Adapting Language Modeling Methods for Expert Search to Rank Wikipedia Entities

Jiepu Jiang, Wei Lu, Xianqian Rong, and Yangyan Gao

Center for Studies of Information Resources,
School of Information Management, Wuhan University, China
{jiepu.jiang,reedwhu,rongxianqian,gaoyangyan2008}@gmail.com

Abstract. In this paper, we propose two methods to adapt language modeling methods for expert search to the INEX entity ranking task. In our experiments, we notice that language modeling methods for expert search, if directly applied to the INEX entity ranking task, cannot effectively distinguish entity types. Thus, our proposed methods aim at resolving this problem. First, we propose a method to take into account the INEX category query field. Second, we use an interpolation of two language models to rank entities, which can solely work on the text query. Our experiments indicate that both methods can effectively adapt language modeling methods for expert search to the INEX entity ranking task.

Keywords: entity retrieval, entity ranking, language model, expert search.

1 Introduction

In this paper, we focus on how to adapt language modeling methods for expert search to the INEX entity ranking task (XER), which aims at finding a list of relevant entities according to a search query. A typical search query may involve several fields:

1. title: a text query field that describes the user's search needs;
2. category: a structural field specifying Wikipedia categories of relevant entities.

For example, a typical INEX XER search query can be:

<title>songs of Bob Dylan</title>
<categories>
<category id="40340">bob dylan songs</category>
</ categories >

The XER task shares a lot of similarities with the TREC expert search task, which can be considered as a special entity ranking task for persons only. Both tasks face the challenge of finding and utilizing descriptive information of entities in the documents. As a result, it is reasonable to adopt methods for expert search in the XER task.

Language modeling methods have been widely adopted in the expert search task. We have applied two widely used language modeling methods for expert search (i.e. model 1 and model 2 [1]) to the XER task. However, our experiments indicated that both methods cannot effectively distinguish entity types. As a result, we mainly focus on resolving this limitation.

S. Geva, J. Kamps, and A. Trotman (Eds.): INEX 2008, LNCS 5631, pp. 264–272, 2009.

First, we propose a method to take into account the INEX category query field, which can be applied to both model 1 and model 2. Second, we interpolate the entity model in model 1 with an entity category model, which solely works on the text query. In our experiments, it is indicated that both methods can effectively distinguish entity types. The first method was also adopted in our participation in INEX 2008. But our experiments indicate that the second method is much more effective.

Although the INEX entity-ranking track involves two tasks, i.e. the entity ranking task (XER) and the entity relation search task (ERS), we only discuss the XER task here due to the lack of evaluation for the ERS task. For our methods taken for the list completion task and ERS task, please refer to the pre-proceedings.

The remainder of this paper is organized as follows: section 2 reviews on language modeling methods for expert search and methods adopted in the INEX entity ranking task; in section 3, we describe our methods; section 4 evaluates the proposed methods; in section 5, we draw a conclusion.

2 Related Works

Language modeling methods are widely adopted for the expert search task. The most widely used language modeling framework for expert search was defined by Balog et al. [1] as model 1 and model 2. Further, refinements were made from various aspects. Petkova et al. [2] considered the dependency between candidates and terms. Balog et al. [3] elaborated candidate-document association. Serdyukov et al. [4] explored the relevance propagation. Balog et al. [5] used non-local information in the collection. For a complete review, please refer to [6].

Compared with expert search, less attention has been paid to the task of searching general entities of various types. In 2007, INEX provided the first collection for entity ranking, which is based on Wikipedia and involves a lot of useful features for entity ranking: entities are manually labeled with categories; the hierarchy of categories is given; entity occurrences are partly labeled in the documents.

Most of the methods adopted in INEX rely on the INEX category query field and Wikipedia category labels to distinguish entity types. Vercoustre et al. [7] used a set-based measure to calculate similarity between the INEX category query and the entity Wikipedia categories. Demartini et al. [8] expanded the category set using YAGO to improve the matching of entity types. Tsikrika et al. [9] adopted expert search model in [4] for entity ranking, and expanded category matching with child categories.

In section 3, we propose two methods to adapt language modeling methods for expert search to the INEX entity ranking task.

3 Models

In this section, we describe our methods. First, we propose a method to take into account the INEX category query field in both model 1 and model 2. Second, we interpolate the entity model estimated in model 1 with a category model, which can help model 1 better understand category query terms in the text query.

3.1 Language Modeling Methods for Expert Search

In section 3.1, we briefly describe two frequently used language modeling methods for expert search, i.e. model 1 and model 2 [1]. Both methods rank entities (experts) by $p(e|q)$, and use co-occurrence information of entities to estimate the probability. Assuming the same prior probability for each entity e, we can rank entities by $p(q|e)$.

For model 1, an entity model θ_e is inferred for each entity e. We can estimate $p(q|e)$ as Eq.(1):

$$p(q \mid e) = p(q \mid \theta_e) = \prod_{t \in q} p(t \mid \theta_e)^{tf(t,q)} \tag{1}$$

In Eq.(1), $tf(t,q)$ is the frequency of t in the query q. Further, θ_e can be inferred using co-occurrence information of e in the collection.

For model 2, the estimation of $p(q|e)$ is divided into each sub event space of d:

$$p(q \mid e) = \sum_{d} p(q \mid d, e) \times p(d \mid e) \tag{2}$$

Since there have been a lot of discussions on model 1 and model 2, we do not go further here. Please refer to [1] for details.

3.2 Considering the INEX Category Query Field

In section 3.2, we propose a method to consider the INEX category query field, which can be applied to both model 1 and model 2. We can represent the whole query as Q, which contains two parts: the text query q and the INEX category query q_{cat}. Then, we rank entities by $p(Q|e)$, which can be transformed as Eq. (3):

$$p(Q \mid e) = p(q, q_{cat} \mid e) = p(q \mid e) \times p(q_{cat} \mid e, q) \tag{3}$$

Assuming q and q_{cat} are independent, $p(q_{cat}|e,q)$ can be simplified to $p(q_{cat}|e)$:

$$p(Q \mid e) = p(q, q_{cat} \mid e) = p(q \mid e) \times p(q_{cat} \mid e) \tag{4}$$

In (4), $p(q|e)$ can be estimated using model 1 or model 2. As a result, the rest of the task is to estimate $p(q_{cat}|e)$.

In the INEX Wikipedia collection, entities are labeled with a list of categories. As a result, we can represent e's labeled categories as a category set, i.e. $CAT_e\{cat_i\}$. Also, we can represent the INEX category query field as a category set, i.e. $CAT_q\{cat_j\}$. Further, assuming that cat_j in CAT_q is generated independently, we estimate $p(q_{cat}|e)$ in Eq. (5):

$$p(q_{cat} \mid e) = p(CAT_q \mid CAT_e) = \prod_{cat_j \in CAT_q} p(cat_j \mid CAT_e) \tag{5}$$

It should be noted that in (5) we adopt q_{cat} as a sequence of categories, although it is a set and may be more reasonable to be estimated in Eq. (6):

$$p(q_{cat} \mid e) = \left(\prod_{cat_j \in CAT_q} p(cat_j \mid CAT_e) \right) \times \left(\prod_{cat_j \notin CAT_q} \left\{ 1 - p(cat_j \mid CAT_e) \right\} \right) \quad (6)$$

Here, we adopt Eq.(5) for the following considerations: on the one hand, it is controversial to model categories that do not exist in CAT_q, since the category query field are not ensured to be accurate, and Wikipedia labels are also not completely accurate; on the other hand, a thorough estimation involving a large amount of unseen categories in (6) will consume a lot of computational resources.

In (5), $p(cat_j|CAT_e)$ is estimated using a maximum likelihood estimation with a Jelinek Mercer smoothing. Then, $p(cat_j|CAT_e)$ can be further considered using each cat_i in CAT_e:

$$p(cat_j \mid CAT_e) = (1-\lambda_1) \times \sum_{cat_i \in CAT_e} \frac{p(cat_j \mid cat_i)}{|CAT_e|} + \lambda_1 \times p(cat_j) \quad (7)$$

In (7), $p(cat_j)$ is the probability of cat_j in the collection, which is estimated in (8). In Eq. (8), $ct(cat_j)$ is the number of entities in the collection that are labeled with cat_j.

$$p(cat_j) = \frac{ct(cat_j)}{\sum_{cat_i} ct(cat_i)} \quad (8)$$

For $p(cat_j|cat_i)$, we estimate it using some rule-based methods:

1. If $cat_j = cat_i$, or cat_j is cat_i's parent category, we set $p(cat_j|cat_i)$ to 1;
2. If cat_j is cat_i's child category, we set $p(cat_j|cat_i)$ to $1/|cat_i|$ ($|cat_i|$ is the number of child categories of cat_i);
3. For other circumstances, $p(cat_j|cat_i)$ is set to 0.

Using this method, we provide a solution to consider the category query field into current language modeling methods for expert search, which can help expert search models better distinguish entity categories. This method can be applied to both model 1 and model 2.

3.3 Understanding Category Terms in Search Query

In section 3.3, we use an interpolation of an entity model and a category model to understand category terms in the text query.

After manually checking all the search queries in INEX 07 and 08, we come to the following conclusion: text query for the INEX entity ranking task consist of two kinds of terms, i.e. ***topic terms*** and ***category terms***.

We define topic terms as terms describing topical information of relevant entities, while category terms are used to specify categories of relevant entities. For example, for the query "songs of Bob Dylan", relevant entities are topically relevant with "Bob Dylan", and should be songs. So, "Bob" and "Dylan" are topic terms, while "songs" is a category term. Among the 95 queries in INEX 07 and 08, only 3 queries (Topic 50, 52 and 105) do not conform to our conclusion of entity ranking queries.

In contrast, queries for expert search only consist of topic terms. For example, the user will propose the query "wheel motor" to search for experts related to the topic "wheel motor". Category terms are omitted in expert search, since it is unnecessary to distinguish categories in the expert search task.

The difference between expert search queries and entity ranking queries is essential in explaining why language modeling methods for expert search are not effective in distinguishing entity categories. In language modeling methods for expert search, we infer entity (expert) models using co-occurrence information of entities. Although the entity models inferred are effective for expert search, considering that expert search queries only consist of topic terms, the entity models inferred may only indicate an approximation of probability distribution for topic terms. Thus, it is not surprising that expert search models are not very effective in understanding the category information need in the text query.

Thus, we infer two models for each entity e: T_e is the distribution model for topic terms, and C_e is the distribution model for category terms. Then, we can estimate $p(t|e)$ using an interpolation between T_e and C_e:

$$p(t \mid e) = \lambda_2 \times p(t \mid T_e) + (1-\lambda_2) \times p(t \mid C_e) \tag{9}$$

In Eq.(9), λ_2 is a prior probability that a term will be generated from T_e. Though it is more reasonable to set different λ_2 for different entities, we adopt a constant value for λ_2 as a simplification. T_e can be inferred using model 1. In the INEX Wikipedia collection, we can infer C_e using labeled categories of the entities.

For each entity e, we represent its labeled categories in the Wikipedia as a category set CAT_e $\{cat_i\}$, in which cat_i is each labeled category in the set CAT_e. Then, we can use CAT_e to estimate C_e, which can be further considered using each cat_i in CAT_e:

$$p(t \mid CAT_e) = \sum_{cat_i \in CAT_e} p(t \mid cat_i, CAT_e) \times p(cat_i \mid CAT_e) \tag{10}$$

Assuming that the generation of t from cat_i is independent with CAT_e, $p(t|cat_i,CAT_e)$ can be simplified to $p(t|cat_i)$:

$$p(t \mid CAT_e) = \sum_{cat_i \in CAT_e} p(t \mid cat_i) \times p(cat_i \mid CAT_e) \tag{11}$$

For $p(t|cat_i)$, we simply estimate it using category name of cat_i by a maximum likelihood estimate in Eq.(12).

$$p(t \mid cat) \approx p_{mle}(t \mid cat) = \frac{tf(t)}{\sum_{t_i \in cat} tf(t_i)} \tag{12}$$

For $p(cat_i|CAT_e)$, we assign all categories with the equal weight and estimate it as $1/|CAT_e|$, where $|CAT_e|$ is the number of categories in the category set CAT_e. In the end, we can represent $p(t|C_e)$ as Eq.(13):

$$p(t \mid C_e) = \frac{\sum_{cat_i \in CAT_e} p_{mle}(t \mid cat_i)}{|CAT_e|} \tag{13}$$

Compared with the former method, this method can solely work on the text search query, which can resolve the limitation of using the INEX category query field. But a main limitation for this method is that C_e is estimated based on Wikipedia category labels. This limitation is left as a future work.

4 Evaluation

4.1 Experiment Settings

In our experiments, we adopt the INEX Wikipedia collection to evaluate our methods. Both INEX 2007 and 2008 queries are used. The INEX 2007 queries can be divided into two groups: one group consists of queries generated from the INEX ad hoc task (INEX 07 adhoc), and the other group consists of the genuine INEX 2007 XER query (INEX 07 xer).

The INEX Wikipedia collection is a subset of Wikipedia, which contains lots of semantic information. In this collection, entity occurrences are partly labeled in the documents. Thus, we do not further recognize named entities. Besides, each entity is also labeled with some categories. Category hierarchies are given. In our experiments, we have found 659,388 entities labeled with 75,601 Wikipedia categories (113,483 categories are provided in total, but some of them are not labeled with any entity).

In the pre-processing stage, we remove XML tags. The indexing process removes common stop words. Words are stemmed using Porter-Stemming algorithm.

Though official results in INEX 08 are evaluated using xinfAP, we use MAP as the main evaluation measure in order to be consistent with INEX 2007 (we do not have any method to evaluate xinfAP results for INEX 07 queries). The evaluation tool is trec_eval.

4.2 Expert Search Models

In section 4.2, we will evaluate the effectiveness of expert search models in the entity ranking task. In our experiments, we try to apply model 1 and model 2 to the INEX entity ranking task. Please refer to [1] for details about these models. In both models, we set the smoothing parameter λ to 0.5.

Table 1 shows evaluation results for model 1 and model 2 in the INEX 07 and 08 query sets, which are our baseline runs. It is indicated that both model 1 and model 2 are not very effective in the INEX entity ranking task. Besides, although previous researches indicated that model 2 is more effective than model 1 in the expert search task, model 1 apparently outperforms model 2 in all query sets of INEX.

Table 1. Evaluation results for model 1 and model 2 in the INEX entity ranking task

Query Set	Model 1		Model 2	
	MAP	xinfAP	MAP	xinfAP
INEX07	**0.2059**	--	0.1635	--
INEX07 adhoc	**0.2588**	--	0.1783	--
INEX07 xer	**0.1614**	--	0.1511	--
INEX08	**0.1189**	**0.1189**	0.0885	0.0885

4.3 Considering the INEX Category Query Field

In section 4.3, we evaluate the effectiveness of the method proposed in section 3.2, which considers the INEX category query fields into expert search language models. For a simplification, we set λ_1 in Eq.(7) to 0.5. For efficiency consideration, we only re-rank the top 500 entities returned by model 1 and model 2 when using the method proposed in section 3.2.

Table 2. MAP results for the method that considers the INEX category query field

Query Set	Mode 1	Model 1 + Method in 3.2		Mode 2	Model 2 + Method in 3.2	
INEX07	0.2059	**0.2522**	(+ 22.49%)	0.1635	0.2167	(+ 32.54%)
INEX07 adhoc	0.2588	**0.3374**	(+ 30.37%)	0.1783	0.2776	(+ 55.69%)
INEX07 xer	0.1614	**0.1806**	(+ 11.90%)	0.1511	0.1656	(+ 09.60%)
INEX08	0.1189	**0.2106**	(+ 77.12%)	0.0885	0.1627	(+ 83.84%)

Table 2 shows evaluation results for the method that considers the INEX category query field. It is indicated that, for both model 1 and model 2, this method can greatly enhance the effectiveness. In INEX 08, we adopted a combination of this method and model 1[1].

4.4 Considering Category Terms in Search Query

In section 4.4, we further consider category query terms into expert search model 1. For a simplification, the parameter λ_2 in (9) is set to 0.5. In our experiments, we try to investigate the following problem:

1. Can the adaptation method proposed in 3.3 help expert search model?
2. Compared with the method in 3.2, can the method using only text query be more effective?

Table 3 shows evaluation results for the method proposed in section 3.3 (the INEX category query field is not used). In Table 3, it is indicated that the method proposed in section 3.3 can also greatly enhance the effectiveness. Besides, compared with the

[1] Due to a coding error, our officially submitted run 1_CSIR_ER_TC_mandatoryRun had used a measure of $p(cat_j|CAT_e)$ different from the method proposed in 3.2. But we mean to use the method in 3.2. Results in Table.2 strictly conform to the method in section 3.2.

Table 3. MAP results for the method proposed in section 3.3

Query Set	Model 1	Model 1 + Method in 3.3	Model 1 + Method in 3.2
INEX07	0.2059	**0.2952** (+ 43.37%)	0.2522 (– 14.57%)
INEX07 adhoc	0.2588	**0.3585** (+ 38.53%)	0.3374 (– 05.89%)
INEX07 xer	0.1614	**0.2420** (+ 49.94%)	0.1806 (– 25.37%)
INEX08	0.1189	**0.2942** (+147.43%)	0.2106 (– 28.42%)

method that considers the INEX category query field (in section 3.2), the interpolation of two models is evidently more effective in all query sets.

It may indicate some problems of using the INEX category query field in the entity ranking task. First, for a large collection containing a huge amount of entity categories (such as the INEX Wikipedia collection), it is difficult and impractical for the user to specify precisely all possible categories of relevant entities. Thus, when the user fails to select out some possible categories for relevant entities, some relevant entities will be excluded. Second, since the categories of relevant entities are specified in the text query, it is also unnecessary to learn it using the structural category query field.

Further, we combine both methods into model 1. Table 4 shows evaluation results of considering both methods into model 1, which means to estimate $p(q|e)$ in Eq.(3) using Eq.(8). However, in the experiments, it is indicated that the combination of two methods is not ensured to receive better effectiveness than using the method proposed in 3.3 only. This problem is left as a future work for us to discover.

In table 5, we gives out xinfAP for each method in INEX 08 query set.

Table 4. MAP results of considering both methods into model 1

Query Set	Model 1	Model 1 + Method in 3.2	Model 1 + Method in 3.3	Model 1 + Method in 3.2 & 3.3
INEX07	0.2059	0.2522	**0.2952**	0.2838
INEX07 adhoc	0.2588	0.3374	0.3585	**0.3755**
INEX07 xer	0.1614	0.1806	**0.2420**	0.2067
INEX08	0.1189	0.2106	0.2942	**0.3042**

Table 5. xinfAP results

Query Set	Model 1 + Method in 3.2	Model 2 + Method in 3.2	Model 1 + Method in 3.3	Model 1 + Method in 3.2 3.3
INEX08	0.2106	0.1627	0.2942	**0.3042**

5 Conclusion

In this paper, we describe two methods to adapt language modeling methods for the expert search task to the INEX entity ranking task. First, we propose a method to take into account the INEX category query field, which can be applied to both model 1 and

model 2. Second, we use an interpolation between the entity model and the category model to understand category terms in the text query.

In our experiments, it is indicated that both methods can effectively adapt language modeling methods for expert search to the INEX entity ranking task. Compared with the method that considers the INEX category query field, the method using category terms (section 3.3) is more effective. However, a combination of both methods is not ensured to further enhance the effectiveness.

References

1. Balog, K., Azzopardi, L., de Rijke, M.: Formal Models for Expert Finding in Enterprise Corpora. In: Proceeding of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006), Seattle, Washington, USA, pp. 43–50 (2006)
2. Petkova, D., Croft, W.B.: Proximity-Based Document Representation for Named Entity Retrieval. In: Proceedings of the 16th ACM conference on information and knowledge management (CIKM 2007), Lisbon, Portugal, pp. 731–740 (2007)
3. Balog, K., de Rijke, M.: Associating People and Documents. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 296–308. Springer, Heidelberg (2008)
4. Serdyukov, P., Rode, H., Hiemstra, D.: Modeling multi-step relevance propagation for expert finding. In: Proceedings of 17th ACM conference on Information and knowledge management (CIKM 2008), Napa Valley, California, USA, pp. 1133–1142 (2008)
5. Balog, K., de Rijke, M.: Non-Local Evidence for Expert Finding. In: Proceedings of the 17th ACM conference on information and knowledge management (CIKM 2008), Napa Valley, California, USA, pp. 731–740 (2008)
6. Vercoustre, A., Thom, J.A., Pehcevski, J.: Entity Ranking in Wikipedia. In: Proceedings of the 2008 ACM symposium on Applied computing (SAC 2008), Fortaleza, Ceara, Brazil (2008)
7. Demartini, G., Firan, C.S., Iofciu, T.: L3S Research at INEX 2007: Query Expansion for Entity Ranking Using a Highly Accurate Ontology. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX 2007. LNCS, vol. 4862, pp. 252–263. Springer, Heidelberg (2008)
8. Tsikrika, T., Serdyukov, P., Rode, H., Westerveld, T., Aly, R., Hiemstra, D., de Vries, A.P.: Structured Document Retrieval, Multimedia Retrieval, and Entity Retrieval Using PF/Tijah. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX 2007. LNCS, vol. 4862, pp. 306–320. Springer, Heidelberg (2008)