Hierarchical Target Type Identification for Entity-oriented Queries

Krisztian Balog
Krisztian.Balog@idi.ntnu.no

Robert Neumayer
Robert.Neumayer@idi.ntnu.no

Department of Computer and Information Science
Norwegian University of Science and Technology
Sem Sælands vei 7-9,Trondheim, Norway

ABSTRACT

A significant portion of information needs in web search target entities. These may come in different forms or flavours, ranging from short keyword queries to more verbose requests, expressed in natural language. We address the task of automatically annotating queries with target types from an ontology. The identified types can subsequently be used, e.g., for creating semantically more informed query and retrieval models, filtering results, or directing the requests to specific verticals. Our study makes the following contributions. First, we formalise the task of hierarchical target type identification, argue that it is best viewed as a ranking problem, and propose multiple evaluation metrics. Second, we develop a purpose-built test collection by hand-annotating over 300 queries, from various recent entity search benchmarking campaigns, with target types from the DBpedia ontology. Finally, we introduce and examine two baseline models, inspired by federated search techniques. We show that these methods perform surprisingly well when target types are limited to a flat list of top level categories; finding the right level of granularity in the hierarchy, however, is particularly challenging and requires further investigation.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

Keywords

Entity retrieval, semantic search, query classification

1. INTRODUCTION

A better understanding and processing of user queries is of vital interest to a number of information management and retrieval tasks. A good deal of effort has been invested in recent years into various aspects of this topic, including query segmentation [24], named-entity recognition [12, 19], semantic tagging [18], structural annotation [6], intent discovery [16, 26], and topical classification [5, 13].

In this paper, we focus on queries targeting entities. These typically come in two flavours: (i) looking for a specific entity, or (ii) asking for a list of entities that are of a particular type or class. It has been shown that more than 50% of queries in web search fall into one of these categories [20]. Complementing keyword queries with explicit type information—a scenario studied at the TREC and INEX Entity Ranking tracks [4, 9]—has been shown to significantly improve retrieval performance. Target types could be used, among others, for building semantically more informed query and retrieval models [3, 15], filtering results [8, 21], or identifying relevant verticals [1, 26]. In practice, however, the scenario assumed at the above benchmarking initiatives, i.e., the user specifying the target type, is a rather unrealistic one; common web users prefer simple interfaces with a single search box. This motivates the need for automatic methods for the target type identification of entity-oriented queries.

In this paper, we introduce the task of *hierarchical target type identification*: given a query, identify the type of relevant results with respect to a given ontology. Specifically, we aim to find the single most specific type within the ontology that is general enough to cover all relevant entities. There are two key differences between this task and prior work on the topic of classifying entity types of queries [15, 25]: (i) our types are not a flat list, but are organised into a hierarchical structure, and (ii) we require "instance of" relations between the target type and relevant entities, instead of mere "relatedness." Finding types with the appropriate granularity or specificity opens up a number of novel application possibilities, for example, in faceted browsing or result presentation.

The hierarchical target type identification task can naturally be formulated as a ranking problem and evaluated using standard information retrieval metrics. However, taking the correctness of results to be a binary decision would not account for "near misses," such as returning items that are too general or too specific. Therefore, we also consider a lenient evaluation, in which types on the same path with the correct answer are also rewarded.

We develop a purpose-built test collection by taking a large number of queries from various recent entity search benchmarking campaigns and hand-annotating them with target types from the DBpedia ontology. Finally, we propose and examine two baseline models, inspired by federated search techniques. We find that even simple baselines can perform very well on identifying types from a flat list, while the hierarchical case proves to be challenging.

In summary, this paper makes the following contributions: (1) we identify the task of hierarchical query type identification, (2) we develop a test set and evaluation methodology, (3) we introduce two baseline methods and perform an experimental evaluation. The resources we developed (query set and type annotations) are made publicly available at `http://bit.ly/SdpbZh`.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

2. RELATED WORK

Query type classification has been studied for web document retrieval to categorise searches according to their geographical locality [11], goal (such as informational or navigational) [14], vertical intent (e.g., product, image, video) [1, 16], or topicality [17]. Of these, vertical intent discovery is the closest to our task in spirit; however, intents are usually limited to a handful a categories (2 in [16] and 18 in [1]) and are not hierarchically organised. A related task, question classification in a community-based question answering portal is presented in [23]. The authors use a three-level hierarchy of categories, however, questions are only associated with leaf level categories.

Little work has been done on classifying entity types of queries. Vallet and Zaragoza [25] introduce the *entity type ranking* task: find the most important types related to the query results. Their approach ranks passages, extracts entities from them, and use the types associated with these entities. There are two important differences between the task in [25] and ours: (1) they consider multiple target types that are related to the query, but the query does not necessarily have to fall into any of them, and (2) they use a flat set of (64) types. Kaptein et al. [15] rank entities in Wikipedia and assign Wikipedia categories automatically to the query by considering the most frequent categories associated with the top 10 results. Again, they do not consider the hierarchical structure of categories (due to the fact that categorisation in Wikipedia is not a well-defined "is-a" hierarchy).

3. PROBLEM STATEMENT

We formulate the problem of *hierarchical target type identification* as follows:

> Given an entity-oriented input query, find the single most specific type from an ontology that is general enough to cover all entities that are relevant to the query.

By entity-oriented queries we mean information needs where the user's intent is to find (i) a specific entity or (ii) a list of entities that are of a particular type or class. It has to be noted that, even for queries with a very clear entity intent, it may not be possible to identify a single common category, apart from the root concept ("Thing" in the DBpedia ontology), as we will show in Section 4. For our task, however, we only consider queries for which there exists a clearly preferred target type; automatic identification of these queries is a non-trivial exercise, and an interesting problem for future research, but it is outside the scope of this work.

We cast the hierarchical target type identification task as a ranking problem (rather than a classification one) as this allows us to give credit for "near-misses," i.e., types that are too specific or too general, instead of merely treating them as incorrect. Moreover, this makes it possible for us to gain a better understanding of both the task and the developed models, as we are not focusing only on a single class label, but also consider the other types returned for the query.

Our task is then summarised as follows:

- INPUT: a query q and an ontology O.[1]
- OUTPUT: a ranked list of types $(t_1, \ldots, t_n)$ where $t_i \in O$.
- EVALUATION: each returned type t_i is labeled with a score (independently of t_j, $i \neq j$) with respect to q. The individual type scores are aggregated, based on their position in the ranked list, into a single score for the query.

4. EVALUATION METHODOLOGY

In this section we introduce the set collection we developed for our task.

4.1 Queries

We collected queries from a number of recent benchmarking evaluation efforts:

- 120 from the TREC Entity track (2009-2011); these focus on specific relationships between entities (e.g., "Airlines that currently use Boeing 747 planes") [4].
- 55 from the INEX Entity Ranking track (2009) that seek a list of entities (e.g., "US presidents since 1960") [9].
- 142 from the Semantic Search Challenge (2010-2011) entity search task, which refer to one particular entity, albeit often an ambiguous one (e.g., "Ben Franklin," which is both a person and a ship) [7].
- 50 from the Semantic Search Challenge (2011) list search task; these, again, target a group of entities that match certain criteria (e.g., "Axis powers of World War II") [7].

As seen from the examples above, these queries cover a broad range of information needs related to entities and amount to a good number of queries to experiment with (a total of 367). Moreover, these topic sets share a peculiarity with pragmatic importance: (a significant portion of) known relevant answers come either from DBpedia or can relatively easily be mapped to DBpedia. This could aid us in the type annotation process, as we will explain next.

4.2 Target Type Annotation

Queries were labelled with types from the DBpedia ontology (version 3.7). The ontology contains 358 categories (out of which only 282 are actually used), organised into a hierarchy of 6 levels. The root category element of all types (on the 0th level) is *Thing*; this, we disregarded from the set of possible types, as it would have no practical value for an actual application. There are 32 categories on the first level of the hierarchy (including *Person*, *Organisation*, *Work*, *Species*, etc.); we refer to these as *top-level types*.

Annotation was done manually by the two authors of the paper. For each query we looked at the known relevant results (and the topic narrative, where available) to clarify the intent. It is important to emphasise that the target type was chosen with respect to the query intent, not based on the qrels (which, occasionally, were found to contain errors). The guiding principle was to pick a single type that is as specific as possible, yet general enough to cover all correct answers. This was possible in 67% of the cases. The remainder of the queries could not be used for three main reasons:

- The query falls into multiple top-level categories and only the root *Thing* class would be general enough to cover them all. This was typically the case for many of the Semantic Search Challenge queries that targeted one particular entity. For example, "Ben Franklin" is a *Person*, but could also refer to the ship (*MeanOfTransportation*) or the musical (*Work*).
- The type of entity targeted in the query cannot be mapped to any class within the ontology. For example, the query "food seed brands belonging to Monsanto" looks for brands; the closest concept is *Organisation*, but it would be incorrect to classify it as an such (or as any sub-class of it).
- In a few cases, it was not possible to decide (or agree on) the query intent or it was not entity-related, e.g., "bookwork" or "banana paper making."

[1] Our notion for an ontology here is simply taxonomic: a hierarchical categorisation of types (or classes) of entities.

Table 1: Overview of query annotation.

Query type	Count	Percentage
Single target type	262	71.3%
Multiple (top-level) target types	30	8.2%
Target type missing from ontology	46	12.5%
Query not interpretable	29	7.9%
Total	367	100%

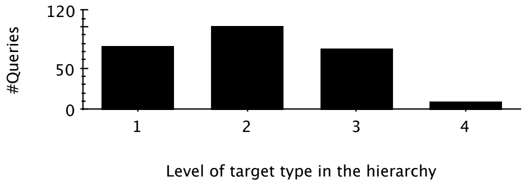

Figure 1: Distribution of target types over the hierarchy.

Table 1 contains a summary.

Figure 1 displays the distribution of target types over the levels of the hierarchy. Interestingly, for more than 70% of the queries the target type lies beyond the top-level categories and can go as far as 4 levels deep.

4.3 Evaluation Metrics

We consider two types of evaluation: (i) *strict*, where judgments are binary, crediting only the correct answer, and (ii) *lenient*, where the relevance scores are graded and near-misses are also rewarded.

For strict evaluation we use mean reciprocal rank (MRR) and success rate at the top rank (S@1). For lenient evaluation we measure the degree of misclassification, as opposed to merely measuring correctness, by considering the distance between the returned type (t) and the correct type (t_q) in the hierarchy. We set the distance function $d(t, t_q)$ to be the number of steps between two types in the hierarchy, if they lie on the same path (which is 0 if $t = t_q$) and to ∞ otherwise. We then turn this distance function into a gain measure $G(t)$ by considering linear and exponential decay functions. If $d(t, t_q) = \infty$ we take $G(t)$ to be 0, otherwise:

- *Linear:* $G(t) = 1 - d(t, t_q)/h$, where h is the depth of the hierarchy (6 in our case).
- *Exponential:* $G(t) = b^{-d(t,t_q)}$, where b is the base of the exponent (which we set to 2).

This way the correct type has $G(t_q) = 1$, more specific and more general types on the same path are rewarded proportional to their distance to the target type, and all other types get $G(t) = 0$. Using the gain values defined above, we compute normalized discounted cumulative gain (nDCG) at two rank cutoff points: 1 and 5.

5. BASELINES

Types have no direct textual representation, apart from their label. To be able to rank them with respect to their relevance to an input query, we rely on entities from a knowledge base that are associated with the given type. A parallel can be drawn between this task and that of ranking resources (collections) in a federated search setting; each type can be considered as a collection of entities and our target is to provide a relevance ranking of these collections (representing types). Note that the same analogy can also be made to other well-studied information retrieval tasks, namely expert finding [2] and blog distillation [10, 22]. In all these cases, two principal approaches are used: (1) representing types as a single "large document," by concatenating all entity descriptors associated with the type, and (2) treating entities as individual retrieval units and aggregating their retrieval scores into a type-level ranking.

Table 2: Strict evaluation with binary judgments.

	Top-level only		Hierarchical			
Model	MRR	S@1	MRR	S@1	nDCG@1	nDCG@5
Type-centric	0.5275	0.3469	0.2987	0.1918	0.1918	0.3089
Entity-centric	0.6951	0.5020	0.3507	0.1633	0.1633	0.3967

We assume that each entity e in the knowledge base has a textual description, e_d, and a set of types, $e_t = (e_t^1, \ldots, e_t^n)$, associated with it, where the types come from an ontology ($e_t^i \in O$). Further, it is assumed that if an entity is assigned to a given type t then it is also assigned to all ancestors of t; for example, if an entity is of type *President*, then it is also of types *Politician* and *Person*. In principle, an entity could be assigned to types that are on different paths within the hierarchy, but in practice that is rarely the case.

For each input query, consisting of a sequence of terms, $q = (w_1, \ldots, w_{|q|})$, and for each possible type in the ontology, $t \in O$, we estimate the probability that the query was generated by the given type, $P(q|t)$, and rank types in decreasing order of this probability. Next, we formalise the two strategies discussed above using language modeling techniques.

Type-centric model. For each type we build a single large document by concatenating the descriptions of all entities that are labelled with that type. Once such a pseudo-document is generated for each type, we can rank types much like documents. Following the standard language modeling approach, we put:

$$P(q|t) = \prod_{i=1}^{|q|} P(w_i|\theta_t) = \prod_{i=1}^{|q|} ((1-\lambda)P(w_i|t) + \lambda P(w_i)), \quad (1)$$

where θ_t is the type language model, computed as a mixture of an empirical model, $P(w_i|t)$, and a background language model, $P(w_i)$. The latter is a standard maximum-likelihood estimate; the former is estimated by aggregating the term probabilities from all entities of that type:

$$P(w|t) = \sum_{e:t \in e_t} P(w|e_d)P(e|t), \quad (2)$$

where $P(w|e_d)$ is the maximum likelihood estimate of term w in the document representation of entity e; $P(e|t)$ is the probability of an entity given a type. For the sake of simplicity, we take this to be uniform, i.e., $P(e|t) = 1/|\{e : t \in e_t\}|$.

Entity-centric model. Instead of creating a direct term-based representation of types, our second approach models and queries individual entities, then aggregates their relevance estimates:

$$P(q|t) = \sum_{e:t \in e_t} P(q|e)P(e|t). \quad (3)$$

The probability of the query given the entity is estimated using a standard query likelihood scoring for document language modeling: $P(q|e) = \prod_{i=1}^{|q|} P(t|\theta_{e_d})$, where θ_{e_d} is a smoothed language model of the entity description. As before, $P(e|t)$ is set uniformly across all entities labeled with t.

6. RESULTS AND ANALYSIS

As a first step, we perform strict evaluation, where the judgments are binary and there is a single correct answer. Table 2 reports the results in terms of mean reciprocal rank (MRR) and success rate

Table 3: Lenient evaluation rewarding near-misses.

	Linear decay		Exponential decay	
Model	**nDCG@1**	**nDCG@5**	**nDCG@1**	**nDCG@5**
Type-centric	0.3265	0.3440	0.2612	0.3287
Entity-centric	0.4143	0.4542	0.2939	0.4173

at rank 1 (S@1).[2] In one set of experiments, we limit ourselves to finding the top-level type for each query (columns 2–3). The results show that even simple baselines can be successful in performing this task with high accuracy. Then, we address the hierarchical version of the type identification task. Not surprisingly, the numbers are much lower here (columns 3–7). One interesting finding is that the type-centric model can more often return the correct type at the top rank (S@1) than the entity-centric approach, while in overall the latter method is more effective.

Next, in Table 3, we present results for the lenient evaluation. We test two ways of accounting for near-misses: linear decay (columns 2–3) and exponential decay (columns 4–5). The latter one is increasingly less tolerant as the distance increases between the returned type and the correct type; therefore, absolute values are lower for this metric. The relative differences between the two runs, however, are found to be stable for both metrics and cutoff values. Compared to the strict case (Table 2), we observe substantial improvements in terms of nDCG@1, while the improvements for nDCG@5 are moderate. This indicates that the top ranked type is often on the same path with the correct answer, however it is not of the right granularity. When examining the types returned at the level of individual queries, we observe some interesting differences between the two approaches. The type-centric model tends to return more specific categories, whereas the entity-centric model rather assigns more general types. This is is indeed the expected behaviour, considering the strategies underlying these methods.

The numbers indicate that the entity-centric model is a clearly preferred choice; this is also in line with findings on the resource selection task in federated search. Our query-level observations, however, suggest that the two approaches should be combined; one possibility for future work would be to use a discriminative framework that employs both entity-level and type-level features.

7. CONCLUSIONS

In this paper we introduced the task of hierarchical target type identification for entity-oriented queries. We outlined the relevance of the task to a range of IR problems, developed an evaluation methodology and a test set based on a number of query sets used in recent entity-oriented benchmarking initiatives. Building on approaches from resource selection in federated search, we proposed two baseline approaches and performed an experimental evaluation. Our main finding is that even simple baselines can perform surprisingly well, when target types are limited to a flat list. The hierarchical case, however, has proven to be more difficult. Our analysis revealed that the top ranked type is often on the same path with the correct answer, but it is not of the right granularity.

Throughout this paper we focused on queries for which a clearly preferred target entity type was available. This was ensured through a manual selection of queries. In future work we will investigate automatic means of making this selection, i.e., deciding whether the query has a clearly defined target type or not.

[2] Note that in case of a single relevant result S@1 is the same as nDCG@1. Nevertheless, we included both to help comparing the numbers in Tables 2 and 3.

References

[1] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *Proc. of SIGIR'09*, pages 315–322, 2009.

[2] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Inf. Process. Man.*, 45(1):1–19, 2009.

[3] K. Balog, M. Bron, and M. De Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.*, 29:22:1–22:31, 2011.

[4] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2011 entity track. In *Proc. TREC'11*, 2012.

[5] S. M. Beitzel, E. C. Jensen, O. Frieder, D. D. Lewis, A. Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *Proc. of ICDM'05*, pages 42–49, 2005.

[6] M. Bendersky, W. B. Croft, and D. A. Smith. Structural annotation of search queries using pseudo-relevance feedback. In *Proc. of CIKM '10*, pages 1537–1540, 2010.

[7] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. T. Duc. Entity search evaluation over structured web data. In *Proc. of EOS'11*, 2011.

[8] M. Bron, K. Balog, and M. de Rijke. Ranking related entities: Components and analyses. In *Proc. of CIKM'10*, pages 1079–1088, 2010.

[9] G. Demartini, T. Iofciu, and A. de Vries. Overview of the inex 2009 entity ranking track. In *Focused Retrieval and Evaluation*, volume 6203, pages 254–264. 2010.

[10] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *Proc. of SIGIR'08*, pages 347–354. ACM, 2008.

[11] L. Gravano, V. Hatzivassiloglou, and R. Lichtenstein. Categorizing web queries according to geographical locality. In *Proc. of CIKM'03*, pages 325–333, 2003.

[12] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proc. of SIGIR'09*, pages 267–274, 2009.

[13] B. J. Jansen and D. Booth. Classifying web queries by topic and user intent. In *Proc. of CHI EA'10*, pages 4285–4290, 2010.

[14] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *Proc. of SIGIR'03*, pages 64–71, 2003.

[15] R. Kaptein, P. Serdyukov, A. De Vries, and J. Kamps. Entity ranking using wikipedia as a pivot. In *Proc. of CIKM'10*, pages 69–78, 2010.

[16] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *Proc. of SIGIR '08*, pages 339–346, 2008.

[17] Y. Li, Z. Zheng, and H. K. Dai. Kdd cup-2005 report: facing a great challenge. *SIGKDD Explor. Newsl.*, 7(2):91–99, 2005.

[18] M. Manshadi and X. Li. Semantic tagging of web search queries. In *Proc. of ACL '09*, pages 861–869, 2009.

[19] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In *Proc. of CIKM'07*, pages 683–690, 2007.

[20] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proc. of WWW'10*, pages 771–780, 2010.

[21] S. Schlobach, M. Olsthoorn, and M. De Rijke. Type checking in open-domain question answering. In *Proc. of ECAI'04*, pages 398–402, 2004.

[22] J. Seo and W. B. Croft. Blog site search using resource selection. In *Proc. of CIKM'08*, pages 1053–1062, 2008.

[23] A. Singh and K. Visweswariah. Cqc: classifying questions in cqa websites. In *Proc. of CIKM'11*, pages 2033–2036, 2011.

[24] B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. In *Proc. of WWW'08*, pages 347–356, 2008.

[25] D. Vallet and H. Zaragoza. Inferring the most important types of a query: a semantic approach. In *Proc. of SIGIR'08*, pages 857–858, 2008.

[26] K. Zhou, R. Cummins, M. Halvey, M. Lalmas, and J. M. Jose. Assessing and predicting vertical intent for web queries. In *Proc. of ECIR'12*, pages 499–502, 2012.