Entity Retrieval by Hierarchical Relevance Model, Exploiting the Structure of Tables and Learning Homepage Classifiers

Yi Fang, Luo Si
Department of Computer Sciences, Purdue University
West Lafayette, IN, 47907, USA
{fangy,lsi}@cs.purdue.edu

Zhengtao Yu,Yantuan Xian,Yangbo Xu
School of Information Engineering and Automation
Kunming University of Science and Technology
Kunming, China, 650051

ABSTRACT

This paper gives an overview of our work done for the TREC 2009 Entity track. We propose a hierarchical relevance retrieval model for entity ranking. In this model, three levels of relevance are examined which are document, passage and entity, respectively. The final ranking score is a linear combination of the relevance scores from the three levels. Furthermore, we 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. To find entity homepages, we train logistic regression models for each type of entities. A set of templates and filtering rules are also used to identify target entities. The key lessons that we learned by participating this year's Entity track include: 1) our special treatment of table and list data is well rewarding; 2) The high accuracy of homepage finding is crucial in this track; 3) Wikipedia can serve as a valuable knowledge resource for different aspects of the related entity finding task.

1. INTRODUCTION

As the Web has evolved into a data-rich repository, both commercial systems and the information retrieval community have shown increasing interest in not just returning documents, but specific entities in response to a user's query. The task of the Entity track is to investigate the problem of related entity finding. Specifically, given the name and homepage of an entity, as well as a context described in natural language text, the retrieval system needs to find target entities with homepages that are of the specified type. In this year’s task, entity types are limited to people, organizations, and products.

In fact, related entity finding incorporates both expert finding and homepage finding tasks from previous TREC tracks. In this paper, we propose a hierarchical relevance retrieval model for entity ranking. In particular, relevance is examined at three levels such as document, passage and entity. The final ranking score is a linear combination of the relevance scores from the three levels. Moreover, as many entities are stored in a structural or semi-structure form such as tables and lists, we exploit the structure of these data to reduce the uncertainty in entity recognition and finding. For entity homepage finding, in addition to utilize Wikipedia as a direct reference, we treat it as a classification problem by training logistic regression models for the three target entity types respectively. Because it is very time consuming to manually identify the positive training instances from a huge pool of candidate web pages, we use an incremental learning strategy to train the classifiers.

2. APPROACHES

2.1 SYSTEM ARCHITECTURE

The architecture of our participant system can be seen in Figure 1. Similar to many question answering (QA) systems, our system also includes three major components: input (question) analysis, document/passage retrieval, and entity (answer) extraction. These three components have been widely adopted to build QA systems. However, there are several key operations in our system that were rarely performed by QA. The first one is the table/list processing component which exploits the structure of tables and lists to make joint decisions on all the elements of the tables and lists. The second key component is to extract the constraints on target entities from the query topics and to utilize the constraints to better locate the related entities. The entity ranking component is the hierarchical relevance model that we proposed to consider the relevance at different levels. Another component unique for this entity track is the homepage detection algorithm that identifies the homepages for the extracted entities. The following sections explain the details of the individual components of our system.

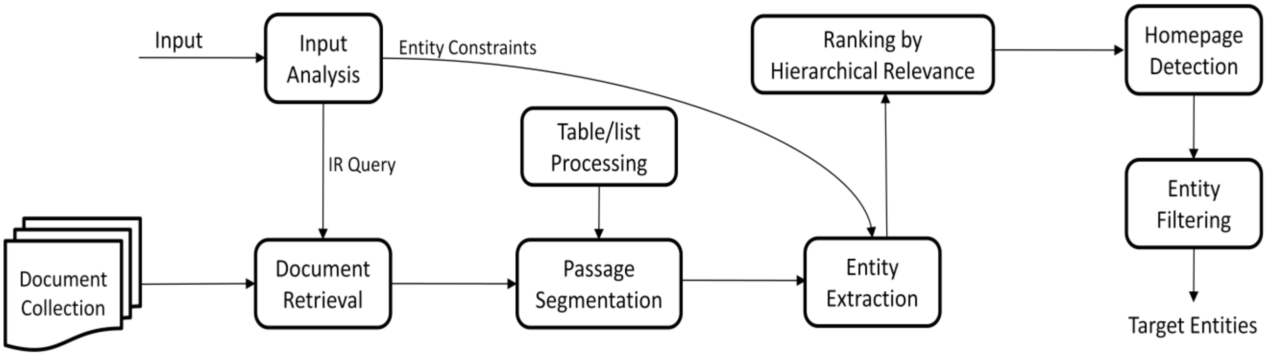

Figure 1: Architecture of Our Entity Retrieval System

2.2 HIERACHICAL RELEVANCE RETRIEVAL MODEL

We state the problem of identifying candidates who are related entities as follows: what is the probability of a candidate e being the target entity given a query q and target type t? That is, we determine $p(e|q,t)$, and rank candidate e according to this probability. The top k candidates are deemed the most probable entities. By applying Bayes' Theorem and the chain rule, we can decompose $p(e|q,t)$ into the following form

$$p(e|q,t) \propto \sum_{d}\sum_{s} p(q|d)\, p(q|s,d) p(e|q,t,s,d)$$

where s denotes a supporting passage in a supporting document d. The first quantity $p(q|d)$ on the right hand side is the probability that the query is generated by the supporting document, which reflects the association between the query and the document. Similarly, the first quantity $p(q|s,d)$ reflects the association between the query and the supporting passage. The last quantity $p(e|q,t,s,d)$ is the probability that a candidate entity e is the related entity given passage s, type t and query q. In summary, this probabilistic retrieval model considers the relevance at three different levels: document, passage and entity. However, accurately estimating these probabilities is difficult for generative probabilistic language modeling techniques. Instead, motivated by the idea of the hierarchical relevance model, we use a linear combination of relevance scores from these three levels to yield the final ranking score $f(e|q,t)$ as follows:

$$f(e|q,t) = \sum_{d}\sum_{s} \alpha f(q|d) + \beta f(q|s,d) + \gamma f(e|q,t,s,d)$$

where α, β and γ are the combination coefficients for the retrieval scores $f(q|d)$, $f(q|s,d)$ and $f(e|q,t,s,d)$. In the subsequent sections, we show how to calculate these individual retrieval scores. All the retrieval scores are finally normalized by the maximum score in their level.

2.2.1 DOCUMENT RETRIEVAL

The main role of the document retrieval stage is to retrieve a hopefully very small subset of the entire collection which will be processed in detail by the entity extraction and other components. The formulation of query from a natural language narrative should maximize the performance of document retrieval. Many entities exist in the documents or queries in the form of acronym such as IU for Indiana University. We expand the original narrative query to include the acronym or full name of the source entity which is likely to cause more retrieved documents containing related entities. Without external resources, to find the acronym of an entity can sometimes be difficult such as LVMH for Moët Hennessy-Louis Vuitton. In our runs, we resort to Farlex Free Dictionary[1] to find

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

acronyms. To retrieve documents from the data collection, we experimented to use the INDRI toolbox[2] and Google respectively. In one run which is not submitted, we use the following INDRI query to retrieve the top 100 documents for each topic:
#weight(3.0 @odN(source entity) 3.0 @odN(keyword) 2.0 @odN(phrase) 1.0 (each term) 1.5 (acronym or full name of source entity) 1.0 (synonym and antonym of keyword))

where N is the number of words in the phrase. The keyword is the term in the narrative that reflects the main property of the target entity.

We found in our experiments the INDRI document retrieval was generally not effective in locating supporting documents. In our submitted runs, we obtain the top 1000 results from Google by the query similar to the above INDRI query except no weights associated with terms. Then we remove those pages that are not in the Clueweb09 test collection and use the top 5 pages in the remaining pages as the candidate documents to extract related entities.

2.2.2 COMPUTING $f(q|d)$

$f(q|d)$ reflects the similarity between the query topic/narrative and the supporting document. Given q and d, it can be computed by retrieval models such as BM25 and language modeling. Because the title of a web page is usually a good and concise summary of the whole document, d in our runs is represented by the TITLE element of the web page instead of using the full text. This strategy seems especially effective for this year's topics, as the titles of many supporting documents exhibit strong correlations with the query narratives.

2.2.3 COMPUTING $f(q|s,d)$

$f(q|s,d)$ reflects the similarity between the query and the supporting passage. Passages are sentences segmented by a set of predefined rules. For tables, the elements of the same attributes belong to a passage along with the attribute name, the header and preceding sentence of the table. In our runs, $f(q|s,d)$ is computed by summing over all the similarity scores R between the terms w_q in the query and the terms w_s in the passage as follows:

$$f(q|s,d) = \sum_{w_q \in q} \sum_{w_s \in s} R(w_q, w_s)$$

where the similarity score R is computed based on the distance defined by WordNet[3].

2.2.4 COMPUTING $f(e|q,t,s,d)$

$f(e|q,t,s,d)$ reflects the confidence that the entity e is the target entity given the corresponding evidence, which is the relevance score at the entity level. First of all, we use Stanford Named Entity Recognizer[4] (NER) and LBJ Named Entity Tagger[5] to extract entities of the target type from the passage. These NERs can directly recognize persons and organizations, but not products. We train a CRF model for named product recognition by labeling 4000 documents about product entities.

After named entity recognition, we then compute the relevance score $f(e|q,t,s,d)$. Specifically, we examine how consistent that the extracted entity satisfies the property specified in the query by looking at the homepage of the entity. In our runs, we calculate the frequency f_1 of the keyword appearing in the entity's Wiki page if the page exists. The keyword is simply the first term or phrase of each query narrative since we found the first terms characterize the target entities (e.g., carriers in Topic 1 "Carriers that Blackberry makes phones for"). We further compute the frequency f_2 of the keyword appearing in the "Categories" section in the bottom of the Wiki page as it can reflect a higher degree of confidence. For example, in Topic 7, the narrative is "Airlines that currently use Boeing 747 planes" and thus the keyword is "airlines". The Wiki page of the candidate entity "Air China" gives a high score f_2 , because "airlines" appears in the "Categories" section which clearly shows the entity is an airline. If

[2] http://www.lemurproject.org/indri/

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

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

[5] http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=FLBJNE

the Wiki page does not exist, we instead use its homepage found by our homepage detection algorithm described in the next section.

We also consider the proximity of the entity and the query terms in a passage as follows:

$$f_3 = \sum_{w_e \in e} \frac{M - \frac{1}{|s|} \sum_{w_q \in s \cap q} D(w_e, w_q) R(w_e, w_q)}{M}$$

where $D(w_e, w_q)$ counts the number of words between the entity name w_e and the query term w_q in the passage. $R(w_e, w_q)$ measures the similarity between w_e and w_q as described in Section 2.2.3. $|s|$ is the total number of words in the passage and M is a constant to scale the quantity to be positive.

Consider the above three factors, the calculation of $f(e|q,t,s,d)$ is as follows

$$f(e|q,t,s,d) = \delta_1 f_1 + \delta_2 f_2 + \delta_3 f_3$$

where δ_1, δ_2 and δ_3 are the combination weights (with $\delta_2 \gg \delta_1$).

2.3 ENTITY EXTRACTION FROM TABLES AND LISTS

The Web contains a large amount of structured data embedded in natural language text, tables, lists, and other forms. A lot of entities exist in these structural forms and this is also manifested in this year's related entity finding task as many targeted entities are stored in the tables and lists. On the one hand, this poses challenges to entity extraction because most NERs utilize the context information to recognize the named entities (e.g., Conditional Random Fields or Hidden Markov Fields) while there is few context for the elements in tables. Without the context, to accurately recognize the types of the entities in the tables is difficult. To acquire the context of the entities, we use the element names as queries in Google to retrieve relevant documents, and then use NERs to recognize the types of the elements in these documents by majority voting.

On the other hand, the structure of tables and lists can facilitate the entity extraction. For example, in a table, all the elements with the same attribute have similar properties and are likely to share the same entity types. Moreover, they are likely altogether to be the target entities. Therefore, we can utilize this structural information to reduce the uncertainty on a single entity and to make a joint decision on all the entities with the same attribute. In our submitted runs, we utilize this fact by assuming that if the majority of the elements with the same attribute are of the same type or identified as target entities, all these elements have the same type or are the target entities.

2.4 ENTITY EXTRACTION BY SURFACE TEXT PATTERNS

The power of surface text patterns has been demonstrated in previous TREC QA tracks [1]. In this Entity track, we can define a set of simple templates which describe the relation between target and source entities. For example, we define a template as follows: <TARGET> is <narrative>. If an entity can be extracted from templates, it will be directly placed at the top of the ranked list while other candidate entities are ranked by the three-level relevance retrieval model. After completing the ranked list, the homepage finding procedure is invoked to find the homepages for the entities and the filtering rules are applied to refine the final results.

2.5 ENTITY HOMEPAGE FINDING

In this track, an entity is uniquely identifiable by one of its primary homepages. After extracting the names of related entities, we need to find their homepages. Wikipedia can be directly used to find Wiki homepages as well as primary homepages through the external links on the Wiki pages. To identify other homepages that go beyond Wikipedia, we treat homepage finding as a classification problem by training logistic regression (LR) models for the three target entity types respectively. We use the 11 training topics provided by the track and also select 421 persons, 568 organizations and 216 products from Google Directory[6] for training. The Google Directory lists each entity's homepages. To obtain a set of negative training instances, we input each selected entity in the general Google search and manually label the top 5 returned results. Table 1 contains the features used in logistic regression.

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

It is relatively easy to find the primary official homepage and its Wiki homepage of an entity. For example, if we type the entity name in Google, the official and Wiki homepages (if they exist) usually show at the top of the returned results. However, for the secondary non-wiki homepages, it may rank very low (probably because of their low PageRanks). Therefore, it is time consuming to find and label these secondary descriptive homepages. Motivated by active learning, we first label a small number of secondary homepages (labeled on the top 5 Google returned results) and use the trained homepage classifiers to classify the web pages from a large pool of the retrieved documents. Then we only look at the positively labeled pages, manually classify them and incorporate these pages as the training data to retrain the model. In this way, the true positive instances are more likely to be labeled. As more positive instances are included, the classification performance is expected to improve.

Another issue is the choice of threshold in LR models where 0.5 is usually used as a cut-off probability to determine whether a page is homepage or not. In one run (KMR3PU), we use 0.45 as the threshold in an attempt to include possibly more relevant entities. To adopt a conservative strategy, we forcefully put the newly found entities at the end of the ranked lists.

Table 1: Features used in logistic regression for homepage classification

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

2.6 ENTITY FILTERING

After obtaining a list of ranked entities, we use a set of heuristic rules below to further refine the results.

- We remove the entities that do not have non-wiki homepages
- We experimented to use nearest neighbor clustering to merge the same entities that have different names such as Deborah Estrin and as D. Estrin.
- Limit the length of entity names for different target types (i.e., 3 words for person, 5 words for organization and 8 words for product)
- Remove the target entity names that heavily overlap with the source entity (e.g., remove “BlackBerry Bold” for the test Topic 1)

If there is not enough entities remaining after filtering, the system goes back to retrieve more documents (i.e., taking the top 10 pages as the candidate documents) or refine the query, and repeats the process as in [2].

3. RESULTS

We submitted the following three runs, based on different homepage classification strategies:

- KMR1PU: Run produced using the complete approaches described in the previous sections;
- KMR2PU: Run without iteratively and incrementally learning homepage classifiers (i.e., only training the LR models one time using the collected data from Google);
- KMR3PU: Run with the threshold of 0.45 set in LR models as described in Section 2.5.

Table 2 lists the results for these six runs. From the table, in KMR1PU vs KMR2PU, we can see that by iteratively learning the homepage classifiers, the system could identify more relevant entities and even more primary homepages and thus the nDCG_R value increased. On the other hand, in KMR1PU vs KMR3PU, adjusting the threshold of the LR models did not find more relevant entities and homepages although more entities were retrieved, and as a result, nDCG_R deteriorated a bit.

Table 2: Performance of the three submitted runs. The columns of the table (from left to right) are: whether Wikipedia received a special treatment, whether any external resources were used, the total number of entities retrieved (num_ret), the number of relevant and primary entities retrieved (rel_ret and pri_ret), P@10 scores (P10), and NDCG@R.

	WP	Ext.	num_ret	rel_ret	pri_ret	P10	nDCG_R
KMR1PU	Y	Y	214	126	61	0.2350	0.3061
KMR2PU	Y	Y	192	115	56	0.2350	0.2916
KMR3PU	Y	Y	260	126	61	0.2350	0.3060

4. CONCLUSIONS

Our submitted runs are ranked at the top in P@10 and NDCG@R among all the submissions, which shows the effectiveness of our approaches. There are several key lessons that we have learned by participating this year's entity track. First of all, we found in our experiments that with a small number of documents retrieved, the documents retrieved by Google seem much more likely to contain the related target entities than those retrieved by INDRI do. In consequence, only a small number of candidate documents are considered in our runs, which results in relatively fewer false negative entities. Secondly, our special treatment of table or list data is well rewarding. In the test topics, over half of them have target entities existing in tables or lists. This may reflect the trend in the evolving Web data: more and more entities are stored in a structural form and there is also a lot of effort to convert unstructured data to the structured such as Freebase[7], Factual[8] and Semantic Web. Thirdly, the high accuracy of homepage finding is crucial in this track, because by definition an entity is uniquely defined by its primary homepage, not by its name. Both precision and recall of the homepage classifiers are important in yielding good final results. Fourthly, in our experiments we found Wikipedia can serve as a valuable knowledge resource to verify whether the candidate entities satisfy the property specified in the topic. It is also very useful in locating certain homepages with high accuracy. Last but not the least, engineering effort and heuristic rules can further improve the overall retrieval performance.

ACKNOWLEDGEMENT

Yi Fang and Luo Si have been supported by a research grant from National Science Foundation (IIS-0746830), a research grant from Indiana Economic Development Corporation, and a research grant from Purdue University. Zhengtao Yu, Yantuan Xian and Yangbo Xu are supported by National Nature Science Foundation (No. 60863011) of China and The Key Project of Yunnan Nature Science Foundation (No. 2008CC023) of China. We would like to thank Lina Li, Lei Su, Junjie Zou, Yihao Zhang, Xianming Yao, Chaosheng Zhang, Shaoming Zhang, Jianyi Guo, and Cunli Mao, all of Kunming University of Science and Technology, for their help and assistance for the work. Any opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Ravichandran, D. and Hovy, E. Learning surface text patterns for a question answering system. In: ACL, 2002.

[2] Pasca, M. and Harabagui, S. High performance question/answering. In: SIGIR, 2001.

[7] http://www.freebase.com/

[8] http://www.factual.com/