Incorporating Hierarchical Domain Information to Disambiguate Very Short Queries

Hamed Bonab
University of Massachusetts Amherst
bonab@cs.umass.edu

John Foley
Smith College
jfoley@smith.edu

Mohammad Aliannejadi
Università della Svizzera italiana (USI)
mohammad.alian.nejadi@usi.ch

James Allan
University of Massachusetts Amherst
allan@cs.umass.edu

ABSTRACT
Users often express their information needs using incomplete or ambiguous queries of only one or two terms in length, particularly in the Web environments. The ambiguity of short queries is a recognized problem for information retrieval (IR) systems. In this study, we investigate various approaches for incorporating hierarchical domain information into IR models such that the domain specification resolves the ambiguity. To this end, we develop practical models for constructing evaluation datasets from existing corpora. In terms of effectiveness, we further study the trade-off between a short query and its domain specification information. In doing so, we find that domains with the highest number of relevant documents are not always the best ones to select. We also evaluate the utility of a domain hierarchy and find that incorporating the hierarchical structure of a collection into the retrieval model could have a high impact on short query disambiguation.

ACM Reference Format:

1 INTRODUCTION
The way in which a user describes a desired information need is through a query. It is intuitive that retrieval performance is highly dependent on the quality of the submitted query and that the quality of the query varies for a wide range of reasons. Belkin et al. [4], for example, in their work on human-centered information retrieval and the hypothesis of Anomalous State of Knowledge (ASK), realized that in many cases, users of search systems are unable to precisely formulate their queries as they lack some vital knowledge such as vocabulary and the need of being more general or more specific. In such cases, it is more suitable to attempt to describe a user’s anomalous state of knowledge than to ask the user to specify their need as a request to the system. Figure 1 presents an example of such a situation where the user has submitted the query “Amazon” to the system. As we see, the user’s actual information need is relevant to “Amazon Rain Forest,” and the incorporated domain hierarchy is utilized in the disambiguation process.

In this study, we investigate the ambiguity of very short queries and the incorporation of hierarchical domain information for disambiguation purposes. We explore the effect of the domain hierarchy of a query in cases where it is already specified or inferred from a user’s actions. In particular, we take the first step in studying the effect of such information on retrieval performance, assuming that such information can be obtained effectively. Our contributions can be summarized as follows:

- We propose a methodology for constructing an evaluation dataset from existing corpora, and develop several approaches to creating query-domain pairs from existing collections and make two sample collections publicly available.
- We explore some representative models for incorporating domain information into a retrieval model such as: leveraging domain hierarchy, using domain taxonomy of documents, and the domain name.
- We study the effect of incorporating domain knowledge into a retrieval model while investigating its optimal trade-off and effectiveness on query disambiguation.

1https://ciir.cs.umass.edu/download
The experimental results and analyses provide useful insights and intuitions into the task of query disambiguation and the potential impact of hierarchical domain knowledge on retrieval performance.

2 RELATED WORK

The problem of short query disambiguation is a long-lasting problem in the information retrieval literature [5, 17]. Two main categories of solutions are implicitly and explicitly addressing the diversity problem [15]. Many studies used word sense for short query disambiguation [19]. However, the effectiveness of these methods—particularly for very short queries—is questionable, given that for this type of queries there are no additional words for providing context information to clear up the confusion [3, 19]. Another well-studied solution is the clustering of retrieved document set [6]. The existing literature on clustering either investigates the clustering directly or explores issues with clustering, especially within interactive search [2, 12, 13, 17]. There are some more recent studies focused on using user-specific features, like search log data [14]. Here, we focus on studying the effect of domain knowledge.

Other studies have explored the modeling of concept hierarchies for providing a hierarchical map of words [3, 10, 11, 16]. More recently, much work has been done on search result diversification based on facet extraction [9] as well as incorporating hierarchical domain information [7, 18]. Search result diversification is a well-studied technique that aims to cover multiple facets of ambiguous and faceted queries in the result page, so that the user is able to find documents that are relevant to their actual information need. Our work is distinguished from this line of research as it analyzes the effect of hierarchical domain knowledge on disambiguation of very short queries, which can potentially be beneficial both for search result diversification and other tasks such as query suggestion [8] and conversational search [1].

3 DATA COLLECTION

Corpus. All experiments in this study are conducted using data derived from the TREC 2017 Core Track dataset, denoted by $Q_{trec}$. It has over 1.8 million New York Times (NYT) annotated articles. Each article is classified under a set of domains, specified as nodes within a hierarchy of labels. The domain hierarchy presented in Figure 1 shows some example domains taken from the corpus. For example the selected domain (D) is “Top/Travel/Destinations/Central & South America.” There is an average of four taxonomic labels for each article. These tags were automatically assigned and manually verified by the NYTimes.com production staff.

Queries. To the best of our knowledge, there is no available judged set of ambiguous query and hierarchical domain pairs. For our experiments, we designed a procedure for constructing pairs of a very short query (Q) together with a single hierarchical domain (D) specification. To this end, we used the 50 Core Track topics that were judged by both the NIST assessors and crowd workers. We appointed four expert annotators and instructed them to convert each query to a short ambiguous query, coupled with a domain specification. We emphasized generating the data in a way such that one could reconstruct the original query with the generated query and the domain specification. Therefore, we retained the original TREC relevance judgments. Next, we describe two strategies that we followed to construct the dataset.

From domain to query (Rel). The first strategy uses the relevance judgments to find the domains with the most relevant documents for each query. We construct a domain tree whose nodes identify the domain of relevant documents for a given query. Therefore, for every query, each node refers to a number of relevant documents. We rank the nodes of the tree based on two factors: (i) Coverage of the node in relation to the all relevant documents (similar to the recall measure) as measured by (1) and (ii) Specificity, in terms of the depth of the node in the relevant tree as presented in (2).

$$\text{Coverage(node)} = \frac{\text{relevant documents in the node}}{\text{total relevant documents}}$$

$$\text{Specificity(node)} = \frac{\text{depth(node)}}{\text{max depth of the relevant tree}}$$

Coverage captures nodes with many relevant documents, thus maximizing recall. Specificity, on the other hand, aims to optimize precision because in the hierarchy, the deeper a node is, the more precise it becomes. Therefore, non-relevant documents are less likely to appear in more specific nodes. For every query, we calculate a score for each node using the harmonic mean of these two factors and consider the highest scoring node as the domain specifier.

For collecting the ambiguous queries, we presented $Q_{trec}$ and the domain specifiers to the annotators. Then we asked them to generate short ambiguous queries (possibly a single term). Note that we instructed the annotators to take into account the domain specifier and generate the queries such that the domain specifier could disambiguate the generated query (i.e., the original query can be guessed or reconstructed). For instance, we see in Figure 1 that the original query “Amazon Rain Forest” is given to an annotator and the generated short query is “Amazon,” presuming that together with its domain specifier the original query can be inferred.

From query to domain (RetRel). This approach is an effort to bias results toward prototypical system outputs. First, we show $Q_{trec}$ to the annotators and asked them to generate a very short ambiguous query for each of the 50 topics. Unlike the Rel collection, we do not show the annotators the domain specifiers; therefore, the query is not guaranteed to align with a domain. We retrieved 1K documents for each generated query using a flat retrieval model (see Section 4). Similar to the Rel collection, we define a ranking function based on the ranking position of the relevant documents as well as the retrieved ones. Specifically, based on these two sets of documents (i.e., retrieved and relevant), we construct a tree whose nodes contain a list of retrieved and relevant documents. Then for each node, we calculate the recall and precision and rank the nodes based on the harmonic mean ($F_1$) of the scores.

From this ranked list of plausible nodes, we instructed the annotators to select a proper domain specifier from the top five ranked domains. For instance, the original query “Radio Waves and Brain Cancer”, an annotator generated “Radio Waves” as the short query.

\footnote{https://trec-core.github.io/2017/}

\footnote{Containing nearly every published article between January 1, 1987 and June 19, 2007}

\footnote{https://catalog.ldc.upenn.edu/LDC2008L19}

\footnote{We define the specificity, since otherwise the root node would always be selected.}
While all of these techniques could be adapted to the settings of our domain, we explain two retrieval models that incorporate the domain knowledge. Here, we study incorporating the hierarchical structure of the corpus for retrieval. One could think of several solutions for retrieval. Since there is no guarantee that all the relevant documents are only in the given domain node, we introduce a smoothing parameter to leverage the domain hierarchy. We re-score the documents based on the maximum number of relevant documents, the local retrieval model helps the model to diversify the results. Therefore, giving more weight to the global retrieval model is less diverse. Therefore, giving more weight to the global retrieval model helps the model to diversify the results. This supports the hypothesis that incorporating the hierarchical information is more effective than a flat retrieval strategy with query expansion. Figure 2, right plot, shows the impact of the hierarchical information on the expanded query. We see that that better results for the Rel set are achieved as we give more weight to the global retrieval model; whereas the results on the RetRel dataset is less impacted. Considering flat retrieval results for both of the collections, we observe a low retrieval performance when only Q is used. However, with adding D the retrieval performance improves significantly. Moreover, we see that (Q + D)RetRel exhibits a higher performance gain, as opposed to (Q + D)Rel. This suggests that expert annotators are more effective in identifying useful domain specifiers, as opposed to the domains that extracted based on the number of relevant documents. Moreover, it indicates that selecting the most relevant domain specifier reduces the ability of the model to diversify the results, hence resulting in lower improvement. Considering the hierarchical retrieval results for QRel and QRetRel, retrieval performances are significantly improved compared to the flat retrieval scenarios on each query-domain set. This is an interesting finding and suggests that respecting the hierarchy could lead to a higher retrieval performance.

Impact of hierarchical structure. Figure 2, left plot, shows the impact of the hierarchical structure on the retrieval performance. For both query-domain sets, the model performs better with $\alpha < 0.5$, indicating the high importance of the query terms in the domain node compared to the whole collection. On the other hand, hierarchical retrieval with expansion, i.e., $Q + D$, do not exhibit significant improvement compared to short queries. Even for the Rel set, adding the domain label decreases the retrieval performance, reducing the values of $\text{P@10}$, $\text{P@20}$, and $\text{rel}_\text{rel}$. This suggests the hypothesis that incorporating the hierarchical information is more effective than a flat retrieval strategy with query expansion. Figure 2, right plot, shows the impact of the hierarchical information on the expanded query. We see that that better results for the Rel set are achieved as we give more weight to the global retrieval model; whereas the results on the RetRel dataset is less impacted by the global retrieval. This suggests that since the Rel was created based on the maximum number of relevant documents, the local retrieval model is less diverse. Therefore, giving more weight to the global retrieval model helps the model to diversify the results more effectively; hence achieving higher performance. Comparing the two plots of Figure 2 shows that the model that expand queries (i.e., the right plot) is less sensitive to the values of $\alpha$, compared to

$$f(q, d) = \sum_{w \in q, d} c(w, q) \log (1 + \frac{c(w, d)}{\mu p(w|D)}) + |q| \log \frac{\mu}{\mu + |d|}$$

$$f(q, d)_{\text{BP}} = \alpha f(q, d)_C + (1 - \alpha)f(q, d)_{\text{ID}}$$

### 4 DOCUMENT RETRIEVAL SCENARIOS

We explain two retrieval models that incorporate the domain knowledge for improved retrieval. One could think of several solutions already proven to be effective in the literature—Latent Dirichlet Allocation (LDA), relevance weighting of query terms, incorporation of external knowledge bases like WordNet and Wikipedia [2]. While all of these techniques could be adapted to the settings of our problem, our main focus is on unique features of a hierarchical domain such as the selected label text and the hierarchical taxonomy of the collection.

#### 4.1 Flat Retrieval

Here, we study the gain that the label of a selected domain can bring about. We expand the query with its domain label at the root node of the hierarchy by adding the label terms to the query terms. For a given query $q$ and document $d$, the ranking function is defined based on the Dirichlet Prior Smoothing.

$$f(q, d)_C = \sum_{w \in q, d} c(w, q) \log (1 + \frac{c(w, d)}{\mu p(w|C)}) + |q| \log \frac{\mu}{\mu + |d|}$$

where $c(w)$ is the term frequency function and $p(w|C) = (\sum_{w' \in C} c(w', C))$ is the maximum likelihood estimate of word $w$ in the collection. $C$, $\mu \in [0, +\infty)$ is the smoothing parameter.

#### 4.2 Hierarchical Retrieval

Here, we study incorporating the hierarchical structure of the corpus for retrieval. Since there is no guarantee that all the relevant documents are only in the given domain node, we introduce a smoothing parameter to leverage the domain hierarchy. We retrieve a ranked list of documents categorized with the domain’s leaf node as local retrieval by the scoring function of (4), and combine it with a corpus-wide ranking of documents as global retrieval, presented in (3). For a given query $q$, domain $D$, and document $d$, the scoring function is a bi-polarization between local and global retrieval, as shown in (5). It uses the smoothing parameter $\alpha \in [0, 1]$ as the trade-off in the effectiveness of local and global retrieval. When $\alpha = 1$ it can be seen as flat global retrieval and $\alpha = 0$ presents flat local retrieval. $p(w|D) = (c(w, D) + \epsilon)/(\sum_{w' \in D} c(w', D))$ is the maximum likelihood estimate of word $w$ in the given domain node.$^6$

$$f(q, d, D) = \sum_{w \in q, d} c(w, q) \log (1 + \frac{c(w, d)}{\mu p(w|D)}) + |q| \log \frac{\mu}{\mu + |d|}$$

$$f(q, d, D)_{\text{BP}} = \alpha f(q, d)_C + (1 - \alpha)f(q, d, D)_{\text{ID}}$$

### 5 RESULTS AND DISCUSSIONS

#### Experimental setup.

For our experiments, we extended the Galago toolkit and used it for document indexing, query retrieval, and incorporating the domain. We report MAP, Prec@10, Prec@20, and the average number of relevant retrieved documents. For pairwise comparisons, we determine statistically significant differences using the two-tailed paired $t$-test with a 95% confidence interval ($p < 0.05$).

#### Performance comparison: flat vs. hierarchical.

Table 1 presents the experimental results for flat and hierarchical retrieval models. Considering flat retrieval results for both of the collections, we observe a low retrieval performance when only $Q$ is used. However, with adding $D$ the retrieval performance improves significantly. Moreover, we see that $(Q + D)_{\text{RetRel}}$ exhibits a higher performance gain, as opposed to $(Q + D)_{\text{Rel}}$. This suggests that expert annotators are more effective in identifying useful domain specifiers, as opposed to the domains that extracted based on the number of relevant documents. Moreover, it indicates that selecting the most relevant domain specifier reduces the ability of the model to diversify the results, hence resulting in lower improvement. Considering the hierarchical retrieval results for $Q_{\text{Rel}}$ and $Q_{\text{RetRel}}$, retrieval performances are significantly improved compared to the flat retrieval scenarios on each query-domain set. This is an interesting finding and suggests that respecting the hierarchy could lead to a higher retrieval performance.

#### Table 1: Experimental Results. The superscript * denotes significant differences ($p < 0.05$)

<table>
<thead>
<tr>
<th>Query Set</th>
<th>$\alpha$</th>
<th>MAP</th>
<th>P@10</th>
<th>P@20</th>
<th>rel_ret</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q_{\text{Rel}}$</td>
<td>1.0</td>
<td>0.114</td>
<td>0.248</td>
<td>0.234</td>
<td>61.16</td>
</tr>
<tr>
<td>$(Q + D)_{\text{Rel}}$</td>
<td>1.0</td>
<td>0.142</td>
<td>0.244</td>
<td>0.252</td>
<td>63.56</td>
</tr>
<tr>
<td>$Q_{\text{RetRel}}$</td>
<td>1.0</td>
<td>0.110</td>
<td>0.242</td>
<td>0.227</td>
<td>58.32</td>
</tr>
<tr>
<td>$(Q + D)_{\text{RetRel}}$</td>
<td>1.0</td>
<td>0.171*</td>
<td>0.336*</td>
<td>0.316*</td>
<td>62.44*</td>
</tr>
<tr>
<td>Hier.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q_{\text{Rel}}$</td>
<td>0.3</td>
<td>0.170</td>
<td>0.360</td>
<td>0.352</td>
<td>70.21*</td>
</tr>
<tr>
<td>$(Q + D)_{\text{Rel}}$</td>
<td>0.8</td>
<td>0.162</td>
<td>0.288</td>
<td>0.273</td>
<td>67.58</td>
</tr>
<tr>
<td>$Q_{\text{RetRel}}$</td>
<td>0.2</td>
<td>0.185</td>
<td>0.412*</td>
<td>0.375*</td>
<td>68.94</td>
</tr>
<tr>
<td>$(Q + D)_{\text{RetRel}}$</td>
<td>0.6</td>
<td>0.189*</td>
<td>0.368</td>
<td>0.350</td>
<td>66.28</td>
</tr>
<tr>
<td>* CombRel</td>
<td>0.5</td>
<td>0.191</td>
<td>0.364</td>
<td>0.365</td>
<td>72.15</td>
</tr>
<tr>
<td>CombRetRel</td>
<td>0.5</td>
<td>0.194*</td>
<td>0.416*</td>
<td>0.375*</td>
<td>76.34*</td>
</tr>
</tbody>
</table>

To make the probability always non-zero, a very small number, $\epsilon = 0.001$, is added to the term frequency in the domain.
We investigated the incorporation of hierarchical domain information into document retrieval models to tackle the problem of short query ambiguity. We proposed a practical solution to construct an evaluation dataset from existing corpora and developed two approaches for creating a query and domain pair from a longer and less ambiguous query. We studied the trade-off in effectiveness between the effect of short query and the domain specification on the performance and found that domain knowledge that is based on the highest number of relevant documents can have the opposite effect on the performance as it reduces the ability of the model to diversify the results. Moreover, we studied the impact of incorporating a piece of hierarchical domain knowledge into the retrieval model and found that using such information can be beneficial for short query disambiguation as it conveys rich knowledge of the domain. In the future, we plan to expand our data collection methodology to other domains and use data collections that are aimed for search result diversification (e.g., TREC Web track). We plan to test our hypotheses on neural retrieval approaches such as the ones based on huge contextual pre-trained models (e.g., BERT).

**Acknowledgements.** This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Air Force Research Laboratory (AFRL) and IARPA under contract #FA8650-17-C-9118 under subcontract #14775 from Raytheon BBN Technologies Corporation. 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.

**REFERENCES**


