Incorporating Hierarchical Domain Information to Disambiguate Very Short Queries

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

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

John Foley
Smith College
jjfoley@smith.edu

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:
Hamed Bonab, Mohammad Aliannejadi, John Foley, and James Allan. 2019. Incorporating Hierarchical Domain Information to Disambiguate Very Short Queries. In *The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19), October 2–5, 2019, Santa Clara, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3341981.3344251

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.

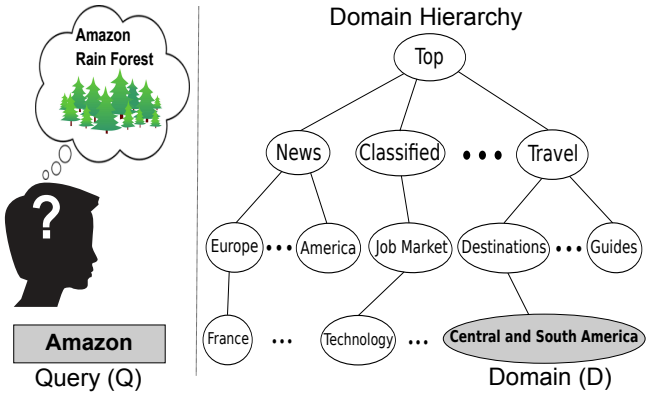

Figure 1: An example of very short query disambiguation using hierarchical domain information.

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.[1]
- 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.

[1] https://ciir.cs.umass.edu/download

ICTIR '19, October 2–5, 2019, Santa Clara, CA, USA

ACM ISBN 978-1-4503-6881-0/19/10…$15.00
https://doi.org/10.1145/3341981.3344251

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,[2] denoted by Q_{trec}. It has over 1.8 million New York Times (NYT) annotated articles[3]. 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[4].

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).[5]

$$\textit{Coverage}(\text{node}) = \frac{|\text{relevant documents in the node}|}{|\text{total relevant documents}|} \tag{1}$$

$$\textit{Specificity}(\text{node}) = \frac{\text{depth(node)}}{\text{max depth of the relevant tree}} \tag{2}$$

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.

[2] https://trec-core.github.io/2017/

[3] Containing nearly every published article between January 1, 1987 and June 19, 2007

[4] https://catalog.ldc.upenn.edu/ldc2008t19

[5] We define the specificity, since otherwise the root node would always be selected.

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

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

Then, based on the ranked list of candidate domains, the annotator selected "*Top/News/Health/Diseases, Conditions & Health Topics/Brain Cancer/*" as the proper domain specifier.

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.

$$f(q,d)_C = [\sum_{w \in q,d} c(w,q) \log [1 + \frac{c(w,d)}{\mu p(w|C)}]] + |q| \log \frac{\mu}{\mu + |d|} \quad (3)$$

where $c(w,.)$ is the term frequency function and $p(w|C) = c(w, C)/\sum_{w' \in C} c(w', C)$ 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].

$$f(q,d,D)_D = [\sum_{w \in q,d} c(w,q) \log [1 + \frac{c(w,d)}{\mu p(w|D)}]] + |q| \log \frac{\mu}{\mu + |d|} \quad (4)$$

$$f(q,d,D)_{BP} = \alpha f(q,d)_C + (1 - \alpha) f(q,d,D)_D \quad (5)$$

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 $(Q + D)_{RetRel}$ 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 Q_{Rel} and Q_{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.

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 P@10, P@20, and rel_ret. 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 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 α, compared to

[6]To make the probability always non-zero, a very small number, $\epsilon = 0.001$, is added to the term frequency in the domain.

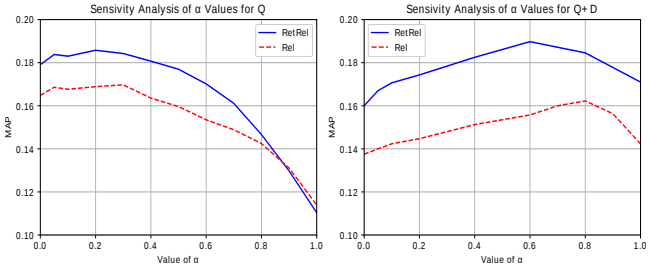

Figure 2: Impact of the hierarchical structure on the retrieval performance (i.e., α).

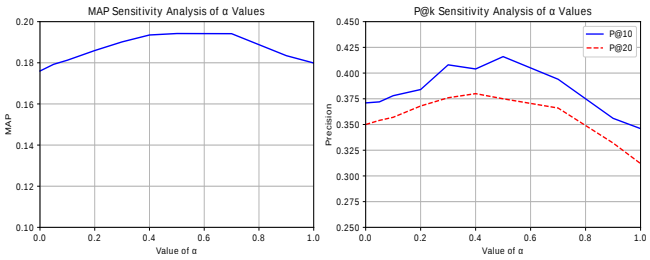

Figure 3: Sensitivity analysis of α values for $Comb_{RetRel}$

the model that retains the short query but incorporates the hierarchical information (i.e., the left plot). This behavior is expected since expanded queries are already disambiguated to some extent; therefore, incorporating global retrieval has a far less effect on its performance. Short queries, on the other hand, can be interpreted in various ways, so higher values of α can have a negative impact as it could cause misinterpretation of the query, giving a high relevance score to the non-relevant facets of the query.

Performance of combined retrieval. Based on these experiments and analyses, one might conclude that the expanded query model is performing better for global retrieval while the short query model is performing better for a local retrieval. To test this hypothesis, we designed a hierarchical retrieval variant in which the query for the local retrieval scoring function is only a short query and the query for global retrieval scoring function is the expanded query. We call this model *Comb* in our experiments, as it is a combination of short and expanded queries. The experimental results in Table 1 for *Comb* query sets demonstrate the validity of this hypothesis in our experiments. For all the evaluation measures, $Comb_{RetRel}$ outperforms other methods significantly. $Comb_{Rel}$ also performs significantly better compared to Rel set performances for flat and hierarchical scenarios. Moreover, Figure 3 presents the effect of the *alpha* parameter on the performance of $Comb_{RetRel}$. We see that $\alpha = 0.5$ is nearly the best value for all the evaluation measures, indicating both global and local retrieval scores contribute equally to the performance of the combined method.

6 CONCLUSIONS AND FUTURE WORK

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

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *SIGIR*. ACM.
[2] James Allan, Anton Leuski, Russell Swan, and Donald Byrd. 2001. Evaluating combinations of ranked lists and visualizations of inter-document similarity. *Information Processing & Management* 37, 3 (2001), 435–458.
[3] James Allan and Hema Raghavan. 2002. Using part-of-speech patterns to reduce query ambiguity. In *SIGIR*. ACM, 307–314.
[4] Nicholas J Belkin, Robert N Oddy, and Helen M Brooks. 1982. ASK for information retrieval: Part I. background and theory. *Journal of documentation* 38, 2 (1982), 61–71.
[5] J. Bhogal, Andrew MacFarlane, and P. Smith. 2007. A review of ontology based query expansion. *Inf. Process. Manage.* 43, 4 (2007), 866–886.
[6] Christopher Boston, Hui Fang, Sandra Carberry, Hao Wu, and Xitong Liu. 2014. Wikimantic: Toward effective disambiguation and expansion of queries. *Data Knowl. Eng.* 90 (2014), 22–37.
[7] Sha Hu, Zhicheng Dou, Xiao-Jie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search result diversification based on hierarchical intents. In *CIKM*. ACM, 63–72.
[8] Makoto P. Kato and Katsumi Tanaka. 2016. To suggest, or not to suggest for queries with diverse intents: Optimizing search result presentation. In *WSDM*. ACM, 133–142.
[9] Weize Kong and James Allan. 2016. Precision-oriented query facet extraction. In *CIKM*. ACM, 1433–1442.
[10] Dawn J. Lawrie and W. Bruce Croft. 2000. Discovering and comparing topic hierarchies. In *RIAO*. CID, 314–330.
[11] Dawn J. Lawrie, W. Bruce Croft, and Arnold L. Rosenberg. 2001. Finding topic words for hierarchical summarization. In *SIGIR*. ACM, 349–357.
[12] Anton Leuski. 2001. Evaluating document clustering for interactive information retrieval. In *CIKM*. ACM, 33–40.
[13] Anton Leuski. 2001. *Interactive information organization: Techniques and evaluation.* Technical Report. Univ. of Massachusetts Amherst, Dept. of Computer Science.
[14] Lilyana Mihalkova and Raymond J. Mooney. 2009. Learning to disambiguate search queries from short sessions. In *ECML/PKDD (2) (Lecture Notes in Computer Science)*, Vol. 5782. Springer, 111–127.
[15] Mark Sanderson. 2008. Ambiguous queries: Test collections need more sense. In *SIGIR*. ACM, 499–506.
[16] Mark Sanderson and Dawn Lawrie. 2002. Building, testing, and applying concept hierarchies. In *Advances in information retrieval*. Springer, 235–266.
[17] Ellen M. Voorhees. 1993. Using WordNet to disambiguate word senses for text retrieval. In *SIGIR*. ACM, 171–180.
[18] Xiao-Jie Wang, Ji-Rong Wen, Zhicheng Dou, Tetsuya Sakai, and Rui Zhang. 2018. Search result diversity evaluation based on intent hierarchies. *IEEE Trans. Knowl. Data Eng.* 30, 1 (2018), 156–169.
[19] Zhi Zhong and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *ACL (1)*. The Association for Computer Linguistics, 273–282.