Searching for Entities: When Retrieval Meets Extraction

Qi Li, Daqing He

School of Information Sciences, University of Pittsburgh
Pittsburgh, Pennsylvania, U. S.
{qili, daqing}@sis.pitt.edu

Abstract. Retrieving entities inside documents instead of documents or web pages themselves has become an active topic in both commercial search systems and academic information retrieval research. Our method of entity retrieval is based on a two-layer retrieval and extraction probability model (TREPM) for integrating document retrieval and entity extraction together. The document retrieval layer finds supporting documents from the corpus, and the entity extraction layer extracts the right entities from those supporting documents. We theoretically demonstrate that the entity extraction problem can be represented as TREPM model. The TREPM model can reduce the overall retrieval complexity while keeping high accuracy of locating target entities. The experiment is based on the document retrieval and entity extraction as well as the overall task. The preliminary results are promising and deserve for further exploration.

Keywords: entity retrieval, document retrieval, entity extraction

1 Introduction

Search engines, returning results as a ranked list of documents, may not provide answers directly to users' information needs, especially when the documents are long. In such situation, entity retrieval, by focusing on finer granularity units called entities, acts as a useful alternative.

Our participation in the TREC 2010 entity track is driven by several research goals. One of the goals is to find entities in an effective and efficient manner. That is, given a query, all identified entities should be ranked in such way that highly relevant entities are ranked above less relevant and non-relevant ones. In order to achieve this goal, we proposed to combine retrieval model and extraction model. Document retrieval model aims at quickly and effectively finding the supporting documents containing the answer entities while extraction model is in charge of correctly extracting the answer entities. By modeling entity retrieval problems as a problem of document retrieval and that of entity extraction, it helps to reduce the complexity of entity retrieval into two separate sub-tasks: document retrieval and entity extraction. Our hypothesis is that entity retrieval can be simplified into two local optimization problems. The advantages of this localized optimization are not only lower the computational complexity, but also to adapt more state-of-the-art techniques in document retrieval and entity extraction into entity retrieval. This model also summaries the previous common workflows in the same task into a theoretical framework. Almost all participants in last year follow the same steps: finding the snippets containing the topic entities and topics, and then extracting the answer entities from them. Here, on one hand, we will demonstrate that this kind of snippets can be constrained on one document or scatter across several pages, and how we can effectively find these snippets. On the other hand, we will state that, for some topics, the identification of the entities can be easily and correctly by ready-to-use named entities identification tools or some trained named entities identification tools with the aid of such knowledge bases as Wikipedia, but, for the other topics, it can also be hard because of the lack of syntactical or semantic signs in the snippets.

2 The Two-layer Retrieval and Extraction Probability Model

The overall architecture of our Entity Retrieval system is shown in Figure 1. The inputs include documents that are HTML pages or plain texts, and information needs represented as search task descriptions and the required entity type. The output is a set of entities with their URIs/URLs. Our Two-layer Retrieval and Extraction Probability Model (TREPM) consists two major components: document retrieval and entity (answer) extraction, which have been widely adopted to build in entity retrieval systems in previous years.

The Document Retrieval Layer is a retrieval engine that aims at finding as many supporting documents as possible. The Entity Extraction Layer tries to identify the entities from those documents by analyzing supporting

patterns inside the documents. The final score, which represents the probability of a candidate entity being the answer, combines document relevance and entity relevance to the topic. The score is then used to rank entities in the set.

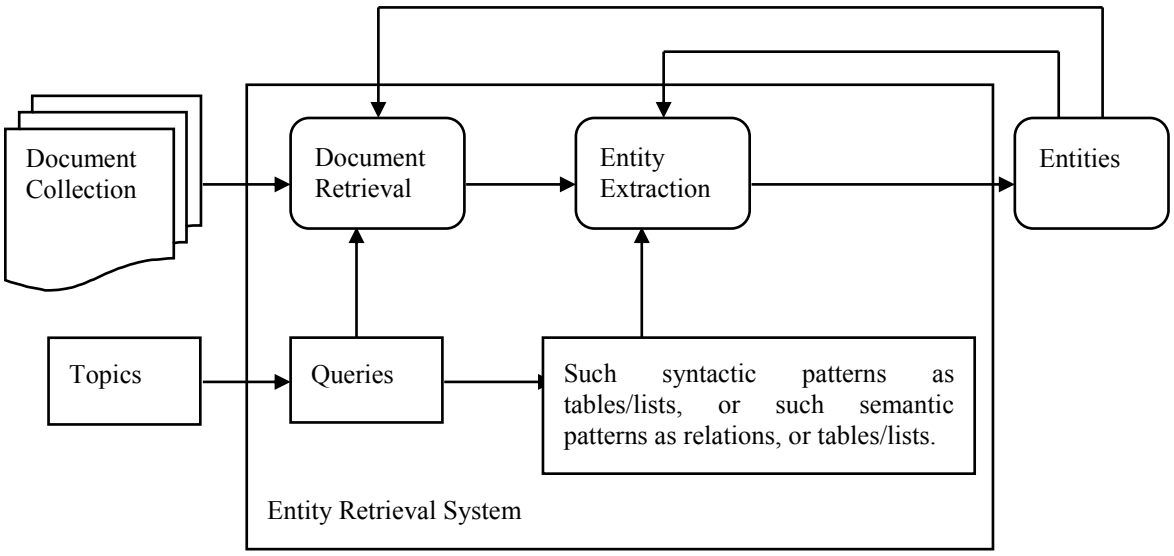

Figure 1: Over all architecture of the Two-layer Retrieval and Extraction Probability Model (TREPM)

Using probability theory, we formalize the entity retrieval question as: the probability of a candidate entity e being the answer entity with target type t given a query q, that is $p(e|q,t)$. Considering the document d and the context c surrounding entity e, $p(e|q,t)$ can be defined as follows Equation 1.

$$\begin{aligned} & p(e|q,t) \\ &= \sum_d \sum_c p(e,d,c|q,t) \\ &= \sum_d \sum_c p(d|q,t)\,p(c|d,q,t)p(e|c,d,q,t) \\ &= \sum_d p(d|q,t) \sum_c p(c|d,q,t)p(e|c,d,q,t) \end{aligned}$$

Equation 1: Probability Model of Entity Retrieval (TREPM model)

Equation 1 by considering the documents and contexts of the entities successfully decomposes the entity retrieval problem $p(e|q,t)$ into two parts, $\sum_d p(d|q,t)$ and $\sum_c p(c|d,q,t)p(e|c,d,q,t)$.

The first part of TREPM model, $\sum_d p(d|q,t)$, is the document retrieval layer. $p(d|q,t_t)$ is the probability that document d is generated by query q with target entity type t. This is to estimate the similarity of a document and a query in the traditional language model. In the general process, the entity retrieval needs to consider all the documents in the corpus and them calculate the probability of the entity generated by the topic. But in the practical time, it is impossible to iterate all the documents to calculate the entity generative probability, so that we apply the heuristic method of only considering the supporting documents $d_{supporting}$ for entity generative, that is, $\sum_d p(d|q,t) \approx \sum_{d_{supporting}} p(d_{supporting}|q,t)$. The output of this step is the supporting documents which we hope with topic entities mentioned in the topics and the answer entities co-occurring in the same document. Here we use "supporting" documents instead of "relevant" documents because we want to emphasis that final answers are the entities to be extracted from these documents. Moreover, we need to note that this kind of homepages may or may not be the same as the homepage of answer entity. For example, Topic 5 in TREC is what are "Products of Medimmune, Inc." The answer entities are Synagis, FluMist, and Ethyol. The homepages are such as http://www.ethyol.com/ which is also a supporting document since it includes the sentence of "ETHYOL® is a registered trademark held by MedImmune, LLC, a member of the AstraZeneca group of companies" with topics and answers occurring in the same page. But Topic 49 in TREC 2010 asks

"What countries does Eurail operate in?" Its answer entities are Austria, Italy, Germany, etc. The homepage for these entities are http://www.germany-tourism.de/, http://www.france.com/, etc. But the valuable homepage can be the introduction of Eurail about the countries it pass (http://www.eurail.com/eurail-global-pass?currency=eur). In this case, the supporting documents are quite different from the answer entity homepage.

The second part of TREPM model is $\sum_c p(c \mid d,q,t)p(e \mid c,d,q,t)$ representing entity extraction. Entity extraction can also be view as two parts. $p(c \mid d,q,t)$ is the context generation probability. That is the generative probability of the context c in a given the document d with the query q and the target entity type t. The second quantity $p(e \mid c,d,q,t)$ is the entity extraction probability in special context. It is the probability that a candidate entity e is the answer, given the context c in document d with the query q and the target entity type t. The same reason as mentioned in document retrieval, it is impossible to iterate all the contexts, so that we only consider the most effective contexts, $c_{supporting}$, for extraction task. Therefore, we have $\sum_c p(c \mid d,q,t)p(e \mid c,d,q,t) \approx \sum_{c_{supporting}} p(c_{supporting} \mid d,q,t)p(e \mid c_{supporting},d,q,t)$. Here, the context $c_{supporting}$ can be both syntactical and semantic context. Examples of the syntactical context can be the tables of InfoBox in Wikipedia with a field indicating products for Company page, whereas the example of the semantic context can be sentences like "ETHYOL® is a registered trademark held by MedImmune, LLC". Both examples can be the context of the product, "ethyol", and the company, "MedImmue, LLC".

In summary, TREMP considers the relevance between entities and the topics at two layers: document and context. In order to retrieval the target entities, a retrieval system will rank all the candidate entities by summing up the scores of all combination of all documents and all candidate context.

3. Evaluation

TREPM model assumes the entity retrieval task returned the entities as answer, but the TREC entity task required the homepage of entities as answers. As part of development of our algorithm, we evaluated each step in TREMP model which includes document retrieval and entity extraction as well as homepage identification (see Figure 2). Here, we treat homepage identification as an extra step in TREC entity task which is not covered in the TREPM model.

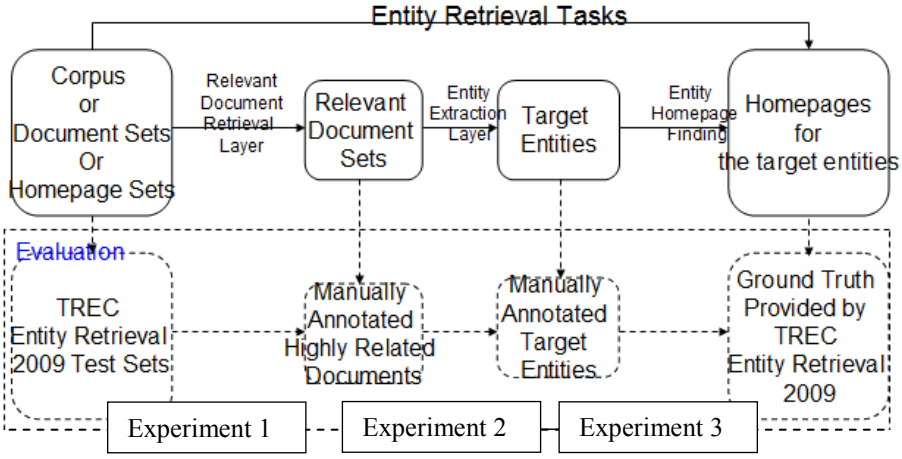

Figure 2: Evaluation of TREMP in three levels: documents retrieval, entity retrieval and homepage finding

To evaluate each step, two annotators manually marked up the ground truth sets for supporting document sets, relevant entities sets and answer entity homepage sets. The requirement for supporting documents markup is to find at least one supporting document which can provide the answers for the topics. If there are any Wikipedia articles containing the answers, they are also required to be marked up. The steps for supporting document annotations are as follows: first annotators generate proper queries to search on search engine to find the possible supporting documents, and then according to the rank retuned by the search engine, two annotators

evaluate if the hit can support further the answer entity extraction. Every topic must find at least one supporting document; and if there are more than one supporting document found, annotators only judge the first 10 hits.

Entity ground truth sets were marked up by two annotators based on the supporting document sets. Entities can include various formats without merging all synonyms into a standard format. For example, the answer entities for "the campus of Indiana University" can be Indiana University East or IU East which are both seen in our supporting document sets.

The requirement for homepage identification annotation is to find the homepage of the entities according to the ground truth from last year. If there is no website can be claimed as homepage for this entity, the focus pages from news or other websites were marked up as the homepage.

The goal of the document retrieval level is to find as many relevant supporting documents as possible. The supporting documents here mean the documents contain potential target entities and can use for further extractions. In the entity extraction level, the task is to extract the answer entities from the supporting documents. The homepage identification is to locate the homepages for those identified target entities. For example, Topic 5 asks about "Products of Medimmune, Inc.". The supporting documents can be homepages that introduce the products of the company (such as a page from Medimmune website: http://www.medimmune.com/about_us_products.aspx), or a Wikipedia page that is the entry of Medimmune Inc. (i.e., http://en.wikipedia.org/wiki/MedImmune). From these related supporting homepage, three products -- Synagis, FluMist and Ethyol -- can be identified as the answers for this topic. The homepages for the answer sets of Synagis, FluMist and Ethyol http://www.synagis.com/, http://www.flumist.com/ and http://www.ethyol.com/ respectively.

3.1 EXPERIMENT 1: DOCUMENT RETRIEVAL

We first evaluated document-level retrieval, finding the highly supporting documents which contain the answer entities for further extractions. In previous works, people use various methods to find the relevant documents or snippets. Although some teams treat the document retrieval as normal tasks using BM25 (Zhai, Cheng, Guo, Xu, & Liu, 2009), language model (Wu & Kashioka, 2009), some teams use structured retrievals. (Fang, Si, Yu, Xian, & Xu, 2009) applies the structured retrieval on document, passage and entity level; (McCreadie, Macdonald, Ounis, Peng, & Santos, 2009) applies the same idea on webpage title and body level; (Zheng, Gottipati, Jiang, & Fang, 2009) also uses the language model but on document and snippet (50 words) level. Some other teams consider the query constructions to refine the queries issued to search engine. For example, (Vinod Vydiswaran, Ganesan, Lv, He, & Zhai, 2009) tries to identify the information need as a structured query to be represented as a relationship which includes the relation terms, the entity of focus, and the entity of interest, and the task is to parse the narrative to a format of relations. (Yang, Jiang, Zhang, & Niu, 2009) also does some query constructions by adding the synonym of topic entities into the query for search.

Although many teams used the advanced document retrieval methods, in our annotating the ground truth of the supporting documents, when we did the extraction we found the supporting documents for all 20 TREC 2009 topics by only searching on search engines with the proper queries. We, therefore, in this experiment use yahoo search engine to retrieval the Web with queries from the narrative of topics. We evaluate whether we can use the simple queries for search engine to find the supporting documents, and how many numbers of the results are the proper number for finding the supporting documents in order to further extraction task. That is the first part of the Formula 1. We test top 100 results return from Yahoo, and compare the precision, recall and F-measure. The results are in Table 1.

Table 1: Precision, Recall, and F-measure for Yahoo search for supporting documents

Rank	P	R	F	Rank	P	R	F
100	0.008	0.416667	0.015668	10	0.06	0.3	0.098834
90	0.008333	0.391667	0.016281	9	0.066667	0.3	0.107727
80	0.009375	0.391667	0.018264	8	0.075	0.3	0.118384
70	0.010714	0.391667	0.020796	7	0.085714	0.3	0.131389
60	0.0125	0.391667	0.024144	6	0.1	0.3	0.147619
50	0.015	0.391667	0.028776	5	0.1	0.258333	0.141667
40	0.01875	0.391667	0.035609	4	0.125	0.258333	0.165238
30	0.025	0.391667	0.046698	3	0.166667	0.258333	0.198333
20	0.0375	0.391667	0.067824	2	0.225	0.233333	0.223333

	1	0.25	0.116667	0.158333

With the score of F-measure, the top two documents are the most valuable documents, when using Yahoo as a way to search for the supporting document. The simple search only can finds small part of supporting documents. Our annotation results for supporting documents shows that almost half of the topics have a Wikipedia page as one of the supporting documents which means Wikipedia is a good source for extracting the targeted entities. In the future work, we will consider use Wikipedia as source for part of topics search. With the recall at 100 (0.42), we found that this approach can only find less than half of the supporting documents. We compare two types of queries: queries generated by topic entities (e.g., Medimmune, Inc.) and queries generated by descriptions (e.g., products of Medimmune, Inc.). There are 9 topics (out of 20) which should generate the supporting document sets from description-as-queries; there are 4 topics (out of 20) which should generate the supporting document sets from topic-entities-as-queries; there are another 7 topics which can generate the supporting document sets from either topic-entities-as-queries or description-as-queries. Why are entity retrieval topics various on the query sources? The reason for no difference between topic-entities-as-queries description-as-queries is that the descriptions for topic entities fail. In this case, the descriptions of the topics are the entity type requirements for answer entities which usually do not appear in the documents. For example, Topic 3 is "students of Claire Cardie". If there is no special webpage indicating this entity, the system will fail at finding the supporting documents, and the results for query "students of Claire Cardie" will be the same as the results for query "Claire Cardie". The reason that topic-entities-as-queries runs better than description-as-queries is that the descriptions hurt search results. For example, Topic 6 is "organizations that award Nobel prizes". The description (e.g., "organizations that award") for topic entity (e.g., "Nobel prizes") can cause the error results from the similar concepts (e.g., "Nobel prize awarded organizations"), especially when there is no special pages that discuss about the target entities. Except for the query sources, there are some cases at failing representing the relations between topic entities and answer entities. For example, Topic 58 is "What are some of the spin-off companies from the University of Michigan?" The narrative part to represent the relation between the topic entity (i.e., the University of Michigan) and the answer entities is "spin-off companies", which will be more effective when using their synonyms of "spun of/from/of from". These analyses also confirm that document-retrieval-in-ER can only deal with term co-occurrence problems since the retrieval model is built on the assumption of bag-of-words (i.e., the independence of the terms in the document). All the semantic related analyses, therefore, should be postponed into entity-extraction-in-ER. For some topics, it is difficult to generate the proper queries to find a supporting document.. How can we include more supporting document will be our further work.

3.2 EXPERIMENT 2: ENTITY EXTRACTION

Most of the work in TREC 2009 reports use named entity recognition tools to identify the entities. Stanford NER tool is the most popular one, but unfortunately it can only identify the named entity of "People" and "Organization", but not "Products". Therefore, such teams as (Yang, Jiang, Zhang, & Niu, 2009) and (Serdyukov & de Vries, 2009) use external knowledge base like Wikipedia to train named entity tool by expanding the product lists. The same as (Vinod Vydiswaran, Ganesan, Lv, He, & Zhai, 2009) and (McCreadie, Macdonald, Ounis, Peng, & Santos, 2009) relies on a dictionary of company names and a pre-defined set of patterns for product recognition. The team like (Zheng, Gottipati, Jiang, & Fang, 2009) follows the same idea but treat proper nouns as candidate product entities. Most of teams do further entity re-ranking since the results directly from the named entity recognition are not promising. (Wu & Kashioka, 2009) specially evaluates the re-ranking process by calculating the similarities between input query, supporting snippets, and related entities. Such work as (Fang, Si, Yu, Xian, & Xu, 2009) also mentions structured of tables and lists can facilitate the entity extraction.

Experiment 2 is to evaluate the entity extraction level. That is, how can we find the answer entities given the supporting documents? We also adapt the basic approaches from last year's work for entity extraction, using the named entities tools to identify the entity of Organization and Person, and use noun phrases as Products. The corpus of supporting documents is preprocessed by removing all html tags. The results of precision, recall and F-measure are as follows in Table 2.

Named entity tools are critical in this step since the results of Organization and Person are much better than Product. Precision of Products is only 0.01 which means treating noun phrases as the products will bring too many noise. Even with some trained data and rules for Organization, the precision is also very low. We consider use Wikipedia as entities repository to filter out the non-entities.

Entities extracted from the home pages are better than the ones from Wikipedia pages. Here we notice that we simply treat all the webpages and Wikipedia pages as HTML pages and remove the HTML tags for them. In fact, however, there are still a lot of non-relevant contains in the same page. For example, in the Wikipedia

pages, there are some category information in the bottom and language information in the left which will be the noise for entity extraction. Therefore, we also write a simple parser which removes the head and foot parts of the Wikipedia. The experiment is to evaluate if removing the noise in the context can improve the results. The results show that the overall precision and recall rise to 0.103 and 0.44 respectively, and the overall F-measure improves (0.144). We can see that narrowing down the context and removing the noise does help to improve the results.

The previous results are based on each document. If we consider the entities by topics, that is, summary the entities across the document in the same topic, we find that precision improves (0.17), although recall drops (0.37). The F-measure still improves (0.16).

Table 2: Precision, Recall and F-measure of Entity Extraction from Supporting Documents

	Precision	Recall	F-measure
Overall	0.103	0.419	0.144
Product	0.012	0.2959	0.023
Person	0.248	0.546	0.337
Organization	0.077	0.411	0.111
Homepage	0.114787	0.369279	0.155062
Wikipage	0.083	0.5204	0.1269

We further investigate the contexts of these answer entities in the supporting documents. The structures of the context include the sentences, lists, or tables in the web pages. For example, in the Wikipedia page of "Medimmune, Inc.", the table of Infobox contains the answer for the product of "Medimmune, Inc." In the web page about the "Product of Medimmume, Inc.", there is a product list which presents the answer for this topic. Also the sentence, "It (MedImmune, LLC) produces Synagis, a drug for …", in Wikipedia, also use another way to indicate the product of MedImmune. The results are in Table 3. We find that most entities from these supporting documents are in the tables or lists, and only few of them are in the sentences. This means the current named entity tools which are trained by the corpus maybe fail at identifying the entities from tables and lists. Also as we reviewed out supporting documents ground truth set, we found that the numbers of ground truth entities across 20 topics are various among different topics, so that it will be difficult to use a simple threshold to limit the number of answer sets. Most of the entities exist in tables and lists, and we have much more work for correctly identifying them. Another observation is that for the most topics the answer entities concentrate in one document/page appearing in tables or lists or sentences, but there are several topics where the entities scatter in several pages. In this case, topics with answers in one page can comparatively easier than the ones with answer entities in several pages. How to correctly find all the supporting documents is another key problem.

Table 3: the Context Structure of the Entities

# of Topics	*Wikipedia*		*Web Page*		*Sentences*
	Table	List	Table	List	
In one page	7	6	3	30	5
In several pages				2	3

3.3 EXPERIMENT 3: HOMEPAGE IDENTIFICATION

In our review the work on entity homepage finding, there are three approaches, relying on search engines, building a classifier, and mining from knowledge base. The work like (Vinod Vydiswaran, Ganesan, Lv, He, & Zhai, 2009) is to build a structured index with more weights on title and headline fields, and then retrieve on the index to find the homepage for the entities. Some works such as (McCreadie, Macdonald, Ounis, Peng, & Santos, 2009) and (Kaptein & Kamps, 2009) use Wikipedia or DBpedia to extract homepages for the target entities. The third method is to build a classifier for homepage identification, such as logistic regression in (Yang, Jiang, Zhang, & Niu, 2009) and (Fang, Si, Yu, Xian, & Xu, 2009). In this study, we also adapted the classification method and features presented in (Fang, Si, Yu, Xian, & Xu, 2009) to train a classifier for homepage identification, but the result was not very promising. The results from JRIP using WEKA indicate that the rules are similar to the top results from search engines. Therefore, in the follow up experiment, we focused on how many results from search engines can be the homepage of entities. We relied on the Yahoo boss to search entities and find the homepage for the entities. The results are in Table 4. In the finally submission version, the homepages of the entities are from the heuristic rule: if we can find the homepage link from

corresponding Wikipedia entity homepage, we will use them as homepage; otherwise, we will use the first hit from search engine (Yahoo!Boss) as homepages.

Table 4: Precision, Recall and F-measure for Homepage

	# of correct entity	# of ground truth	# of entities hits	Precision	Recall	F-measure
Top 5	53	167	5760	0.058	0.386	0.076
Top 4	50	167	4554	0.07	0.35	0.090
Top 3	50	167	3401	0.094	0.348	0.082
Top 2	50	167	2286	0.1376	0.348	0.1103
Top 1	45	167	1168	0.21	0.3	0.13

The homepage identification by search engines can only find one fifth homepage. Although knowledge bases such as Wikipedia can also provide the answer for another one third, there are still more than half entities that couldn't find the homepages for them. One of the reasons for homepage identification failure is that the identical entities can be represented as different text surfaces. For example, both Indiana University East and IU East can represent the same entity which can be referred to the same homepage, http://www.iue.edu. In some cases, the abbreviation format of the entities will cause the difficulty of homepage identification. Another difficult is from the definition of the homepage. Some entities only have webpages to describe them. For example, we analysis the homepage sets for Topic 5 "Products of Medimmune, Inc.", as shown in Table 5. The homepage of a product can be news, or product-related company's homepage, or the product's introduction page from company, or products homepage. In this case, it will be hard to define the homepage for some entities, such as country.

Table 5: Entity Homepage Set for Topic 5 "Products of Medimmune, Inc."

Docno	URL	Type of the URL
clueweb09-en0000-27-129352	http://baltimore.bizjournals.com/baltimore/stories/2009/01/05/daily20.html	News
clueweb09-en0006-42-198412	http://www.ethyol.com/	Products Homepage
clueweb09-en0006-41-111382	http://www.flumist.com/	Products Homepage
clueweb09-en0008-26-393002	http://www.medimmune.com	Company Homepage
clueweb09-en0008-26-393062	http://www.medimmune.com/about/history.asp	Company Introduction page
clueweb09-en0008-26-393262	http://www.medimmune.com/products/ethyol/index.asp	Company Introduction page
clueweb09-en0008-26-393282	http://www.medimmune.com/products/flumist/index.asp	Company Introduction page
clueweb09-en0008-26-393302	http://www.medimmune.com/products/synagis/index.asp	Company Introduction page

4 SUMMARY

This work summaries our study of the TREMP model in TREC entity retrieval. Our idea is to decompose entity retrieval problem into a document retrieval problem and entity extraction problem. In the procedure of supporting document annotation and entity answers annotation, we have some interesting findings. The supporting documents usually include all answer entities to a topic in one document. Only in few cases, we need to collect the answers one by one. Wikipedia is an important source for the answer sets since it can provide the answers for about half of the topics. Although we annotated the ground truth supporting documents for all the topics with only retrieval on search engines, according to our experiment 1 result, the simply query on search engine can only find small part of the supporting documents. Therefore, how to improve the supporting document retrieval will be the interesting topic. Entities appearing in the supporting documents can be in various contexts. The different entity surfaces also cause the difficulties to find the homepage for them further. In the comparing the entities with the homepage of the entities provided by TREC, we can there are big gaps between them. The homepages can include the website with name of "homepage", or the news or articles concentrating on some special entities. Therefore, we would like to argue that we need to have further understanding of the definition of homepage. In the analysis of the two-layer model, we find that some topics are easy to find the supporting documents while some have the answers scatting across several documents which we note them as retrieval-hard topics. For some topics, the answer entities are in the HTML page with few hints to indicate it as answers which we note them as extraction-hard problem. In the future study, we will focus on how to identify retrieval-hard topics and extraction-hard topics, how to improve the search results on retrieval-hard topics, and how to improve the extraction result for extraction-hard topics.

References

Balog, K., & de Vries, A. P. (2009). Overview of the TREC 2009 Entity Track. *TREC 2009*.

Craswell, N., Demartini, G., Gaugaz, J., & Iofciu, T. (2009). L3S at INEX 2008: Retrievaing Entities Using Structured Information. *INEX 2008*.

Fang, Y., Si, L., Yu, Z., Xian, Y., & Xu, Y. (2009). Entity Retrieval with Hierarchical Relevance Model,Exploiting the Structure of Tables and Learning Homepage Classiers. *theEighteenth Text REtrieval Conference (TREC 2009)*. Gaithersburg, MD.

Jiang, J., Lu, W., Rong, X., & Gao, Y. (2009). Adapting Language Modeling Methods for Expert Search to Rank Wikipedia Entities. *INEX 2008*.

Kaptein, R., & Kamps, J. (2009). Finding Entities in Wikipedia using Links and Categories. *the 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2008). 5631*. Heidelberg: Springer.

Koolen, M., Kaptein, R., & Kamps, J. (2010). Focused Search in Books and Wikipedia: Categories, Links and Relevance Feedback. *INEX 2009*.

McCreadie, R., Macdonald, C., Ounis, I., Peng, J., & Santos, R. L. (2009). University of Glasgow at TREC 2009: Experiments with Terrier. *the Eighteenth Text Retrieval Conference (TREC 2009)*. Gaithersburg, MD.

Rode, H., Hiemstra, D., de Vries, A., & Serdyukov, P. (2009). Efficient XML and Entity Retrieval with PF/Tijah: CWI and University of Twente at INEX 2008. *the 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2008). 5631*. Heidelberg: Springer.

Serdyukov, P., & de Vries, A. (2009). Delft University at the TREC 2009 Entity Track: Ranking Wikipedia Entities. *the Eighteenth Text REtrieval Conference (TREC 2009)*. Gaithersburg, MD.

Vercoustre, A.-M., Pehcevski, J., & Naumovski, V. (2009). Topic Difficulty Prediction in Entity Ranking. In Proceedings of the 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2008). *5631*. Heidelberg: Springer.

Vercoustre, A.-M., Pehcevski, J., & Thom, J. A. (2008). Using Wikipedia Categories and Links in Entity Ranking. *the 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007). 4862*. Heidelberg: Springer.

Vinod Vydiswaran, V., Ganesan, K., Lv, Y., He, J., & Zhai, C. (2009). Finding Related Entities by Retrieving Relations: UIUC at TREC 2009 Entity Track. *the Eighteenth Text REtrieval Conference (TREC 2009)*. Gaithersburgh, MD.

Wu, Y., & Kashioka, H. (2009). NiCT at TREC 2009: Employing Three Models for Entity Ranking Track. *the Eighteenth Text REtrieval Conference (TREC 2009)*. Gaithersburg, MD.

Yang, Q., Jiang, P., Zhang, C., & Niu, Z. (2009). Experiments on Related Entity Finding Track at TREC 2009. *TREC 2009*.

Zhai, H., Cheng, X., Guo, J., Xu, H., & Liu, Y. (2009). A Novel Framework for Related Entities Finding: ICTNET at TREC 2009 Entity Track. *TREC 2009*.

Zheng, W., Gottipati, S., Jiang, J., & Fang, H. (2009). UDEL/SMU at TREC 2009 Entity Track. *TREC 2009*.