CSIR at INEX 2008 Entity-Ranking Task: Entity Retrieval and Entity Relation Search in Wikipedia

Jiepu Jiang[1], Wei Lu[1], Xianqian Rong[1], Yangyan Gao[1]

[1] Center for Studies of Information Resources (CSIR),
School of Information Management, Wuhan University, P. R. China
{jiepu.jiang, reedwhu, rongxianqian, gaoyangyan2008}@gmail.com

Abstract. In this paper, we mainly describe our methods taken in INEX 2008 entity-ranking task. Firstly, we discuss how to extend current expert retrieval models to deal with the category field in INEX entity ranking queries. Then, on such basis, we propose similar solutions to the former list completion task and the entity relation search task.

Keywords: Entity retrieval, entity ranking, language model.

1 Introduction

In recent years, expert retrieval has been focused and widely discussed in various aspects. A few ranking models are created for this issue and have been proved to be effective. In contrast, the issue of entity retrieval as a general case is less discovered until the INEX entity-ranking task in 2007. This task has introduced a perspective of entity retrieval beyond expert retrieval, which involves not only more types of entities in practice, but also various tasks.

One of the main foci in INEX entity-ranking task is to return entities that satisfy a topic with multiple fields (the INEX XER task). Typical topic fields consist of a text query field, which describes the entity search query using natural language, and also a category field specifying possible categories of relevant entities in Wikipedia and an example field which contains a small and incomplete list of relevant entities. Two mandatory runs are required for the INEX XER task: one should use the text field and the category field, while the other should make use of the text field and the example entity list (the former list completion task).

The main difference between the INEX XER task and an expert retrieval task (such as the TREC expert search task) resides in the extra demands for match in entity categories for the former. In the expert search task, we can focus in topical demands of queries since the entity type is pre-defined. As a result, expert retrieval models are aiming at finding topical relevant entities. However, in the INEX XER task, we are facing candidate entities of various categories and the users' demands for category are adhoc instead of pre-defined. Generally, current expert retrieval models are incapable of dealing with such extra demands for category. As a result, we extend current expert search models to incorporate such extra demands for category.

On such basis, the entity relation search task (ERS) task is discussed. The ERS task is a new task in INEX. Different from the XER task, the ERS task aims at finding right entity pairs with some specified relationship. We adopt a 3-step method in this task: firstly, topical relevant entities are retrieved using our proposed extensive model; then, we find for each relevant entity a list of entities have the specified relationship; in the end, those found entity pairs are re-ranked all together.

The remainder of this paper is organized as follows: section 2 reviews models on expert search and INEX entity ranking; in section 3, we describe models of the XER task and the ERS task; section 4 draws a conclusion. In the pre-proceeding session of INEX, evaluation of models is not available and thus is not included in this paper.

2 Related Works

The problem of expert retrieval (finding relevant experts of a specific topic) can be considered as the issue of retrieving entities of some pre-defined category, i.e. people. Early approaches of expert retrieval usually involve empirical methods and structured or semi-structured resources. Since the TREC 2005 expert search task, expert retrieval is focused by IR researchers and a lot of models are created. Specifically, language modeling methods for expert search are proved to be highly effective and thus are widely adopted.

Generally, language models for expert retrieval rank experts by the probability that the query is generated by the experts. In TREC 2005, Cao et al. [1] and Azzopardi et al. [2] introduce two kinds of language models for expert retrieval. These methods are later explained by Balog et al. [3] as candidate model (model 1) and document model (model 2). Fang et al. [4] also described a similar framework, but they had explicitly modeled on relevance and used the probability ranking principle to rank.

Further, some detailed problems are studied under the framework. Serdyukov et al. [5] introduced a method to enhance performance by query modeling. Balog et al. [6] elaborated the estimation method of candidate-document association. Serdyukov et al. [7] explored the relevance propagation in expert search. Petkova et al. [8] explored the dependence between candidate and terms using proximity-based methods. Most recently, Balog et al. [9] fully considered non-local information available in the collection and significantly improved the performance.

In contrast, only limited researchers studied the entity retrieval as a general case. Most of them noticed the problem of categories and proposed feasible solutions. For example, Vercoustre et al. [10] used a Jaccard coefficient between query category set and entity category set for ranking. Demartini et al. [11] expanded the category query field using YAGO, while Jamsen et al. [12] achieved category expansion using Wikipedia hierarchies. Zhu et al. [13] treats entities' categories as a metadata field and searches entities between multi-fields.

In this paper, we will extend current expert retrieval models to deal with the entity retrieval problem under the environment of the INEX entity-ranking task.

3 Models

In this section, we will describe our language modeling process for entity retrieval. A lot of language models for expert retrieval have already been explored and testified to be effective, which constitute an important part of our models. For the XER task, we mainly discuss the problem under the settings of two mandatory runs. For the ERS task, we propose models using both the mandatory fields and some extra features.

3.1 Entity Ranking Task

The query of entity ranking can be multi-fields. In this paper, we mainly discuss the queries of two mandatory runs mentioned in section 1. For the first run, only the text query and the category field can be used. So, we can represent the query as $q(q_{text}, q_{cat})$, where q_{text} refers to the text query, and q_{cat} represent the category query. The second run is studied in the following subsection as the former list completion task.

As a result, from a language model perspective, the problem of entity retrieval can be translated as: given a query q, to find the possible relevant entity e, i.e. to estimate $p(e|q)$ for each entity and accordingly rank entities. Then, $q(e|q)$ can be transformed to (1). Assuming each entity shares the equal probability, $q(e|q)$ is proportional to $p(q|e)$. Thus, we can use $p(q|e)$ to rank entities.

$$p(e \mid q) = \frac{p(q \mid e) * p(e)}{p(q)} \tag{1}$$

Considering q consists of two parts (q_{text} and q_{cat}), we can simply assume them to be independent and transform $p(q|e)$ as (2). Then, we can treat $p(q_{text}|e)$ and $p(q_{cat}|e)$ separately.

$$p(q \mid e) = p(q_{text}, q_{cat} \mid e) = p(q_{text} \mid e) * p(q_{cat} \mid e) \tag{2}$$

For the $p(q_{text}|e)$, we adopt the expert search model 2 [3] to estimate it. Assuming e is independent for the generation of q_{text} in d, we can also use (4) as a simplification. Then, we estimate $p(q_{text}|d)$ using text retrieval models as in (5). In (5), each query terms in q_{text} is assumed to be independent, $p_{ml}(t|d)$ is the max likelihood estimation of t in d, $p_c(t)$ is the probability of t in the whole corpus, λ is a parameter which is set to 0.5 in our runs.

$$p(q_{text} \mid e) = \sum_{d \in D} p(q_{text} \mid d, e) * p(d \mid e) \tag{3}$$

$$p(q_{text} \mid e) = \sum_{d \in D} p(q_{text} \mid d) * p(d \mid e) \tag{4}$$

$$p(q_{text} \mid d) = \prod_{t \in q_{text}} p(t \mid d) = \prod_{t \in q_{text}} \left\{ (1-\lambda) * p_{ml}(t \mid d) + \lambda * p_c(t) \right\} \tag{5}$$

For the $p(d|e)$, we also adopt a general method, i.e. (6). In (6), D' refers to a subset of documents that are associated with e, and $a(d_i,e)$ is the association between d_i and e. In our model, $a(d_i,e)$ is simply measured as the frequency of e in d_i. For elaborated methods, Balog et al. [6] had a thorough exploration.

$$p(d \mid e) = \frac{a(d,e)}{\sum_{d_i \in D'} a(d_i,e)} \tag{6}$$

As for $p(q_{cat}|e)$, q_{cat} can be treated as a category set, where every single element category is generated independently from the entity's labeled category set CAT_e. Then, we can estimate $p(q_{cat}|e)$ as (7):

$$p(q_{cat} \mid e) = \prod_{cat_i \in q_{cat}} p(cat_i \mid CAT_e) \tag{7}$$

$$p(q_{cat} \mid e) = \left(\prod_{cat_i \in q_{cat}} p(cat_i \mid CAT_e) \right) * \left(\prod_{cat_j \notin q_{cat}} \left\{ 1 - p(cat_j \mid CAT_e) \right\} \right) \tag{8}$$

It should be noted that in (7) we adopt the q_{cat} as a sequence of categories, though it seems more reasonable to treat it as a set and estimate it in (8). But we choose (7) here for the following reasons: on the one hand, the category queries are not ensured to be accurate, thus it is also arguable to model other categories as not generated by CAT_e; on the other hand, a thorough estimation of a large amount of unseen categories in (8) consumes much computational resources, which is not efficient.

For (7), we estimate each $p(cat_i|CAT_e)$ using a maximum likelihood estimation with a smooth. Then $p(cat_i|CAT_e)$ can be further divided into each categories in CAT_e, which is presented in (9).

$$p(cat_i \mid CAT_e) = (1-\lambda) * \sum_{cat_j \in CAT_e} \frac{p(cat_i \mid cat_j)}{|CAT_e|} + \lambda * p(cat_i) \tag{9}$$

In (9), $p(cat_i)$ is the corpus probability of cat_i, which is used for smoothing. Then we discuss several possible circumstances for $p(cat_i|cat_j)$ in a rule-based case: when cat_i is exactly cat_j, or cat_i is cat_j's parent category, we set $p(cat_i|cat_j)$ to 1; when cat_i is cat_j's child category, we set $p(cat_i|cat_j)$ to $1/|cat_j|$, where $|cat_j|$ is the size of the cat_j's child category set; for other circumstances, $p(cat_i|cat_j)$ is set to 0.

3.2 List Completion Task

Another mandatory run of the XER task is formerly known as the list completion task, which provides a list of example entities instead of the category field in retrieval. So, we can represent the query as $q(q_{text}, q_{lc})$, in which q_{text} refers to the text query and q_{lc} represents the list of example entities. We can rank each entity e according to $p(q|e)$, which can also be transformed to (10) if we assume the independence between q_{text} and q_{lc}.

$$p(q \mid e) = p(q_{text}, q_{lc} \mid e) = p(q_{text} \mid e) * p(q_{lc} \mid e) \quad (10)$$

In (10), we can also estimate $p(q_{text}|e)$ in (3). Then, the problem resides in $p(q_{lc}|e)$, which can be transformed to (11). In (11), cat_i refers to any category labeled with entities in the list *lc*. Then, $p(cat_i|CAT_e)$ can be estimated in (9).

$$p(q_{lc} \mid e) = \prod_{cat_i \in lc} p(cat_i \mid CAT_e) \quad (11)$$

In INEX 2008, we submit two mandatory runs for the XER task:

(1) Run *1_CSIR_ER_TC_mandatoryRun*. This run uses methods described in 3.1, which makes use of the title field and the category field.

(2) Run *1_CSIR_LC_TE_mandatoryRun*. This run uses methods described in 3.2, which makes uses of the title field and the example list.

3.3 Entity Relation Search

The entity relation search (ERS) task is a new task for entity ranking, which aims at finding entity pairs with appropriate relation. For this task, the problem can be solved in three steps: firstly, searching for a list of entities relevant to the topic, as stated in 3.1 and 3.2; then, for each relevant entity *e* retrieved, finding a group of target entities that have the specified relation with *e*; in the end, re-rank entity pairs all together.

As a result, the main problem is: given entity *e*, to find a list of target entities that have relation q_r with e_0 and match category query q_{cat}. So, queries can be represented as a multi-field query $q(e_0, q_r, q_{cat})$, in which e_0 is the entity that retrieved entities are associated with, q_r represents the relation text query, q_{cat} represents the category of the target entities. Then, we can rank each entity according to $p(q|e)$, which can also be transformed to (12) if we assume independence between any two parts of *q*.

$$p(q \mid e) = p(e_0, q_r, q_{cat} \mid e) = p(e_0 \mid e) * p(q_r \mid e) * p(q_{cat} \mid e) \quad (12)$$

Then, we can estimate $p(q_r|e)$ as (3) and $p(q_{cat}|e)$ as (7). As for the $p(e_0|e)$, we can also adopt models similar to expert search models in estimation. By treating e_0 as a term, we can also use (3) to estimate it.

In INEX 2008, we submit two runs for the ERS task:

(1) Run *1_CSIR_ERS_TC_R_TC_ERSWITHCATE*. This run uses methods in (12), which makes use of the title field and the category field to find each e_0, and use relation title and target entity categories to find each target entities.

(2) Run *1_CSIR_ERS_TC_R_T_mandatoryRun*. Compared with (12), this run only makes use of the relation title when finding target entities for each e_0.

4 Conclusion

In this paper, we describe our methods of entity ranking and entity relation search in the INEX 2008 entity-ranking task. Some simple methods are discussed to understand the difference between expert search and entity retrieval, i.e. to deal with the demand

for category in entity retrieval task. Refinements of models are left as future works, which includes the independence assumptions and estimation methods. Besides, we plan to future discover methods of dealing with categorical demands under some more relaxed settings, for example, to use only the text query field. The estimation of models takes advantages of Wikipedia features, which is also one of the limitations and needs to be relaxed.

References

1. Y. Cao, J. Liu, S. Bao, H. Li. Research on Expert Search at Enterprise Track of TREC 2005. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), 2005.
2. L. Azzopardi, K. Balog, M. de Rijke. Language Modeling Approaches for Enterprise Tasks. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), 2005.
3. K. Balog, L. Azzopardi, M. de Rijke. Formal Models for Expert Finding in Enterprise Corpora. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, 2006: 43-50.
4. H. Fang, C. Zhai. Probabilistic Models for Expert Finding. In Proceedings of the 29th annual European Conference on Information Retrieval Research, Rome, Italy, 2007: 418-430.
5. P. Serdyukov, Sergey Chernov, Wolfgang Nejdl. Enhancing Expert Search through Query Modeling. In Proceedings of the 29th annual European Conference on Information Retrieval Research, Rome, Italy, 2007: 737-740.
6. K. Balog, M. de Rijke. Associating People and Documents. In Proceedings of the 30th annual European Conference on Information Retrieval Research, Glasgow, Scotland, 2008: 296-308.
7. P. Serdyukov, Henning Rode, Djoerd Hiemstra. University of Twente at the TREC 2007 Enterprise Track: Modeling relevance propagation for the expert search task. In Proceedings of the 16th Text REtrieval Conference (TREC 2007), 2007.
8. D. Petkova, W. B. Croft. Proximity-Based Document Representation for Named Entity Retrieval. In Proceedings of the 16th ACM conference on Conference on information and knowledge management, Lisbon, Portugal, 2007: 731-740.
9. K. Balog, M. de Rijke. Non-Local Evidence for Expert Finding. In Proceedings of the 17th ACM conference on Conference on information and knowledge management, Napa Valley, California, USA, 2008: 731-740.
10. A. Vercoustre, J. A. Thom, J. Pehcevski. Entity Ranking in Wikipedia. In Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, 2008.
11. G. Demartini, C. Firan, T. Iofciu. L3S Research Center at the INEX Entity Ranking Track. In Proceedings of the INEX 2007, 2007.
12. J. Jamsen, T. Nappila, P. Arvola. Experiments on Category Expansion at INEX 2007. In Proceedings of the INEX 2007, 2007.
13. J. Zhu, D. Song, S. Ruger. Integrating Document Features for Entity Ranking. In Proceedings of the INEX 2007, 2007.