2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Bipartite Graph based Entity Ranking for Related Entity Finding

Lei Cao, Jiafeng Guo, Xueqi Cheng
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{*leicao, guojiafeng*}*@software.ict.ac.cn, cxq@ict.ac.cn*

***Abstract*—Related entity finding (REF) is a promising application, which aims to return a list of related entities given a query that describes the source entity, the specific type of target entities, and the relation between the source entity and target entities. One typical entity ranking strategy is to rank the candidate entities based on the co-occurrence between the candidate entities and the given query. However, such a strategy is inadequate to rank entities properly especially for those related but unpopular entities. In this paper, we propose a bipartite graph based entity ranking method, which leverage the Co-List relationship between candidate entities (i.e., entities co-occurring in the same structured/unstructured lists) to help improve the entity ranking. Specifically, we first estimate the initial relevance scores for the candidate entities based on a generative probabilistic model. We then construct a bipartite graph based on Co-List relation between candidate entities, and apply an iterative refinement process analogous to heat diffusion on the graph to propagate the relevance scores over entities. Finally the candidate entities are ranked according to their refined scores. We further develop an optimization framework for the iterative refinement process in our ranking method. Experimental results on the data collection from the TREC 2010 Entity Track show the effectiveness of our proposed method.**

***Keywords*-entity ranking; bipartite graph model; related entity finding**

I. Introduction

As the World Wide Web has been growing rapidly to be a huge knowledge repository, more and more users seek the information from the Web. Traditional information retrieval systems usually return documents relevant to users' information need. However, it is inadequate for a wide range of query tasks. In many situations, users' information need may be better answered by specific entities instead of documents. For example, an auto-racing fan may want to find the teammates of Michael Schumacher, or a statesman may want to know the member countries of OPEC. It would be very laborious and time-consuming for users to look through the relevant documents one by one to find the right answers (i.e., related entities). Therefore, it is necessary to study how to find the entities related to users' information need automatically.

The problem of *related entity finding*(REF) focuses on returning a ranked list of related entities for a given query. Different from the keyword queries in traditional document retrieval, a query for REF needs to specify the type and the relevance criteria of the target entities. Similar to the setup in TREC 2009 Entity Track[1], a query for REF can be formulated as a structured query which consists of the source entity, the specific type of target entities, and the relation between the source entity and the target entities. For example, a user who wants to know the member countries of OPEC may formulate a query as illustrated in Fig.1. In this query, the source entity is "OPEC", the specific type is "Location" and the relation is "member countries of". Thus the system needs to automatically find a list of countries which are the members of the OPEC, such as "Saudi Arabia", "Iraq", "United Arab Emirates" and so on.

```
<query>
    <source_entity>Organization of Petroleum Exporting
                Countries (OPEC)</source_entity>
    <target_type>Location</target_type>
    <relation>
        Find countries that are members of OPEC.
    </relation>
</query>
```

Figure 1. The example query

The REF problem discussed here is similar to the main task proposed by TREC Entity Track[1]. The only difference is that the official task of TREC Entity Track requires the system to return both the related entities and their homepages. Since the homepage finding component can be naturally separated from the REF task and studied as a standalone task[2], we only focus on the REF problem in this paper.

Several ranking methods[3][4][5][6] have been proposed for the REF problem. Most of them rank candidate entities based on the co-occurrence between candidate entities and the given query. However, the co-occurrence based ranking strategy is inadequate to rank entities properly especially for those related but unpopular entities (i.e., related entities which has limited co-occurrence with the given query). On the contrary, some unrelated entities which happen to co-occur with the given query may easily obtain higher rank in existing approaches.

To overcome aforementioned limitations, we propose to take the relation between candidate entities into consideration. As we observe, many related entities for a given query often co-occur in the same structured/unstructured lists. For example, the member countries of OPEC may

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.60

appear in the same HTML Tables, or enumerated in the same unstructured text snippets. Such co-occurrence relation between entities, named as *Co-List Relation*, conveys a large amount of knowledge of the semantic coherence or similarity between entities. Therefore, we can leverage the Co-List relation to help boost those related but unpopular entities in the REF task.

In this paper, we propose a bipartite graph based entity ranking method which leverages the Co-List relation between entities for related entity finding. Given a query, the candidate entities are extracted from relevant documents, and we estimate the initial relevance scores for the candidate entities based on a generative probabilistic model. We then construct a bipartite graph based on Co-List relation between candidate entities, and apply an iterative refinement process analogous to heat diffusion on the graph to propagate the relevance scores over entities. Finally the candidate entities are ranked according to their refined scores. We further develop an optimization framework for the iterative refinement process in our ranking method. Experiments on the data collection from the TREC 2010 Entity Track show that our method is more effective than traditional ranking strategies, which are simply based on co-occurrence between candidate entities and the given query.

The main contributions of the paper are multi-fold:

- We propose to use the Co-List relation for REF task
- We design a bipartite graph based ranking method to leverage the Co-List relation for entity ranking
- We develop an optimization framework for the iterative refinement process in our ranking method

The remainder of the paper is organized as follows. Section II discusses related work. Section III introduces the Co-List relation between entities. In Section IV we describe our ranking method for REF. In section V the experimental results are provided. In section VI we conclude and suggest future work.

II. Related Work

The REF task is similar to the traditional Question Answering (QA) task [7], in particular, the problem of list question answering [8]. However, REF is different from QA. Firstly, the collection of documents used in traditional QA systems is a relative small set of newswire and newspaper articles whereas related entity finding uses a large, noisy Web corpus, which makes it hard to apply deep linguistic analysis. Secondly, the query in QA is often in the form of natural language while the query in REF is often in a structured form as illustrated in Fig. 1. Thirdly, REF focuses more on the entities which engage in the specified relation with source entity while the answers of QA are not necessary to be named entities.

REF is also related to Expert Finding [9]. Since the Expert finding task only focuses on a specific type ”Person” and a specific relation ”expert of”, the REF task can be considered as a more general problem. Fang et al. [10] proposed a general probabilistic framework for studying expert finding problem and derived two families of generative models. Cao et al. [11] proposed two-stage language model for expert finding, and further utilized the relation between people to enhance the expert search results. The relation they used is simple co-occurrence between people, which is different from the Co-List relation we consider in our work.

The INEX Entity Ranking track which uses Wikipedia as the data collection has introduced two tasks, entity ranking and list completion[12]. In both of the two tasks, the system should return the relevant entities, which are represented by their corresponding Wikipedia pages. Several approaches have been proposed for the task. Balog et al. [13] developed a generative language modeling approach for the entity ranking, and explored various ways of estimating these components in the model. Vercoustre et al. [14] utilized the known categories, the link structure of Wikipedia, as well as the link co-occurrences with the examples (when provided) to improve the effectiveness of entity ranking. Kaptein et al. [15] proposed to use Wikipedia as pivot for find entities on the web, and exploit the structure of Wikipedia to improve entity ranking effectiveness.

More recently, TREC Entity Track introduced the task of related entity finding [1]. Some methods have been proposed for the task.Fang et al. [3] proposed a hierarchical relevance retrieval model for entity ranking. Furthermore, they exploit the structure of tables and lists to identify the target entities from them by making a joint decision on all the entities with the same attribute. McCreadie et al. [4] built semantic relationship support for the Voting Model, by considering the co-occurrences of query terms and entities in a document as a vote for the relationship between these entities. Zhai et al. [5] proposed a novel probabilistic model for the task related entities finding in a Web collection. Bron et al. [6] proposed for REF a framework consisting of four core components: co-occurrence models, type filtering, context modeling and homepage finding. Yang et al.[16] reconstructed logical sitemap and applied it in Related Entity Finding Task to find the relevant pages by integrating additional site level information . Wang et al. [17] proposed the document-centered model and entity-centered model for related entity finding.

Bipartite graph based ranking method has been applies in different applications. Rui et al. [18] proposed a bipartite graph reinforcement model for image annotation. Deng et al. [19] proposed a generalized Co-HITS algorithm based on bipartite graph for query suggestion, and they further investigated the algorithm based on two frameworks, including the iterative and the regularization frameworks. The iterative process in our work is different from theirs. Their iterative process is akin to random walk process while our iterative process is analogous to heat diffusion. Zhou et al. [20] introduced a hybrid method which combined ”heat-

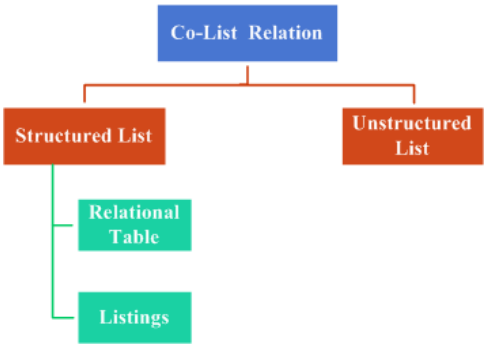

Figure 2. the Taxonomy for Co-List Relation between Entities

Figure 4. The HTML list for Co-List between entities

spreading" process and "probabilistic spreading" process to solve the diversity-accuracy dilemma of recommender systems.

III. Co-List Relation between Entities

As observed, many related entities for a given query often co-occur in the same structured/unstructured lists(e.g. tables or text enumerations), and the entities in the same list are usually semantically similar or coherent. Here we define such co-occurrence relation between entities as Co-List relation, which is the key concept in our work. We will leverage the Co-List relation to propagate the relevance over entities, thus to boost those related but unpopular entities .

The Co-List relation can be encoded in different forms. In order to better understand the Co-List relation, we propose a taxonomy to further specify such a relation as illustrated in Fig. 2. The Co-List relation can be categorized into two major types, i.e. Structured List and Unstructured List, according to whether the Co-List relation is encoded in the structured form or unstructured form.

Structured List can be further divided into two sub-categories: Relational Table and Listings. Relational Table denotes the set of HTML tables which contain relational knowledge(i.e. entities and attributes). Such HTML tables are often regarded as a two-dimension representation of logical relationship between groups of data. For example,

Current members

OPEC has twelve third world member countries: six in the Middle East, four in Africa, and two in South America.

Country	Region	Joined OPEC[22]	Population (July 2008)[23]	Area (km²)[24]
Algeria	Africa	1969	33,779,668	2,381,740
Angola	Africa	2007	12,531,357	1,246,700
Ecuador	South America	2007[A 1]	13,927,650	283,560
Iran	Middle East	1960[A 2]	75,875,224	1,648,000
Iraq	Middle East	1960[A 2]	28,221,180	437,072
Kuwait	Middle East	1960[A 2]	2,596,799	17,820
Libya	Africa	1962	6,173,579	1,759,540
Nigeria	Africa	1971	158,259,000	923,768
Qatar	Middle East	1961	824,789	11,437
Saudi Arabia	Middle East	1960[A 2]	28,146,656	2,149,690
United Arab Emirates	Middle East	1967	4,621,399	83,600
Venezuela	South America	1960[A 2]	26,414,816	912,050
Total			369,368,429	11,854,977 km²

Figure 3. The structured HTML table for Co-List between entities

the HTML table in Fig. 3 lists the objects relevant to the query as illustrated in Fig. 1. The related entities such as Algeria, Angola and Iran are listed in the first column of the table, and some attributes of the related entities are listed in other columns. Listings denotes the set of HTML lists which contain similar or coherent entities. Listings can be further divided into Vertical Listings, Horizontal Listings and 2-D Listings. For example, for the query illustrated in Fig. 1, the related entities may be enumerated in a list, which belongs to the type Vertical Listings.(see Fig. 4).

Unstructured List denotes the set of text snippets in which entities are enumerated. Such text snippets are common over the web, and they can be found in many web pages. For example, for the query illustrated in Fig. 1, the related entities are enumerated in a text snippet and separated by comma. (see Fig. 5)

> The 12 OPEC members are Algeria, Angola, Indonesia, Iran, Iraq, Kuwait, Libya, Nigeria, Qatar, Saudi Arabia, the United Arab Emirates (UAE) and Venezuela.

Figure 5. The unstructured text snippet for Co-List between entities

We apply different methods to extract the *instances of Co-List relation* (i.e. specific forms of Co-List relation in the web pages such as relational HTML tables, HTML lists and text snippets). For Structured List, we apply the rule-based method to extract the corresponding HTML tables or HTML lists which may contain several candidate entities. The features we use are similar to those used in [21] and [22]. For Unstructured List, we use a set of query-independent generic patterns to identify the corresponding text snippets (see Fig. 6). For example, if a sentence contains the keywords in the query followed by "such as", followed by a list of simple noun phrases which may contain several candidate entities, it would be recognized as the instance of Co-List relation. In order to ensure the quality of extraction, the context of instance of Co-List relation is required to be relevant to the given query.

Pattern 1 : ..... such as A, B, C, ...
Pattern 2 : ..... are/were A, B, C, ...
Pattern 3 : ..... including A, B, C, ...
Pattern 4 : ..... includes A, B, C, ...
.....

Figure 6. The example generic patterns

IV. Related Entity Ranking

A. Problem Setup and Notation

Formally, let D=$\{d_1, d_2, ..., d_n\}$ denotes the document collection, E_t=$\{e_{t_1}, e_{t_2}, ..., e_{t_m}\}$ denotes a set of entities with the type T, and $r = (e_i, e_j)$ denotes the relation between two entities e_i and e_j. We define a query $Q = (e_s, r, T)$ for REF, where e_s is the source entity, T is the specific type of target entities, and r is the relation between the source entity and target entities. Therefore, the problem of related entity finding can be formalized as follows: Given the query Q, the goal is to find a set of target entities E_t from document collection D.

B. Initial Entity Relevance Estimation

Given a query, we first extract relevant text snippets from the web documents. We then apply an off-the-shelf named entity recognizer such as Stanford NER tagger to recognize named entities with the specific type. Due to the difference between the training corpus and our web documents, the off-the-shelf NER tagger may fail to recognize the named entities. Therefore, we also extract the anchor texts from the relevant text snippets as candidate entities since many named entities will be formed as anchor texts in web documents.

we propose a generative probabilistic model to estimate the initial relevance score. Given a candidate entity, its relevance score to the given query is defined as

$$score(e_t, Q) = p(e_t|Q) \tag{1}$$

where $Q = (e_s, r, T)$. Based on Bayes' rule, we can easily derive that

$$\begin{aligned} p(e_t|Q) &\propto p(e_t, Q) \\ &= p(e_t, e_s, r, T) \\ &= p(T|e_t, e_s, r)p(e_t, e_s, r) \\ &= p(T|e_t, e_s, r)p(e_t|e_s, r)p(e_s, r) \\ &\approx p(T|e_t)p(e_t|e_s, r) \end{aligned} \tag{2}$$

In Eqn. (2), we assume the target type T is independent of the relation r and the source entity e_s. We also drop $p(e_s, r)$ in Eqn. (2) as it is assumed to be uniform, thus does not influence the ranking. There are two components to be estimated: $p(T|e_t)$ and $p(e_t|e_s, r)$.

$p(T|e_t)$ is the probability that e_t mentions the specific type T. This component is estimated as follows,

$$p(T|e_t) = \frac{count(T, e_t)}{count(e_t)} \tag{3}$$

where $count(T, e_t)$ is the count of text snippets in which e_t is recognized with the target type T by the NER tagger, and $count(e_t)$ is the count of text snippets in which e_t occurs. It is assumed that if e_t is recognized with the specific type T by the NER tagger in more text snippets, it is more likely to belong to the specific type T.

$p(e_t|e_s, r)$ is the probability that (e_s, r) mentions e_t. It is estimated as follows,

$$p(e_t|e_s, r) = \frac{count(e_t, e_s, r)}{\sum_{e'_t} count(e'_t, e_s, r)} \tag{4}$$

where $count(e_t, e_s, r)$ is the count of text snippets in which e_t, e_s and r co-occur. This component shows the strength of popularity of related entity e_t which engage in the relation r with the source entity e_s.

C. Bipartite Graph based Refinement

We construct a bipartite graph based on candidate entities and instances of Co-List relation, and perform an iterative refinement process on the graph to boost the performance of ranking.

1) Graph Construction: We construct a bipartite graph based on the candidate entities and instances of Co-List relation between entities. The candidate entities and instances of Co-List relation are regarded as the two disjoint sets of graph vertices. If one candidate entity exists in an instance of Co-List relation, the vertices corresponding to them will be connected by an edge. However, the bipartite graph may be disconnected because some candidate entities may not exist in any instance of Co-List relation, or we fail to find any instance of Co-List relation for the given query. Therefore, we build a virtual instance of Co-List relation. Every vertex representing candidate entity is connected to the vertex representing the virtual instance of Co-List relation. Thus vertices from different disjoint sets are all connected using edges with proper weight.

Let $G = (U \bigcup V, E)$ denotes the bipartite graph we construct. U=$\{u_1, u_2, ..., u_m\}$ denotes the set of candidate entities. V=$\{v_1, v_2, ..., v_n\}$ denotes the set of instances of Co-List relation. For two vertices u_i, v_j of one edge, the weight of the edge is defined as follows

$$w_{ij} = \begin{cases} 1 & \text{if } u_i \text{ is connected to } v_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Every vertex in the bipartite graph has an initial weight. For the vertices representing the candidate entities, their initial weights are set to the scores obtained from Eqn.(2). For the vertices representing the instances of Co-List relation, their initial weights are set to the normalized uniform

values (i.e. $1/n$ where n is the count of instances of Co-List relation). The initial weight of the vertex representing the virtual instance of Co-List relation is set to be zero.

2) Refinement over the Graph: Once the bipartite graph is constructed, we apply the iterative process analogous to heat diffusion on it for the refinement. The equation for the iteration is as follows,

$$\begin{cases} X_{t+1} = \alpha M_1 Y_t + (1-\alpha) X_0 \\ Y_{t+1} = \beta M_2 X_{t+1} + (1-\beta) Y_0 \end{cases} \tag{6}$$

where $M_1 = D_X^{-1} A$ and $M_2 = D_Y^{-1} A^T$.

In Eqn. (6), A is the bi-adjacency matrix of bipartite graph G. A^T is the transpose of A. D_X is the diagonal matrix with its (i:i)-element equal to the sum of the i-th row of A; D_Y is the diagonal matrix with its (i;i)-element equal to the sum of the i-th column of A. X_0 is the vector of initial scores of candidate entities. Y_0 is the vector of initial scores of instances of Co-List relation. X_t and Y_t are the ranking value vectors after t iterations. α and β are the weights which range from 0 to 1. The first row in Eqn. (6) addresses the ranking update of candidate entities, and the second row in Eqn. (6) indicates the ranking update of instances of Co-List relation.

3) Convergence: We show the sequences $\{X_t\}$ and $\{Y_t\}$ in the iterative refinement process converge. Considering the sequence $\{X_t\}$ firstly, by the iteration in Eqn. (6), we have

$$X_{t+1} = [\alpha(1-\beta)M_1 Y_0 + (1-\alpha)X_0] \sum_{k=0}^{t} (\alpha\beta M_1 M_2)^k + (\alpha\beta M_1 M_2)^{t+1} X_0 \tag{7}$$

Since $0 < \alpha, \beta < 1$, and the eigenvalues of $M_1 M_2$ are in [-1, 1], we have

$$\begin{cases} \lim_{t\to\infty} (\alpha\beta M_1 M_2)^{t+1} = 0 \\ \lim_{t\to\infty} \sum_{k=0}^{t} (\alpha\beta M_1 M_2)^k = (I - \alpha\beta M_1 M_2)^{-1} \end{cases} \tag{8}$$

Hence,

$$X^* = \lim_{t\to\infty} X_t = (I - \alpha\beta M_1 M_2)^{-1} [\alpha(1-\beta) M_1 Y_0 + (1-\alpha) X_0] \tag{9}$$

Similarly, we can get

$$Y^* = \lim_{t\to\infty} Y_t = (I - \alpha\beta M_2 M_1)^{-1} [\beta(1-\alpha) M_2 X_0 + (1-\beta) Y_0] \tag{10}$$

Now we can compute X^* and Y^* directly without iterations.

4) Optimization Framework: We develop optimization framework for the iteration in Eqn. (6). The cost function associated with X and Y is defined to be

$$\begin{aligned} Q(X,Y) = & \frac{1}{2} \sum_{i,j=1}^{m} w_{ij}^{xx} \|x_i - x_j\|^2 \\ & + \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}^{yy} \|y_i - y_j\|^2 \\ & + \frac{\mu}{2} \Big\{ \sum_{i=1}^{m} d_i^x \|x_i - f_i^0\|^2 + \sum_{j=1}^{n} d_j^y \|y_j - g_j^0\|^2 \Big\} \end{aligned} \tag{11}$$

where μ is the regularization parameter and

$$\begin{cases} w_{ij}^{xx} = \sum_{k=1}^{n} \frac{a_{ik} a_{kj}}{d_k^y} \\ w_{ij}^{yy} = \sum_{k=1}^{m} \frac{b_{ik} b_{kj}}{d_k^x} \\ F^0 = \frac{\alpha(1-\beta) M_1 Y_0 + (1-\alpha) X_0}{1-\alpha\beta},\ f_i^0 \in F^0 \\ G^0 = \frac{\beta(1-\alpha) M_2 X_0 + (1-\beta) Y_0}{1-\alpha\beta},\ g_j^0 \in G^0 \end{cases} \tag{12}$$

In Eqn. (11) and Eqn. (12), a_{ik} is the (i:k)-element of matrix A, b_{ik} is the (i:k)-element of matrix A^T, d_k^x is the sum of k-row of matrix A, d_k^y is the sum of k-column of matrix A, f_i^0 is the i-th element of vector F^0, and g_j^0 is the j-th element of vector V^0. Thus the iteration in Eqn. (6) can be transformed into the following optimization problem,

$$(X^*, Y^*) = argmin Q(X,Y) \tag{13}$$

Differentiating $Q(X,Y)$ with respect to X and Y respectively, we have

$$\begin{cases} \frac{\partial Q(X,Y)}{\partial X}|_{X=X^*} = X^* - M_1 M_2 X^* + \mu(X^* - F^0) = 0 \\ \frac{\partial Q(X,Y)}{\partial Y}|_{Y=Y^*} = Y^* - M_2 M_1 Y^* + \mu(Y^* - G^0) = 0 \end{cases} \tag{14}$$

If we set

$$\alpha\beta = \frac{1}{1+\mu} \tag{15}$$

finally we have

$$\begin{cases} X^* = (I - \alpha\beta M_1 M_2)^{-1} [\alpha(1-\beta) M_1 Y_0 + (1-\alpha) X_0] \\ Y^* = (I - \alpha\beta M_2 M_1)^{-1} [\beta(1-\alpha) M_2 X_0 + (1-\beta) Y_0] \end{cases} \tag{16}$$

From the above procedure, we prove that Eqn. (11) is the corresponding optimization framework for the iteration in Eqn. (6).

V. Experiment

A. Experimental Setting

All the experiments in our work are conducted on a standard collection from TREC 2010 Entity Track. The collection includes: (1) ClueWeb09 Category A (2) 47 test topics. In ClueWeb09 Category A, there are about 500 million English web pages. The test topics we used here are from topic#21 to topic#70 excluding three topics (#35, #46 and #59). Note that test topics #35, #46 and #59 are not judged in the evaluation of the TREC 2010 Entity Track. Every test topic consists of the source entity, the target entity type and the desired relation described by free text. The test topics are listed in Table I. Due to the lack of space we only list the first ten test topics.

For each test topic, our system returns a list of top 100 ranked entities. The top 100 ranked entities are labeled by five annotators. Each candidate entity is labeled as one of two levels: Relevant (Score 1), Irrelevant (Score 0). We evaluate the performance based on three measures: NDCG(i.e. normalized Discounted Cumulative Gain), P@10 (i.e. Precision at rank 10) and MAP (i.e. Mean Average Precision). After computing the measures for ranked entity

Table I
THE LIST OF FIRST TEN TEST TOPICS. TARGET TYPE ARE ORG=ORGANIZATION,PER=PERSON,LOC=LOCATION, PRO=PRODUCT.

ID	Source Entity	Relation	Target Type
21	Bethesda,Maryland	What art galleries are located in Bethesda, Maryland?	LOC
22	OPEC	Find countries that are members of OPEC	LOC
23	The Kingston Trio	What recording companies now sell the Kingston Trio's songs?	ORG
24	Jazz at Lincoln Center Orchestra	Find the members of the Jazz at Lincoln Center Orchestra.	PER
25	U.S. Supreme Court	From what schools did the Supreme Court justices receive their undergraduate degrees?	ORG
26	Cray XT computer	Who has installed (taken delivery of) a Cray XT computer?	ORG
27	Department of Mathematics, Montgomery College, Rockville	Who are the publishers of the text books used in this department?	ORG
28	IEEE Engineering in Medicine and Biology Society	Find journals published by the IEEE Engineering in Medicine and Biology society.	PRO
29	Dow Jones	Find companies that are included in the Dow Jones industrial average	ORG
30	Ocean Spray Cranberries, Inc.	Find U.S. states and Canadian provinces where Ocean Spray growers are located.	LOC
.....	.....	.....	.....

list of every query, we can average them to obtain an overall performance evaluation of the entity ranking method. The parameters α and β in Eqn. (6) are set to 0.8 and 0.5 respectively, and their sensitivities will be analyzed in detail in section V-E.

Here, we note that the evaluation methodology in our work is slightly different from that used in the TREC Entity Track. The Entity Track considered the problem of finding both related entities and their homepages, and the evaluation measures are based on the homepages of the entities. However, our focus is related entity finding. Thus the evaluation measures in our work are only based on entities.

B. Baseline methods

In the experiments, we compare our proposed Bipartite Graph based Entity Ranking method(BGER) to the following methods:

- Naive Method (Naive). In the naive method, we first extract the relevant text snippets from the document collection. We then apply the off-the-shelf NER tagger directly to recognize the entities with the specific type. Finally we rank the candidate entities based on the co-occurrence between the candidate entity and the given query. We regard naive method as the baseline.
- Generative Probabilistic Model based Entity Ranking (GPMER). We use the scores obtained from initial estimation in section IV-B. GPMER is similar to most of traditional strategies, which are based on the co-occurrence between entities and the given query.

C. Overall Performance

The experimental results are shown in Table II. From the results, we can see that GPMER slightly outperform the naive method. Due to the difference between the training corpus to build the NER tagger and the web document collection, the off-the-shelf NER tagger may generate lots of noise. By leveraging the anchor texts and generative language model, GPMER can improve the performance for entity ranking.

Table II
THE PERFORMANCE FOR METHODS NAIVE, GPMER AND BGER

	Naive	GPMER	BGER
Average P@10	0.1745	0.1830	0.2787
Average NDCG	0.3784	0.3960	0.5174
MAP	0.1260	0.1342	0.2265

From Table II, we can see that BGER can outperform GPMER in terms of all three measures. Compared to GPMER, the Average P@10, Average NDCG and MAP in BGER is improved by about 52.30%, 30.66%, 68.78% respectively. We also conduct the T-Test and find that the improvement is significant (p-value$<$0.01). It indicates that BGER can boost those related but unpopular entities by leveraging the Co-List relation.

D. Case Study

We take test topic#29 as an example. The top 10 entities returned by Naive, GPMER and BGER are listed in table III respectively. Note that related entities are in bold. In table III, GPMER returns 7 related entities in the top 10. We have observed that the related entities for test topic#29 often be listed in a HTML table (see Fig. 9) to be represented to the users. We can make use of the phenomenon to propagate the relevance over the entities to boost the those, which are related but less popular. After the refinement, BGER will returns 9 related entities in top 10.

Table III
TOP 10 ENTITIES FOR TOPIC#29 GENERATED BY METHODS NAIVE, GPMER AND BGER RESPECTIVELY.RELEVANT ENTITIES ARE IN BOLD.

Naive	*GPMER*	*BGER*
Dow Jones Industrial Average	New York Stock Exchange	**Bank of America**
New York Stock Exchange	**Bank of America**	**IBM**
S&P	**IBM**	**Alcoa**
Bank of America	Dow Jones Industrial Average	**American Express**
Wall Street Journal	**Alcoa**	**Microsoft**
Honeywell	S&P	**Intel**
Standard & Poor	**American Express**	**Boeing**
Alcoa	**Microsoft**	**Pfizer**
Reuters	**Intel**	**Hewlett-Packard**
Federal Reserve	**Pfizer**	General Motors

Although the method BGER can improve the performance for entity ranking, there are still noisy entities at the top of ranking list. For example, the entity "General Motors" in the list returned by BGER is unrelated to the test topic#29. This caused by the following reason. "General Motors" was once one of the components of Dow Jones Industry Average. But on June 8, 2009 it was replaced by other companies. Because the web document collection we use contains lots of history

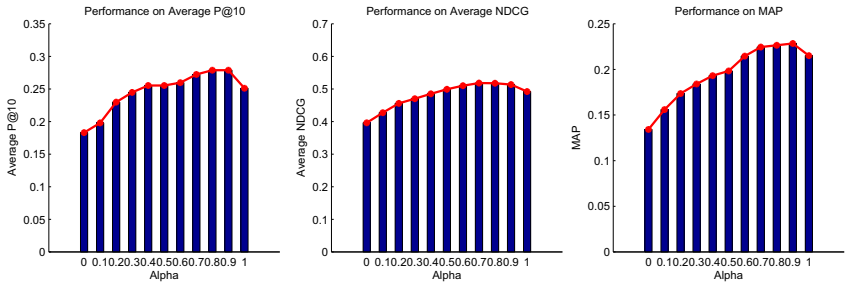

Figure 7. the performance of BGER with respect to the parameter α

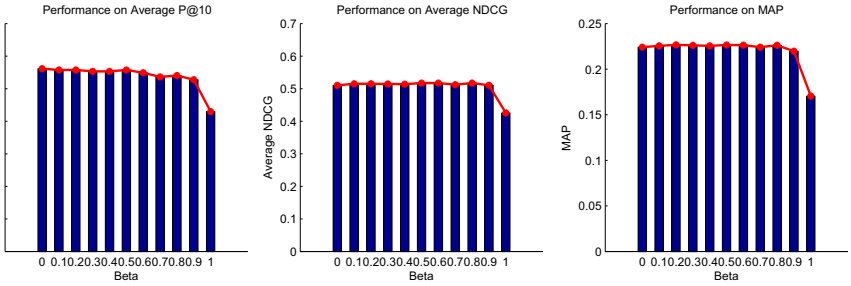

Figure 8. the performance of BGER with respect to the parameter β

information, such an error may happen. This kind of error may be eliminated when we consider the time factor of the web pages.

Components

The Dow Jones Industrial Average currently consists of the following 30 companies:[4]

Company	Symbol	Industry	Date Added
3M	MMM	Conglomerate	1976-08-09 (as Minnesota Mining and Manufacturing)
Alcoa	AA	Aluminum	1959-06-01 (as Aluminum Company of America)
American Express	AXP	Consumer finance	1982-08-30
AT&T	T	Telecommunication	1999-11-01 (as SBC Communications)
Bank of America	BAC	Banking	2008-02-19
Boeing	BA	Aerospace and defense	1987-03-12
Caterpillar	CAT	Construction and mining equipment	1991-05-06
Chevron Corporation	CVX	Oil & gas	2008-02-19
Cisco Systems	CSCO	Computer networking	2009-06-08
Coca-Cola	KO	Beverages	1987-03-12
DuPont	DD	Chemical industry	1935-11-20 (also 1924-01-22 to 1925-08-31)
ExxonMobil	XOM	Oil & gas	1928-10-01 (as Standard Oil)
General Electric	GE	Conglomerate	1907-11-07
Hewlett-Packard	HPQ	Technology	1997-03-17
The Home Depot	HD	Home improvement retailer	1999-11-01
Intel	INTC	Semiconductors	1999-11-01
IBM	IBM	Computers and technology	1979-06-29
Johnson & Johnson	JNJ	Pharmaceuticals	1997-03-17
JPMorgan Chase	JPM	Banking	1991-05-06 (as J.P. Morgan & Company)
Kraft Foods	KFT	Food processing	2008-09-22
McDonald's	MCD	Fast food	1985-10-30
Merck	MRK	Pharmaceuticals	1979-06-29
Microsoft	MSFT	Software	1999-11-01
Pfizer	PFE	Pharmaceuticals	2004-04-08
Procter & Gamble	PG	Consumer goods	1932-05-26
Travelers	TRV	Insurance	2009-06-08
United Technologies Corporation	UTX	Conglomerate	1939-03-14 (as United Aircraft)
Verizon Communications	VZ	Telecommunication	2004-04-08
Wal-Mart	WMT	Retail	1997-03-17
Walt Disney	DIS	Broadcasting and entertainment	1991-05-06

Figure 9. the example Html table for test topic #29

We also note that BGER doesn't work well for some test topics compared to GPMER. This may be caused by two reasons. (1) For some test topic we may fail to find any instance of Co-List relation. In this situation, the final ranking list will at least retain the original order obtained from the GPMER. (2) For some test topic we may extract noisy instances of Co-List relation. Thus the performance of ranking for that topic will be worsened. Such kind of error may be eliminated by using more effective methods for the extraction of instances of Co-List relation. However, BGER can bring better order than GPMER for most of the test topics, and the overall performance will be improved finally.

E. Parameters Sensitivity Analysis

In BGER, there are two parameters α and β (see Eqn. (6)). The parameter α is a trade-off to balance the contributions of the initial relevance estimation and the Co-List relation between entities. Similar to α, β is also the trade-off to balance the contributions of the initial scores and candidate entities. We conduct experiments to analyze their sensitivity.

We first set β to 0.5 and range α from 0 to 1, an increase of 0.1 each. Fig. 7 illustrates the results for average P@10, average NDCG and MAP. From Fig. 7, we find that the performance is smooth when α varies in a range [0.6, 0.9] for the method BGER.

Similarly we set α to 0.8 and range β from 0 to 1, an increase of 0.1 each. Fig. 8 illustrates the results. From Fig. 8 we find that the performance is smooth when β varies in a range [0, 0.9] for the method BGER.

VI. Conclusion and Future Work

Related entity finding is a very promising application. In this paper, we represent bipartite graph based method for the relate entity ranking, where we leverage the Co-List relation

to boost those related but unpopular entities. Given a query, we first estimate the initial relevance scores for the candidate entities. We then construct a bipartite graph based on the Co-List relation between the entities, and perform refinement to propagate the scores over entities. Finally all the candidate entities are ranked according to their refined ranking score. Experimental results demonstrate the effectiveness of our method.

In future work, we will investigate other methods for named entity recognition, relation recognition and extraction of instances of Co-List relation. We will also try to extract the target entities from the instances of Co-List relation with high scores to improve the recall.

ACKNOWLEDGMENT

This research work was funded by the National High-tech R&D Program of China under grant No. 2010AA012500, the National Natural Science Foundation of China under Grant No. 61003166 and Grant No. 60933005.

REFERENCES

[1] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld, "Overview of the trec 2009 entity track," in *TREC*, 2009.

[2] N. Craswell, D. Hawking, R. Wilkinson, and M. Wu, "Overview of the trec 2003 web track," in *TREC*, 2003, pp. 78–92.

[3] Y. Fang, L. Si, Z. Yu, Y. Xian, and Y. Xu, "Entity retrieval with hierarchical relevance model,exploiting the structure of tables and learning homepage classiers," in *TREC*, 2009.

[4] R. McCreadie, C. Macdonald, I. Ounis, J. Peng, and R. L. Santos, "University of glasgow at trec 2009: Experiments with terrier," in *TREC*, 2009.

[5] H. Zhai, X. Cheng, J. Guo, H. Xu, and Y. Liu, "A novel framework for related entities finding: Ictnet at trec 2009 entity track," in *TREC*, 2009.

[6] M. Bron, K. Balog, and M. de Rijke, "Ranking related entities: components and analyses," in *CIKM*, 2010, pp. 1079–1088.

[7] E. M. Voorhees, "The trec-8 question answering track report," in *TREC*, 1999.

[8] E. M. Voorhees, "Overview of the trec 2003 question answering track," in *TREC*, 2003, pp. 54–68.

[9] N. Craswell, A. P. de Vries, and I. Soboroff, "Overview of the trec 2005 enterprise track," in *TREC*, 2005.

[10] H. Fang and C. Zhai, "Probabilistic models for expert finding," in *ECIR*, 2007, pp. 418–430.

[11] Y. Cao, J. Liu, S. Bao, and H. Li, "Research on expert search at enterprise track of trec 2005," in *TREC*, 2005.

[12] A. P. de Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas, "Overview of the inex 2007 entity ranking track," in *INEX*, 2007, pp. 245–251.

[13] K. Balog, M. Bron, and M. de Rijke, "Category-based query modeling for entity search," in *ECIR*, 2010, pp. 319–331.

[14] A.-M. Vercoustre, J. Pehcevski, and J. A. Thom, "Using wikipedia categories and links in entity ranking," in *INEX*, 2007, pp. 321–335.

[15] R. Kaptein, P. Serdyukov, A. P. de Vries, and J. Kamps, "Entity ranking using wikipedia as a pivot," in *CIKM*, 2010, pp. 69–78.

[16] Q. Yang, P. Jiang, C. Zhang, and Z. Niu, "Reconstruct logical hierarchical sitemap for related entity finding," in *TREC*, 2010.

[17] Z. Wang, C. Tang, X. Sun, H. Ouyang, R. Lan, W. Xu, G. Chen, and J. Guo, "Pris at trec 2010: Related entity finding task of entity track," in *TREC*, 2010.

[18] X. Rui, M. Li, Z. Li, W.-Y. Ma, and N. Yu, "Bipartite graph reinforcement model for web image annotation," in *ACM Multimedia*, 2007, pp. 585–594.

[19] H. Deng, M. R. Lyu, and I. King, "A generalized co-hits algorithm and its application to bipartite graphs," in *KDD*, 2009, pp. 239–248.

[20] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," in *PNAS*, 2010, pp. 4511–4515.

[21] M. J. Cafarella, A. Y. Halevy, Y. Zhang, D. Z. Wang, and E. W. 0002, "Uncovering the relational web," in *WebDB*, 2008.

[22] Y. Wang and J. Hu, "A machine learning based approach for table detection on the web," in *WWW*, 2002, pp. 242–250.