Integrating Document Features for Entity Ranking

Jianhan Zhu, Dawei Song, Stefan Rüger

Knowledge Media Institute
The Open University, United Kingdom
{j.zhu, d.song, s.rueger} @open.ac.uk

Abstract. The Knowledge Media Institute of the Open University participated in the entity ranking and entity list completion tasks of the Entity Ranking Track in INEX 2007. In both the entity ranking and entity list completion tasks, we have considered document features in addition to a basic document content based relevance model. These document features include categorizations of documents, relevance of category names to the query, and hierarchical relations between categories. Furthermore, based on our TREC2006 and 2007 expert search approach, we applied a co-occurrence based entity association discovery model to the two tasks based on the assumption that relevant entities often co-occur with query terms or given relevant entities in documents. Our initial experimental results show that, by considering the predefined category, its children and grandchildren in the document content based relevance model, the performance of our entity ranking approach can be significantly improved. Consideration of the predefined category's parents, a category name based relevance model, and the co-occurrence model is not shown to be helpful in entity ranking and list completion, respectively.

Keywords: entity ranking, list completion, entity retrieval, categories

1 Introduction

In this year's Entity Ranking Track, there are two related tasks, i.e., entity ranking and entity list completion, on the Wikipedia dataset. A special feature of the Wikipedia dataset is that each document corresponds to an entity. Given a query topic, the aim of entity ranking is to find a list of entities that are relevant to the query topic. A category as part of the query topic specifies the type of entities that should be returned. Some entities have been labeled with certain categories in the dataset. Since entity labeling has been done collaboratively and voluntarily by users, there is no guarantee that all entities are labeled, and all entities are correctly labeled. Therefore, we assume that the categories for each entity can only be used as a guideline. We identify four types of entities that are potentially relevant to a query topic in terms of their categorization. First, the entities are labeled with the specified category. Second, the entities are labeled with categories related to the specified category. Third, the entities are labeled with neither the specified categories nor any category related to the specified category. Fourth, the entities are not labeled.

The Entity Ranking Track is related to the Expert Search task in the TREC (Text REtrieval Conference) 2005, 2006, and 2007 Enterprise Search tracks [1][2][3]. Given a query topic, the aim of expert search is to find a ranked list of experts from a list of candidates in an organization or domain. We successfully used a two-stage model in expert search in TREC2006 and 2007 Expert Search tasks. The two-stage model consists of a document relevance model where a number of documents relevant to the query topic are discovered, and a co-occurrence model where experts' relevance to the query topic are measured by their co-occurrences with query terms in a text window in these relevant documents. The two-stage model is also compatible with how users search for experts on the web, i.e., they find relevant documents on a topic through a search engine, and then read these documents in order to find out experts in these documents. Based on the similarity between entity ranking and expert search, we use the two-stage model as one component in entity ranking.

Entity ranking is more general than expert search since in entity ranking, entities of any types can be retrieved for a topic. The nature of Wikipedia dataset makes the entity ranking track different from expert search task, since in entity ranking each document corresponds to an entity while in expert search expert names are mentioned in documents and named entity recognition tools need to be employed in identifying these occurrences of expert names.

Entity list completion can be seen as a special case of entity ranking task. In entity list completion, a few entities relevant to a query topic are given. These entities can be used as relevance feedback information for finding other relevant entities. We think there are mainly two ways for using this relevance feedback information. First, use these entities and their corresponding documents as relevance feedback information. Second, based on the observation that these entities may often co-occur with other entities that are also relevant to the query topic, we propose to use a co-occurrence model for measuring the relevance between new entities and these given entities.

We think that entity ranking is sensitive to multiple document features that need to be taken into account in finding relevant entities on the Wikipedia dataset. Therefore, a number of components considering these document features in our approach include: 1. Document content based relevance to the query topic, 2. Specified category in the query topic, 3. Sub-categories and parents of the specified category, 4. The content based relevance of category names of each document to the query topic, and 5. a novel multiple-window based co-occurrence model.

We proposed the multiple-window based co-occurrence model in TREC 2006 and 2007 for expert search [4][5]. Similarly, we have applied the multiple-window based approach to entity ranking. Entities are mentioned in other documents. The contexts of these occurrences of entities often include query terms in the query topic. We assume that there are associations between an entity and query terms on multiple levels, i.e., from phrase, sentence, paragraph, etc., up to document levels. All these levels of associations need to be considered in the co-occurrence model. Increased window sizes often lead to more coverage of associations while introducing noise. We propose a novel weighted multiple window size based approach as opposed to a single fixed window size based approach in previous association discovery research [6]. In entity list completion, we have considered the co-occurrences of given relevant entities and new entities.

The rest of this paper is organized as follows. In Section 2, we introduce our entity ranking approach. We extend our entity ranking approach for entity list completion in Section 3. We report our experimental results, and submitted runs on Wikipedia dataset in Section 4 and 5, respectively. Finally, we conclude and discuss future work in Section 6.

2 ENTITY RANKING

For each document, which corresponds to an entity, we use its content based relevance to the query topic as the baseline model. We enhance the baseline model by taking into account multiple document features, i.e., the entity's categories' relations with the specified category, the entity's categories' content based relevance to the query topic, and the entity's co-occurrences with the query terms in other documents.

2.1 Content based Relevance

If an entity is relevant to a topic, the content of the document representing the entity is likely to contain terms in the query topic. We used three standard relevance models, i.e., Boolean, BM25, and Lucene's span relevance models, for judging the relevance of the document content to the topic.

BM25 is a probabilistic IR model. We used the BM25 equation of Okapi [7] for the relevance model. Given a query q and document d, we get

$$p(d \mid q) \propto \sum_{T \in q} w \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} + k_2 |q_i| \frac{avdl-dl}{avdl+dl} \quad (1)$$

where $w=log((N-n+0.5)/(n+0.5))$ is the IDF of T; N is the number of documents in the dataset; n is the number of documents where T appears; K is $k_1((1-b)+b*dl/avdl)$; k_1, b, k_2 and k_3 are parameters; tf is the frequency of T in d; qtf is the frequency of T in q; dl is the length of d; and $avdl$ is the average document length. Based on the suggested parameter values in Okapi [7], we set the values of k_1, b, k_2 and k_3 as 1.4, 0.6, 0.0, and 8.0, respectively.

Boolean query model specifies that all query terms must occur in a document.

$$p(d \mid q) \propto coord \cdot \frac{1}{\sqrt{\sum_{T \in q} idf^2}} \sum_{T \in q} tf \cdot idf \cdot \frac{avdl}{dl} \quad (2)$$

where *coord* is the number of query terms that are found in d divided by the total number of terms in the query, and *idf* is $1+log(N/(n+1))$.

Span query model is based on co-occurrences of all query terms in text windows. The score for a matching span s is as follows:

$$p(s \mid q) \propto coord \cdot sloppyFreq(s) \cdot \frac{1}{\sqrt{\sum_{T \in q} idf^2}} \sum_{T \in q} idf \cdot \frac{avdl}{dl} \quad (3)$$

where *sloppyFreq(s)=1/(slop+1)* is a factor that decreases as the sloppiness of the matching span increases[1]; the effect is to favor more exact matches.

In order to help equalize the scoring between documents with many matching spans and those with few matches, the score for all matching spans in a document *d* is taken as the square root of the total score of all matches as follows:

$$p(d \mid q) \propto \sqrt{\sum_{s \text{ in } d} p(s \mid q)} \qquad (4)$$

We apply our relevance model to entity ranking in Section 4.

2.2 Entity's Categories

An entity's category information can help entity retrieval in mainly three aspects.

First, since a preferred category is specified as part of a query topic, if three is a match between an entity's categories and the preferred category, the relevance of the entity to the query topic should be largely boosted.

Second, since the categorization of entities is not done completely and some relevant entities may not be labeled with the preferred category, we need to find categories which are relevant to the preferred category and were used to label other relevant entities. We propose to find the sub-categories and parents of the preferred category. If there is a match between these categories and an entity's categories, the relevance of the entity to the query topic will still be boosted although the entity's categories may not contain the preferred category.

In the hierarchy of categories for the Wikipedia dataset, the links between categories do not always represent an "is-a" relationship between two categories, i.e., the child may not be a sub-class of the parent sometimes. In order to avoid the "concept drift" in the hierarchy, categories related to the preferred category are only limited to its parents and children in our approach, although we will investigate the effect of incorporating more distantly linked categories in future work.

Third, if an entity is relevant to a query topic, the names of the entity's categories can often contain terms in the query topic. We propose to create a metadata field for an entity by joining its' categories' names together, and use a standard relevance model, such as BM25, Boolean, or span models, to measure the relevance between this metadata field and the query topic.

We envisage that the categorization information associated with the Wikipedia dataset can significantly assist entity retrieval. The assumption can be tested based on an anatomy of our entity retrieval system studying the effect of multiple document features in entity retrieval that will be shown in Section 4.

[1] "The maximum allowable positional distance between terms to be considered a match is called slop. Distance is the number of positional moves of terms to reconstruct the phrase in order" [8]

2.3 Entity's Co-Occurrences with Query Terms

So far, entity ranking task is similar to a document ranking problem, i.e., judging the relevance between a number of documents and a query topic for producing a ranked list of documents. Categorization information associated with documents can be used as filtering or weighting search results based on the similarity between their categories and the predefined category. However, we propose an entity co-occurrence model, which is based on the context information of each entity in documents, to further enhanced entity ranking.

The proposed entity co-occurrence model is very similar to our co-occurrence model, which takes into account contextual information of experts in documents, in the TREC expert search task. Similarly, each entity occurs in a number of documents, and the contexts of these occurrences can help us estimate the relevance between the query and the entity.

In TREC2006 and 2007, we have successfully employed a novel two-stage multiple window based approach for expert search. Now we propose to apply the two-stage model to the entity ranking task. Given a query q, an entity e's relevance to the query in a co-occurrence model is $p_{co\text{-}occ}(e|q)$. We get:

$$p_{co-occ}(e \mid q) = \sum_{d} p(e, d \mid q) = \sum_{d} p(d \mid q)\, p(e \mid d, q) \qquad (5)$$

where d is a document, $p(d|q)$ is the document relevance model, and $p(e|d,q)$ is the co-occurrence model.

We use one of the three relevance models presented in Section 2.1 for the first stage. In the second stage, an entity's relevance to the query topic is judged based on the co-occurrences of the entity and query topic terms in documents.

Since entity's association with a query topic can be of multiple levels, from phrase, sentence, paragraph, up to document levels, we propose a novel multiple window based approach to capture all these levels of associations. We assume that smaller text windows lead to more accurate associations and larger windows may introduce noise thus leading to less accurate associations. Therefore, we take a weighted sum of the relevance between an entity and a topic based on a number of text windows, where smaller windows are given higher weights and larger windows are given lower weights.

Suppose that, in a document d, there are M occurrences of an entity e as $\{e_k\}$ ($k=1,\ldots, M$). We use L windows with incremental sizes, i.e., $\{W_j\}$ ($j=1,\ldots, L$), for associating each entity occurrence e_k with query terms in d. For e_k, the smallest window in $\{W_j\}$, SW_k, which can enable e_k to co-occur with all query terms in SW_k, is used to measure the association between e_k and the query; if such a window does not exist, the association score between e_k and the query is zero. For example, suppose that we use three windows {20, 40, 80}. If one occurrence of an expert, e_k, does not co-occur with all query terms within the 20-sized window but does co-occur with all of the query terms within the 40-sized window, then we use the window size 40 to measure their associations. Therefore, for different occurrences of experts, different window sizes may be used for association discovery. This gives us more flexibility than the use of one fixed sized window only. Thus, in d, the association between e and the query is a weighted sum of the association scores between all the occurrences of e with the query, respectively, as follows:

$$p(e \mid d, q_i) \propto \sum_{\substack{k=1,\ldots,M \\ e_k \text{ and } q_i \text{ co-occur} \\ SW_k \text{ is the smallest}}} f(SW_k) \cdot P(e_k \mid d, q_i, SW_k) \quad (6)$$

$f(SW_k)$, as a function of the window size, is the weight for the association score between e_k and the query in d. Generally, the smaller the window size, the higher the weight, thus the weight is inversely proportional to the window size.

We extend the co-occurrence model proposed by Cao et al. [9] to our multiple-window-based co-occurrence model and define $P(e_k|d,q_i, SW_k)$ as:

$$p(e_k \mid d, q_i, SW_k) \propto \mu \frac{pf(e_k, SW_k)}{pf_{total}(SW_k)} + \frac{1-\mu}{df_e} \sum_{d_i : e \in d_i} \frac{1}{n_e} \sum_{\substack{e_j : e_j \in d_i \\ e_j \text{ and } q_i \text{ co-occur} \\ SW_j \text{ is the smallest}}} \frac{pf(e_j, SW_j)}{pf_{total}(SW_j)} \quad (7)$$

where $pf(e_k, SW_k)$ is the frequency of e_k in window SW_k, $pf_{total}(SW_k)$ is the total frequency of entities in SW_k, df_e is document frequency of e, n_e is the number of occurrences of e in d_i. We use a Dirichlet prior to smooth parameter μ:

$$\mu = \frac{pf_{total}(SW_k)}{pf_{total}(SW_k) + \kappa}$$

Here κ is the average of term frequency of all occurrences of all entities inside all windows in the dataset.

We test the effectiveness of our co-occurrence model in Section 4.

2.4 Our Combined Entity Ranking Approach

Our overall entity ranking approach integrates the document relevance model in Section 2.1, weighting function based on entity's categories and relevance of an entity's category names in Section 2.2, and innovative co-occurrence model in Section 2.3.

Given an entity e, a predefined category c, and a query q, therefore, the overall relevance of e given q is:

$$p_{overall}(e \mid q, c) = w_c(w_{content} p(d_e \mid q) + w_{name} p(name_e \mid q) + w_{co-occ} p_{co-occ}(e, q)) \quad (8)$$

where c is the predefined category, d_e is the document representing entity e, $name_e$ is the joint category names of e, w_c is a weight based on the relation between the the entity's categories and the predefined category, $w_{content}$, w_{name}, and $w_{co\text{-}occ}$ are the weights for the document relevance model, category names based relevance model, and co-occurrence model, respectively. By adjusting $w_{content}$, w_{name}, and $w_{co\text{-}occ}$, we can tune the effect of the three models in entity ranking, and by adjusting w_c, we can tune the effect of predefined category, its parents, and its children in entity ranking. Finally, the overall relevance of e given q is used to rank entities.

3 ENTITY LIST COMPLETION

Entity list completion can be seen as a special case of entity ranking where a few given relevant entities can be used as relevance feedback information. We have incorporated the given relevant entities in our two-stage approach. We assume that entities relevant to the query topic tend to co-occur often with the given entities in documents. Again, we adopted the novel multiple-window based approach for integrating multiple levels of associations between an entity and any of the given entity. Based on Equation 10, we get

$$p_{overall}(e \mid q, c) = w_c(w_{content} p(d_e \mid q) + w_{name} p(name_e \mid q) + w_{co-occ} p_{co-occ}(e, q) + w_{co-occ,given} \sum_j p_{co-occ}(e, e_j)) \quad (9)$$

where $w_{co\text{-}occ}$ is the weight for the co-occurrence model for the entity and given entities, and e_j is a given entity.

4 EXPERIMENTAL RESULTS

The aim of our experiments is to test the effect of the basic document relevance model and different document features, i.e., categorizations of documents, relevance of category names to the query, and hierarchical relations between categories in entity ranking and list completion.

We pre-processed the dataset by removing HTML tags. We indexed and searched the dataset using Lucene. We used a pure document content based Boolean relevance model as the baseline shown in Table 1, i.e., in Equation 8, w_c is set as 1.0, $w_{content}$ is set as 1.0, w_{name} is set as 0, and $w_{co\text{-}occ}$ is set as 0. We improve the baseline by adding categorization information and/or the co-occurrence model in getting other runs shown in Table 1.

Table 1. Experimental results for entity ranking

Runs	MAP	R-Prec	Bpref	P@10	Num_rel_ret
baseline	0.1943	0.2239	0.2697	0.2174	623
Cat1	0.2712	0.3036	0.3530	0.3000	596
Cat2	0.2609	0.2763	0.3796	0.2804	600
Cat3	0.3116	0.3351	0.3907	0.3543	655
Cat4	0.3306	0.3584	0.4156	0.3652	669
Cat5	0.3206	0.3457	0.4047	0.3478	654
Cat6	0.2475	0.2799	0.3331	0.2870	594
Cat-CoOcc1	0.3069	0.3447	0.3995	0.3457	687
Cat-background1	0.2827	0.3357	0.3882	0.3450	452
Cat-background2	0.3298	0.3532	0.4160	0.3587	668
Cat-background3	0.3313	0.3639	0.4151	0.3652	671
Cat-background4	0.3308	0.3650	0.4132	0.3652	672

Cat1: we assume that entities labeled with the predefined category should be given higher weight, and set w_c as 1.0 for entities labeled with the predefined category, and 0.3 otherwise for run Cat1 in Table 1. We can see that MAP, R-Prec,

Bpref and P@10 are all significantly improved compared with the baseline showing that categorization information is very helpful in entity ranking. However, the number of relevant entities discovered decreases from 623 to 596, showing that the integration of categorization information helps put relevant entities near the top of ranked lists at the expense that some entities not labeled with the predefined category are put lower down ranked lists.

Cat2: we take into account the relevance of entity's category names to query topics in the model by setting w_{name} as 0.4 for run Cat2 in Table 1. We can see that the results degrade due to the combination. The reason might be that terms in an entity's category names may often be mentioned in the entity's document already and simply combining the relevance scores linearly may not be very helpful in entity ranking.

Cat3: we assume that entities labeled with the children of the predefined category should also be considered. Therefore, we set w_c as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, and 0.3 otherwise for run Cat3 in Table 1. We can see that MAP, R-Prec, Bpref and P@10 are all significantly improved compared with Cat1. It is worth noting that the number of relevant entities retrieved also significantly improves compared with that for Cat1. This proves that some entities are not labeled with the predefined category directly but labeled with the children of the predefined category. By taking into account children of the predefined category, the retrieval performance significantly improves.

Cat4: we assume that entities labeled with the grandchildren of the predefined category should be considered. Therefore, we set w_c as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the grandchildren of the predefined category, and 0.3 otherwise for run Cat4 in Table 1. We can see that all performance measures improve compared with those for Cat3, showing that grandchildren of the predefined category are helpful in entity ranking.

Cat5: we assume that entities labeled with the grand-grandchildren of the predefined category should be considered. Therefore, we set w_c as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the grandchildren of the predefined category, 0.55 for entities labeled with the grand-grandchildren of the predefined category, and 0.3 otherwise for run Cat5 in Table 1. We can see that all performance measures degrade compared with those for Cat4, showing that the introduction of grand-grandchildren of the predefined category may introduce more noise than helpful information in entity ranking. This is also consistent with the observation of concept drift in the categorization hierarchy.

Cat6: we assume that entities labeled with the parents of the predefined category should be considered. Therefore, we set given higher weight, and set w_c as 1.0 for entities labeled with the predefined category, 0.7 for entities labeled with the parents of the predefined category, and 0.3 otherwise for run Cat6 in Table 1. We can see that all performance measures degrade compared with those for Cat1, showing that parents of the predefined category are not very helpful in entity ranking due to the reason that they are probably too general.

We further improve the categorization information enhanced baseline by integrating with the co-occurrence model. We trained our co-occurrence model on the TREC2006 expert search test collection. On the basis of run Cat4, we get:

Cat-CoOcc1: we set $w_{co\text{-}occ}$ as 0.3 for run CatCoOcc1 in Table 1. We can see that MAP, R-Prec, Bpref, and P@10 all degrade compared with those for Cat4. However, the number of relevant entities retrieved improves from 669 to 687, showing that the integration of the co-occurrence model helps find more relevant entities at the expense of putting many relevant entities lower down the ranked lists than Cat4 does.

Furthermore, we study the effect of the background document relevance model in entity ranking. On the basis of the best performing run, Cat4, we get:

Cat-background1: we assume that entities not labeled with the predefined category, its children, or grandchildren are not relevant, i.e., we set w_c as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the grandchildren of the predefined category, and 0 otherwise for run Cat-background1 in Table 1. We can see that all performance measures degrade compared with those for Cat4. Especially, the number of relevant entities retrieved decreases sharply from 669 to 452, showing that entities in the dataset are not completely labeled, and it is necessary to include a background model for more successful entity ranking.

We study the effect of the weight for the background model in the following runs:

Cat-background2: on the basis of Cat4, we set w_c as 0.2 for the background model in run Cat-background2 in Table 1. We can see that the MAP degrades slightly compared with that for Cat4.

Cat-background3: on the basis of Cat4, we set w_c as 0.35 for the background model in run Cat-background3 in Table 1. We can see that the MAP improves slightly compared with that for Cat4.

Cat-background4: on the basis of Cat4, we set w_c as 0.4 for the background model in run Cat-background4 in Table 1. We can see that the MAP degrades slightly compared with that for Cat4.

We further study how to use the given relevant entities for entity list completion using the co-occurrence model proposed in Section 3.

Table 2. Experimental results for entity list completion

Runs	MAP	R-Prec	Bpref	P@10	Num_rel_ret
Cat-CoOcc-feedback1	0.2725	0.3005	0.3471	0.2935	558
Cat4	0.2727	0.3005	0.3473	0.2935	558

In Table 2, on the basis of run Cat4, we integrate the co-occurrence model for the following run:

Cat-CoOcc-feedback1: in Equation 9, we set w_c as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the grandchildren of the predefined category, and 0.3 otherwise, $w_{co\text{-}occ}$ as 0, and $w_{co\text{-}occ,\ given}$ as 0.4.

For comparison purpose, we remove given relevant entities from run Cat4, which does not use any relevance feedback information, and get the results for Cat4 in Table 2. We can see that the introduction of the co-occurrence model does not help improve the performance of entity ranking. We think this may be due to the reason that each entity's document already contain detailed and complete information, therefore, the

co-occurrence model introduce information that is already covered in the entity's document. We will study more effective use of relevance feedback information in future work.

5 OUR SUBMITTED RUNS

We submitted three entity ranking runs and one list completion run to the Entity Ranking track, and their results are shown in Table 3. The descriptions of our four runs are as follows.

ou_er01: Boolean model, w_c is set as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the parent of the predefined category, or 0.3 otherwise, $w_{content}$ is set as 1.0, w_{name} is set as 0.4, and $w_{co\text{-}occ}$ is set as 0.2.

ou_er02: Boolean model, w_c is set as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the parent of the predefined category, or 0.3 otherwise, $w_{content}$ is set as 1.0, w_{name} is set as 0.6, and $w_{co\text{-}occ}$ is set as 0.4.

ou_er03: Boolean model, w_c is set as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.8 for entities labeled with the parent of the predefined category, or 0.5 otherwise, $w_{content}$ is set as 1.0, w_{name} is set as 0.4, and $w_{co\text{-}occ}$ is set as 0.2.

ou_lc01: Boolean model, w_c is set as 1.0 for entities labeled with the predefined category, 0.8 for entities labeled with the children of the predefined category, 0.65 for entities labeled with the parent of the predefined category, or 0.3 otherwise, $w_{content}$ is set as 1.0, w_{name} is set as 0.4, $w_{co\text{-}occ}$ is set as 0.2, and $w_{co\text{-}occ,given}$ is set as 0.5.

Table 3. Submitted runs for entity ranking and list completion

Runs	MAP	R-Prec	Bpref	P@10
ou_er01	0.2582	0.2958	0.3855	0.2913
ou_er02	0.2306	0.2583	0.3639	0.2630
ou_er03	0.2306	0.2583	0.3639	0.2630
ou_lc01	0.2072	0.2213	0.2384	0.2389

6 CONCLUSIONS

We have participated in both entity ranking and list completion tasks in INEX2007. Based on the assumption that entity ranking is sensitive to multiple document features, we propose a novel approach for integrating multiple document features for effective entity ranking. In our approach, we have considered the content of the document describing an entity, matching between the entity's categories and the preferred category, the effect of hierarchical relations between categories, and the content of categories. In addition, we integrate a co-occurrence model, which considers multiple levels of associations between an entity and a query topic, in entity ranking.

We treat entity list completion as a special case of entity ranking by using the given relevant entities as relevance feedback information for incorporation into our co-occurrence model, which considers multiple levels of associations between an entity and each given relevant entity.

Our experimental results show that a document content based relevance model can be significantly improved by considering the categorization information of documents. In particular, consideration of the predefined category, its children, and grandchildren is helpful in entity ranking, while consideration of the predefined category's grand-grandchildren seems not very helpful. Consideration of the predefined category's parents is not help in entity ranking. We think the reason may be due to "concept drift" in the category hierarchy.

On the other hand, entity ranking based purely on documents labeled with the predefined category, its children, and grandchildren can be significantly improved by integrating with the baseline, showing that there are still a number of entities which are not labeled with the predefined category or its children and grandchildren which still are relevant to the query topic.

Interestingly, the incorporation of both category name based relevance model and our co-occurrence model is not helpful in both entity and list completion, respectively. We think this may be due to the reason that each entity's document already contain detailed and complete information, therefore, both the category name based relevance model and our co-occurrence model introduce information that is already covered in the entity's document. We will carry out more systematic research to re-confirm our findings in our experimental results. We will also study more effective approach of using relevance feedback information in the form of given relevant entities in entity list completion.

ACKNOWLEDGEMENTS

The work reported in this paper is funded in part by an IBM 2007 UIMA innovation award and the JISC (Joint Information Systems Committee) funded DYNIQX (Metadata-based DYNamIc Query Interface for Cross(X)-searching content resources) project.

References

[1] Bailey, P., Craswell, N., de Vries, A.P., and Soboroff, I.(2007) Overview of the TREC 2007 Enterprise Track (DRAFT). In Proc. of The Sixteenth Text REtrieval Conference (TREC 2007), Gaithersburg, Maryland USA.

[2] Craswell, N., de Vries, A.P., Soboroff, I. (2005) Overview of the TREC-2005 Enterprise Track. In Proc. of The Fourteenth Text REtrieval Conference (TREC 2005).

[3] Soboroff, I., de Vries, A.P. and Craswell, N. (2007) Overview of the TREC 2006 Enterprise Track. In Proc. of The Fifteenth Text REtrieval Conference (TREC 2006), Gaithersburg, Maryland USA.

[4] Zhu, J., Song, D., Rüger, S., Eisenstadt, M. and Motta, E. (2007) The Open University at TREC 2006 Enterprise Track Expert Search Task. In Proc. of The Fifteenth Text REtrieval Conference (TREC 2006).
[5] Zhu, J., Song, D., Rüger, S., Eisenstadt, M. and Motta, E. (2007) The Open University at TREC 2006 Enterprise Track Expert Search Task. In Proc. of The Sixteenth Text REtrieval Conference (TREC 2007) Notebook.
[6] Conrad, J.G., Utt, M.H. (1994) A System for Discovering Relationships by Feature Extraction from Text Databases. In Proc. of SIGIR 1994: 260-270.
[7] Robertson, S.E., Walker, S., Beaulieu, M.M., Gatford, M., Payne, A. (1995): Okapi at TREC-4. In NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-04): 73-96.
[8] Hatcher, E. and Gospodnetic, O. (2004) Lucene in Action. Manning Publications Co, ISBN: 1932394281.
[9] Cao, Y., Liu, J., Bao, S. and Li, H. (2005) Research on Expert Search at Enterprise Track of TREC 2005. In Proc. of TREC 2005.