NiCT at TREC 2010: Related Entity Finding

Youzheng Wu, Chiori Hori, Hisashi Kawai
Spoken Language Communication Group, MASTAR Project
National Institute of Information and Communications Technology (NiCT)
2-2-2 Hikaridai, Keihanna Science City, Kyoto 619-0288, Japan
{youzheng.wu, chiori.hori, hisashi.kawai}@nict.go.jp

Abstract

This paper describes experiments carried out at NiCT for the TREC 2010 Entity track. Our studies mainly focus on improving the NE Extraction and Ranking Entity modules, both of them play vital roles in Related Entity Finding system. In our last year's system, only a Named Entity Recognition tool is used to extract entities that match coarse-grained types of target entities such as organization, person, etc. In this year, dependency tree-based patterns learnt automatically are employed to filter out entities that do not match fine-grained types of target entities such as university, airline, author, etc. In the Entity Ranking part, we propose a dependency tree-based similarity method and incorporate homepage information to improve ranking.

1 Introduction

The TREC Related Entity Finding (REF) track is defined as follows:

> Given an input entity, by its name and homepage, the type of the target entity, as well as the nature of their relation, described in free text, find related entities that are of target type, standing in the required relation to the input entity.

The goal of the entity track is to perform entity-oriented search tasks on the Web [2]. In this year, target entity types are only limited to organization, people, location, and product.

2 Architecture

The NiCT's participant system demonstrated in Figure 1, is a cascade of the following five components.

⊡ The *Relevant Page Retrieval* extracts keywords from $entity\_name$ and $narrative$ fields to retrieve some related Web pages or documents. In implementation, we first employ Yahoo BOSS API (`http://developer.yahoo.com/search/boss/`) to search Web pages from the Web and then map them to documents in the ClueWeb09 test collection. Because one lesson from the TREC 2009 is that commercial search engines such as Google and Yahoo are generally superior in locating supporting documents to the search engine we built using Indri.

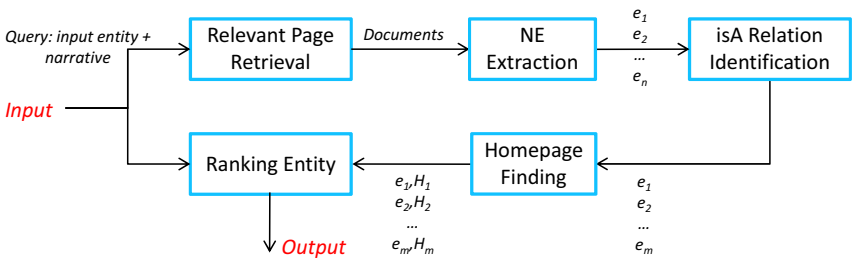

Figure 1: Architecture of NiCT REF System.

⊡ The *NE Extraction* extracts entities that match the given target type from the retrieved documents, which is supported by a NER tool developed by the Cognitive Computation Group at UIUC (`http://l2r.cs.uiuc.edu/~cogcomp`). Particularly, phrases/words tagged with PER, ORG, LOC and MISC tags are extracted when target entities are person, organization, location, and product, respectively. A list of entities $\{e_1, e_2, ..., e_n\}$ is generated.

The *Relevant Page Retrieval* and *NE Extraction* modules are implemented similar to other participant systems. Our studies mainly focus on the other three modules that will be presented in section 3, 4, and 5, respectively.

3 IsA Relation Identification

The *NE Extraction* can only extract coarse-grained types of entities such as organization, location, etc. However, users' queries sometimes require fine-grained types of entities such as airline, university, actor, etc. On the other hand, many incorrect entities are extracted by the NER tool. Main reason lies in: the NER tool is trained on newspaper, however, we use it to tag web data. Therefore, it is necessary to filter out entities that do not match fine-grained entity types.

The *IsA Relation Identification* is designed to filter out entities that does not match fine-grained entity types using dependency tree-based patterns. For example, this module can hopefully remove the extracted entities that are not airlines for the TREC 2009 test question: Airlines that currently use Boeing 747 planes.

At offline phase, the dependency tree-based patterns are learnt via the following steps.

- Extracting Yago IsA relation examples as training pairs. Here are some instances, $\langle Michael\ Schumacher, driver\rangle$, $\langle Vientiane\ Times, newspaper\rangle$, etc. For simplicity, $Michael\ Schumacher$, $Vientiane\ Times$ are called entities, $driver$ and $newspaper$ are called fine-grained entity types.
- For each pair,
  1. Composing query by combining words in pair and retrieving Yahoo snippets for the pair from the Web.
  2. Parsing snippets using Lin's dependency parser, Minipar (`http://webdocs.cs.ualberta.ca/~lindek/minipar.htm`).
  3. Extracting shortest dependency path between entity and its fine-grained type.
- Choosing the extracted pathes with high frequencies as IsA Relation patterns. Figure 2 shows two example patterns.

At online phase, the following steps are conducted.

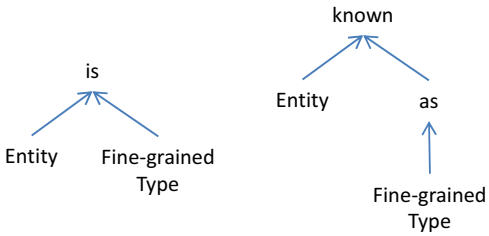

Figure 2: Dependency tree-based patterns.

- Recognizing fine-grained type of target entity from $narrative$ field according to our heuristic rule: head of first non-stop noun phrase is fine-grained type. For example, $gallery$ in "What art galleries are located in Bethesda, Maryland?" is the fine-grained type.
- For each extracted entity in the *NE Extraction* module,
    1. Composing query by combining entity and the fine-grained type to retrieve Yahoo snippets from the Web.
    2. Parsing snippets using Lin's Minipar.
    3. Removing entities for which no IsA Relation patterns appear in the retrieved Yahoo snippets.

Finally, a list of remained entities $\{e_1, e_2, ..., e_m\}$, where, $(m \leq n)$, is generated.

4 Homepage Finding

To find entity homepage, a binary SVM classifier is trained. The DBpedia Homepages data[1] is used as positive examples, which contains a set of $\langle entity, homepage \rangle$ pairs. Simply, $\langle entity, 50thpage \rangle$ pairs are regarded as negative, $50thpage$ denotes the 50th page returned by Yahoo API for query $entity$.

The classification features used include:

- URL-type features: URLs are classified into four categories [3][5], i.e., *root*, e.g. http://www.nict.go.jp; *subroot*, e.g. www.nict.go.jp/research; *path*, e.g. http://www2.nict.go.jp/x/x151/; *file*, e.g. http://www.nict.go.jp/research/network-e.html.
- BINARY features: whether URL contains URL-specific-keywords (such as index, default, main); whether title of page contains homepage-specific-keywords (such as home, homepage, official, main); whether page contains homepage-specific-keywords; whether meta data of page contains homepage-specific-keywords; whether URL contains variants of entity;
- OTHER features: number of characters such as question marks, underscores, etc in URL; ratio of entity words in URL; ratio of entity words in title of page; ratio of title words in entity, pagerank score.

At online stage, we first retrieves top 10 pages via Yahoo API for each entity e_i, and then match them to documents in the ClueWeb09 test collection. Lastly, we employ the trained SVM model to find the homepage H_i of the entity e_i. The module outputs a list of $\langle e_i, H_i \rangle$ pairs, $i = 1, ..., n$.

[1] `http://wiki.dbpedia.org`

In many TREC 2009 participants' systems, the *Homepage Finding* module follows the *Ranking Entity* module. Our TREC 2010 system, however, reversely connects them, which makes the Ranking Entitiy use homepage information.

5 Ranking Entity

Our system regards the ranking task as a problem of estimating the probability $p(e_i, H_i|Q)$ of generating a related entity e_i and its homepage H_i given input query Q, which can be modeled by,

$$p(e_i, H_i|Q) \quad = \quad p(e_i|Q) \times p(H_i|Q) \tag{1}$$

where, e_i is independent of H_i.

$$p(H_i|Q) \quad = \quad \frac{p(Q|H_i) \times p(H_i)}{p(Q)} \propto p(Q|H_i) \tag{2}$$

$$p(e_i|Q) \quad = \quad \sum_{D_i} p(D_i|Q) * p(e_i|D_i, Q) \tag{3}$$

where, D_i represents a supporting sentence of entity e_i.

Combining Equation (1), (2) and (3), we can get,

$$p(e_i, H_i|Q) \quad = \quad \sum_{D_i} p(D_i|Q) * p(e_i|D_i, Q) \times p(Q|H_i) \tag{4}$$

To compute $p(D_i|Q)$, a dependency tree-based similarity algorithm is proposed, which consists of the following steps.

1. Parsing $entity\_name$ and $narrative$ field of input query and supporting sentence D_i into dependency trees using Minipar, i.e, T_q, T_s.
2. Representing trees in terms of their substructures/subtrees, any nodes along with all its children. Here, DP_q and DP_s represent set of sub-trees of input query, and set of sub-trees of a supporting sentence, respectively.
3. Calculating similarity between trees using Equation (5).

$$p(D_i|Q) \quad = \quad \frac{DP_q \cap DP_s}{\sqrt{|DP_q| \times |DP_s|}} \tag{5}$$

To compute $p(Q|H_i)$, BM25 is used. In some cases, homepages such as Michael Schumacher's homepage (`http://www.michael-schumacher.de/`) do not contain any valuable information. Thus, we retrieve snippets HS_i from the homepage site using Q as query.

$$p(Q|H_i)' \quad = \quad \gamma \times p(Q|H_i) + (1 - \gamma) \times p(Q|HS_i) \tag{6}$$

Due to time constraints, $p(e_i|D_i, Q)$ is set to 1. In related studies, proximity model [4] is employed.

6 Experiments

6.1 Submitted Runs

Four runs are submitted for the TREC official evaluation. The configurations are listed in Table 1. The values of $p(Q|H_i)$ are set to 1 in the RUN-1 and the RUN-2, which means that homepage information is not used for ranking. To understand the contribution of the *IsA Relation Identification*, the RUN-1 and the RUN-3 do not incorporate it, while the RUN-2 and the RUN-4 do.

	RUN-1	RUN-2	RUN-3	RUN-4
Value of $p(Q\|H_i)$	1	1	BM25	BM25
IsA Relation Identification	not used	used	not used	used

Table 1: Configurations in the four RUNs

6.2 Official Results

Table 2 lists the results for the four runs. Here, *Best* and *Median* mean the best and the median scores among all participants' systems, respectively. Figure 3 demonstrates nDCG@R score for each of test queries.

	RUN-1	RUN-2	RUN-3	RUN-4	*Best*	*Median*
nDCG@R	.1237	.1245	.1696	.1655	≈ 0.38	≈ 0.12
P@R	.909	.991	.1453	.1446	-	-
MAP	.647	.703	.953	.971	-	-
P@10	.894	.1064	.1447	.1574	-	-
pri_ret/rel_ret	150/85	143/76	187/74	174/64	-	-

Table 2: Official results of the submitted runs. pri_ret means the number of primary homepages retrieved, rel_ret means the number of relevant pages retrieved, R means the number of primary and relevant homepages for a query.

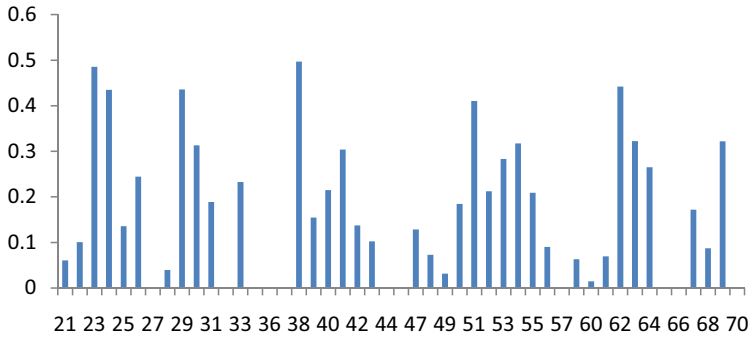

Figure 3: nDCG@R scores for each of test queries.

This experimental results indicate that: (1) 11 out of 47 queries got zero. The problem might be from the Page Retrieval and/or the NE Extraction modules. In addition, errors from the Homepages Finding module also account for a certain proportion, e.g., homepages of countries can not be identified in our system. (2) The IsA Relation Recognition can slightly improve P@10 and MAP scores by filtering out noise entities, which, however, wrongly removes correct entities in some cases indicated by pri_ret scores. (3) Homepage information can greatly improve the REF system, e.g., the largest nDCG@R and P@10 improvements are 4.6%, and 5.6%, respectively.

7 Conclusion

This paper describes NiCT's participant system for the TREC 2010 Entity track. Given input query, extracted entities and their supporting sentences, we mainly focus on improving quality of extracted entities by removing noise from entities, and computing similarity between input query and supporting sentences of entities.

The official evaluation results indicate: Homepage information can greatly improve the REF system, while, the enhancement from the IsA Relation Recognition is not significant. In future study, we aims at improving the IsA Relation Recognition and recall of the NE extraction via mining tables and lists in pages.

References

[1] Youzheng Wu, and Hideki Kashioka. NiCT at TREC 2009: Employing Three Models for Entity Ranking Track. *In Proc. of TREC 2009*.

[2] Krisztian Balog, Arjen P.de Vries, Pavel Serdyukov, Paul Thomas, Thijs Westerveld. Overview of the TREC 2010 Entity Track. *In Proc. of TREC 2010*.

[3] Yi Fang, Luo Si, Zhengtao Yu, Yantuan Xian, Yangbo Xu. Entity Retrieval with Hierarchical Relevance Model. *In Proc. of TREC 2009*.

[4] Tao Tao, and ChengXiang Zhai. An Exploration of Proximity Measures in Information Retrieval. *In Proc. of SIGIR-2007*, Amsterdam, The Netherlands, 2007.

[5] Trystan Upstill, Nick Craswell, and David Hawking. Query-Independent Evidence in Home Page Finding. *In ACM Transaction on Information Systems*, Vol.21, No.3, July 2003, Pages 286-313.

[6] Oren Etzioni, Michael Cafarella, Doug Downey, etc. Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. *In Proc. of the 19th national conference on Artifical intelligence*, pp391–398, California, 2004.