PRIS at TREC2013 Knowledge Base Acceleration Track

Chunyun Zhang, Weiran Xu, Ruifang Liu, Weitai Zhang, Dai Zhang, Janshu Ji, Jing Yang

Abstract:

This paper details the participation of Pattern Recognition and Intelligent System lab of BUPT in CCR and SSF task of TREC 2013 Knowledge Base Acceleration track. In the CCR task, The PRIS system focuses attention on query expansion and similarity calculation. The system uses DBpedia as external source data to do query expansion and generates directional documents to calculate similarities with candidate worth citing documents. In the SSF task, The PRIS system utilizes a pattern learning method to do relation extraction and slot filling. Patterns of regular slots which are same to TAC-KBP slots are learned from KBP Slot Filling corpus. Other slots are filled by following some generalized patterns learned from external source data including homepages of some famous people and facilities. Experiments show that the CCR system gives a good performance above the median value. The pattern learning method for SSF task gives an outstanding performance.

1. Introduction

The goal of KBA track is filtering a large stream of text to find documents that can help update a knowledge base like Wikipedia. The KBA2013 includes two tasks: CCR task and SSF task. For the CCR task, given a fixed list of target entities from Wikipedia and Twitter, systems should filter documents worth citing in a profile of the entity. For the SSF task, given a slot for each of the target entities, systems should detect changes to the slot value, such as location of next performance or founder of an organization. Our team participated in both of tasks.

The PRIS-CCR system focuses attention on query expansion and similarity calculation. The PRIS-SSF system utilizes a pattern learning method to do relation extraction and slot filling. We group all slots into two classes. The system learns patterns for the two classes of slots with different training data.

2. Cumulative Citation Recommendation task (CCR)

The CCR system focuses attention on query expansion and similarity calculation. The framework of our system is shown in Figure 1.

2.1 Query Expansion

In the CCR task, we utilize DBpedia and entity supporting documents to do two layers' query expansion. Expansion terms are given tags of Name, Label, Key and Link.

2.1.1 Query Expansion in Class Level

Based on sources of their supporting documents, the system classifies all entities into two classes: Wikipedia entity and Twitter entity. For entities in the two classes, do query expansion as following:

For each Wikipedia entity, the system chooses values of property name and values of

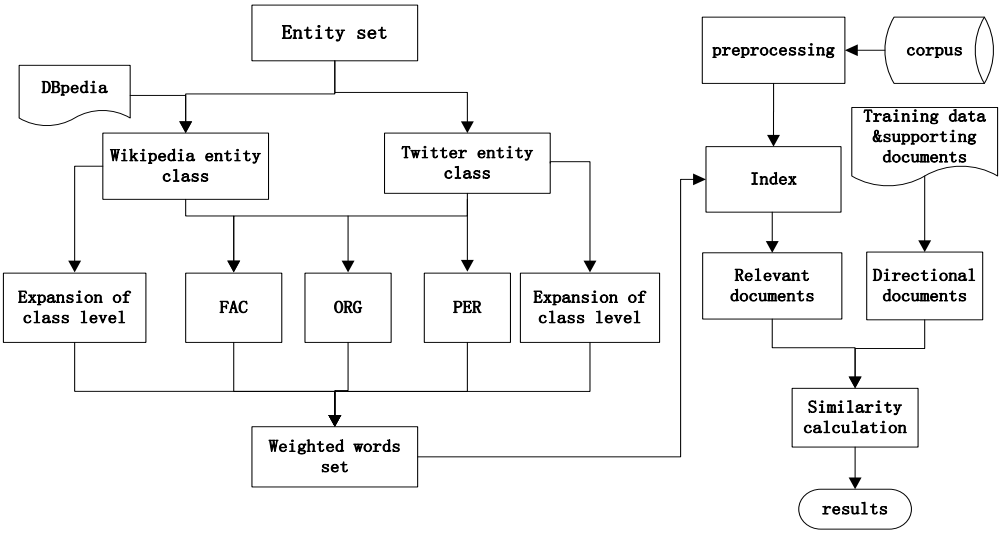

Figure 1 the framework of PRIS system for CCR task

property label as expansion terms from the corresponding DBpedia page and label these terms as Name and Label respectively; for each Twitter entity, the system chooses alternative names and the link of homepage as expansion terms from the corresponding twitter page and then visits the homepage to extract key words of homepages as expansion terms too. Here, label alternative names as Label, the link of homepage as Link and key words as Key.

2.1.2 Query Expansion in Type Level

The system respectively adopts different query expansion methods according to different types of entity. The different methods are described as following:

For each FAC entity, the system chooses the location and representative features as expansion terms from its corresponding Wikipedia page or homepage and label these terms with the tag of Key; for each ORG entity, the system chooses the chairman or CEO as expansion terms from corresponding Wikipedia page or homepage and label these terms with the tag of Key; for each PER entity, the system chooses the outstanding contributions of the entity from corresponding Wikipedia pages and homepages as expansion terms, such as magnum opuses of a writer, albums of a singer and political offices of a statesman. Then give these terms a tag of Key.

With the two level query expansions, we label these expansion term set as e_q with all expansion terms in. Specially, record the value of Name and Label with the set e_{name}, obviously, $e_{name} \subset e_q$.

2.2 Similarity Calculation

After query expansion, we next retrieve relevant documents based on these query terms. By computing similarities with directional documents, choose the most similar documents as vital documents for each entity.

2.2.1 Retrieval Relevant Documents

Based on the query expansion terms, we retrieve relevant documents from index. To

rank all relevant documents, we reference the work [2] described by University of Delaware in the KBA 2012 paper and allocate different weights to different expansion terms and calculate scores of all relevant documents as following:

$$Retrieval\_Score(e_q, d) = \begin{cases} Weighted\_Score(e_q, d) & mention(e_{name}, d) = 1 \\ 0 & mention(e_{name}, d) = 0 \end{cases} \quad (1)$$

where $mention(e_{name}, d)$ is an function which identifies the document *d* mentions e_{name} and is defined as:

$$mention(e_{name}, d) = \begin{cases} 1 & \text{if } e_{name} \cap d \neq \varnothing \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$Weighted\_Score(e_q, d)$ denotes the occurrence of e_q in *d*. α and β are the coefficients which assign different weights to different score components to balance their influences.

$$Weighted\_Score(e_q, d) = \alpha \cdot \sum_{name \in e_{name}} name + \beta \cdot \sum_{\substack{k \in e_q \\ k \notin e_{name}}} k \quad (3)$$

The condition $mention(e_{name}, d)$ checks whether entity *e* is discussed in *d*, and the function body $Weighted\_Score(e_q, d)$ serves as the complementary information to the relevance score under the assumption that the more expansion terms occur in *d*, the more likely *d* is relevant to entity *e*.

2.2.2 Directional documents generation

For each entity, the supporting document and training data include much important information and can be used to judge whether a new document is worth citing or not. The system builds directional documents as the criterion for each entity. The system builds three kinds of directional document: supporting directional document, keywords directional document and expansion directional document.

Supporting Directional Document (Sd): for entity *i*, filter the supporting document by removing stop words to generate the supporting directional document with the name of Sd_i.

Keywords Directional Document (Kd): take all 170 supporting documents as training data to calculate the tf-idf values of each document and choose the top 80% words of the corresponding support document of entity *i* to generate a directional document with the

name of Kd_i.

Expansion Words Directional Document (Ed): for entity *i*, integrate the supporting document and its training documents to a new document d_i. Then take these new 170 documents as training data to calculate the tf-idf value of each document and choose the top 80% words of each document to generate a direction document of entity *i* with the name of Ed_i.

2.2.3 Similarity calculation

For the purpose of generating final recommended documents from the candidate documents in section 2.2.1, we utilized Jaccard coefficient to calculate the similarity between candidate documents and directional documents of each entity. The Jaccard similarity coefficient is a statistic used for comparing the similarity and diversity of sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets. We used a variation of the traditional Jaccard formula for our specific task showing as follows:

$$Similarity(e_d, d) = \frac{e_d \cap d}{e_d} \quad (3)$$

$e_d = Sd \cup Kd \cup Ed$ is the set of directional documents of each entity. The equation (3) demonstrates that the more terms that a candidate document and a directional document share, the bigger of the similarity.

2.3 Results

Table 2 The best result of runs of PRIS.

Run	P	R	F	SU
Vital+useful cutoffstep=1	0.457	0.365	0.406	0.364
max			0.670	0.575
median			0.440	0.395
mean			0.388	0.395
Vital cutoffstep=1	0.156	0.333	0.213	0.210
Vital cutoffstep =10	0.108	0.232	0.148	0.142
max			0.338	0.286
median			0.205	0.210
mean			0.188	0.214

3. Stream Slot Filling (SSF)

For the SSF task, our system utilizes a pattern learning method to do relation extraction and slot filling. The PRIS system groups all slots into two classes: KBP slots and other slots. Patterns of regular slots which are same to TAC-KBP slots are learned from KBP Slot Filling corpus. Other slots are filled by following some generalized patterns learned from external source data including homepages of some famous people and facilities.

3.1 Pattern Learning Method

3.1.1 Trigger words mining

In information extraction tasks, a specified relation pattern is mostly triggered by some trigger words. So, based on the idea of activation force of Jun Guo [2], we define trigger force as a criterion for trigger words mining. The trigger force is described in details in [3].

3.1.2 Pattern generation Based Trigger Words

In our system, the type of generated pattern is dependency pattern, which refers to the shortest dependency path centered by trigger words and linking an entity & attribute-value pairs.

For each slot, retrieval sentences containing trigger words and parse these sentences by using the Stanford Parser to create dependencies in the "collapsedDependency" format. Based on these dependencies, let the trigger word as center word and find the shortest path linking to the entity and attribute-value.

For example, for a given slot CauseOfDeath, suppose trigger words have been extracted as "died", "dies". The sentence "John died of cancer." containing trigger word "died" can be captured. Based on the dependency relationship, we use the shortest connecting path centered by trigger word "died" to represent the relation between them:

$$John \underset{nsubjpass}{\rightarrow} \mathbf{died} \underset{prep\_of}{\leftarrow} cancer$$

Then a dependency pattern can be generated as following:

PER: CauseOfDeath <PER> nsubjpass died prep_of <A_disease>

3.1.3 Relation pattern learning

To minimize semantic drift in both generations of entity-values and patterns, the system proposes a semantic drift analysis algorithm [] .

After finding candidate patterns, these patterns are ranked according to:

$$C(p_i) = \frac{seen(p_i)}{all(p_i)} \log_2(seen(p_i))$$

Where seen(p) is the number of entity-values(by type) extracted with patterns p that are already in the seed pair class and all(p) is the total number of entity-values (by type) extracted with pattern p.

3.2 Patterns for Other Slots

For the four other slots different to KBP slots, the system learns some generalized patterns from external source data including homepages of some famous people and facilities. The method is same to pattern leaning method described above without the trigger word mining step.

For the FAC slot Contact_Meet_Entity, firstly, find the transitional word T which have relation prep with Entity; secondly, find Core words which have relation nsubj with T word; lastly, take the subtree of Core words as the answer for the slot.

For the PER slot Contact_Meet_PlaceTime, firstly, find the transitional word T which has the relation nsubj with entity; secondly, find Core words which have relation prep or tmod

with T word; lastly, check whether the Core words can be connect with preps or not, if can be connect, combine all subtree of Core words as answer; if not, discard all Core words. Specially, the core word must be Date or Location for this slot.

For the slot Affiliate, we just find all related PER or ORG with entities involved in three layer relationship in dependency tree for each entity.

3.3 Result

Table 2. The best result is shown in table 2.

parameters	P	R	F	SU
DOCS	0.0617	0.305	0.103	0.0245
max			0.103	0.333
OVERLAP	0.522	0.452	0.484	0.494
max			0.672	0.670
FILL	0.234	0.099	0.139	0.306
max			0.159	0.359
DATE_HOUR	0.582	0.577	0.579	0.453
max			0.720	0.711

Reference

[1] Vectomova, Olga; Wang, Ying (2006). A study of the effect of term proximity on query expansion (Abstract). Journal of Information Science 32 (4): 324–333. doi:10.1177/0165551506065787.

[2] Xitong Liu, Hui Fang(2012), Entity Profile based Approach in Automatic Knowledge Finding.

[3] Guo J, Guo H, Wang Z. An activation force-based affinity measure for analyzing complex networks[J]. Scientific reports, 2011, 1.

[4] Chunyun Zhang, Dai Zhang(2013), A Trigger Word Mining Method Based on Activation Force, ready to publish.