Purdue at TREC 2010 Entity Track: a Probabilistic Framework for Matching Types between Candidate and Target Entities

Yi Fang, Luo Si, Naveen Somasundaram,
Salman Al-Ansari
Department of Computer Sciences, Purdue University
West Lafayette, IN, 47907, USA
fangy@cs.purdue.edu

Zhengtao Yu,Yantuan Xian
School of Information Engineering and Automation
Kunming University of Science and Technology
Kunming, China, 650051

ABSTRACT

This paper gives an overview of our work for the TREC 2010 Entity track. The goal of the TREC Entity track is to study entity-related searches on Web data, which has not been sufficiently addressed in prior research. For both the Related Entity Finding (REF) task and the Entity List Completion (ELC) task in this track, we propose a unified probabilistic framework by incorporating the matching between target entity types and candidate entity types. This framework is motivated by the observation that much more specific type information than the given type can be inferred from the query narratives. These fine-grained types can help narrow down candidate entities. Specific probabilistic models can be derived from this general framework. For the REF task, besides the type matching component, we generally follow our previous work on TREC Entity 2009. For the ELC task, we apply the same framework and the resulting model combines structured document retrieval with type matching.

1. INTRODUCTION

The aim of the TREC Entity track is to evaluate entity-related searches on Web data. This year continues on the last year's main task [1]: related entity finding (REF), but includes a few changes. The data collection is the English subset of ClueWeb category A, which is ten times larger than last year. This enables retrieval systems to find more relevant entities, but potentially mixed with more irrelevant ones. The number of topics is increased to 70 (including last year's 20) and a new entity type, i.e., location, is also added. In addition to the main task, this year introduces a pilot task: entity list completion (ELC), which is to perform entity search on the semantic data. The data collection is the Billion Triple Challenge 2009 dataset and the evaluation topics are from TREC Entity 2009.

For both REF and ELC tasks, we propose a unified probabilistic framework by incorporating the matching between target entity types and candidate entity types. Specifically, we infer the types of target entities from the query topic and infer the types of candidate entities from their profiles, and then match the two types. In addition, in REF, we generally follow our previous work on TREC Entity 2009 to calculate the relevance between topics and entities. The structures of tables and lists are further investigated to extract related target entities from them. In ELC, we leverage the IR techniques to store the semantic data about entities into documents and then to retrieve the entities by structured document retrieval. With the proposed framework, we also perform type matching between target entity types and candidate entity types.

2. PROBABILISTIC FRAMEWORK

Both REF and ELC tasks aim to return a ranked list of relevant entities *e* for a query *Q*. To tackle the REF task, different approaches have been proposed to estimate the probability $p(e|Q)$. In the query *Q*, the target type *T* is given as one of the 4 types: people, products, organizations and locations. However, the provided type information is too coarse. More specific and fine-grained type information is desirable if it can be inferred from the query narratives and be matched with the candidate entities. This observation is also indicated and (heuristically) utilized by other TREC Entity participants such as [8, 9, 10].

In this TREC work, we propose a unified probabilistic framework by introducing a binary matching variable *m* between the target entity type t_q and the candidate entity type t_e. Specifically, we use $p(m = 1|t_q, t_e)$ to denote probability that t_q matches with t_e. t_q and t_e are derived from the query *Q* and the candidate entity *e*, respectively. We formalize the tasks as estimating the probability $p(e, m = 1|Q)$. We derive our ranking formula as follows:

$$\begin{aligned} p(e, m = 1|Q) &= p(e|Q)p(m = 1|e, Q) \\ &= p(e|Q) \sum_{t_q, t_e} p(t_q|Q)p(t_e|e)p(m = 1|t_q, t_e) \end{aligned} \quad (1)$$

The graphical model representation is shown in Figure 1.

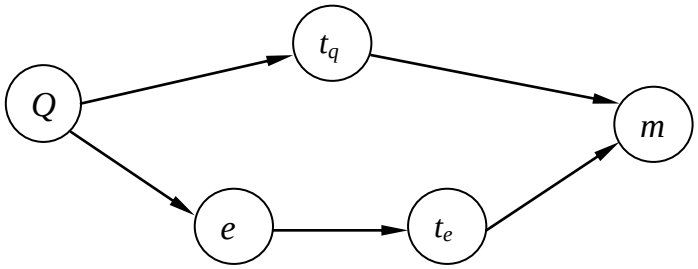

Figure 1: Graphical model representation of the type matching probabilistic framework.

This probabilistic framework is so general that it can be used for both REF and ELC tasks. Furthermore, many models for REF can be seen as its special cases (e.g., if we treat $p(m = 1|t_q, t_e) = 1$). In the next sections, we show how to utilize the framework to address the REF and ELC tasks.

3. RELATED ENTITY FINDING

3.1 PROBABLISTIC MODELS

Similar to many other probabilistic models in the Entity track [5, 6, 7], the probabilistic framework in Eqn. (1) has the component $p(e|Q)$ to measure the query-entity relevance. In our TREC runs, we use the hierarchical relevance model [2] to calculate $p(e|Q)$. In the hierarchical relevance model, the problem of identifying relevant entities is formulated to estimate the probability of a candidate e being the target entity given a topic q and target type T. That is, we determine $p(e|q, T)$, and rank candidate e according to this probability. T belongs to one of the 4

types and is specified in the query topic. The top k candidates are deemed the most probable entities. We can decompose $p(e|q,T)$ into the following form

$$p(e|q,T) \propto \sum_d \sum_s p(q|d)\, p(q|s,d) p(e|q,T,s,d) \quad (2)$$

where s denotes a supporting passage in a supporting document d. The first quantity $p(q|d)$ on the right hand side is the probability that the query is generated by the supporting document, which reflects the association between the query and the document. Similarly, the first quantity $p(q|s,d)$ reflects the association between the query and the supporting passage. The last quantity $p(e|q,T,s,d)$ is the probability that a candidate entity e is the related entity given passage s, and query q. In sum, this probabilistic retrieval model considers the relevance at three different levels: document, passage and entity.

After obtaining $p(e|Q)$, another essential component in Eqn. (1) is to compute the probability of type matching as follows:

$$p(m=1|e,Q) = \sum_{t_q,t_e} p(t_q|Q) p(t_e|e) p\left(m=1\middle|t_q,t_e\right) \quad (3)$$

where $p\left(t_q\middle|Q\right)$ calculates the probability that the query topic is looking for entities with the type t_q. Similarly, $p(t_e|e)$ is the probability that the candidate entity e has the type t_e. $p\left(m=1\middle|t_q,t_e\right)$ calculates the matching between the target entity type t_q and the candidate entity type t_e. As shown in Eqn. (1), the final ranking score is the combination of Eqn. (2) and Eqn. (3).

We generally follow [2] to calculate the quantities $p(q|d)$, $p(q|s,d)$ and $p(e|q,T,s,d)$ in Eqn. (2). In the subsequent sections, we show how to compute the quantities in Eqn. (3).

3.2 COMPUTING $p(t_q|Q)$

$p(t_q|Q)$ reflects the types of the entities that the query looks for. In fact, the query topic provides the type information in the <target_entity> field, which belongs to one of the four types: i.e., people, product, organization and location. We can directly utilize the information to obtain the target entity type. Specifically, $p\left(t_q\middle|Q\right) = 1$ if t_q is the provided type and otherwise $p\left(t_q\middle|Q\right) = 0$. However, this categorization may be too coarse and would potentially return many irrelevant entities. In fact, more specific type information is indicated in the <narrative> field. For example, in Topic 29 "Find companies that are included in the Dow Jones industrial average", we know that the target entity should not only be an organization, but more specifically be a company. In our automatic run KMR1PU, we compute $p(t_q|Q)$ by calculating the similarity score between the type in <target_entity> and the word in <narrative>, and choose the word with the highest similarity as the target entity type. The similarity is computed based on the distance defined by WordNet[1]. This method is simple and has limitations for certain queries. For example, some types are in a phrase instead of a word such as in Topic 62 "What cruise lines have cruises originating in Baltimore?". Moreover, a better type word may go beyond the words in the query and could be inferred from the query. For example, in Topic 61 "Who are institutional members

[1] http://wordnet.princeton.edu/

of the Association for Symbolic Logic (ASL)?", a better type could be "university" instead of "institution", if we consider "institution" in the context of ASL. This needs further investigation in the future work. In our manual run KMR3PU, we manually choose the target entity types.

3.3 COMPUTING $p(t_e|e)$

Similar to $p(t_q|Q)$, $p(t_e|e)$measures the strength that the candidate entity e has the type t_e (The candidate entities are extracted by Named Entity Recognizers from support documents as described in [2]). t_e is the type that can categorize the entity. The step of choosing t_e can be viewed as the task of entity profiling [3]. In other words, t_e should be a good summary of the entity and can potentially categorize the entity based on the entity's profile documents. In our run KMR1PU, we utilize Wikipedia as one source to profile an entity by looking at the "category" section of the entity's Wiki page, since the majority of the target entities have their Wiki pages. The distribution over the categories is assumed uniform, i.e., $p(t_e|e) = 1/n$, where n is the number of related Wiki categories. If the Wiki page does not exist for the entity, we use the original type in <target_entity> as t_e.

3.4 COMPUTING $p(m = 1|t_q, t_e)$

$p(m = 1|t_q, t_e)$ reflects the similarity between the target entity type t_q and the candidate entity type t_e. The type of relevant entities is expected to be consistent with the target entity type. This probability enables us to perform fuzzy match between the two types by considering their semantics. For example, if the target entity type is "institution" and the candidate entity type is "university", they are quite match in terms of semantics. In our run KMR1PU, we compute $p(m = 1|t_q, t_e)$ by normalizing the word similarity obtained from WordNet. Specifically, the similarity score s_{eq} is inversely proportional to the number of nodes along the shortest path between the synsets, and then the normalization is done through $p(m = 1|t_q, t_e) = \frac{s_{eq}}{\sum_e s_{eq}}$.

3.5 OTHERS

We apply the same set of techniques as those in our TREC Entity 2009 work for the other parts of the retrieval system: i.e., query transformation, entity extraction from tables and lists, entity homepage finding and result filtering. Readers can refer to [2] for the details.

4. ENTITY LIST COMPLETION

4.1 BACKGROUND

The motivation of the ELC task is close to that of the main task, but instead of finding entities on the Web, the task is to find these entities on the Semantic Web. The query topics here are the 14 topics from the TREC Entity 2009 REF task. In the ELC task, we intend to address the following research questions: 1) How to leverage the IR techniques for semantic search?; 2) How can structured semantic information be utilized to IR problems? 3) How to use the type matching to further constraint entity search on the semantic data?

4.2 DATA PROCESSING

The dataset for the ELC task is the Billion Triple Challenge dataset[2]. Since the data is from many different semantic data sources, it contains many different ontologies. This poses challenges to the retrieval task. The dataset is in the Resource Description Framework (RDF) format with a series of triples: <Subject> <Predicate> <Object>. Each subject can be treated as an entity, represented by a URI. Objects can either be textual nodes or entities. The subject is related to the object through the predicate. We group the same subject together to form a document and then treat entity search on the semantic data as document search. The RDF data was converted into the TRECTEXT format. The RDF predicates were mapped to the field names and the RDF objects were treated as field values. The resulting TRECTEXT documents were then indexed using the Indri[3] toolbox. Table 1 shows the statistics of the data. Following the work in [4], 4 fields of predicates are also indexed: <name>, <title>, <dbpedia-title>, and <text>. Indexing these fields allows utilizing the rich Indri structural query language such as field weighting and restriction. No stop words were removed and Porter stemming was applied during indexing.

4.3 PROBABLISTIC MODELS

As shown in Eqn. (1), we use the probabilistic framework to combine the evidence from documents and from type matching:

$$p(e, m = 1|Q) = p(e|Q)p(m = 1|e, Q)$$

where $p(e|Q)$ measures the similarity between query e and entity Q, and $p(m = 1|e, Q)$ is the probability calculated based on type matching. Since the entity is represented by a document, any document retrieval model can be used to compute $p(e|Q)$. We use the Indri structured document retrieval model to calculate $p(e|Q)$ as follows:

$$p(e|Q) = \sum_{b \in q} \sum_{i=0}^{4} \omega_i \left(\alpha_T f_{iT}(b) + \alpha_O f_{iO}(b) + \alpha_U f_{iU}(b)\right)$$

where $f_i(b)$ denotes the Jelenik-Mercer smoothed log probabilities for the query term b. ω_i is the weighting parameter for the 4 selected attributes and the whole document, respectively. T is the set of query terms, O is the set of ordered query terms, and U is the set of unordered query terms. α is the corresponding parameters. All the parameter values are set to those suggested in [4].

For computing $p(m = 1|e, Q)$, we use the same decomposition with Eqn. (3). The approach to calculating the quantities $p(t_q|Q)$, $p(t_e|e)$ and $p\left(m = 1\middle|t_q, t_e\right)$ for the ELC task follows the same as those described in Section 3.3, 3.4 and 3.5. More sophisticated methods can be further developed based on the semantic structures of the data. For example, all the entities already have their profile documents. We can assign the types for those entities without Wiki categories by

[2] http://vmlion25.deri.ie/

[3] http://www.lemurproject.org/indri/

training classifiers on the Wiki categories. In addition, when computing $p(m = 1|t_q, t_e)$, we can utilize the ontologies that the entities come from, since they contain more precise type information than WordNet. We will explore the extensions in the future work.

5. EXPERIMENTS

5.1 SUBMITTED RUNS FOR REF

On the REF task, we submit two runs for the TREC official evaluation. KMR1PU is an automatic run and KMR3PU is a manual run in which the types of target entities are manually chosen. Table 1 shows the results for the two runs. Figure 3 demonstrates the nDCG@R scores for each of the 50 test queries. By comparing KMR3PU with KMR1PU, we can see that more accurate identification of target entity types can substantially improve the performance. Out of 47 queries, 9 queries got zero nDCG@R score in KMR1PU, and 6 queries got zero in KMR3PU. The problem could come from named entity recognition, document retrieval or homepage finding. The exact reasons need further investigation.

Table 1. Official results of the two submitted runs for the REF task. "pri_ret" means the number of primary entities retrieved.

	nDCG@R	Rprec	MAP	P@10	pri_ret
KMR1PU	0.2485	0.2099	0.1555	0.2511	246
KMR3PU	0.2917	0.2505	0.1916	0.2894	296

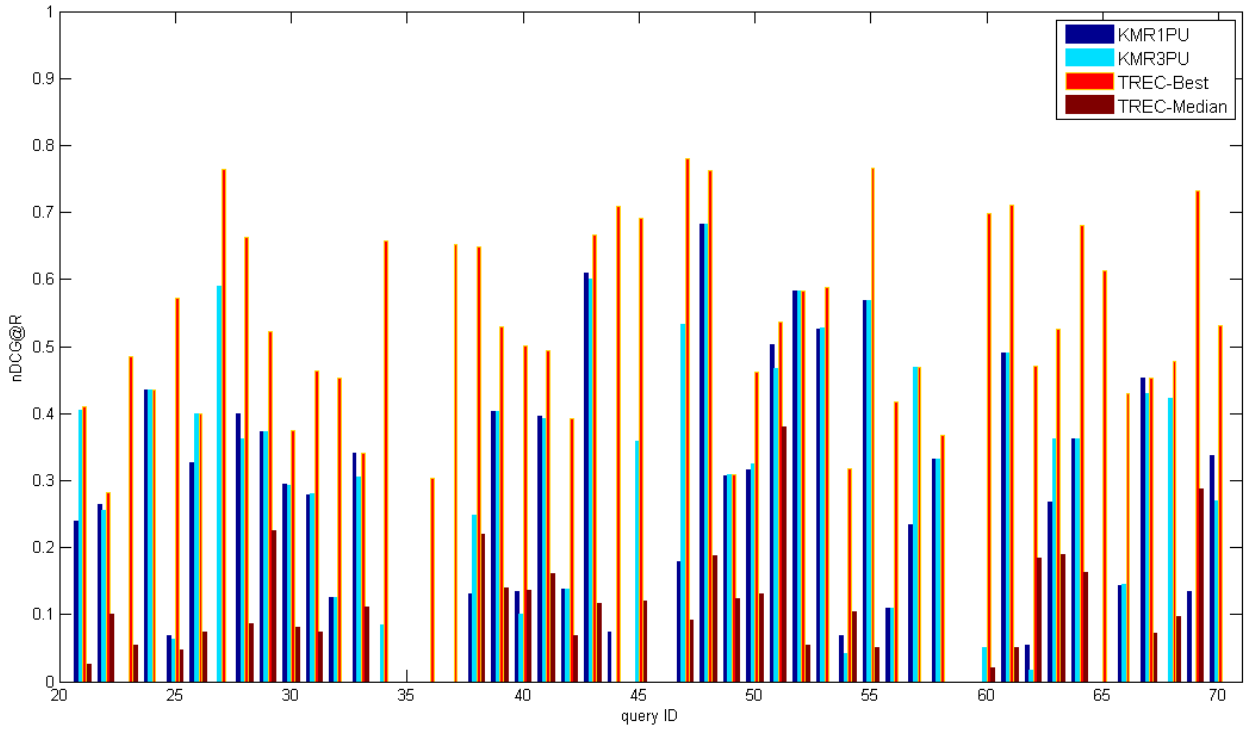

Figure 2. nDCG@R scores for each of the test queries

5.2 SUBMITTED RUNS FOR ELC

Table 2 shows the results for our submitted run KM5PU on the ELC task. Compared with all the competing runs (whose MAP and Rprec are below 0.12), KM5PU achieves a significantly better performance, which indicates the effectiveness of the structured document retrieval with type matching for this task. One limitation of the proposed approach is that the example entities given in the query topics are not exploited. In the future work, we will consider re-ranking the candidate entities by utilizing the relations between the candidate entities and the example entities.

Table 2. Official results of the submitted run KM5PU for the ELC task.

Query ID	num_ret	num_rel	rel_ret	MAP	Rprec
4	31	5	5	0.3579	0.4000
5	15	1	1	1.0000	1.0000
7	48	25	9	0.1646	0.3200
11	9	8	0	0.0000	0.0000
12	25	17	1	0.0294	0.0588
15	25	3	1	0.1667	0.3333
17	44	21	16	0.3719	0.3810
20	60	1	0	0.0000	0.0000
All	257	81	33	0.2613	0.3116

6. CONCLUSIONS AND FUTURE WORK

This paper describes the participation of Purdue University in the TREC 2010 Entity track. We propose a probabilistic framework by combining both query-entity relevance and candidate-target entity type matching. We derive specific methods from the framework to address both REF and ECL tasks in this track. The preliminary results have shown the effectiveness of the proposed approaches. In the future work, we will derive different probabilistic models from this framework by varying the way of computing the individual components/probabilities in the framework. Better calculations of these components can be naturally plugged and combined into the framework.

ACKNOWLEDGEMENT

Yi Fang and Luo Si have been supported by research grants from National Science Foundation (NSF IIS-0746830) and (NSF IIS-#1017837), a research grant from Indiana Economic Development Corporation, and a research grant from Purdue University. Zhengtao Yu and Yantuan Xian are supported by National Nature Science Foundation (No. 60863011) of China and The Key Project of Yunnan Nature Science Foundation (No. 2008CC023) of China. Any

opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsor.

REFERENCES

[1] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 Entity track. In The Eighteenth Text REtrieval Conference (TREC-18), 2009.
[2] Y. Fang, L. Si, Z. Yu, Y. Xian, and Y. Xu. Entity retrieval with hierarchical relevance model. In The Eighteenth Text REtrieval Conference (TREC-18), 2009.
[3] K. Balog and M. de Rijke. Determining expert profiles (with an application to expert finding). In IJCAI, 2007.
[4] J. Dalton and S. Huston. Semantic entity retrieval using web queries over structured RDF data. In SemSearch Workshop, 2010.
[5] Y. Wu and H. Kashioka. NiCT at TREC 2009: Employing three models for Entity ranking track. In The Eighteenth Text REtrieval Conference (TREC-18), 2009.
[6] H. Zhai, X. Cheng, J. Guo, H. Xu, and Y. Liu. A novel framework for related entities finding: ICTNet at TREC 2009. In The Eighteenth Text REtrieval Conference (TREC-18), 2009.
[7] M. Bron and K. Balog and M. de Rijke. Ranking related entities: components and analyses. In CIKM, 2010.
[8] R.L.T. Santos, C. Macdonald, and I. Ounis. Voting for related entities. In: Proceedings of RIAO 2010.
[9] O. Vechtomova. Related entity finding: University of Waterloo at TREC 2010 Entity track. In The Ninteenth Text REtrieval Conference (TREC-19), 2010.
[10] R. Kaptein, P. Serdyukov, A. de Vries, and J. Kamps. Entity ranking using Wikipedia as a pivot. In CIKM, 2010.