on the accuracy of selected concepts, while the lower threshold has the main effect on the number of examined concepts.



(a) TREC7-8 (b) ROBUST04 (c) GOV

Figure 5: MAP of the proposed method in terms of β_U and β_L at the 2$^{\text{nd}}$ concept layer.

Table 6: Comparison of retrieval performance of the proposed method with the baselines. $*$ and † indicate statistically significant improvement in terms of MAP and P@20 according to Wilcoxon signed rank test over `SDM/LCE` with $p < 0.05$ and $p < 0.1$, respectively. Percentage differences in retrieval performance of Method A relative to `SDM/LCE` as well as the proposed method relative to `SDM/LCE` and Method A are shown in parentheses.

Without PRF							
Collection	**Evaluation Metric**	**QL**	**SDM**	**Method A HAL**	**Method A CNet**	**Proposed HAL**	**Proposed CNet**
TREC7-8	MAP	0.1982	0.2124	0.2282∗† (7.44%)	0.2335∗† (9.93%)	0.2457∗† (15.68%/7.67%)	0.2484∗† (16.95%/6.38%)
	P@20	0.3540	0.3765	0.3762 (-0.08%)	0.3783 (0.48%)	0.3785∗ (0.53%/0.61%)	0.3796∗ (0.82%/0.34%)
ROBUST04	MAP	0.2359	0.2510	0.2764∗† (10.12%)	0.2851∗† (13.59%)	0.2898∗† (15.46%/4.85%)	0.2963∗† (18.05%/3.93%)
	P@20	0.3339	0.3667	0.3679 (0.33%)	0.3773∗† (2.89%)	0.3802∗† (3.68%/3.34%)	0.3795∗† (3.49%/0.58%)
GOV	MAP	0.2184	0.2333	0.2491∗ (6.77%)	0.2524∗† (8.19%)	0.2578∗† (10.5%/3.49%)	0.2591∗† (11.06%/2.65%)
	P@20	0.0389	0.0451	0.0476 (5.54%)	0.0493∗ (9.31%)	0.0558∗† (23.73%/17.23%)	0.0552∗† (22.39%/11.97%)
With PRF							
Collection	**Evaluation Metric**	**RM**	**LCE**	**Method A* HAL**	**Method A* CNet**	**Proposed* HAL**	**Proposed* CNet**
TREC7-8	MAP	0.2151	0.2423	0.2503∗ (3.3%)	0.2558∗† (5.57%)	0.2642∗† (9.04%/5.55%)	0.2672∗† (10.28%/4.46%)
	P@20	0.3641	0.3836	0.3883 (1.23%)	0.3927∗ (2.37%)	0.3934∗† (2.55%/1.31%)	0.4035∗† (5.19%/2.75%)
ROBUST04	MAP	0.2683	0.2826	0.2935∗ (3.86%)	0.2979∗ (5.41%)	0.3034∗† (7.36%/3.37%)	0.3053∗† (8.03%/2.48%)
	P@20	0.3561	0.3785	0.3826∗ (1.08%)	0.3834∗ (1.29%)	0.3893∗† (2.85%/1.75%)	0.3965∗† (4.76%/3.42%)
GOV	MAP	0.2403	0.2678	0.2693 (0.56%)	0.2730∗ (1.94%)	0.2793∗† (4.29%/3.71%)	0.2811∗† (4.97%/2.97%)
	P@20	0.0483	0.0566	0.0583 (3.00%)	0.0617∗ (9.01%)	0.0706∗ (24.73%/21.1%)	0.0720∗† (27.21%/16.69%)

4.4 Comparison of Methods

Table 5 provides comparison of performance of the proposed method with the baselines described in Section 4.1. As follows from this table, the best performing baseline is Method A, which is the most similar to the proposed method, since Method A and the proposed method both minimize the number of examined concepts. This can potentially reduce the effect of topic drift, which results in superior performance of these methods.

The outermost concept layer, in which a method is able to identify the concepts that can increase the precision of retrieval results in another interesting criterion for comparison of the methods. A method that is able to identify effective expansion concepts in remote concept layers is more robust, since these layers include large number of noisy concepts. As follows from Table 5, the average outermost layer across different collections (rounded to the nearest integer), in which the baselines A-D and the proposed method were able to identify effective expansion concepts is 3, 3, 2, 2 and 4, respectively. Therefore, it can be concluded that the proposed method and the methods that have multiple thresholds tend to perform better than the methods that have a single threshold. The other conclusion that can be made from this table is that the average outermost layer across different collections (rounded to the nearest integer), in which the 4 baselines and the proposed method were able to discover effective concepts, are 2 and 3 for the collection- and ConceptNet-based concept graphs, respectively. Overall, it can be also seen that the methods using ConceptNet-based concept graph (CNet) obtain higher MAP than the methods using collection-based concept graphs automatically constructed using HAL (HAL).

In Table 6, the performance of the proposed method is compared with `QL`, `RM`, `SDM`, `LCE` and the best performing methods in Table 5 that use collection- and ConceptNet-based concept graphs. As opposed to the upper part of

Table 6, all the methods in its lower part also use unigram concepts from the top retrieved documents for query expansion, in addition to the concepts from the concept graphs (i.e. a query is first expanded using the methods in the top part of the table and then using `RM`). The same collection- and ConceptNet-based concept graphs were used to obtain the results in the lower and upper parts of Table 6.

Several conclusions can be made from Table 6. First, Method A provides significant improvement over `QL` and `SDM` when the concept graph is generated from ConceptNet, while the proposed method has significant improvements over the `QL` and `SDM` baselines in case of both the collection- and ConceptNet-based concept graphs. Second, Method A provides a significant improvement over `SDM` in the 5 cases, when it does not incorporate `PRF` concepts, however it provides a significant improvement over `LCE` only in one of the cases, when it uses `PRF` concepts. Although the proposed method provides a smaller improvement over `LCE`, when it uses `PRF` concepts, than over `SDM`, when it does not use `PRF` concepts, the improvements that are achieved in these two cases are significant. Finally, although the parameters are estimated with the goal of maximizing MAP, the proposed method demonstrates significant improvement over the baselines (`QE` and `SDM`) also in terms of P@20.

5. SUMMARY AND CONCLUSIONS

The main contribution of this work is a two-stage method for sequential selection of effective concepts for query expansion from the concept graph. The proposed method is formulated as an optimization problem with the goal of evaluating the least possible number of candidate concepts needed to ensure a given precision of retrieval results. In the first stage of the proposed method, the candidate concepts are sorted using a number of computationally inexpensive features. This sorting is utilized in the second stage to sequentially select expansion concepts by using computationally expensive features. Experimental evaluation using TREC collections indicates that the proposed method outperforms state-of-the-art baselines, which instead of minimizing the number of *evaluated concepts*, aim to minimize the number of *selected concepts* or maximize a concept *quality measure*. We also found out that the proposed method and the baselines produce more accurate results using ConceptNet-based than collection-based concept graph. We believe that applying the proposed method to the case of entity-based queries and knowledge graphs is an interesting future direction for extending this work.

6. REFERENCES

[1] R. Anand and A. Kotov. Improving difficult queries by leveraging clusters in term graph. In *Proceedings of the 11th AIRS*, pages 426–432, 2015.

[2] S. Balaneshin-kordan and A. Kotov. An empirical comparison of term association and knowledge graphs for query expansion. In *Proceedings of the 38th ECIR*, pages 761–767, 2016.

[3] S. Balaneshin-kordan and A. Kotov. Optimization method for weighting explicit and latent concepts in clinical decision support queries. In *Proceedings of the 2nd ACM ICTIR*, 2016.

[4] M. Bendersky, D. Metzler, and W. B. Croft. Parameterized concept weighting in verbose queries. In *Proceedings of the 34th ACM SIGIR*, pages 605–614, 2011.

[5] A. Bookstein. Information retrieval: A sequential learning process. *Journal of the American Society for Information Science*, 34(5):331–342, 1983.

[6] C. Burgess, K. Livesay, and K. Lund. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25(2-3):211–257, 1998.

[7] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st ACM SIGIR*, pages 243–250, 2008.

[8] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th ACM SIGIR*, pages 365–374, 2014.

[9] F. Diaz. Pseudo-query reformulation. In *Proceedings of the 38th ECIR*, pages 521–532, 2016.

[10] F. Hasibi, K. Balog, and S. E. Bratsberg. Entity linking in queries: tasks and evaluation. In *Proceedings of the 2015 ICTIR*, pages 171–180, 2015.

[11] A. Kotov and C. Zhai. Interactive sense feedback for difficult queries. In *Proceedings of the 20th ACM CIKM*, pages 163–172, 2011.

[12] A. Kotov and C. Zhai. Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries. In *Proceedings of the 5th ACM WSDM*, pages 403–412, 2012.

[13] N. Lao and W. W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

[14] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th ACM SIGIR*, pages 120–127, 2001.

[15] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM SIGIR*, pages 3–12, 1994.

[16] N. Limsopatham, C. Macdonald, and I. Ounis. Inferring conceptual relationships to improve medical records search. In *Proceedings of the 10th OAIR*, pages 1–8, 2013.

[17] H. Liu and P. Singh. Conceptnet – a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[18] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[19] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th ACM SIGIR*, pages 472–479, 2005.

[20] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *Proceedings of the 30th ACM SIGIR*, pages 311–318, 2007.

[21] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[22] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of the 21st ACM SIGIR*, pages 206–214, 1998.

[23] L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *Proceedings of the 29th ACM SIGIR*, pages 91–98, 2006.

[24] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM SIGIR*, pages 275–281, 1998.

[25] D. Rao, P. McNamee, and M. Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 93–115. Springer, 2013.

[26] L. Soldaini, A. Cohan, A. Yates, N. Goharian, and O. Frieder. Retrieving medical literature for clinical decision support. In *European Conference on Information Retrieval*, pages 538–549. Springer, 2015.

[27] A. Tian and M. Lease. Active learning to maximize accuracy vs. effort in interactive information retrieval. In *Proceedings of the 34th ACM SIGIR*, pages 145–154, 2011.

[28] C. Xiong and J. Callan. Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM CIKM*, volume 6, pages 3–1, 2015.

[29] C. Xiong and J. Callan. Query expansion with freebase. In *Proceedings of the 1st ICTIR*, pages 111–120, 2015.

[30] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd ACM SIGIR*, pages 59–66, 2009.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM SIGIR*, pages 334–342, 2001.

[32] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4(2):260–268, 2002.