World Wide Web (2015) 18:521–543
DOI 10.1007/s11280-013-0267-8

# Related entity finding by unified probabilistic models

**Yi Fang · Luo Si**

Received: 23 March 2013 / Revised: 11 August 2013 /
Accepted: 22 October 2013 / Published online: 21 November 2013


**Abstract** Both the WWW research community and industry have shown increasing interests in not just finding relevant documents, but specific objects or entities to satisfy more sophisticated user information needs. TREC launched an Entity Track in 2009 to investigate the task of related entity finding. This paper proposes two novel probabilistic models to integrate several components into a unified modeling process. In particular, the type matching component can characterize the degree of matching between the expected entity type that is inferred from query and the candidate entity type that is inferred from entity profile. Another important component can incorporate prior knowledge about entities into the retrieval process. The main difference of the two models is that the second model explicitly considers the effect of source entity while the first one does not. A comprehensive set of experiments were conducted on the TREC Entity Track testbeds from 2009 to 2011 with careful design to show the contributions of individual components. The results demonstrate that both the type matching component and the entity prior modeling component can effectively boost the entity retrieval performance. Furthermore, the second model performs better than the first one in all the settings, indicating the benefits of explicitly modeling source entity in related entity finding. Both models generate better or competitive results than the state-of-the-art results in the TREC REF tasks. In addition, the proposed unified probabilistic approach is applied to the TREC Entity List Completion task and also demonstrates good performance.

**Keywords** Related entity finding · Probabilistic models · Entity search

Y. Fang (✉)
Department of Computer Engineering, School of Engineering, Santa Clara University,
500 El Camino Real, Santa Clara, CA 95053, USA
e-mail: yfang@scu.edu

L. Si
Department of Computer Science, Purdue University, 300 N. University Street,
West Lafayette, IN 47907, USA
e-mail: lsi@purdue.edu

## 1 Introduction

Entity oriented search deals with the retrieval problems about entities, in order to satisfy some sophisticated information needs that go beyond document search. As the Web is evolving into a data-rich repository, both the commercial market and the information retrieval research community have shown increasing interest in not just finding documents, but specific objects or entities in response to a user's query. Many entity search engines have recently emerged to identify specific types of entities of interest such as people, locations and products. For example, by typing the query "Italian restaurant" into Google, the first hit on the result is a list of Italian restaurants, along with their homepages, telephone numbers, and their locations on the map. While current web search engines are capable of handling certain simple entity related queries, there is still a long way to go towards general entity search that can address a wide range of queries. Text REtrieval Conference (TREC) launched a new Entity track in 2009 to investigate the problem of related entity finding (REF) [5]. In its pilot task, given the name and homepage of a source entity, as well as a context/relation described in natural language text, the retrieval system needs to find target entities with homepages that are of the specified type. In 2010 and 2011, TREC continues the Entity track to investigate the REF task and also proposes new tasks based on semi-structured data with the same set of query topics [7, 9]. Below shows one REF query of TREC Entity 2009 and one query of TREC Entity 2010.

This emerging area of entity search differs from traditional document retrieval. Unlike documents, entities are not directly represented and need to be identified and recognized in the mixed space of structured and unstructured Web data. Much prior work in entity search applied standard document retrieval methods to textual representations of entities. For example, one simple approach is to rank candidate entities by considering the retrieval scores of their associated documents with respect to the query. More refined methods exploit the types of entities. The TREC Entity track in 2010 organizes target entities into four types such as people, product, organization and location. However, this categorization is often too coarse to indicate the desired type of target entity (e.g., a more specific type for Topic 29 could be company rather than organization). Therefore, it is desirable to estimate more appropriate type information of target entity with finer granularity. One the other hand, the type of a candidate entity and also its relationship with the target entity often contain uncertainty. Furthermore, some prior knowledge about candidate target entities would be valuable. For example, some more common candidate entities or candidate entities more related with source entity may tend to be better choices in entity search.

Based on the above observations, this paper proposes two novel probabilistic models to integrate several individual components in a unified modeling framework for entity search. The new formal methods estimate the type information of target entity (i.e., expected entity type) and of candidate entity with finer granularity by utilizing knowledge from WordNet[1] and Wikipedia. Based on the type information, these methods improve entity search results with a matching component that investigates the consistency between candidate entity type and expected entity type, which is valuable for distinguishing more relevant candidate entities from less relevant ones. Furthermore, the two probabilistic methods model

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

prior knowledge of candidate entities with either occurrence information or relationship with source entity. Our contributions can be summarized as follows:

1. To the best of our knowledge, this is the first research work that proposes formal unified probabilistic models for entity search by integrating entity relevance, entity type estimation, type matching, entity prior and co-occurrence information with source entity together. While the existing work considers some of the components such as entity relevance, co-occurrence model, and entity prior in [12, 53], and entity type estimation in [27], none of them has taken all the components into account in a unified model. Section 2 has more discussions on the related work.
2. The paper shows that both the type matching component and the entity prior modeling component can effectively boost the entity retrieval performance.
3. The results suggest that it is beneficial to explicitly consider relationship of source entity and candidate entities for improving entity search results.
4. Experiments are conducted based on the standard experimentation paradigm of TREC Entity 2009–2011. The proposed methods are applied to both the TREC tasks of related entity finding and entity list completion. The proposed methods generate better or competitive results than the state-of-the-art results in the TREC Entity tasks.

The next section discusses related work. Section 3 introduces our proposed models for entity search. Section 4 presents some other components used in our entity search systems. Section 5 introduces more advanced components and heuristics that are commonly used in TREC Entity to further improve the entity retrieval performance. Section 6 explains our experimental methodology and Section 7 presents the experimental results. Section 8 concludes and points out some future work.

## 2 Related work

Entity oriented search started out with ranking entities of a specific type, i.e., expert search [2]. The expert finding task, which was run at the TREC Enterprise track [15], focuses on a single type ("person") and relation ("expert in"). Various probabilistic models have been proposed including generative language models [4, 20] and discriminative models [22]. Other popular methods include voting models [32] and graph based models [42]. These methods aim at modeling the relevance of experts through the bridge of documents. A comprehensive survey on expert search is presented by [10].

The more general entity search problem was introduced by TREC Entity track launched in 2009 [5], which targeted on three types of entities, i.e. persons, locations, and organizations. The first edition featured the related entity finding task. In 2010 and 2011, TREC continues the Entity track to investigate the REF task and also proposes new tasks based on (semi)structured data with the same set of query topics [7, 9]. Motivated by the approaches in expert search, similar methods were proposed for entity search. For example, several generative language modeling approaches were employed to rank entities, where the entity model was constructed from snippets containing the entity and the relation is used as a query [49, 51, 52, 54]. Discriminative learning approach was adopted in [31] to rank candidate entities. Voting models were also applied in [39] by considering the occurrence of an entity among the top ranked documents for a given query as a vote for the existence of a relationship between this and the entity in the query. Beyond entity ranking, entity search is a complex problem with several subtasks in the retrieval process. Some approaches first

collect text snippets from documents relevant to the REF query, next obtain entities by performing named entity recognition on the snippets, implement some sort of ranking step and finally find homepages [21, 35, 46, 48]. A number of approaches rely heavily on Wikipedia; as a repository of entity names, to perform entity type filtering based on categories and to find homepages through external links [26, 34, 40].

Zhai [53] proposes a probabilistic framework to estimate the probability of an entity given a REF query, with two components: the probability of the relation given an entity and source entity, and the probability of an entity given the source entity and target type. [12] propose a probabilistic framework including co-occurrence models, type filtering, and context modeling. While this approach is close to ours, it only focuses on the target types specified in the query topics such as people, product, and organization, which are often too coarse to indicate the desired type of target entity. The work in [27] automatically identifies refined target entity types from natural language queries. It uses the KL divergence for calculating the similarity between categories, but this is not desired to be combined with generative probability scores as the KL divergence score is not between 0 and 1. This may be one reason of its relatively low performance on TREC Entity Track 2009 dataset in [27] compared to the best TREC runs. To the best of our knowledge, there is no prior research work that proposes formal probabilistic models for integrating entity relevance, entity type estimation (with finer granularity), type matching, entity prior and co-occurrence information with source entity together in a single probabilistic framework. Moreover, we go beyond the Wikipedia subcollection and conduct a comprehensive evaluation on the TREC Entity testbeds of three years.

INitiative for the Evaluation of XML Retrieval (INEX) launch an entity ranking track in 2007, using Wikipedia as the test collection [18] so that each entity corresponds to a Wikipedia page. Two tasks were introduced: task 1 (entity ranking), with the aim of retrieving entities of a given category that satisfy a topic described in natural language text; and task 2 (list completion), where given a topic text and a small number of entity examples, the aim was to complete this partial list of answers. Since each entity is represented by a Wikipedia page, standard document retrieval can be readily applied to obtain a list of relevant entities for a query. Many approaches further exploit the known categories [25, 50], the link structure of Wikipedia [16, 45, 55], as well as the link co-occurrences [43] with the examples (when provided) to improve the effectiveness of entity ranking. The majority of the approaches use set overlap between expected categories and candidate entity categories to derive a score for the category component. Category expansion was also applied based either on the Wikipedia category structure [43, 50] or on lexical similarity between category labels and the query topic [45]. Balog et al. [8] propose a probabilistic model to further exploit the category information and show the advantage of a category-based representation over a term-based representation. While much work has been done on investigating the types/categories of queries and entities in the INEX evaluations, relatively little related work exists based on the TREC testbeds which go beyond the Wikipedia documents.

Another closely related area is Question answering (QA) which was investigated at the TREC QA track [47]. A comprehensive survey on QA can be found in [37]. The TREC QA track recognized the importance of search focused on entities with factoid questions and list questions. In order to answer list questions, participating systems have to return instances of the class of entities that match the description in the question. Recently, Vechtomova [44] proposes an domain-independent entity retrieval approach which is evaluated on both the testbeds of TREC Entity 2010 REF and QA track list questions from TREC 2005 and 2006. The "list" subtask indeed resembles the TREC Entity tasks, but they differ in important ways as pointed out in [12]: 1) QA list queries do not always contain an entity; and 2) TREC

Entity queries impose a more specific relation between the source entity and the target entities.

The semantic Web (SW) community has also studied a similar problem, which is more often called semantic search. For example, the semantic search engine NAGA [28] builds on a knowledge base that consists of millions of entities and relationships extracted from Web-based corpora. A graph based query language enables the formulation of queries with additional semantic information such as entity types. The search engine ESTER combines full-text on Wikipedia with ontology search in YAGO [11]. A major challenge with current IR approaches to entity retrieval is that they often cannot produce interpretable descriptions of the found entities or of the relationships between them. The SW approach mainly targets on Linked Open Data (LOD)[2] which may have the potential of providing the required semantic information. TREC Entity has started to investigate the problem of entity search over semantic data since 2010 with the tasks of ELC and REF LOD variant [7, 9]. Balog et al. [6] explore the potential of combining IR with SW technologies to improve the end-to-end performance on a specific entity search task. Ad-hoc entity search has also been investigated in the database community and some work exploits the type information of queries and entities [13, 14]. This paper focuses on the EOS tasks defined by the TREC conferences in the IR community.

## 3 Probabilistic models for related entity finding

The goal of the REF task is to return a ranked list of relevant entities $e$ for a query, where a query consists of a source entity $S$, target type $T$ and a relation $R$ [5]. In TREC Entity 2009, the type information $T$ is provided in the <target_entity> field, which belongs to one of the three types: i.e., *people*, *product*, *organization*. In TREC 2010, an additional type, *location*, was added. However, this categorization may be too coarse and would potentially return many irrelevant entities. In fact, more fine-grained target type information is indicated in the <narrative> field. For example, in TREC 2010 Topic 29 (Figure 1), we know that the target entity should not only be an organization, but more specifically be a company.

In this section, we propose two probabilistic models by introducing a binary matching variable $m$ indicating the degree of matching between expected target entity type and candidate entity type ($m = 1$ represents relevance and $m = 0$ irrelevance). The prior knowledge about entities is also incorporated in the models. The two models differ in whether the information of source entity $S$ is taken into account.

### 3.1 Model A

In the first model, which is called Model A, we formalize REF as the task of estimating the probability $P(e, m = 1|R, S)$. This setup is similar to document search where relation $R$ is treated as a query and entities are ranked according to the relevance to the query. This probability is difficult to estimate, due to the lack of training data. Thus, we turn to a generative model, by applying Bayes' Theorem and rewrite

$$p(e, m = 1|R, S) = \frac{p(R, S, m = 1|e)p(e)}{p(R, S)}$$

[2] http://linkeddata.org/

```
<query>
<num>9</num>
<entity_name>The Beaux Arts Trio</entity_name>
<entity_URL>clueweb09-en0005-08-02741</entity_URL>
<target_entity>person</target_entity>
<narrative>Members of The Beaux Arts Trio.</narrative>

<num>29</num>
<entity_name>Dow Jones</entity_name>
<entity_URL>clueweb09-en0006-73-08332</entity_URL>
<target_entity>organization</target_entity>
<narrative>Find companies that are included in the Dow Jones
          industrial average.</narrative>
</query>
```

**Figure 1** Query Topic No. 9 of TREC entity 2009 (*above*) and Query Topic No. 29 of TREC entity track 2010 (*bottom*)

We then drop the denominator as it does not influence the ranking of entities, and derive our ranking formula as follows:

$$\begin{aligned} p(e, m = 1|R, S) &\propto p(R, S, m = 1|e)p(e) \\ &= p(m = 1|e, R, S)p(R, S|e)p(e) \\ &= p(e, R, S)p(m = 1|e, R, S) \\ &= p(R|e, S)p(e|S)p(S)p(m = 1|e, R, S) \end{aligned} \tag{1}$$

Next, we introduce a latent variable $t_R$ indicating the expected entity type which is inferred from relation $R$. Similarly, we use another latent variable $t_e$ to denote the type of the candidate entity $e$. Assuming $S$ is conditionally independent of $m$ given $e$ and $R$, $p(m = 1|e, R, S)$ can be then decomposed as follows

$$p(m = 1|e, R, S) = \sum_{t_R} \sum_{t_e} p(m = 1|t_e, t_R)p(t_e|e)p(t_R|R) \tag{2}$$

where $p(m = 1|t_R, t_e)$ denotes the probability that $t_R$ matches with $t_e$. $t_R$ and $t_e$ are derived from the relation $R$ and the candidate entity $e$, respectively. We drop $p(S)$ in (1) because it is a constant for candidate entities. Then by plugging (2) into (1), we can obtain

$$p(e, m = 1|R, S) \propto p(R|e, S)p(e|S) \sum_{t_R} \sum_{t_e} p(m = 1|t_e, t_R)p(t_e|e)p(t_R|R)$$

The graphical model representation is shown in Figure 2a. Model A includes the following components: 1) entity relevance $p(R|e, S)$ where $S$ plays a role, 2) co-occurrence between source and target entities $p(e|S)$, 3) candidate entity type $p(t_e|e)$, 4) expected entity type $p(t_R|R)$, and 5) type matching $p(m = 1|t_e, t_R)$. In Sections 3.3–3.8, we show how to measure these individual components.

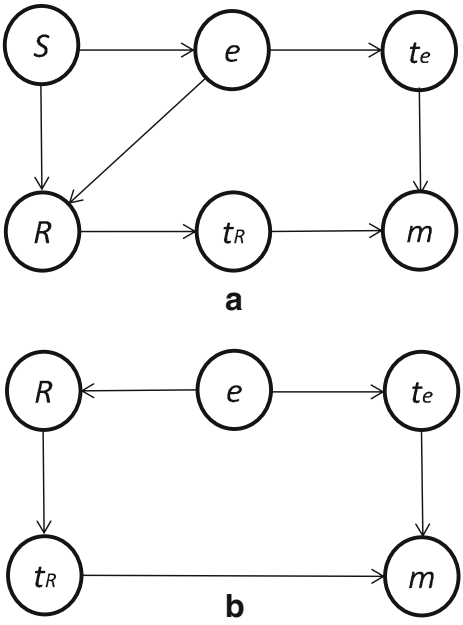


**Figure 2** Graphical model representation of (**a**) Model A and (**b**) Model B

### 3.2 Model B

In TREC Entity tracks, source entity $S$ is provided in the query. However, in real applications, this information is usually not available. For example, web search users tend to express their information needs by natural language text without specifying source entity. In this section, we introduce Model B without taking $S$ into consideration. Consequently, the target probability becomes $p(e, m = 1|R)$, which can be decomposed in a similar way as Model A, as follows:

$$\begin{aligned} p(e, m = 1|R) &= \frac{p(R, m = 1|e)p(e)}{p(R)} \\ &\propto p(R, m = 1|e)p(e) \\ &= p(m = 1|e, R)p(R|e)p(e) \\ &= p(e, R)p(m = 1|e, R) \\ &= p(R|e)p(e)p(m = 1|e, R) \\ &\propto p(R|e)p(e) \sum_{t_R} \sum_{t_e} p(m = 1|t_e, t_R)p(t_e|e)p(t_R|R) \end{aligned}$$

Model B shares three identical components with Model A, $p(t_e|e)$, $p(t_R|R)$, and $p(m = 1|t_e, t_R)$. On the other hand, Model B has two different components: 1) entity relevance $p(R|e)$ without $S$ playing any role, and 2) entity prior $p(e)$. The graphical model representation is shown in Figure 2b.

### 3.3 Expected entity type

$p(t_R|R)$ reflects the types of the entities that the query looks for. We can directly utilize the information to obtain the expected target entity type. Specifically, $p(t_R|R) = 1$ if $t_R$

is the provided type and otherwise $p(t_R|R) = 0$. However, the specified type in TREC 2009 and 2010 only contains three or four categories, which may be too coarse and would potentially return many irrelevant entities. Instead we obtain $t_R$ 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 one of the target entity types. The similarity is computed based on the distance defined by WordNet. This word is also labeled as the keyword which will be assigned a different weight to form the query for document retrieval (see Section 7.3). TREC 2011 provides a more fine-grained target type which is directly used as one of the candidate expected types. Moreover, a better type may go beyond the words appearing in the query and could be inferred from the query. Thus, we add other types into $t_R$ by classifying the query into categories. Specifically, we retrieve the top 5 Wikipedia documents for a given query and choose two most frequent Wikipedia categories of the documents as the added types for the query. This process is similar to pseudo-relevance feedback and also similar to the method in [27]. The distribution over the expected types is assumed uniform, i.e., $p(t_R|R) = 1/n_R$, where $n_R$ is the number of the candidate expected types for $R$.

### 3.4 Candidate entity type

$p(t_e|e)$ measures the probability that candidate entity $e$ is of type $t_e$. $t_e$ 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. Specifically, we utilize Wikipedia as one source to profile an entity by looking at the "category" section of the entity's Wiki page. While the vast majority of the target entities have their Wiki pages (we found over 92 % of the entities in the training data have wiki pages), some candidate entities do not. Similar to the pseudo-relevance feedback method in Section 3.3, we also retrieve the top 5 Wikipedia documents by using the given candidate entity's name as the query. We add two most frequent Wiki categories of the documents into the set of candidate types. The distribution over the types is also assumed uniform.

### 3.5 Type matching

$p(m = 1|t_R, t_e)$ reflects the similarity between the expected entity type $t_R$ and the candidate entity type $t_e$. The type of relevant entities is expected to be consistent with the expected 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 form a good match in terms of semantics. We compute $p(m = 1|t_R, t_e)$ by utilizing the word similarity obtained from WordNet. Specifically, $p(m = 1|t_R, t_e)$ is inversely proportional to the number of nodes $d_{eR}$ along the shortest path between the synsets of $t_R$ and $t_e$, i.e., $p(m = 1|t_R, t_e) \propto \frac{1}{d_{eR}}$.

### 3.6 Entity relevance

$p(R|e, S)$ in Model A or $p(R|e)$ in Model B measures the relevance of entity with respect to query. The estimation of the quantity is the focus of most of the prior work in the literature. This section presents two methods, based on language modeling and hierarchical relevance model, respectively.

### 3.6.1 Language modeling

Similar to the candidate models [4] in expert finding, we can build entity-centric language models to estimate $p(R|e)$. An entity is represented by snippets extracted from relevant documents. We represent the relation by an entity language model ($\theta_e$), a distribution over terms taken from the snippets. By assuming independence between the terms in the relation $R$ we arrive at the following estimation:

$$p(R|e) = p(R|\theta_e) = \prod_{t \in R} p(t|\theta_e)^{n(t,R)}$$

where $n(t, R)$ is the number of times $T$ occurs in $R$. To estimate the entity language model $\theta_e$, we aggregate term probabilities from the entity profile which is the new document composed of snippets of an entity $e$.

$$p(t|\theta_e) = \frac{n(t, e) + \alpha p(t)}{\sum_{t'} n(t', e) + \alpha}$$

where $n(t, e)$ is the number of times $T$ appears in entity profile $e$, $p(t)$ is the collection language model, and $\alpha$ is the Dirichlet smoothing parameter, set to the average document length in the collection [29].

$p(R|e, S)$ can be estimated in a similar way, by replacing the entity language model by the co-occurrence language model in the above estimation [12].

### 3.6.2 Hierarchical relevance model

An alternative method to estimate $p(R|e)$ is proposed in [21], which is called hierarchical relevance model. In this model, three levels of relevance are examined which are document, passage and entity, respectively. The final ranking score is a linear combination of the relevance scores from the three levels. Specifically, $p(R|e)$ can be decomposed into the following form

$$p(R|e) \propto \sum_{d} \sum_{u} p(R|d) p(R|u, d) p(e|R, u, d)$$

where $u$ denotes a supporting passage in a supporting document $d$. The first quantity $p(R|d)$ is the probability that the query is generated by the supporting document, which reflects the association between the query and the document. Similarly, the second quantity $p(R|u, d)$ reflects the association between the query and the supporting passage. The last quantity $p(e|R, u, d)$ is the probability that a candidate entity $e$ is the related entity given passage $u$, and relation $R$. The advantage of adding the passage based relevance is that it allows us to detect highly relevant information embedded in a possibly lengthy document. Passage retrieval has been widely used in Question Answering [37]. In sum, this probabilistic retrieval model considers the relevance at three different levels: document, passage and entity. The entity relevance $p(R|e, S)$ in Model A can also be calculated in a similar way by only using the supporting documents $d$ in which source entity $S$ and candidate entity $e$ co-occur.

## 3.7 Entity prior

$p(e)$ is the a prior probability of an entity being relevant (independent of the query). If a query is difficult and the retrieval system cannot find enough evidence to decide which entity is relevant, entity prior may play an important role in ranking entities. It has been

shown that utilizing candidate expert importance is effective in expert finding [2, 33, 41]. Thus, entity prior is expected to be an informative component in entity search. To estimate it, we may reasonably assume that a candidate that has been mentioned many times has a high prior probability of being an answer entity. Therefore, we choose $p(e) = \frac{c(e)}{\sum_{e'} c(e')}$ where $c(e)$ is the count of mentions of the candidate $e$ in the collection of the top documents retrieved. Although this estimation seems simple and straightforward, the experimental results in Section 7.1 show that the entity prior component can help boost the entity retrieval performance.

### 3.8 Source and candidate entity co-occurrence

$p(e|S)$ is the co-occurrence component that indicates the association between source and candidate entities. It can be computed based on the co-occurrences between $e$ and $S$ in documents, independent of the actual content of the documents [12]. Specifically, $p(e|S)$ can be estimated as follows:

$$p(e|S) = \frac{f(e, S)}{\sum_{e'} f(e', S)}$$

where $f(e, S)$ is a function to calculate the strength of the co-occurrences. There exists various forms for the function $f$. A simple one could be the Maximum likelihood estimate (MLE) by computing the relative frequency of co-occurrences between $e$ and $S$ (regardless of their distance in the document). Bron et al. [12] compared four methods: MLE, $\chi^2$ hypothesis test, Pointwise mutual information, and Log likelihood ratio. They found that $\chi^2$ hypothesis test generally showed better performance than the other methods. Thus, in the experiments we adopt the $\chi^2$ hypothesis test in the same way as in [12]. The $\chi^2$ hypothesis test determines if the co-occurrence of two entities is more probably than just by chance. It is worth noting that $p(e|S)$ can be viewed as a refined estimation of entity prior by considering the association between candidate entity and source entity.

## 4 Other components in the pipeline

The pipeline of our entity search is similar to many TREC participants. We first collect text passages from documents relevant to the query, and then obtain entities by performing named entity recognition on the passages. The probabilistic models in Section 3 (possibly combined with engineering heuristics in Section 5) are then applied to rank the entities. Finally, the homepages of the entities are identified, since the evaluations in TREC Entity are based on the homepages. While the probabilistic models in Section 3 form the core part for REF, document retrieval, named entity recognition, and homepage finding are also indispensable components in the pipeline. This section introduces these components.

### 4.1 Document retrieval

The initial step of the whole process is document retrieval with respect to the given query topic. In particular, the query is formed as a set of keywords extracted from the source entity $S$ and relation $R$. The formulation of query from a natural language narrative should maximize the performance of document retrieval. With a part-of-speech tagger, we parse the relation to obtain the verb or noun to form the set of keywords. Many entities exist in the documents or queries in the form of acronym such as IU for "Indiana University". We augment the query by including the synonyms of the keywords from WordNet.

In addition, we also add acronyms or full name of the source entity, which is likely to result in more documents containing related entities. Without external resources, to find the acronym of an entity can sometimes be difficult such as LVMH for "Moet Hennessy-Louis Vuitton". In our experiments, we resort to Farlex Free Dictionary [3] to find acronyms. With the formed query, we retrieve the top pages from Google, and then remove those pages that are not in the TREC test collection. The remaining pages are used as the candidate documents to extract related entities. The default number of documents used in the experiments is 10 which we found is an empirically good choice based on the training data.

### 4.2 Named entity recognition

After the relevant documents are retrieved, text passages, which are individual sentences, are extracted from the documents. We use Stanford Named Entity Recognizer (NER)[4] and LBJ Named Entity Tagger[5] to extract entities from the passages. The purpose of using two NERs is to increase the recall of extracting relevant entities. These NERs can directly recognize persons, organizations, and locations. To recognize products, we train a Conditional Random Field (CRF) model [30] for named product recognition. In the experiments, a total of 4000 documents were randomly selected from Open Directory Project[6] under the category of product. Two graduate students then annotated each word of the documents, and Stanford Named Entity Recognizer was used to train the new NER model. The inter-annotator agreement percentage at word level was 85.6 %.

### 4.3 Homepage finding

According to the TREC Entity track, an entity is uniquely identifiable by one of its primary homepages. After extracting the names of related entities, we need to find their homepages. We treat homepage finding as a classification problem. The positive training examples can be collected from the homepages provided by website directory services such as Open Directory Project. The negative training examples are obtained by removing the positive examples from the top Google pages returned with respect to the candidate entity's name. We then train logistic regression models with a number of features specified in [21] which has proved effective in homepage detection. The trained models are then applied to the candidate homepages which come from Google's top returned non-Wikipedia pages (that are also in the TREC test collection). In the experiments, we selected 421 persons, 568 organizations, 216 products, and 321 locations from Google Directory[7](now merged to Open Directory Project) as the positive examples for training. The number of negative examples roughly keeps the same. Table 1 contains the features used in the logistic regression model.

[3] http://acronyms.thefreedictionary.com

[4] http://nlp.stanford.edu/software/CRF-NER.shtml

[5] http://cogcomp.cs.illinois.edu/page/softwareD_view/4

[6] http://www.dmoz.org/

[7] http://directory.google.com/

**Table 1** Features used in logistic regression for homepage finding

| | |
|---|---|
| URL features | Whether includes the full entity name |
| | Whether includes partial of the entity name |
| | Whether includes the entity name behind the last slash |
| | Whether contains keywords such as "wiki", "index", "default", etc. |
| | Whether contains acronyms of the entity name |
| | Length in characters |
| | Numbers of slashes, question marks, underscores and digits |
| Document features | Frequency of the entity name in the document |
| | Whether contains keywords such as "official", "main", "home", etc |
| | Whether the TITLE element contains the entity name |
| | Length in words |
| | PageRank |

## 5 Advanced components

Sections 3 and 4 present the basic components that are necessary to complete the REF task. In this section, we introduce additional advanced components that may be able to further improve the entity retrieval performance. These advanced components have been heavily used in the top runs in TREC Entity (e.g., [7, 9, 21, 48]) to achieve the state-of-the-art performance.

### 5.1 Tables and lists

For a query topic, many related entities exist in structural forms such as in tables or lists. On the one hand, this poses challenges to entity extraction because most NERs utilize the context information to recognize the named entities (e.g., CRF) while there is few context for the elements in tables. To acquire the context of the entities, we use the element names as queries to retrieve relevant documents, and then use NERs to recognize the types of the elements in these documents by majority voting. On the other hand, the structure of tables of lists can facilitate the entity extraction. For example, in a table, all the elements with the same attribute have similar properties and are likely to share the same entity types. Moreover, they are likely altogether to be the target entities. In our experiments, we utilize this fact by assuming that if the majority of the elements with the same attribute are of the same type or identified as answer entities, all these elements have the same type or are the answer entities. Based on the training data, we observed that while some relevant entities could be extracted by utilizing the structure of tables and lists, this procedure may generate many false positives. Placing these entities at the top of the ranked list may significantly decrease the precision. Therefore, the entities detected by this procedure are appended into the bottom of the ranked list.

### 5.2 Surface text patterns

It has been noted in the Question Answering community that certain types of answers can be directly extracted by matching surface text patterns. For example, for questions like "When was Einstein born?", a typical answer is "Einstein was born in 1756." This example suggests the text pattern "<NAME> was born in <BIRTHDATE>" can be used to locate

**Table 2** The eight text surface patterns used in REF

| | |
|---|---|
| 1 | <ANSWER> is <narrative> |
| 2 | <ANSWER> ( <narrative> , |
| 3 | <ANSWER> , <narrative> . |
| 4 | , a <narrative> <ANSWER> , |
| 5 | ( <narrative> <ANSWER> ) , |
| 6 | form of <ANSWER> , <narrative> |
| 7 | for <narrative> , <ANSWER> and |
| 8 | as <narrative> , <ANSWER> and |

the correct answer. Depending on the types of the questions (e.g., BIRTHDATE, DEFINITION, INVENTOR, etc.), different patterns can be used to identify the answers. The power of surface text patterns has been demonstrated in previous TREC QA tracks [38]. Given the similarity of the REF tasks with Question Answering, we can also use this technique in REF. For example, for Topic 9, a reasonable template could be "<PERSON> is a member of the Beaux Arts Trio". Since the REF queries are similar to the DEFINITION type of questions in TREC QA, we use eight templates that were used in [38] for the DEFINITION question type in QA. These text patterns are shown in Table 2. The training queries indicate that the first word(s) of the original <narrative> is either plural or not meaningful, while the target entity should be in the singular form. Thus, we trim it in order to form appropriate patterns. The templates are applied to all the query topics to directly extract answers from the documents. Since the answers identified by surface text pattern matching are usually of high accuracy, they will be placed at the top of the ranked list of candidate entities. After completing the ranked list, filtering rules are applied to further refine the final results.

### 5.3 Entity filtering

After obtaining a list of ranked entities, we conduct a set of operations below to further refine the results.

- There may exist a variety of ways of referring to the same entity. For example, "Deborah Estrin" and "D. Estrin" may both refer to the same person. This problem can be alleviated through the use of entity resolution by looking at textual similarity in the names of the entities. We use the nearest-neighbor clustering approach, also known as agglomerative hierarchical clustering [24], to merge the same entities with different names. The algorithm begins with all the candidate entities as singleton clusters, and successively merge clusters to produce the other ones if the similarity between two clusters is above a threshold. To measure the similarity between two entity names, we calculate the percentage of overlapping words in the names. If the percentage is greater than 0.5, the two clusters/entities will be merged. The empirical choice of the threshold worked well on the training data.
- Limit the length of entity names for different target types, i.e., 3 words for person, 5 words for organization, and 8 words for product. The intuition is that if a candidate name for person is too long, it is unlikely to be the correct answer. The length limits are chosen based on 95 % of the entities in the training data (e.g., 95 % of the training entities are no longer than 3 words).

– Remove the candidate entities whose names largely overlap with the source entity. Based on the <narrative>, the source entity may be retrieved for the answer. For example, "Dow Jones Industrial Average" could be identified as a candidate for Topic 29 with the source entity "Dow Jones", which should be removed. The rationale is that the target entity should be different from the source entity. We use 50 % overlap as the threshold (i.e., if 50 % of the words in the candidate entity is the same with the source entity, the candidate entity will be removed from the linked list). This threshold is chosen based on the empirical observation on the training data.

For some query topics, only very few entities may remain after filtering. In this case, we need to retrieve more documents, repeat the whole process, and hopefully find more related entities. This iterative process is similar to that for the high-performance QA systems [36], which was proven to be effective. In the experiments, if there are only 5 or less entities returned, we will apply this iterative process, because most of the training queries have 5 or more target entities. 50 more documents are then retrieved to identify more candidate entities, which yielded good empirical results on the training data.

## 6 Experimental setup

### 6.1 Testbeds

We use the data and topic sets from TREC Entity 2009, 2010 and 2011 to evaluate the proposed models. To make a fair comparison with TREC participating runs, our experiments adhere to the rules of the respective TREC Entity tracks. The document collection is the ClueWeb09 dataset.[8] TREC Entity 2009 used the Category B subset which includes about 50 million documents, and the other two years used the English portion of ClueWeb, comprising of approximately 500 million pages. The Entity tracks created 20, 50 and 50 topics for the year of 2009, 2010 and 2011, respectively. There exist some differences in the rules of the different years' Entity tracks, but they are mostly minor. We only make minimal effort to adapt the proposed methods to each year and keep them as general as possible.

We also comply with the official TREC evaluation metrics that are used for the respective years. For 2009, we report on NDCG@R, P@10, and the number of primary (#pri) and relevant (#rel) entity homepages retrieved. For 2010, We report on NDCG@R, MAP and R-Precision (R-Prec). For 2011, the official evaluation measures are MAP and R-Prec. All the runs are automatic including those from the top TREC Entity participants for comparison. We also list whether the top TREC runs utilized web search (Y) or not (N) by having a column in the result tables.

The proposed methods involve a set of parameters. In the experiments, all these parameters were tuned and determined on independent training data which are different from the test data. The training data was developed based on the list of training queries provided by TREC 2009. For specific tasks such as homepage classification and named product recognition, we also developed corresponding training datasets. Sections 4.2 and 4.3 provide the details. The heuristic rules used in this paper were also based on the empirical observation on the training data.

[8] http://lemurproject.org/clueweb09/

**Table 3** Experimental results with Model A. The †symbol indicates statistical significance at 0.95 confidence interval against the baseline

| | 2009 NDCG@R | 2010 NDCG@R | 2011 R-Prec |
|---|---|---|---|
| $p(m = 1\|e, R)$ | 0.0580 | 0.0645 | 0.0376 |
| $p(R\|e, S)$ | | | |
| HRM | 0.2001 | 0.2145 | 0.1937 |
| LM | 0.1907 | 0.2252 | 0.2000 |
| $p(R\|e, S)p(e\|S)$ | | | |
| HRM | 0.2228† | 0.2214 | 0.2019 |
| LM | 0.2102† | 0.2369† | 0.2133 |
| *Integrated*-HRM | 0.2544† | 0.2618† | 0.2237† |
| *Integrated*-LM | 0.2462† | 0.2725† | 0.2342† |

### 6.2 Research questions

An extensive set of experiments were designed to address the following questions of the proposed research:

- Is the type matching component effective in improving the retrieval performance? (Section 7.1)
- Can entity prior help rank candidate entities? (Section 7.1)
- Can Model A improve over Model B by considering the information of source entity? (Section 7.1)
- Can the proposed models yield competitive performance against the best TREC participating runs, after incorporating popular engineering heuristics? (Section 7.2)
- How does the use of web search engines for document retrieval impact the end-to-end retrieval performance? (Section 7.3)
- Can the proposed models be applied to other entity oriented search tasks such as those based on more structured semantic data? (Section 7.5)

## 7 Experiments

### 7.1 Effect of individual components

In this section, the proposed models' individual components and their combinations are evaluated. Table 3 contains the experimental results for Model A. $p(m = 1|e, R)$ represents the type matching component (Sections 3.3–3.5). $p(R|e, S)$ is the entity relevance component. Specifically, "HRM" represents the hierarchical relevance model and "LM" represents the language modeling approach (Section 3.6). $p(R|e, S)p(e|S)$ includes an extra component, co-occurrence between source entity and candidate entities $p(e|S)$ (Section 3.8), over $p(R|e, S)$. *Integrated*-HRM and *Integrated*-LM in Table 3 denote Model A that incorporates all the basic components, with HRM and LM as entity relevance model respectively. We can see that *Integrated*-HRM and *Integrated*-LM yielded comparable performance. The notations and semantics in Table 4 are similar for Model B. In Table 3, we conducted

**Table 4** Experimental results with Model B. The †symbol indicates statistical significance at 0.95 confidence interval against the baseline

| | 2009 NDCG@R | 2010 NDCG@R | 2011 R-Prec |
|---|---|---|---|
| $p(m = 1\|e, R)$ | 0.0580 | 0.0645 | 0.0376 |
| $p(R\|e)$ | | | |
| HRM | 0.2109 | 0.2043 | 0.1881 |
| LM | 0.2021 | 0.2152 | 0.1909 |
| $p(R\|e)p(e)$ | | | |
| HRM | 0.2162 | 0.2132 | 0.1939 |
| LM | 0.2078 | 0.2284† | 0.1963 |
| *Integrated*-HRM | 0.2457† | 0.2483† | 0.2101† |
| *Integrated*-LM | 0.2312† | 0.2617† | 0.2182† |

statistical significance test with $p(R|e, S)$-HRM and $p(R|e, S)$-LM as the baseline (i.e., *Integrated*-HRM and $p(R|e, S)p(e|S)$-HRM against $p(R|e, S)$-HRM, and *Integrated*-LM and $p(R|e, S)p(e|S)$-LM against $p(R|e, S)$-LM). Similar tests were conducted for Model B in Table 4. The two-tailed Student's t-test at 0.95 confidence level were used in all the experiments.

We can find that $p(m = 1|e, R)$ alone cannot yield good performance. However, when it is combined with the other components, the integrated models bring substantial improvement, by comparing *Integrated* vs $p(R|e, S)p(e|S)$ in Table 3 or $p(R|e)p(e)$ in Table 4. In addition, by comparing $p(R|e)p(e)$ with $p(R|e)$ in Table 4, we can see that the entity prior component leads to improved performance for all the three years. Similarly, the co-occurrence component $p(e|S)$ in Model A brings gains for the three years as well, if comparing $p(R|e, S)p(e|S)$ against $p(R|e, S)$ in Table 3. These results validate the importance of entity prior in ranking candidate entities. Furthermore, by comparing *Integrated* in Table 3 and Table 4, we can see that Model A is superior to Model B in all the three years. These results indicate that source entities contain informative evidence that can help identify target entities.

**Table 5** Top 10 entities returned for TREC 2010 Topic 29 (Figure 1) by different methods. Relevant entities are in bold and entities with the wrong types are in italics

| $p(m = 1\|e, R)$ | $p(R\|e)p(e)$ | $MA$ | $p(R\|e, S)p(e\|S)$ | $MB$ |
|---|---|---|---|---|
| nasdaq | **microsoft** | **boeing** | **coca cola** | **boeing** |
| bloomberg | **boeing** | **ibm** | **boeing** | **coca cola** |
| **ibm** | *federal reserve* | **pfizer** | *cnnmoney* | **microsoft** |
| news corporation | *european* | **coca cola** | *futures* | nasdaq |
| Yahoo | **coca cola** | **intel** | **microsoft** | **ibm** |
| atari | *uw* | **alcoa** | **pfizer** | **intel** |
| washington post | **ibm** | *cnnmoney* | **alcoa** | **merck** |
| **boeing** | **intel** | **mcdonald's** | **ibm** | **dupont** |
| *stanford* | *futures* | **merck** | *federal reserve* | **caterpillar** |
| enterprise media group | **merck** | **microsoft** | **mcdonald's** | *stanford* |

**Table 6** Comparison of our runs with the results from the top 3 participating groups in TREC Entity 2009. Best results on each metric are highlighted. The †symbol indicates statistical significance at 0.95 confidence interval against $MB$

| | NDCG@R | P@10 | #rel | #pri | WS |
|---|---|---|---|---|---|
| KMR1PU | 0.3061 | 0.2350 | 126 | 61 | Y |
| uogTrEpr | 0.2662 | 0.1200 | **347** | 79 | N |
| ICTZHRun1 | 0.2103 | 0.2350 | 80 | 70 | N |
| TREC Median | 0.0751 | 0.0050 | – | – | – |
| $MA$ | 0.2544 | 0.1970† | 244 | **84** | Y |
| $MB$ | 0.2457 | 0.1840 | 221 | 76 | Y |
| $MA^+$ | **0.3165**† | **0.2600**† | 168 | 72 | Y |
| $MB^+$ | 0.2938† | 0.2330† | 152 | 64 | Y |

**Table 7** Comparison of our runs with the results from the top 3 participating groups in TREC Entity 2010. Best results on each metric are highlighted. The †symbol indicates statistical significance at 0.95 confidence interval against $MB$

| | NDCG@R | MAP | R-Prec | WS |
|---|---|---|---|---|
| bitDSHPRun | **0.3694** | **0.2726** | **0.3075** | Y |
| FduWimET4 | 0.3420 | 0.2223 | 0.2837 | Y |
| KMR1PU | 0.2485 | 0.1555 | 0.2099 | Y |
| TREC Median | 0.1252 | 0.0628 | 0.0983 | - |
| $MA$ | 0.2725 | 0.1928† | 0.2427† | Y |
| $MB$ | 0.2617 | 0.1781 | 0.2285 | Y |
| $MA^+$ | 0.3235† | 0.2144† | 0.2786† | Y |
| $MB^+$ | 0.3030† | 0.2096† | 0.2684† | Y |

**Table 8** Comparison of our runs with the results from the top participating groups in TREC Entity 2011. Best results on each metric are highlighted. The †symbol indicates statistical significance at 0.95 confidence interval against $MB$

| | MAP | R-Prec | WS |
|---|---|---|---|
| TongKeyEN2 | 0.1209 | 0.1972 | Y |
| ICSTmaxSni | 0.0004 | 0.0015 | Y |
| $MA$ | 0.1648 | 0.2342† | Y |
| $MB$ | 0.1544 | 0.2182 | Y |
| $MA^+$ | **0.2039**† | **0.2691**† | Y |
| $MB^+$ | 0.1935† | 0.2598† | Y |

We illustrate the effect of the individual components by using TREC 2010 Topic 29 as an example. Table 5 lists the top 10 entities produced by different methods. $MA$ and $MB$ denote integrated Model A and Model B (with LM), respectively. We can find that $p(R|e)p(e)$ retrieves 6 relevant entities (bold font) mixed with 4 non-relevant entities that are not of the target type "Company" (italic font). On the other hand, although $p(m =$

**Table 9** Performance with web search (Google) versus INDRI for document retrieval

| | 2009 NDCG@R | 2010 NDCG@R | 2011 R-Prec |
|---|---|---|---|
| $MA^+$ Web Search | 0.3165 | 0.3235 | 0.2691 |
| $MA^+$ INDRI | 0.2632 | 0.2758 | 0.2323 |
| $MB^+$ Web Search | 0.2938 | 0.3030 | 0.2598 |
| $MB^+$ INDRI | 0.2588 | 0.2646 | 0.2232 |

$1|e, R)$ can identify 9 entities of correct type, only 2 of them are relevant. After combining the two components, $MB$ can find 8 relevant entities, with only one entity of wrong type. We can find the similar pattern for Model A by looking at $p(R|e, S)p(e|S)$ and $MB$. These observations indicate the effectiveness of the type matching component to downweigh the entities of wrong types in the ranking.

### 7.2 Incorporating advanced components

In this section, we evaluate the proposed models combined with the advanced components in Section 5. We denote the resulting models as $MA^+$ and $MB^+$ for Model A and Model B, respectively. $MA$ and $MB$ represent the same methods with *Integrated* in Tables 3 and 4, respectively. We compare the results with the best TREC participating runs. Tables 6, 7 and 8 show the experimental results. We can see that by incorporating the advanced components, the proposed methods yield better results than the best TREC automatic runs of 2009 and 2011, and generate competitive results for 2010. In addition, $MA^+$ delivers better performance than $MB^+$ for all the three years. These results along with $MA$ vs $MB$ indicate that Model A may be the better choice for the TREC Entity task. It is also worth noting that source entity is often not provided and available in real-world entity search. This may hinder the applications of Model A in practice.

### 7.3 Impact of web search

All the above experiments rely on web search engines for document retrieval (see Section 4.1). This is also the case for most of the TREC Entity participants. We list a column WS in Tables 6, 7, and 8 to show whether the TREC runs used web search (Y) or not (N). In TREC Entity 2011, groups that generate results using web search engines are required to submit an obligatory run, using a ClueWeb online query service based on Lemur[9] (as the time of conducting this experiment, the ClueWeb API is not accessible to the authors). In this section, we evaluate the proposed models based on the index built by the Lemur INDRI toolbox.[10] The following structured INDRI query is used to retrieve documents for each topic:

```
#weight(3.0 @odN(source entity) 3.0 @odN(keyword)
2.0 @odN(phrase) 1.0 (each term)
1.5 (acronym or full name of source entity)
1.0 (synonym and antonym of keyword))
```

[9] http://lemurproject.org/

[10] http://lemurproject.org/indri/

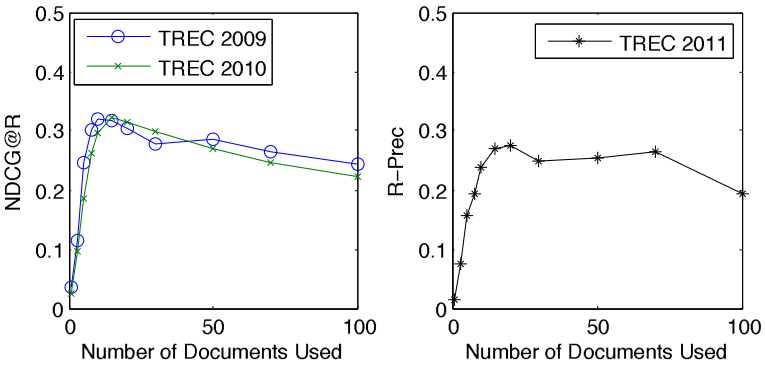


**Figure 3** Evaluation results of $MA^+$ with increasing number of documents used

where N is the number of words in the phrase. The keyword is the term in the narrative that reflects the main property of the target entity. Its extraction is introduced in Section 3.3. Acronym and synonym extraction is discussed in Section 4.1. The weights associated with the terms are manually set according to the perceived importance of the terms.

Table 9 shows the experimental results for $MA^+$ and $MB^+$. From the table, we can find that web search substantially outperforms the INDRI search on all the years. This may be explained by that fact that web search gets better quality documents which result in more related pages. It is also worth noting that $MA^+$ still yields better performance than $MB^+$, with INDRI retrieval.

### 7.4 Impact of number of documents used

This section investigates how the number of documents used affects the end-to-end entity search performance. The documents are retrieved by Google. We then use the top $K$ pages that are also part of the ClueWeb09 collections for other components in the entity search pipeline. Figure 3 shows the evaluation results of $MA^+$ by varying the number of documents used (i.e., $K$). We can see that all the three years only need around 10 to 20 documents to achieve their maximum performance. These results indicate the importance of identifying quality documents for entity search, which may also explain the large gap of the results between Google and INDRI in Section 7.3.

### 7.5 Application to entity list completion

The motivation of the Entity List Completion (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. The dataset for the ELC task is the Billion Triple Challenge dataset. [11] Since the data comes 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

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

**Table 10** Comparison of our runs with the results from the top 3 participating groups in TREC Entity 2010 ELC task. Best results on each metric are highlighted

| | MAP | R-Prec | #rel |
|---|---|---|---|
| KMR5PU | 0.2613 | **0.3116** | 33 |
| ilpsSetOLnar | 0.1152 | 0.0899 | **43** |
| LiraSealClwb | 0.0755 | 0.0494 | 15 |
| *Ours* | **0.2706** | 0.2911 | 40 |

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 toolbox. Following the work in [17], 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.

We use Model B to combine the evidence from documents, from type matching and from entity prior. Since the entity is represented by a document, any document retrieval model can be used to compute $p(R|e)$. We use the INDRI structured document retrieval model to calculate $p(R|e)$ as follows:

$$p(R|e) \propto \sum_{t \in R} \sum_{i=1}^{4} w_i (\gamma_T f_{iR}(t) + \gamma_O f_{iO}(t) + \gamma_U f_{iU}(t)) \tag{3}$$

where $f_i(t)$ denotes the Jelenik-Mercer smoothed log probabilities for the query term $T$. $w_i$ is the weighting parameter for the 4 selected attributes and the whole document, respectively. $R$ is the set of terms in relation $R$, $O$ is the set of ordered query terms, and $u$ is the set of unordered query terms. $\gamma$ is the corresponding parameters. All the parameter values are set to those suggested in [17].

For computing $p(m = 1|t_R, t_e)$, we use the same decomposition with (2). For the entities having Wikipedia entries, the approach to calculating $p(t_R|R)$, $p(t_e|e)$ and $p(m = 1|t_R, t_e)$ generally follows what is described in Section 3. Moreover, by utilizing the entities with known Wikipedia categories, we can train logistic regression models by treating the categories as labels. We can then use the classifiers to assign categories to the entities without Wikipedia entries. The entity prior is set by the frequency based method discussed in Section 3.7.

Table 10 shows the results along with those from the top 3 TREC participating groups. We can see that our method ("Ours") yields competitive results. Specifically, it achieves the best performance on MAP which is the main evaluation measure in the ELC task. The major difference between the proposed method and the best TREC run (KMR5PU) is the use of entity prior in the proposed model. It is also worth noting that our method did not exploit the example entities given in the query topics. The proposed approach can be used (probably more suitable) for the REF Linked Open Data (LOD) task defined in TREC Entity 2011. In the future work, we will consider re-ranking the candidate entities for the ELC task by utilizing the relations between the candidate entities and the example entities.

## 8 Conclusions and future work

Entity oriented search is becoming an important retrieval task in IR, as it has been recognized that the next generation of web search engines will need to move beyond document search and should be aware of entities. This paper proposes unified probabilistic models to formalize the process of related entity finding. The proposed methods incorporate entity relevance, type estimation, type matching, entity prior and entity co-occurrence into a holistic probabilistic framework. We conduct a comprehensive set of experiments on the tasks of TREC Entity tracks from 2009 to 2011. The experiments demonstrate the contributions of the individual components. In particular, we show that both the type matching component and the entity prior modeling component can effectively boost the entity retrieval performance. Moreover, combined with other components and heuristics in the retrieval pipeline, further improvements can be attained to deliver state-of-the-art performance on the end-to-end task of related entity finding. The proposed approach is also applied to another EOS task: entity list completion, and generates good performance. This indicates the wide applicability of the proposed approach in entity oriented search.

The work reported in this paper is an initial step toward a promising research direction. There are many interesting future research problems. First of all, it is interesting to explore the applicability of the proposed models in real-world entity search applications. Although Model A yielded better performance than Model B does in general, Model B may be more suitable for the real-world applications where users express their information needs by a natural language query without specifying any source entity. On the other hand, we can also adapt Model A by first detecting source entity from the query. There exists some prior work on named entity detection in query [1, 19, 23] which can be applied for Model B to perform on natural language queries. Secondly, it is worth exploring to improve the estimations of the individual components of the probabilistic models. For example, entity prior can be estimated by further considering the PageRank of the entity in Wikipedia. The work in expert search that computes candidate importance [41] can be utilized for setting entity prior as well. Entity type estimation and matching can also be improved by leveraging ontology structure of types (e.g., Wikipedia ontology). In addition, the estimations of the individual components need to be refined as well for the presence of semantic data. An important feature of semantic data is that it is densely connected. Web link structure has been successfully exploited by web search engines to improve document retrieval. It is likely that much of that work could be applied to the Semantic Web.

## References

1. Alasiry, A., Levene, M., Poulovassilis, A.: Extraction and evaluation of candidate named entities in search engine queries. In: Web Information Systems Engineering, pp. 483–496. Springer (2012)
2. Balog, K.: People search in the enterprise. In: SIGIR, pp. 916–916. ACM (2007)
3. Balog, K., de Rijke, M.: Determining expert profiles (with an application to expert finding). In: IJCAI'07, pp. 2657–2662 (2007)
4. Balog, K., Azzopardi, L., De Rijke, M.: Formal models for expert finding in enterprise corpora. In: SIGIR, pp. 43–50. ACM (2006)
5. Balog, K., de Vries, A., Serdyukov, P., Thomas, P., Westerveld, T.: Overview of the trec 2009 entity track. In: TREC (2009)
6. Balog, K., Meij, E., de Rijke, M.: Entity search: building bridges between two worlds. In: SemSearch Workshop, p. 9. ACM (2010)
7. Balog, K., Serdyukov, P., de Vries, A.: Overview of the trec 2010 entity track. In: TREC (2010)

8. Balog, K., Bron, M., De Rijke, M.: Query modeling for entity search based on terms, categories, and examples. ACM Trans. Inf. Syst. (TOIS) **29**(4), 22 (2011)
9. Balog, K., Serdyukov, P., de Vries, A.: Overview of the trec 2011 entity track. In: TREC (2011)
10. Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L.: Expertise retrieval. Found. Trends Inf. Retr. **6**(2–3), 127–256 (2012)
11. Bast, H., Chitea, A., Suchanek, F., Weber, I.: Ester: efficient search on text, entities, and relations. In: SIGIR, pp. 671–678. ACM (2007)
12. Bron, M., Balog, K., de Rijke, M.: Ranking related entities: components and analyses. In: CIKM, pp. 1079–1088. ACM (2010)
13. Chakrabarti, S., Puniyani, K., Das, S.: Optimizing scoring functions and indexes for proximity search in type-annotated corpora. In: WWW, pp. 717–726. ACM (2006)
14. Cheng, T., Yan, X., Chang, K.C.C.: Entityrank: searching entities directly and holistically. In: VLDB, pp. 387–398. VLDB Endowment (2007)
15. Craswell, N., de Vries, A., Soboroff, I.: Overview of the trec-2005 enterprise track. In: TREC, pp. 199–205 (2005)
16. Craswell, N., Demartini, G., Gaugaz, J., Iofciu, T.: L3s at inex 2008: retrieving entities using structured information. Adv. Focus. Retr. **5631**, 253–263 (2009)
17. Dalton, J., Huston, S.: Semantic entity retrieval using web queries over structured rdf data. In: SemSearch Workshop (2010)
18. de Vries, A., Vercoustre, A.M., Thom, J., Craswell, N., Lalmas, M.: Overview of the inex 2007 entity ranking track. Focus. Access XML Doc. (INEX) **4862**, 245–251 (2008)
19. Du, J., Zhang, Z., Yan, J., Cui, Y., Chen, Z.: Using search session context for named entity recognition in query. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 765–766. ACM (2010)
20. Fang, H., Zhai, C.X.: Probabilistic models for expert finding. ECIR **4425**, 418–430 (2007)
21. Fang, Y.: Entity retrieval by hierarchical relevance model, exploiting the structure of tables and learning homepage classifiers. In: TREC (2009)
22. Fang, Y., Si, L., Mathur, A.P.: Discriminative models of integrating document evidence and document-candidate associations for expert search. In: SIGIR, pp. 683–690. ACM (2010)
23. Guo, J., Xu, G., Cheng, X., Li, H.: Named entity recognition in query. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 267–274. ACM (2009)
24. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction (2005)
25. Jiang, J., Lu, W., Rong, X., Gao, Y.: Adapting language modeling methods for expert search to rank wikipedia entities. Adv. Focus. Retr. **5631**, 264–272 (2009)
26. Kaptein, R.: Result diversity and entity ranking experiments: anchors, links, text and wikipedia. In: TREC (2009)
27. Kaptein, R., Serdyukov, P., De Vries, A., Kamps, J.: Entity ranking using wikipedia as a pivot. In: CIKM, pp. 69–78. ACM (2010)
28. Kasneci, G., Suchanek, F.M., Ifrim, G., Ramanath, M., Weikum, G.: Naga: searching and ranking knowledge. In: ICDE, pp. 953–962. IEEE (2008)
29. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: SIGIR, pp. 111–119. ACM (2001)
30. Lafferty, J., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML (2001)
31. Lin, B., Rosa, K.D., Shah, R., Agarwal, N.: Lads: rapid development of a learning-to-rank based related entity finding system using open advancement. In: The First International Workshop on Entity-Oriented Search (2011)
32. Macdonald, C., Ounis, I.: Voting for candidates: adapting data fusion techniques for an expert search task. In: CIKM, pp. 387–396. ACM (2006)
33. Macdonald, C., Hannah, D., Ounis, I.: High quality expertise evidence for expert search. In: ECIR, pp. 283–295. Springer-Verlag (2008)
34. McCreadie, R.: University of glasgow at trec 2009: experiments with terrier. In: TREC (2009)
35. Pan, Z., Chen, H.: Tongkey at entity track trec 2011: related entity finding (2011)
36. Pasca, M.A., Harabagiu, S.M.: High performance question/answering. In: SIGIR, pp. 366–374. ACM (2001)
37. Prager, J.: Open-domain question answering. Found. Trends Inf. Retr. **1**(2), 91–231 (2006)
38. Ravichandran, D., Hovy, E.: Learning surface text patterns for a question answering system. In: ACL, Association for Computational Linguistics, pp. 41–47 (2002)

39. Santos, R.L.T., Macdonald, C., Ounis, I.: Voting for related entities. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, pp. 1–8 (2010)
40. Serdyukov, P.: Delft university at the trec 2009 entity track: ranking wikipedia entities. In: TREC (2009)
41. Serdyukov, P., Hiemstra, D.: Modeling documents as mixtures of persons for expert finding. In: ECIR, pp. 309–320 (2008)
42. Serdyukov, P., Rode, H., Hiemstra, D.: Modeling multi-step relevance propagation for expert finding. In: CIKM, pp. 1133–1142. ACM (2008)
43. Tsikrika, T., Serdyukov, P., Rode, H., Westerveld, T., Aly, R., Hiemstra, D., de Vries, A.: Structured document retrieval, multimedia retrieval, and entity ranking using pf/tijah. Focus. Access XML Doc. (INEX) **4862**, 306–320 (2008)
44. Vechtomova, O., Robertson, S.E.: A domain-independent approach to finding related entities. Inf. Process. Manag. (2012)
45. Vercoustre, A.M., Pehcevski, J., Thom, J.: Using wikipedia categories and links in entity ranking. Focus. Access XML Doc. (INEX) **4862**, 321–335 (2008)
46. Vinod Vydiswaran, V.G.: Finding related entities by retrieving relations: Uiuc at trec 2009 entity track. In: TREC (2009)
47. Voorhees, E.M.: The trec-8 question answering track report. In: TREC, vol. 8, pp. 77–82 (1999)
48. Wang, D., Wu, Q., Chen, H., Niu, J.: A multiple-stage framework for related entity rinding: Fdwim at trec 2010 entity track. TREC (2010)
49. Wang, Z.: Bupt at trec 2009: entity track. In: TREC (2009)
50. Weerkamp, W., Balog, K., Meij, E.: A generative language modeling approach for ranking entities. Adv. Focus. Retr. **5631**, 292–299 (2009)
51. Wu, Y., Kashioka, H.: Nict at trec 2009: employing three models for entity ranking track. In: TREC (2009)
52. Yang, Q.: Experiments on related entity finding track at trec 2009. In: TREC (2009)
53. Zhai, H.: A novel framework for related entities finding: Ictnet at trec 2009 entity track. In: TREC (2009)
54. Zheng, W.: Udel/smu at trec 2009 entity track. In: TREC (2009)
55. Zhu, J., Song, D., Rüger, S.: Integrating document features for entity ranking. Focus. Access XML Doc. (INEX) **4862**, 336–347 (2008)