ABSTRACT

Many of the recent, and more effective, retrieval models have incorporated dependencies between the terms in the query. In this paper, we advance this query representation one step further, and propose a retrieval framework that models higher-order term dependencies, i.e., dependencies between arbitrary query concepts rather than just query terms. In order to model higher-order term dependencies, we represent a query using a hypergraph structure—a generalization of a graph, where a (hyper)edge connects an arbitrary subset of vertices. A vertex in a query hypergraph corresponds to an individual query concept, and a dependency between a subset of these vertices is modeled through a hyperedge.

An extensive empirical evaluation using both newswire and web corpora demonstrates that query representation using hypergraphs is highly beneficial for verbose natural language queries. For these queries, query hypergraphs significantly improve the retrieval effectiveness of several state-of-the-art models that do not employ higher-order term dependencies.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms
Algorithms, Experimentation

Keywords
Query hypergraphs, query representation, retrieval models

1. INTRODUCTION

Over the past decade, information retrieval research has undergone a gradual shift of focus from retrieval models that use bag-of-words query representations to retrieval models that incorporate term dependencies. Some recent examples of retrieval models that incorporate term dependencies include, among others, Markov random fields [27], linear discriminant model [14], dependence language model [13], quasi-synchronous dependence model [30], and positional language model [26].

In this paper, we propose a novel retrieval framework that takes a further step toward a more accurate modeling of the dependencies between the query terms. Rather than modeling the dependencies between the individual query terms, our framework models dependencies between arbitrary concepts in the query.

We broadly define a query concept as a syntactic expression that models a dependency between a subset of query terms. Query concepts may model a variety of linguistic phenomena, including n-grams, term proximities, noun phrases, and named entities. Therefore, a dependency between query concepts represents a dependency between term dependencies, i.e., a higher-order term dependency.

To the best of our knowledge, there is little prior work on modeling this type of higher-order term dependencies for information retrieval. Most retrieval models limit their attention to either pairwise term dependencies [11, 26] or, at most, dependencies between multiple terms [2, 27]. In contrast, the retrieval framework proposed in this paper can model dependencies between arbitrary concepts, e.g., a dependency between a phrase and a term. We hypothesize that an accurate modeling of concept dependencies is especially important for verbose natural language queries. This is due to the fact that the grammatical complexity of these queries often challenges the capabilities of the current retrieval models [2, 20].

As an example, consider the natural language query in Figure 1:

"Provide information on the use of dogs worldwide for law enforcement purposes."

Figure 1(a) shows an excerpt from the top document retrieved by a sequential dependence model [27], a state-of-the-art retrieval model that incorporates term dependencies. As evident from this excerpt, the top-retrieved document is non-relevant with respect to the query. Even though it contains many instances of the phrase "law enforcement" as well as the terms provided and information it does not mention the use of dogs.

On the other hand, an excerpt from the document in Figure 1(b) clearly indicates the relevance of the top document.

1In the remainder of this paper, we shall use the definitions “higher-order term dependency” and “concept dependency” interchangeably.

2A description of the TREC topic #426.
retrieved by our method with respect to the query. Even though this excerpt matches less of the query terms than the excerpt in Figure 1(a), it contains a relationship between the term dog and the phrase “law enforcement”, which is highly indicative of its relevance. This relationship cannot be modeled without accounting for higher-order term dependencies.

As Figure 1 shows, the evidence of the concepts co-occurring within a passage of text is a strong indicator of their dependency. This is somewhat akin to term dependencies, which are often modeled based on the frequency of the terms co-occurring next (or close) to each other in the document [27, 39, 26].

In the case of concept dependency, however, instead of relying on the entire document, we only examine a single document passage that is deemed to be the most relevant with respect to the query. This focused evidence can distinguish between relevant documents and documents which simply contain many repeated concept instances, as in Figure 1(a). This approach is reminiscent of the passage retrieval models that often make use of the evidence from the highest-scoring document passage [3, 9, 8, 16, 41].

In contrast to the approach presented in this work, most passage retrieval methods are based on a conjunctive retrieval model and treat a query as a bag of words. However, as the excerpts in Figure 1 demonstrate, such a simple conjunctive retrieval model is not sufficient, especially for verbose, natural language queries.

Instead, the proposed retrieval framework distinguishes between the concepts and the dependencies that are crucial for conveying the query intent, and the concepts and the dependencies of lesser importance. For instance, in the case of the query in Figure 1, the dependency (dog, “law enforcement”) in Figure 1(b) is crucial for expressing the query intent, while the dependency (information and “law enforcement”) in Figure 1(a) is not.

To summarize, unlike any of the current retrieval models, the retrieval framework proposed in this paper integrates three main characteristics that we believe are crucial for improving the effectiveness of retrieval with verbose queries. First, it models arbitrary term dependencies as concepts. Second, it uses passage-level evidence to model the dependencies between these concepts. Finally, it assigns weights to both concepts and concept dependencies, proportionate to the estimate of their importance for expressing the query intent. In this paper, we show that by integrating these characteristics, the proposed retrieval framework can significantly improve the effectiveness of several current state-of-the-art retrieval models.

Table 1: Examples of the possible structures and the concepts they might contain for the query “members of the rock group nirvana” (stopwords removed).

<table>
<thead>
<tr>
<th>Structure σ</th>
<th>Concepts {κ : κ ∈ σ}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>“members”, “rock”, “group”, “nirvana”</td>
</tr>
<tr>
<td>Bigrams</td>
<td>“members rock”, “rock group”, “group nirvana”</td>
</tr>
<tr>
<td>Noun Phrases</td>
<td>“members”, “rock group nirvana”</td>
</tr>
<tr>
<td>Named Entities</td>
<td>“nirvana”</td>
</tr>
<tr>
<td>Dependencies</td>
<td>“members nirvana”, “rock group”</td>
</tr>
</tbody>
</table>

The proposed retrieval framework is based on a query representation using a hypergraph structure – a generalization of a graph, where an edge can connect more than two vertices. A vertex in a query hypergraph corresponds to an individual query concept. The vertices are grouped by structures, which model various linguistic phenomena. For instance, as shown in Table 1, a structure can group together terms, n-grams or noun phrases. Finally, any subset (rather than just a pair as in a standard graph) of vertices can be connected via a hyperedge, which models concept dependencies.

We use the query hypergraph representation to derive a ranking function that incorporates concepts and concept dependencies in a principled manner, based on the factorization of the hypergraph. We then propose two approaches for the parameterization of the ranking function. The first parameterization approach assigns weights to the concepts and the concept dependencies based on their respective structures. The second parameterization approach assigns weights based on a set of importance features associated with each concept and concept dependency.

The remainder of this paper is organized as follows. First, in Section 2 we provide the theoretical underpinnings of the query hypergraph representation and ranking with query hypergraphs. Then, in Section 3 we describe the related work and its connection to query representation using hypergraphs. In Section 4 we report the details of the empirical evaluation of the proposed framework. Section 5 concludes the paper.

2. QUERY HYPERGRAPHS

2.1 Query Representation with Hypergraphs

In this paper, we base the query representation on two modeling assumptions. First, we assume that given a query Q, we can model it using a set of linguistic structures

\[ \Sigma^Q \triangleq \{\sigma_1, \ldots, \sigma_n\}. \]
The structures in the set $\Sigma^Q$ are both complete and disjoint. The completeness of the structure implies that it can be used as an autonomous query representation. The disjointness of the structures means that there is no overlap in the linguistic phenomena modeled by the different structures. In other words, each structure groups together concepts of a single type (e.g., terms, bigrams, noun phrases, etc.).

Second, within each structure, arbitrary term dependencies can be modeled as concepts. In other words, each structure $\sigma_i \in \Sigma^Q$ is represented by a set of concepts

$$\sigma_i \triangleq \{\kappa_i^1, \kappa_i^2, \ldots\}.$$

Each such concept is considered to be an atomic unit for the purpose of query representation. In addition, for convenience, we adopt the notation

$$K^Q \triangleq \bigcup_{i=1}^{n} \sigma_i,$$

to refer to the union of all the query concepts, regardless of their respective structures.

These modeling assumptions, while conceptually simple, create an expressive formalism for hierarchical query representation. This formalism is flexible enough to specify a wide range of specific instantiations. Table 1 shows that it can model a wide spectrum of linguistic phenomena that are often encountered in natural language processing and information retrieval applications.

For instance, as we can see in Table 1, a structure can contain single terms as concepts, resulting in a bag-of-words query representation. A structure can also contain adjacent bigrams or noun phrases. Concepts need not be defined over contiguous query terms, as is demonstrated by the last structure in Table 1, which models a set of linguistic dependency links between the query terms.

For the purpose of information retrieval, we are primarily interested in using the resulting hierarchical query representation to model the relationship between a query $Q$ and a document $D$ in the retrieval corpus. Specifically, given a set of query structures $\Sigma^Q$ and a document $D$, we construct a hypergraph $H(\Sigma^Q, D)^3$.

A hypergraph is a generalization of a graph where an edge can connect an arbitrary set of vertices. A hypergraph $H$ is represented by a tuple $(V, E)$, where $V$ is a set of elements or vertices and $E$ is a set of non-empty subsets of $V$, called hyperedges. In other words, the set $E \subseteq P(V)$ of hyperedges is a subset of the powerset of $V$ [18].

Specifically for the scenario of document retrieval, we define the hypergraph $H$ over the document $D$ and the set of query concepts $K^Q$ as

$$V \triangleq K^Q \cup \{D\}$$

$$E \triangleq \{\{k, D\} : k \in PS(K^Q)\}.$$

Figure 2 demonstrates an example of a hypergraph $H$ for the search query “international art crime”. In this particular example, we have two structures. The first structure contains the query terms denoted $i$, $a$, and $c$, respectively. The second structure contains a single phrase, $ac$. Over these concepts, we can define a set of five hyperedges—four hyperedges connecting document $D$ and each of the concepts, and one hyperedge connecting $D$ and all of the concepts.

Formally, for the hypergraph $H$ in Figure 2, the vertices and the hyperedges are defined as follows

$$V_{Fig. 2} = \{ D, i, a, c, ac \}$$

$$E_{Fig. 2} = \{(i, D), (\{a\}, D), (\{c\}, D),$$

$$ (\{ac\}, D), (\{i, a, c, ac\}, D) \}.$$

Note that this hypergraph configuration is just one possible choice. In fact, any subset of query terms can serve as a query concept, and similarly, any subset of query concepts can serve as a hyperedge, as shown by Equation 1.

### 2.2 Ranking with Query Hypergraphs

In the previous section, we defined the query representation using a hypergraph $H = (V, E)$. In this section, we define a global function over this hypergraph, which assigns a relevance score to document $D$ in response to query $Q$. This relevance score is used to rank the documents in the retrieval corpus.

A factor graph, a form of hypergraph representation which is often used in statistical machine learning [6], associates a factor $\phi_e$ with a hyperedge $e \in E$. Therefore, most generally, a relevance score of document $D$ in response to query $Q$ represented by a hypergraph $H$ is given by

$$sc(Q, D) \triangleq \prod_{e \in E} \phi_e(k_e, D) \overset{\text{rank}}{=} \sum_{e \in E} \log(\phi_e(k_e, D)).$$

\[3\] For conciseness, we use the abbreviation $H \triangleq H(\Sigma^Q, D)$ in the remainder of this paper.
It is interesting to note that Equation 2 is reminiscent of the recently proposed log-linear retrieval models, including the Markov random field model [27] and the linear discriminant model [14]. Similarly to these models, Equation 2 scores a document using a log-linear combination of factors $\phi_k(t_i, D)$.

However, an important difference from these retrieval models is related to the fact that the factors $\phi_k(t_i, D)$ in Equation 2 are defined over concept sets, rather than single concepts, as in previous work [14, 27]. This definition enables the modeling of higher-order dependencies between query terms. Higher-order term dependencies cannot be easily modeled by the existing retrieval models that incorporate term dependencies [4, 14, 26, 27, 30, 39].

Thus far, we have provided only the most abstract definition of the query representation and ranking with query hypergraphs. In the remainder of this section, we provide an in-depth discussion of the query hypergraph induction and a detailed derivation of the ranking function.

First, in Section 2.3, we fully specify the structures, concepts, and hyperedges in the query hypergraph $H$. Then, in Section 2.4, we instantiate the factors $\phi_k(t_i, D)$ in the ranking function in Equation 2 using these specifications. Finally, in Section 2.5, we examine the different parameterizations of the ranking function.

### 2.3 Query Hypergraph Induction

#### 2.3.1 Hypergraph Structures

There are many potential ways in which we could define the set of structures $\mathcal{K}^Q$ in the query hypergraph. In this work, we focus on three types of structures that are successfully used in previous work on modeling term dependencies for information retrieval [4, 5, 27, 32]. We leave a further exploration of other possible hypergraph structures to future work.

1. **QT-structure.** The query term (QT) structure contains the individual query words $t_i$ as concepts. Terms are the most commonly used concepts in information retrieval, both in bag-of-words models [33, 34] and models that incorporate term dependencies [27, 29, 14].

2. **PH-structure.** The phrase (PH) structure contains the combinations of query terms that are matched as exact phrases in the document. Exact phrase matching has often been used for improving the performance of retrieval methods [12, 42]. Most recently, it has been shown that using query bigrams for exact phrase matching is a simple and efficient method for improving the retrieval performance in large scale web collections [4, 5, 27, 29, 32]. Following this finding, we define the concepts in the PH-structure as adjacent query word pairs $(t_i, t_{i+1})$.

3. **PR-structure.** Unlike the PH-structure, the proximity (PR) structure can contain arbitrary subsets of query terms of the form $\{t : t \in Q\}$ as concepts. The PR-structure also differs from the PH-structure in the way the concepts in the structure are matched in the document. In order to match the document, the individual terms in a concept in the PR-structure may occur in any order within a window of fixed length. In this paper, we fix the window size to $4|t|$, where $|t|$ is the number of terms in the concept. This approach follows the definition of term proximity as defined by Metzler and Croft [27].

#### 2.3.2 Hyperedges

As described in Section 2.1, a naïve induction approach may result in an exponential number of hyperedges in a query hypergraph. Instead, for the purpose of this paper, we limit our attention to two types of hyperedges, which have an intuitive appeal from an information retrieval perspective.

1. **Local hyperedges.** For each concept $\kappa \in \mathcal{K}^Q$, we define a hyperedge $((\kappa), D)$. This local edge$^4$ represents the contribution of the concept $\kappa$ to the total document relevance score, regardless of the other query concepts. As we show in the next section, the factors defined over the local edges are akin to the functions that are usually employed in the existing log-linear retrieval models [14, 27].

2. **Global hyperedge.** In addition to the local edges, we define a single global hyperedge $((\kappa), D)$ over the entire set of query concepts $\mathcal{K}^Q$. This global hyperedge provides the evidence about the contribution of each concept $\kappa \in \mathcal{K}^Q$ given its dependency on the entire set of query concepts $\mathcal{K}^Q$. Unlike in the case of local edges, the factors defined over the global hyperedge cannot be easily expressed using the existing log-linear retrieval models.

Figure 2 provides a simple example of these two types of hyperedges. The hyperedges at the bottom of the hypergraph in Figure 2 are the local edges, while the hyperedge at the top is the global hyperedge.

#### 2.4 Factors $\phi_k(t_i, D)$

Following the hyperedge induction process described in Section 2.3.2, in this section we define two types of factors. The local factors – corresponding to the local edges – are defined in Section 2.4.1; the global factor – corresponding to the global hyperedge – is defined in Section 2.4.2.

Both local and global factors incorporate a matching function $f(\kappa, X)$, which assigns a score to the occurrences of the concept $\kappa$ in a text fragment $X$. As a matching function, following some previous work on log-linear retrieval models [4, 14, 27], we use a log of the language modeling estimate for concept $\kappa$ with Dirichlet smoothing [45], i.e.

$$f(\kappa, X) \triangleq \log \frac{tf(\kappa, X) + \mu^{1/2(|\kappa, C|)}}{\mu + |X|},$$

where $tf(\kappa, X)$ and $tf(\kappa, C)$ are the number of occurrences of the concept $\kappa$ in the text fragment and the collection, respectively; $\mu$ is a free parameter; $|X|$ is the number of terms in $X$, and $|C|$ is the total number of terms in the collection.

#### 2.4.1 Local Factors

The local factors are defined over the local edges $((\kappa), D)$. A local factor assigns a score to the occurrences of concept $\kappa$ in the document $D$, regardless of the other query concepts. Therefore, a local factor is defined similarly to the previously proposed log-linear retrieval models [4, 14, 27]

$$\phi((\kappa), D) \triangleq \exp \left( \lambda(\kappa) f(\kappa, D) \right),$$

$^4$From now on, we refer to the local hyperedges simply as edges, since they are defined over a vertex pair, rather than an arbitrary set of vertices.
where $\lambda(\kappa)$ is an importance weight assigned to the concept $\kappa$, and $f(\kappa, D)$ is a matching function between the concept $\kappa$ and the document $D$.

### 2.4.2 The Global Factor

The global hyperedge $(K^Q, D)$ described in Section 2.3.2, represents a dependency between the entire set of query concepts. In this section, we present a global factor that is defined over this hyperedge.

A common way to estimate a dependency between query terms is using a measure of their proximity in a retrieved document [11, 26, 27, 39]. Analogously, we may simply choose to estimate a dependency between query concepts using similar proximity measures. However, there are two notable difficulties that impede an application of this approach to concept dependency.

First, the existing term proximity measures usually capture close, sentence-level, co-occurrences of the query terms in a retrieved document [27, 32, 39]. The dependency range is much longer for concept dependencies. For instance, in the example in Figure 1(b), the concepts dog and law enforcement do not ever appear in the same sentence. However, the dependency between them is revealed when examining their co-occurrences in a larger text passage.

Second, since concepts can be arbitrarily complex syntactic expressions, the probability of observing a concept co-occurrence is much lower than the probability of observing a term co-occurrence, even in large collections. For instance, most documents in the retrieved list for the query in Figure 1, do not contain both of the concepts dog and law enforcement in a context of a single passage.

Therefore, instead of estimating the dependency between query concepts using the standard proximity measures, we leverage a long history of research on passage retrieval [3, 8, 9, 16, 40, 41] for the derivation of the global factor.

In the passage retrieval literature, a document is often segmented into overlapping passages of text of fixed size [16, 17]. The document is then scored using some combination of document-level and passage-level scores. One of the most successful and frequently-used score combinations is the Max-Peg combination, which uses the highest scoring passage to assign a score to the document [3, 8, 16, 24, 41].

Similarly to the Max-Peg retrieval model, we define the global factor using a passage $\pi$, which receives the highest score among the set $\Pi_D$ of passages extracted from the document $D$. Formally,

$$\phi(K^Q, D) = \exp \left( \max_{\pi \in \Pi_D} \sum_{\kappa \in K^Q} \lambda(\kappa, K^Q) f(\kappa, \pi) \right), \quad (5)$$

where $\lambda(\kappa, K^Q)$ is the importance weight of the concept $\kappa$ in the context of the entire set of query concepts $K^Q$, and $f(\kappa, \pi)$ is a matching function between the concept $\kappa$ and a passage $\pi \in \Pi_D$.

Intuitively, the global factor in Equation 5 assigns a higher relevance score to a document that contains many important concepts in the confines of a single passage. Note that the importance weight $\lambda(\kappa, K^Q)$ of a concept in the global factor is determined not only by the concept itself – as in the case of the importance weights $\lambda(\kappa, D)$ in the local factors – but also by the concepts that co-occur together with the concept in the passage $\pi$.

### 2.5 Query Hypergraph Parameterization

In the previous section, we introduced two types of concept weights that parameterize the ranking function in Equation 2. First, there are the independent importance weights $\lambda(\kappa)$ that parameterize the local factors (see Equation 4). Second, there are the importance weights $\lambda(\kappa, K^Q)$ that assign weight to a concept, while taking into account the rest of the concepts in the query (see Equation 5).

In this section, we consider two possible parameterization schemes for these concept weights. In Section 2.5.1, we consider parameterization by structure. Conversely, in Section 2.5.2, we examine parameterization by concept.

#### 2.5.1 Parameterization By Structure

A simple way to parameterize the importance weights $\lambda(\kappa)$ and $\lambda(\kappa, K^Q)$, is to make the assumption that the weights of all the concepts in the same structure are tied. Formally:

$$\forall \kappa_i, \kappa_j \in \sigma : \lambda(\kappa_i) = \lambda(\kappa_j) = \lambda(\sigma)$$

This assumption has the benefit of significantly reducing the number of free parameters in the retrieval model, thereby greatly simplifying the estimation process. Due to its simplicity, parameterization by structure is often used in the log-linear retrieval models [14, 27, 32].

Using parameterization by structure and the definitions of local and global factors in Section 2.4, we can explicitly rewrite the ranking function in Equation 2 as

$$sc(Q, D) = \sum_{\sigma \in \Sigma^Q} \lambda(\sigma) \sum_{\kappa \in \sigma} f(\kappa, D) + \max_{\pi \in \Pi_D} \sum_{\sigma \in \Sigma^Q} \lambda(\sigma, \Sigma^Q) \sum_{\kappa \in \sigma} f(\kappa, \pi). \quad (6)$$

#### 2.5.2 Parameterization By Concept

The main drawback of parameterization by structure is the fact that it implies that all the concepts in the same structure are equally important for expressing the query intent. This implication is not always true, especially for more verbose, natural language queries, which may benefit from assigning varying concept weights [2, 4, 22].

Therefore, we may wish to remove the restriction imposed in the previous section, and parameterize the concept weights based on the concepts themselves rather than their respective structures. Assigning a single weight to each concept is clearly infeasible, since the number of concepts is exponential in the size of the vocabulary. Therefore, we take a parameterization approach proposed in recent work on query modeling [2, 4, 5, 22, 35, 38], and represent each concept using a combination of importance features, $\Phi$, described in Table 2. These importance features are based

<table>
<thead>
<tr>
<th>Feature Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>GF($\kappa$)</td>
</tr>
<tr>
<td>WF($\kappa$)</td>
</tr>
<tr>
<td>CF($\kappa$)</td>
</tr>
<tr>
<td>DF($\kappa$)</td>
</tr>
</tbody>
</table>

Table 2: Concept importance features $\Phi$.  

The main drawback of parameterization by structure is the fact that it implies that all the concepts in the same structure are equally important for expressing the query intent. This implication is not always true, especially for more verbose, natural language queries, which may benefit from assigning varying concept weights [2, 4, 22].

Therefore, we may wish to remove the restriction imposed in the previous section, and parameterize the concept weights based on the concepts themselves rather than their respective structures. Assigning a single weight to each concept is clearly infeasible, since the number of concepts is exponential in the size of the vocabulary. Therefore, we take a parameterization approach proposed in recent work on query modeling [2, 4, 5, 22, 35, 38], and represent each concept using a combination of importance features, $\Phi$, described in Table 2. These importance features are based
on concept frequencies, and can be efficiently computed and cached, even for large-scale collections.

Using these importance features, we can explicitly rewrite the ranking function in Equation 2 as

\[
sc(Q, D) = \sum_{\sigma \in \Sigma^Q} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D) + \
\max_{\pi \in \Pi_D} \sum_{\sigma \in \Sigma^Q} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma, \Sigma^Q) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, \pi).
\]

(7)

### 2.5.3 Parameter Estimation

To estimate the free parameters \(\lambda(\cdot)\) in Equation 6 and Equation 7, we rely on a large and growing body of literature on the learning to rank methods for information retrieval (see Liu [23] for a survey). As a base algorithm for parameter optimization we make use of the coordinate ascent (CA) algorithm proposed by Metzler and Croft [28].

The CA algorithm iteratively optimizes a target metric (in our case, retrieval metric such as MAP) by performing a series of one-dimensional line searches. It repeatedly cycles through each of the parameters \(\lambda(\cdot)\), while holding all other parameters fixed. This process is performed iteratively over all parameters until the gain in the target metric is below a certain threshold.

We use the CA algorithm primarily for its simplicity, efficiency and effectiveness, as demonstrated by the previous work [4, 5, 27]. However, any other learning to rank approach that estimates the parameters for linear models such as RankSVM [15] or RankNet [7] can be adopted as well.

To ensure the scalability of our retrieval model, we compute the global factor (Equation 5) only for the top thousands of documents retrieved by the local factors (Equation 4). Therefore, the setting of the importance weights \(\lambda(\kappa)\) will affect the document ranking, which, in turn, will affect the choice of the highest-scoring passages and subsequently the setting of the importance weights \(\lambda(\kappa, K^Q)\).

Accordingly, we perform the optimization in two stages. We decompose \(sc(Q, D)\) into its local and global components. First, we optimize the local component (i.e., the weights \(\lambda(\kappa)\)). Then, we fix the weights of the local component, and optimize the global component (i.e., the weights \(\lambda(\kappa, K^Q)\)). Each of these optimizations is done using the standard CA algorithm.

### 3. RELATED WORK

In this paper we describe a general retrieval framework that models dependencies between arbitrary query concepts using a query hypergraph. It is important, therefore, to examine the connections between some of the well known retrieval models and query hypergraphs.

#### 3.1 Bag-of-Words Models

As Zobel and Moffat [46] point out, the majority of the standard bag-of-words models in IR can be generally expressed by the following summation:

\[
sc(Q, D) \triangleq \sum_{t \in Q} \lambda(t, Q) f(t, D),
\]

where \(\lambda(t, Q)\) and \(f(t, D)\) are some arbitrary functions (which may include normalization constants) defined over a query term \(t\) and its occurrences in the query and the document, respectively. Examples of such models include, among others, the query likelihood model [33], BM25 [34] and divergence from randomness [1].

Therefore, it is easy to show that all of these bag-of-words models can be straightforwardly modeled using a query hypergraph. To induce such a hypergraph, we simply need to define a single QT-structure \(\sigma_1 = \{t_1, t_2, \ldots\}\), and a set of local edges

\[
E = \{(t, D) : t \in \sigma_1\}.
\]

#### 3.2 Passage Retrieval

There is a long history of passage-based retrieval models in information retrieval [3, 8, 9, 16, 41, 40, 24]. These retrieval models are typically defined using vector space models [9, 16, 17] or language models [2, 24, 40], and employ a simple bag-of-words query representation. One of the most common passage retrieval techniques is Max-Psg, which uses the passage with the highest score for document score derivation [3, 8, 16, 24, 41].

Max-Psg with the bag-of-words query representation is a special case of the general query hypergraph described in this paper. Our model combines the recent advances in retrieval models that go beyond the bag-of-words query representations with passage retrieval models.

In addition, it is important to mention some recent work on query expansion [21] and query reformulation [43] using passage-based evidence, which uses hierarchical graphical representation of the query, similar to the one presented in this paper. This work is orthogonal to ours, as it uses passage evidence to augment the query with new concepts, rather than to model the query and the retrieval function. Combining this work on query expansion and reformulation with the retrieval models based on query hypergraphs is a promising direction for future work.

#### 3.3 Term Dependencies

The advent of large-scale web corpora encouraged the development of retrieval models that employ phrases and proximity matches to model term dependencies [27, 29, 39, 14, 26, 32]. Most of these retrieval models take a log-linear form, and can be modeled using a query hypergraph with the structures described in Section 2.3.1, but without the inclusion of the global hyperedge.

Retrieval models that employ term dependencies usually resort to parameterization by structure [27, 14, 39, 32] (as described in Section 2.5.1). While this assumption significantly reduces the number of the free parameters in the retrieval model, it may be detrimental to the performance of verbose natural language queries that may contain concepts of variable importance.

Recently, researchers started to examine retrieval models that employ parameterization by concept. To avoid learning a separate weight for each concept, these models represent a concept using a set of features [4, 5, 22, 35, 38]. This approach significantly outperforms parameterization by structure, especially for verbose natural language queries. Accordingly, we also employ parameterization by concept in the retrieval with query hypergraphs (see Section 2.5.2).

#### 3.4 Higher-Order Term Dependencies

To the best of our knowledge, there is very little prior work on retrieval with higher-order term dependencies (i.e., de-
dependencies between arbitrary concepts rather than terms). One notable exception is an early work on generalized term dependencies by Yu et al. [44], which derives higher-order dependencies from pairwise term dependencies. However, the model proposed by Yu et al. [44] is infeasible for large-scale collections, since it requires an explicit computation of the probability of relevance for each individual query term, as well as pairs and triples of query terms.

A more recent retrieval model that attempts to incorporate higher-order term dependencies is the Full Dependence (FD) variant of the Markov random field model proposed by Metzler and Croft [27]. The FD model, however, is only able to capture dependencies between multiple terms, rather than multiple concepts. For instance, it can model a dependency between the terms in the triple *(dog, law, enforcement)*, but it cannot model a dependency between the pair of concepts *(dog, "law enforcement")*. 

## 4. EVALUATION

### 4.1 Experimental Setup

All the empirical evaluation described in this section is implemented using Indri, an open-source search engine [37]. The structured query language implemented in Indri natively supports multiple types of concepts, including exact phrases and proximity matches, as well as customizable concept weighting schemes. As a result, Indri provides a flexible and convenient platform for evaluating the retrieval performance of query hypergraphs.

Table 4 presents a summary of the TREC corpora used in our experiments. The corpora vary both by type *(Robust04)* is a newswire collection, *Gov2* is a crawl of the .gov domain, and *ClueWeb-B* is a set of pages with the highest crawl priority derived from a large web corpus), number of documents, and number of available topics, thereby providing a diverse experimental setup for assessing the robustness of retrieval with query hypergraphs.

<table>
<thead>
<tr>
<th>Name</th>
<th># Docs</th>
<th>Topic Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust04</td>
<td>528,185</td>
<td>301-450, 601-700</td>
</tr>
<tr>
<td>Gov2</td>
<td>25,205,179</td>
<td>701-850</td>
</tr>
<tr>
<td>ClueWeb-B</td>
<td>50,220,423</td>
<td>1-100</td>
</tr>
</tbody>
</table>

Table 4: Summary of the TREC collections and topics used for evaluation.

For the Robust04 and Gov2 collections, a standard Porter stemmer is used. In contrast, the ClueWeb-B collection is stemmed using the Krovetz stemmer, which is a “light” stemmer, as it makes use of inflectional linguistic morphology [19] and is especially suitable for web collections where aggressive stemming can decrease precision at top ranks [31]. Stopword removal is performed on both documents and queries using the standard INQUERY stopword list. The free parameter *µ* in the concept matching function *f*(*κ*, *X*) (see Equation 3) is set according to the default Indri configuration of the Dirichlet smoothing parameter.

Since query hypergraphs attempt to capture complex dependencies between query concepts, we apply them to the description portions of the TREC topics. TREC topic descriptions express the information needs behind the topics using verbose natural language queries. For instance, a description portion of the TREC topic entitled “hydrogen as a feasible energy source?” is a question “What is the status of research on hydrogen as a feasible energy source?”.

In order to compute the global factor (Equation 5), we segment each document into semi-overlapping passages of 150 words (i.e., the overlap between the passages is 75 words). As shown in previous work on passage retrieval [3, 9, 8, 16], this passage configuration leads to improved effectiveness on most TREC collections.

The optimization of the free parameters for all the baselines and the proposed retrieval methods is done using three-fold cross-validation with mean average precision (MAP) as the target metric. In addition to MAP, we also report ERR@20, an early precision metric that was adopted as the official retrieval performance metric at the TREC 2010 Web Track [10]. The statistical significance of differences in the performance of the proposed retrieval methods with respect to their respective baselines is determined using a two-sided Fisher’s randomization test [36] with 25,000 permutations and α < 0.05.

### 4.2 Retrieval Performance Evaluation

In this section, we compare the performance of the retrieval with query hypergraphs to a number of state-of-the-art baselines that incorporate exact phrase matches, proximities, and concept weight parameterization. These baselines do not, however, incorporate concept dependencies.

The query hypergraph representation, proposed in this paper, further extends each of these baselines with higher-order term dependencies via the inclusion of the global hyperedge and the corresponding global factor *φ*(*K*, *D*) (see Equation 5). In the remainder of this section, we examine the improvements in the retrieval performance (or lack thereof) of these baselines when they are extended with the query hypergraph representation.

#### 4.2.1 Query Likelihood Baseline

Query likelihood [33] is a popular retrieval method that employs a bag-of-words query representation. In this section, we juxtapose the retrieval performance of the query likelihood baseline (denoted QL) to the performance of a query hypergraph that includes a single QT-structure (structure that contains the individual query terms as concepts). We denote this hypergraph representation *H*-QL. This juxtaposition demonstrates the contribution of the global factor *φ*(*K*, *D*) (see Equation 5) to the retrieval performance.

Table 3(a) shows the comparison between the QL and the *H*-QL methods. The results in Table 3(a) demonstrate that the addition of the global factor *φ*(*K*, *D*) into a bag-of-words representation significantly improves its retrieval effectiveness in all the cases.

Note that the *H*-QL method is equivalent to the Max-Psg method proposed in the previous work [3, 8, 9, 16, 41], which ranks the documents in the collection by a combination of the document score and the score of its highest-scoring passage. The improvements in retrieval performance shown in Table 3(a) are in line with the improvements attained by the Max-Psg method reported in this previous work.

#### 4.2.2 Markov Random Fields Baselines

Markov random fields for information retrieval (MRF-IR) is a state-of-the-art retrieval framework that incorporates...
term dependencies. It was first proposed by Metzler and Croft [27], and was shown to be highly effective, especially for large-scale web collections.

Metzler and Croft propose two instantiations of the general MRF-IR framework. The first instantiation is the sequential dependence model (denoted SD), which incorporates only dependencies between adjacent query terms. The second instantiation is the full dependence model (FD), which incorporates dependencies between all query term subsets. The SD and FD baselines can be extended with a respective hypergraph that includes three structures: QT, PR and PH (refer to Section 2.3.1 for the exact definitions of these structures). We denote these hypergraph representations $H_{SD}$ and $H_{FD}$, respectively. These hypergraphs are parameterized by structure, and their ranking functions are derived according to Equation 6.

Table 3(b) compares the performance of the sequential dependence baseline (SD) and its corresponding hypergraph $H_{SD}$. As evident from Table 3(b), in most cases (except for the ClueWeb-B collection) the retrieval effectiveness (in terms of MAP) is significantly improved by the hypergraph extension. However, these improvements are smaller than in the case of the QL baseline.

Similarly, Table 3(c) compares the performance of the full dependence baseline (FD) and its corresponding hypergraph $H_{FD}$. Comparing Table 3(b) and Table 3(c), we can see that in most cases the FD baseline slightly outperforms the SD baseline. However, these differences were not found to be statistically significant.

When comparing the performance of the FD baseline and its corresponding hypergraph $H_{FD}$, Table 3(c) demonstrates that the inclusion of the global factor results in an improved retrieval effectiveness (in terms of MAP) for all collections, and in statistically significant improvements for the Robust04 and Gov2 collections.

In addition, we can compare between the retrieval performance of the hypergraphs $H_{SD}$ and $H_{FD}$. Similarly to the case of the baselines SD and FD, no statistically significant differences were found in the performance of these hypergraphs. $H_{FD}$ is slightly more effective for the ClueWeb-B and the Gov2 collections, while being slightly less effective for the Robust04 collection.

### 4.2.3 Weighted Sequential Dependence Model

A major drawback of the SD and the FD baselines is that they use the parameterization-by-structure approach (see Section 2.5.1), which ties the importance weights $\lambda(\cdot)$ of all the concepts that belong to the same structure (i.e., all the terms, phrases and proximities get the same respective weights). This parameterization can be detrimental, especially for longer, more verbose queries that may mix concepts of differing importance.

Recently, Bendersky et al. [4] proposed a weighted variant of the sequential dependence model (denoted WSD) that overcomes this drawback. The concept weights in the WSD method are parameterized using a set of importance features, associated with each concept based on its respective structure, as described in Section 2.5.2.

We extend the WSD baseline with a query hypergraph $H_{WSD}$. The $H_{WSD}$ includes the global factor $\phi(K^Q, D)$, which is also parameterized by concept. The ranking function for the $H_{WSD}$ hypergraph is presented in Equation 7.

Table 3(d) compares the retrieval performance of the WSD baseline and its corresponding hypergraph $H_{WSD}$. While the retrieval improvements that stem from this hypergraph extensions are not as pronounced as in the cases of the QL, SD
and FD baselines, the addition of the global factor to the WSD baseline still results in effectiveness gains for all the collections and most of the metrics. For instance, for the Gov2 collection, the \(H\rightarrow WSD\) method improves the performance (in terms of MAP) for 60% of the queries compared to the WSD baseline, while hurting only 30% of the queries. For 7% of the queries MAP is improved by more than 25%, while there is a 25% drop in performance for only 2% of the queries.

4.2.4 Further Analysis

In addition to comparing each individual query hypergraph model to its respective baseline, some general trends can be observed in Table 3. First, it is interesting to compare the relative differences in gains across the baselines, when the global factor is added. The gains are the largest for the QL baseline, which does not include any term dependencies, and decrease as more term dependencies are added by the SD and the FD baselines. As an example, for the Gov2 collection, the effectiveness gain as a result of the global factor inclusion decreases from 6.2% for the QL baseline to 1.6% for the FD baseline.

These diminishing returns demonstrate that there is some degree of overlap between the effect of term dependencies and higher-order term dependencies on the retrieval effectiveness. The overlap is not complete, however, since the addition of the global factor still has a statistically significant impact on the retrieval performance in most cases. This is true even for the FD baseline, which includes term dependencies between all query term pairs and triples.

Finally, we note that the parameterization of the ranking function by concept (as in the WSD baseline) (a) significantly improves the retrieval performance of the ranking function parameterized by structure (as in the SD baseline), and (b) further diminishes the gains obtained through the inclusion of the global factor. While \(H\rightarrow WSD\) is the best-performing retrieval method (in terms of MAP) in Table 3, its average effectiveness gain over the WSD baseline is only 1.3%.

For comparison, the average effectiveness gain of the \(H\rightarrow QL\) method over the QL baseline is 4.7%.

4.3 Parameterization Analysis

In this section we analyze the parameterization of the query hypergraph. We examine both parameterization-by-structure and parameterization-by-concept regimes, which are described in detail in Section 2.5.1 and Section 2.5.2, respectively.

Recall that the parameters of the query hypergraph are optimized using the coordinate ascent algorithm such that the ranking function is decomposed into local and global factors (see Section 2.5.3). In this section, due to the space constraints, we focus our attention on the resulting parameterization for the Robust04 collection. We choose this collection, since it has the largest number of queries, and the learned parameterization is stable across all folds. However, it is important to note that the findings in this section hold for the other two collections as well.

4.3.1 Parameterization by Structure

Table 5 shows the hypergraph parameters for the local factors (\(\lambda(\sigma)\) and the global factor (\(\lambda(\sigma, \Sigma^G)\)), averaged across folds, when the parameterization-by-structure approach is used (see Equation 6). These parameters correspond to the \(H\rightarrow SD\) model, the results for which are shown in Table 3(b).

<table>
<thead>
<tr>
<th>(\lambda(\sigma))</th>
<th>(\lambda(\sigma, \Sigma^G))</th>
</tr>
</thead>
<tbody>
<tr>
<td>QT: +0.129</td>
<td>+0.322</td>
</tr>
<tr>
<td>PH: +0.065</td>
<td>+0.017</td>
</tr>
<tr>
<td>PR: +0.065</td>
<td>-0.011</td>
</tr>
</tbody>
</table>

Table 5: Query hypergraph parameterization by structure (Robust04 collection).

<table>
<thead>
<tr>
<th>(\varphi)</th>
<th>(\lambda(\varphi, \sigma))</th>
<th>(\lambda(\varphi, \sigma, \Sigma^G))</th>
</tr>
</thead>
<tbody>
<tr>
<td>GF: -0.007</td>
<td>QT: 0</td>
<td>PH: +0.005</td>
</tr>
<tr>
<td>WF: +0.017</td>
<td>+0.007</td>
<td>+0.002</td>
</tr>
<tr>
<td>QF: +0.012</td>
<td>0</td>
<td>+0.008</td>
</tr>
<tr>
<td>CF: -0.021</td>
<td>0</td>
<td>-0.001</td>
</tr>
<tr>
<td>DF: -0.018</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AP: +0.540</td>
<td>+0.029</td>
<td>+0.298</td>
</tr>
</tbody>
</table>

Table 6: Query hypergraph parameterization by concept (Robust04 collection).

Note that both for the local and the global factors the weights assigned to the term structure (QT) are the highest, which is in line with other models that incorporate term dependencies [27]. This demonstrates that despite the importance of term dependencies, individual term occurrences are still the most important indicators of relevance.

In addition, in Table 5, the parameters of the local factors are weighted higher than the parameters of the global factor. Recall that the global factor is defined over the highest-scoring passage in the document. Thus, the lower weight of the global factor parameters in line with previous work, where passage evidence is typically weighted lower than the document evidence [3, 41, 16].

Finally, note the negative weight assigned to the proximity (PR) structure in the global factor. While small, this negative weight is consistent across folds, as well as in the other collections. Intuitively, this negative weight indicates that in the highest-scoring passage of the relevant document we expect to encounter exact phrase concepts, rather than unordered proximity concepts.

4.3.2 Parameterization by Concept

Table 6 shows the hypergraph parameters for the local factors (\(\lambda(\varphi, \sigma)\)) and the global factor (\(\lambda(\varphi, \sigma, \Sigma^G)\)), averaged across folds, when the parameterization-by-concept approach is used (see Equation 7). These parameters correspond to the \(H\rightarrow WSD\) model, the results for which are shown in Table 3(d). For the convenience of presentation and to reduce weight sparsity, we combine the weights of the PR and PH structures in the PR+PH column.

Note that a priori constant importance feature AP generally receives the highest weight. This is due to the fact that setting all the other feature weights to zero yields exactly the parameterization-by-structure approach.

Features such as document frequency (DF), collection frequency (CF) and Google frequency (GF) receive, as expected, negative weights in most cases. In contrast, the query frequency (QF) and the Wikipedia title frequency (WF) features get positive weights, which indicates that the appearance of the concept in page title or in a search query is positively correlated to the concept importance.
5. CONCLUSIONS

The retrieval framework proposed in this paper represents a query by means of a hypergraph. In the query hypergraph, each vertex corresponds to a concept, and these concepts are grouped into disjoint structures. A hyperedge in the query hypergraph represents a concept dependency. We describe a principled derivation of a ranking function based on the factorization of the query hypergraph. We then propose two parameterization regimes for the derived ranking function, based on either structures or concepts.

The proposed retrieval framework exhibits three important characteristics. First, it models term dependencies as concepts. Second, it models dependencies between these concepts (i.e., higher-order term dependencies). Finally, it assigns weights to concepts and concept dependencies, proportionate to their importance for expressing the query intent. For verbose natural queries, the proposed retrieval framework significantly improves the retrieval effectiveness of several state-of-the-art retrieval methods that do not incorporate higher-order term dependencies.

6. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by ARRA NSF IIS-9014442. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

7. REFERENCES