PRIS at TAC2012 KBP Track

Yan Li, Sijia Chen, Zhihua Zhou, Jie Yin, Hao Luo, Liyin Hong,
Weiran Xu, Guang Chen, Jun Guo
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
buptliyan@gmail.com

Abstract

Our method to Knowledge Base Population at TAC2012 is described in this paper. An enhanced pattern bootstrapping system is mainly utilized in the Slot Filling task. And for the Entity Linking task, query expansion method, rule-based method and entity similarity ranking strategy are combined.

1 Introduction

The main goal of the Knowledge Base Population (KBP) track at TAC 2012 is to promote research in and to evaluate the ability of automated systems to discover information about named entities and to incorporate this information in a knowledge source. Actually, it is not new for us as we have taken part in the KBP track for three years. We participated in both regular English slot filling and entity linking tasks this year just as before.

The Slot Filling task involves learning a pre-defined set of relationships and attributes for target entities based on the documents in the test collection. Different from our previous years' work, besides the rule-based method, we referenced the work of NYU in KBP 2010 [1] and designed an enhanced pattern bootstrapping system with more comprehensive preprocessing and query expansion and more elaborate base patterns. The good evaluation results show the effectiveness of our system.

The Entity Linking task is to determine for each query, which knowledge base entity is referred to, or if the entity is not present in the reference KB. And the main difficulties of this task are alias detection (that multiple queries may refer to the same entity using different name variants or different doc ids) and entity disambiguation (that the same query name may refer to multiple entities).

In TAC2012-KBP track, we consider Entity Linking task as a retrieval task. In order to resolve the two difficulties mentioned above effectively, we designed some rules for helping make better decisions. In all, the rule-based entity matching method and a collaborating ranking strategy were both used for this year's task.

The remaining of this paper is organized as follows. Section 2 and 3 describe systems about Slot Filling task and Entity Linking task respectively. Section 4 presents our evaluation results of the tasks.

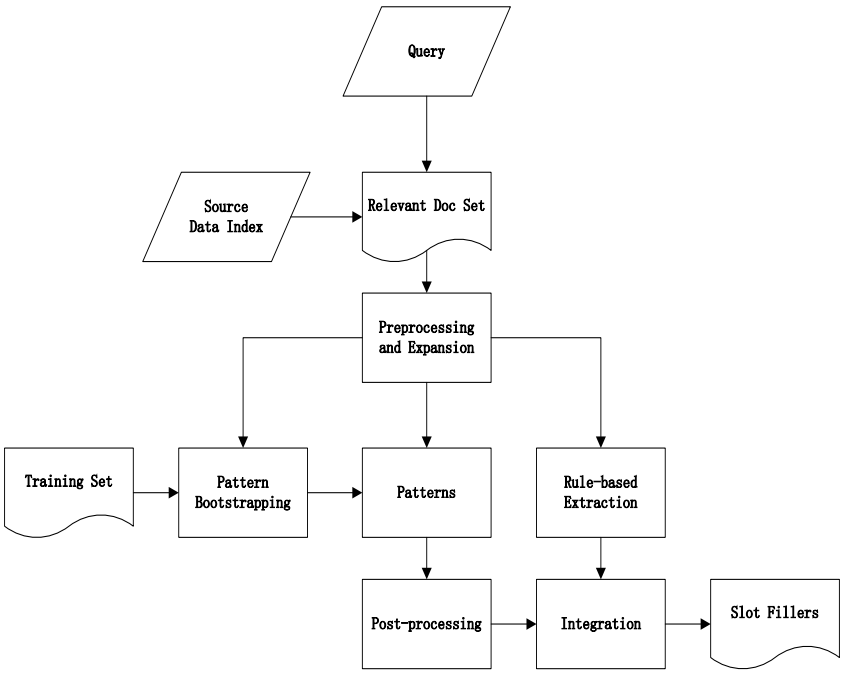

Figure 1: Framework of Slot Filling system

2 English Slot Filling Task

Fig.1 shows the framework of our Slot Filling system. The pattern bootstrapping module, which is the most important one, was trained in advance and generated a set of patterns for each slot. Queries were searched in our index of source data returning top 50 relevant documents for each query to conduct the preprocessing and expansion, which is also needed before training the bootstrapping system. Then the workflow goes into two branches: rule-based information extraction and matching the query-relevant documents with patterns generated by the bootstrapping module. The rule-based method is actually almost the same as our last year's work and will not be described here due to the limited space.

2.1 Preprocessing

We first indexed the source corpora and retrieved top 50 relative documents for each target entity. The Stanford CoreNLP package was used to get complete linguistic annotations of all the natural language texts, which includes the part-of-speech (POS) tagger, the named entity recognizer (NER), SUTime, the dependency parser and the coreference resolution.

All the slot values were categorized into 12 domain types. PERSON, ORGANIZATION and LOCATION are recognized by NER (also known as CRF Classifier). DATE, NUMBER and URL are recognized using a rule-based system. Domain types for TITLE, CRIMINAL CHARGE, CAUSE OF DEATH, NATIONALITY, RELIGION and SCHOOL are mainly from pre-prepared lists.

2.2 Name Expansion

As many sentences might contain

coreferential name variants of a query, we should identify as much mentions of the target query as possible in its relevant documents. Every entity may have many alternate names, which means query expansion became so important to improve system performance in terms of Recall.

We tried various ways as follows for query name expansion of the ORG type. As for the entities of PER type, only step 3 below was adopted:

Ignore the corporate suffixes (”Ltd”, “Corp.”, “PLC”, etc.) if an ORG query entity contains one. As for the rest of the query, we searched in the context and got the corresponding acronym or full name.

Based on the Stanford CoreNLP package, we added the co-reference names of the query entity into the query expansion set.

A rule-based method was also used for identifying query alternate names. For example, “known as”, “called” and “former” were appropriate patterns. And an interpretative entity name in parenthesis after the target query also implied a variant of it.

To increase the recall of the relevant documents, we retrieved the source data with all these expanded query names. By the way, the results of the name expansion were directly used as the values of the alternate_name slots.

2.3 Pattern Bootstrapping

For each slot, there were patterns that implied the relation between the target entity and slot value. Extraction patterns were typically obtained through manually written patterns compiled by experts or automatically generated patterns based on training data. In order to automatically generate more and more patterns for slot

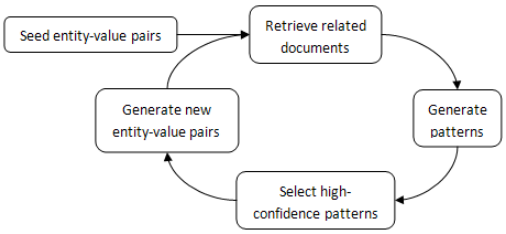

Figure 2: The Pattern Bootstrapping System

filling, we designed an enhanced adaboost pattern-matching system, which referenced the work described by NYU in the KBP 2010 paper (Grishman and Min 2010) and Snowball (Agichtein and Gravano 2000). The pattern adaboost system is described in Fig.2.

The training set was the KBP English Monolingual Slot Filling Evaluation Data in the past three years, which contains 106 ORG entities and 92 PER entities, 1627 entity-value pairs in total. The training examples were used to generate patterns, which in turn resulted in new entity-value pairs extracted from the document collection. At each iteration of the extraction process, the system evaluated the quality of these new-generated patterns and tuples, and then keeps only the reliable ones for the next iteration. In addition, the Stanford Dependency Parser (Marneffe and Manning 2008) was also utilized as part of our extraction approach.

Two types of sentence-level patterns are defined: word sequence patterns and dependency path patterns. A word sequence pattern is the middle context between an entity-value pair. And with dependency structures generated by the Stanford Dependency Parser, a dependency pattern refers to the shortest dependency path which connects an entity-value pair. For the sentence “*Takeshi Watanabe, the first president of the ADB, died in his native Japan.*”, Fig.3

PER: title → <PER> appos <TITLE>
PER: member_of → <PER> appos **president** prep_of <ORG>
PER: country_of_death → <PER> nsubj-1 **died** prep_in <LOC>
PER: countries_of_residence → <PER> **native** <LOC>

Figure 3: Illustration of Generated Patterns

shows some pattern examples that might be generated with named-entity tags.

2.4 Post-processing

For the slots of DATE type, the full date of a temporal expression could be inferred from contextual information, such as the news post date. We used SUTime module of Stanford CoreNLP to identify temporal expressions in the text and normalize them to the required TIMEX2 format of yyyy-mm-dd.

We also normalized fillers for PERSON slots including PER: spouse, PER: children and PER: parents. We actually searched last names from the context and make the fillers more detailed and less ambiguous. For examples, if we found a document containing the text "John Doe's first wife, Ruth", then "Ruth Doe" was regarded as a better filler than "Ruth" for PER: spouse of the target entity "John Doe".

To identify LOCATION slot fillers into countries, states/provinces and cities, we applied a list from Wikipedia, which contained all countries and states or provinces. Fillers not in the gazetteer were classified as city-level.

For fillers of the PER: title, we used a list of about 500 titles with no modifiers. We further included adjectival modifiers (e.g. "financial Minister"), noun compound modifiers (e.g. "police chief") and prepositional modifiers (e.g. "chief of military operations") as part of the slot values.

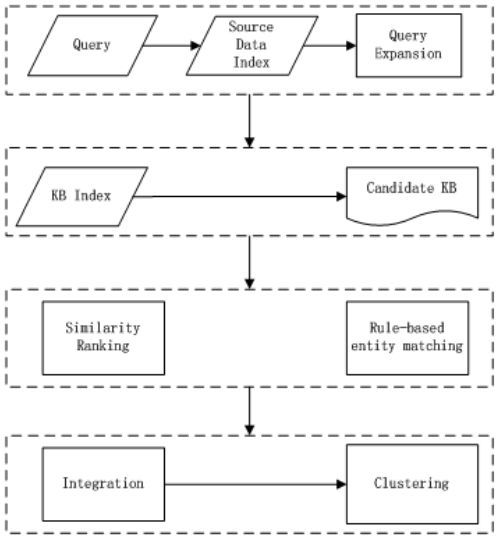

Figure 4: Framework of Entity Linking System

3 English Entity Linking Task

The framework of Entity Linking system is shown in Fig.4. The candidate knowledge base node was first obtained after query expansion. Then the similarity calculation for ranking the candidates and the rule-based entity matching were executed simultaneously. The corresponding answer sets were integrated and the clustering for the NIL entity was finally conducted.

3.1 Query Expansion

The strategy for our query expansion is illustrated in Fig.5.

Each query is attached with one supporting document. We first searched the source data index and returned the top 100 relevant documents. The Stanford Named Entity Recognizer (NER) tool was then utilized for filtering and classifying entities.

Each returned documents and the supporting one were represented by word frequency vectors. And they were clustered by the hierarchical clustering algorithm. Then the cluster which contains the supporting document was selected as

the supporting document set for queries.

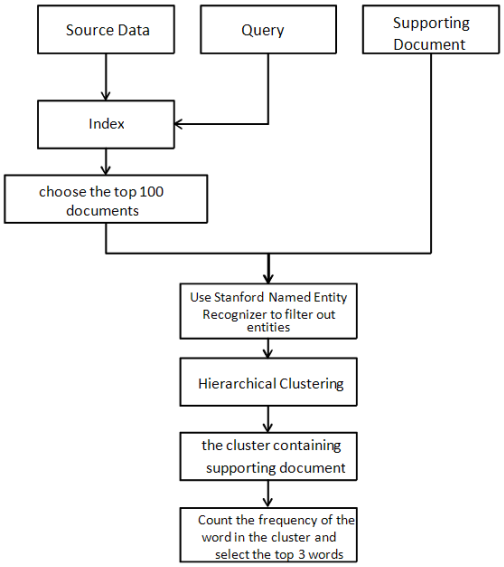

Figure 5: Illustration of Query Expansion

the final top three expansion words were obtained according to their frequencies in the set.

3.2 Rule-based Entity Matching

This module involves matching one of the KB candidates to the query entity based on rules. In particular, the rules were categorized into three types according to the entity type of PER, ORG and GPE.

For the PER queries, We counted the number of the queries in the candidates. If the highest number is bigger than the predefined threshold we regarded this KB as the answer. We note that the person's first name weighs more than the family name.

For the ORG queries, we checked the query title and the KB title field. If all the words in the query title occur in only one KB title, we considered this KB as the answer for the query.

It is usual for the GPE queries that the mentioned small location is placed before the big one. Under this assumption, we separated the KB title by the label ",_" and got the first element. If it matches the query title, we returned this KB as the answer.

Algorithm 1 Collaborating Ranking

Input:

query q; a set of candidate vectors $o^{(q)}$; F^*; a composite $g(.)$

Output:

a set of ranking scores $y^{(q)}$

1: for j = 1; j <=candidate number; j++ do

2: From a feature vector $x_j^{(q)}$.

3: Compute ranking scores: $f_1(x_j^{(q)}), f_2(x_j^{(q)}), f_3(x_j^{(q)})$

4: Compute composite function: $y_j^{(q)} = g(\bullet)$

5: end for

6:return $y^{(q)}$

3.3 Candidate Ranking

This module utilized a collaborating ranking strategy to find the best KB candidate for a query entity.

There are three types of features considered:

• f_1: Words occurred in the KB candidate entry and the supporting document.

• f_2: Entities chosen by the Stanford NER toolkit, which include ORG, PER and GPE.

• f_3: Words occurred in the sentences containing f_2 including the query.

They formed three rankers respectively based on the cosine similarity.

Let $x_j^{(q)}$ be the feature vector of the j-th KB candidate with query q. We denote $F^*=\{f_1, f_2, f_3\}$. We transform the computation of collaboration among rankers into solving the following composite function:

$$y_j^{(q)} = g(f_1(x_j^{(q)}), f_2(x_j^{(q)}), f_3(x_j^{(q)})) \quad (1)$$

where the version of g is:

$$g = \begin{cases} f_2(x_j^{(q)}), \text{ if query type is GPE} \\ NIL, \text{ if } \|f_3(x_j^{(q)})\| < \text{threshold} \\ \max_i \{ f_i(x_j^{(q)}) \}, \text{ otherwise} \end{cases} \quad (2)$$

A general algorithm for collaborating ranking is presented in Algorithm 1.

4 Evaluation Results

4.1 Evaluation Results of Slot Filling

Five runs were submitted for the Slot Filling task this year, and Tab.1 shows the best evaluation results.

4.2 Evaluation Results of Entity Linking

Three runs were submitted for the Entity Linking task this year, and Tab.1 shows the evaluation results on B3+ F1 measure.

References

Ralph Grishman and Bonan Min. New York University KBP 2010 Slot - Filling System. In proceedings of TAC-2010.

Ang Sun et.al. New York University 2011 System for KBP Slot Filling. In proceedings of TAC-2011.

Zheng Chen et.al. CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description. In proceedings of TAC 2009.

Heeyoung Lee et.al. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In proceedings of CoNLL-2011.

Zheng Chen and Heng Ji. Collaborative Ranking: A Case Study on Entity Linking. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 771–781, 2011.

Number of filled slots in key: 1543 Number of filled slots in response: 956 Number correct non-NIL: 646 Number redundant: 51 Number incorrect / spurious: 222 Number inexact: 37	
Precision	0.6757322 (646/956)
Recall	0.41866493 (646/1543)
F1	0.5170068

Table 1: Slot Filling Task Evaluation Results

	All	**In KB**	**Not in KB**	**NW docs**	**WB docs**	**PER**	**ORG**	**GPE**
PRIS1	0.519	0.470	0.574	0.573	0.412	0.686	0.480	0.304
PRIS2	0.474	0.384	0.572	0.525	0.372	0.682	0.433	0.198
PRIS3	0.348	0.234	0.466	0.370	0.304	0.501	0.364	0.096

Table 2: Entity Linking Task Evaluation Results