Retrieval and Ranking of Semantic Entities for Enterprise Knowledge Management Tasks

Chad Cumby
Accenture Technology Labs
161 N Clark St.
Chicago, IL, USA
chad.m.cumby@accenture.com

Katharina Probst
Google, Inc.
Atlanta, GA, USA
katharina.probst@gmail.com

Rayid Ghani
Accenture Technology Labs
161 N Clark St.
Chicago, IL, USA
rayid.ghani@accenture.com

ABSTRACT

We describe a task-sensitive approach to retrieval and ranking of semantic entities, using the domain information available in an enterprise. Our approach utilizes noisy named-entity tagging and document classification, on top of an enterprise search engine, to provide input to a novel ranking metric for each entity retrieved for a task. Retrieval is query-centric, where the user query is the target topic (e.g., a technology needed for a proposal). Named entities are then extracted from the retrieved documents, and ranked according to their similarity to the target topic. We evaluate our approach by comparing to a baseline retrieval and ranking technique that is based on entity occurrence rates, and show encouraging results.

Keywords

Enterprise search , Metadata IR, Information Extraction

1. INTRODUCTION

Current knowledge management systems not only consist of documents, but also of a variety of semantic entities. People, employees, companies, clients, projects, partners, alliances, locations, and competitors are just some examples of such entities. When a worker performs a specific enterprise task, one or more of these semantic entities are often required to fulfill that task. Writing proposals for new contracts with various companies, finding experienced workers within the company to work on new projects, or evaluating different third-party vendor capabilities with respect to various project requirements are just some enterprise tasks that require the use of entities mentioned above. In general, various knowledge management tasks can be greatly simplified or assisted by delivering relevant semantic entities to the person performing them.

Currently, it is very difficult to retrieve such information: while any commercial enterprise search engine will yield a list of relevant documents, they are not currently able to retrieve a reliable ranked list of semantic entities such as companies or experts. Information extraction has focused on extracting certain kinds of entities but Search and IR work has typically focused on ranking of documents. In the cases where entity search has been studied in the IR community [2, 1] it has been restricted to expert finding with specialized heuristics, without utilizing a general class of semantic information. This is limiting in several ways, and we argue that augmenting search systems in order to retrieve semantic entities specific to a given task and ranking them dynamically based on the needs of the user is essential for many enterprise tasks. In order to enable more general entity retrieval and ranking, we present here a system that retrieves and ranks semantic entities that can be specific to a user query, i.e., a topic such as 'Business Intelligence'.

An important difference between content on the Web versus in the enterprise is that in the latter, business processes often produce extensive meta-data about each document in the enterprise knowledge base. This can include straightforward information such as the creator of a document or the time the document was submitted, but also includes a lot of semantic knowledge about the domain such as client companies or locations that are relevant for the document. Using this meta-data to incorporate semantics into search engines is at the center of our approach. With it, we create a more representative profile of a semantic entity to be used in ranking, compared to simple occurance counts of the entity in the corpus.

2. RETRIEVAL AND RANKING

We query the document search engine for the topic t that was specified by the user and use the retrieved documents to extract candidate entities. Our base document search engine indexes automatically extracted entities such as companies, people, keywords, locations, acronyms, etc. as well as manually given entities such as project client, project contact, etc. from all documents in a specific set, associated with occurrence frequency.

The algorithm proceeds by collecting the entities of the desired type, i.e., people in our example, that were extracted for all the returned documents. Each candidate entity e_i is associated with a count cnt_{e_i}. This count indicates how many of the returned documents contain the entity. We first order all candidates by *cnt* and consider only those entities in the top n (100 in our experiments) by occurrence.

For example, let t be a topic query of interest to the user, e.g., *CRM* or *BP drilling*. Let $type_{e_c}$ be the candidate entity type, e.g., people. The query will result in a document set R_t. We then create set of candidate enti-

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.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

ties $E_t = \{e_1, e_2, \ldots e_n\}$ using document counts as described above, up to the predetermined n (100 in our experiments).

After our system retrieves a set of candidate entities $E_t = \{e_1, e_2, \ldots e_n\}$, it next ranks them according to their relevance to the topic t.

In order to compare the candidate entities e_i to the target topic, we again use the metadata that is associated with each document. We first create a model of the target topic by considering all the metadata associated with the result document set for this topic. More formally, we again formulate the query q_t to submit to the document search engine, which returns the document set R_t. For this set of documents $d_j \in R_t$ we construct a multi-dimensional vector of metadata $\mathbf{v}_t$ that models R_t. For our method, we assume independence between metadata categories.

Each metadata value is associated with a count of the number of documents that this value pertains to in the current retrieved set, the count cnt_{mv_i}. In other words, for a query t, the counts represent the number of documents in R_t having this metadata value, e.g., being tagged with a specific location. We then use all the counts to construct a normalized vector $\mathbf{v}_t$ for the target topic, with $v_{t,i} = \frac{cnt_{mv_i}}{|R_t|}$.

In the next step, we use the vector to determine the angle between it and a similar vector constructed for each candidate entity. We create a new query for each candidate entity e_i by concatenating the original topic t to e_i and passing it to our document search engine as query $t + e_i$. The candidate vectors are constructed from the result set R_{t+e_i}. Again, we use all counts associated with all metadata values, and construct the vector $\mathbf{v}_{t+e_i}$ from this. The number of dimensions, i.e., the number of metadata values, is denoted by s. We then compute the cosine distance between each vector $\mathbf{v}_{t+e_i}$ and the target vector $\mathbf{v}_t$ as follows: $d(\mathbf{v}_{t+e_i}, \mathbf{v}_t) = \frac{\sum_j v_{t,j} * v_{t+e_i,j}}{||\mathbf{v}_t|| * ||\mathbf{v}_{t+e_i}||}$

Finally, we rank entities from lowest to highest distance, i.e., $argmin_{e_i \in E_t}(d(\mathbf{v}_{t+e_i}, \mathbf{v}_t))$ will be ranked highest.

3. EVALUATION

The main task of our evaluation is to gauge the relevance of our ranked list of entities, in the metadata-rich enterprise environment. Our experiment compares our performance against two simpler non-metadata based baseline methods on an expert-finding task using an annotated database of skills within our company, augmented with an expertise survey, as the ground truth.

In the experiment, we compare our system (which we refer to in Table 1 as **META**) to a document frequency-based baseline method, in which we rank the candidate entities by the number of hits in the document collection co-occurring with the topic query (refered to as **COUNT**). We also compare to a random baseline, where we randomly permute the rank of the relevant (retrieved) candidate entities (**RAND**).

This experiment aims at finding people within our company who are experts in various domain areas. To obtain a set of expertise data to test our retrieval and ranking system against, we employed a database of 575 specific skills self-selected by about 100000 employees within the company.

For each skill in our test set, we query the document search engine with the skill name, and retrieve the top 100 entities from the result set as described earlier, ordered by entity/topic co-occurrence. We then rank each set of 100 using our metadata based ranking (META) , as well as the topic/entity co-occurrence count based ranking (COUNT) , and finally with a random ranking (RAND). We use several different metrics to compare the performance of our ranking method to the two baselines: mean average precision (MAP), mean reciprocal rank (MRR), precision at several different list size cutoffs (P@x), and two metrics that measure pairwise misrankings (rpref & bpref).

	MAP	rpref	bpref	MRR	P@5	P@10
META	.30	.61	.59	.25	.20	.21
COUNT	.28	.57	.55	.24	.17	.18
RAND	.24	.52	.50	.22	.14	.15

Table 1: Performance for ranking experts extracted from the annotated 'contact' field.

In Table 1 we show the performance of our system (META) versus the baseline methods (COUNT and RAND). For the set of candidate experts retrieved from document metadata and via Named Entity extraction from the documents themselves, our metadata-based re-ranking procedure increases precision in all the relevant metrics over the COUNT and RAND baselines, especially within the first 5-10 results. In addition, the MRR results on our test set show our method on average returns a first expert higher in search results than the two baselines.

4. CONCLUSION

In conclusion, we presented a query-specific semantic entity ranking system that is useful for a number of tasks that are relevant to enterprises. Our system is designed to be general enough to retrieve and rank any type of entity such as people, companies, locations, or other more exotic types of entities. Our algorithm's generality, however, also means that it is not tuned to any specific entity, much unlike previous work in expert retrieval. We would expect that tuning our system to a specific type of entity would result in higher relevance of the retrieved results, leaving room for future improvement by specialization. Our current evaluation shows that even the general system that is not tuned to any type of entity improves over two baselines, by taking advantage of metadata that is either given manually, or automatically extracted by a commercial enterprise search engine.

We plan to extend the evaluation and are currently running large-scale experiments with thousands of users to evaluate our system by providing them with a tool that can help find experts on a given topic. We further plan to extend our system to automatically tag documents for all the metadata categories, so that our algorithm will no longer suffer from data sparseness. We expect that this would allow our algorithm to be more accurate in its retrieval and similarity metric.

5. REFERENCES

[1] K. Balog and M. de Rijke. Determining expert profiles (with an application to expert finding). In *IJCAI 2007*, 2007.

[2] N. Craswell, A. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005.