

PRIS at Knowledge Base Population 2013

Yan Li, Yichang Zhang, Doyu Li, Xin Tong, Jianlong Wang, Naiche Zuo, Ying Wang,
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 2013 is described in this paper. An pattern bootstrapping system with automatic pattern evaluation is mainly utilized in the Slot Filling task. For the Entity Linking task, query expansion method and entity similarity ranking strategy are mainly considered. And for sentiment holder and target detection in the Sentiment Slot Filling task, two CRF models are primarily trained.

1 Introduction

We participated in three tasks of Knowledge Base Population at TAC 2013 including English Regular Slot Filling, Entity Linking and Sentiment Slot Filling.

The Regular Slot Filling task involves learning a pre-defined set of relationships and attributes for target entities based on the documents in the test collection. On the basis of our last year's pattern bootstrapping system, we focused on automatic evaluation for newly generated patterns this year. In addition, we added a new module named WMEB to further filter out noisy fillers.

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. For this year's task, we devoted ourselves to rule-based query expansion and similarity calculation considering four factors.

The goal of Sentiment Slot Filling is to use corpora to collect information regarding sentiment expressed towards or by an entity. We trained two CRF models using four types of features to identify sentiment holders and targets respectively. The determination of sentence sentiment orientations is just based on a general opinion lexicon.

2 English Regular Slot Filling

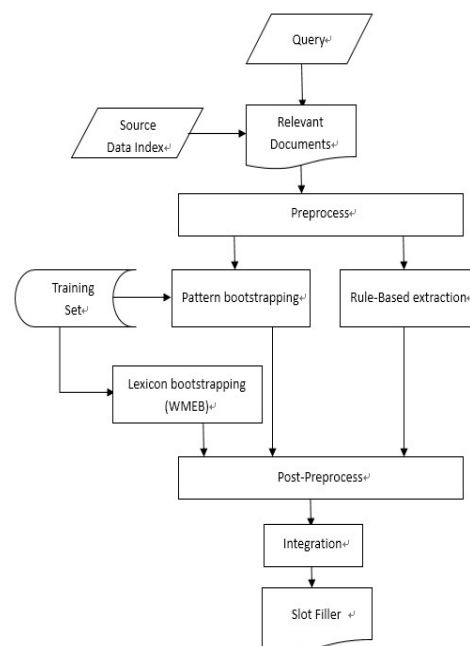


Figure 1. Framework of Slot Filling System

Figure 1 is the overall framework of our system, the most important part is pattern bootstrapping based on dependency parsing. We trained patterns for each slot using the evaluation sets of Slot Filling task from 2009 to 2011. We extract answers for each query with these patterns and the corresponding documents (each query retrieves up to 150 related documents). In addition, the automatic evaluation of newly generated patterns and candidate slot fillers is also this year's focus.

2.1 Pre-processing and Query Expansion

Initially, we indexed the source data using Indri. For each query, we retrieved 150 related documents and parsed them using Stanford CoreNLP system. In particular, we also conducted coreference resolution.

We divided the slots into 12 categories which are the same as our previous year's system.

The query expansion method will be introduced later in the Entity Linking section.

2.2 Pattern Bootstrapping

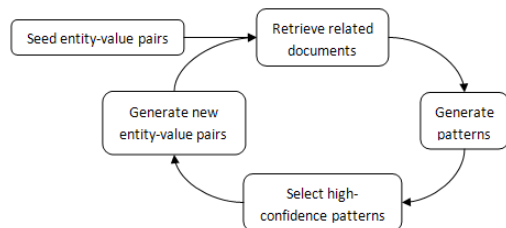


Figure 2. The Pattern Bootstrapping System

We generated patterns and extract slot fillers in a bootstrapping manner based on dependency tree. We regarded the training process as parsing an undirected graph, and selected all paths containing less than 3 vertices from query to candidate filler. The bootstrapping will induce noises if it is not controlled. How to determine the stop

criterion automatically has been proposed for a long time. The flowchart of bootstrapping shown in Figure 2 is similar to our last year's system unless the process of selecting high-confidence pattern. Specifically, we evaluated patterns according to the following formula:

$$\text{score}(p) = \frac{pf-1}{pa} \log(pa)$$

where p represents the pattern, pf stands for the number of the seed entity-filler pairs in each iteration, pa indicates the number of all the pairs found in an iteration and $\log(pa)$ is a weighting factor measuring the pattern's ability to find the pair. Benefiting from this formula, we are able to obtain better patterns in each iteration so as to ensure a larger accuracy.

We excluded the word sequence pattern this year and only used the dependency pattern. The reason is that, from last year's results, the dependency tree has a stronger expression.

2.3 Pattern with Trigger Words

For some certain types of slots (i.e., $\langle PER \rangle$ was born in $\langle Location \rangle$), the trigger words (i.e., *born in*) imply a strong relationship between entities and its fillers. Patterns with these trigger words are expected to have higher weights.

We applied the Word Activation Force model proposed by Jun Guo et al. to find trigger words. We chose the sentences that can extract correct entity-filler pairs, and then extracted the middle parts between entity and filler in the sentences, resulting in a lot of words, phrases, or other strings. We calculated the activation force between these strings and regarded the ones with higher values as the trigger words.

2.4 Post-processing

If the candidate filler confirms to the entity

type, then according to the predefined 12 slot categories, we can determine which type of filler should be filled.

For the case of no entity type information, we get a list of words in different fields based on the WMEB model proposed by T. McIntosh and J. Curran. Then we find their synonyms through WordNet. The union of original vocabulary and synonyms is our final vocabulary. According to the vocabulary, we can determine which kind of relationship the candidate filler should belong to.

3 English Entity Linking

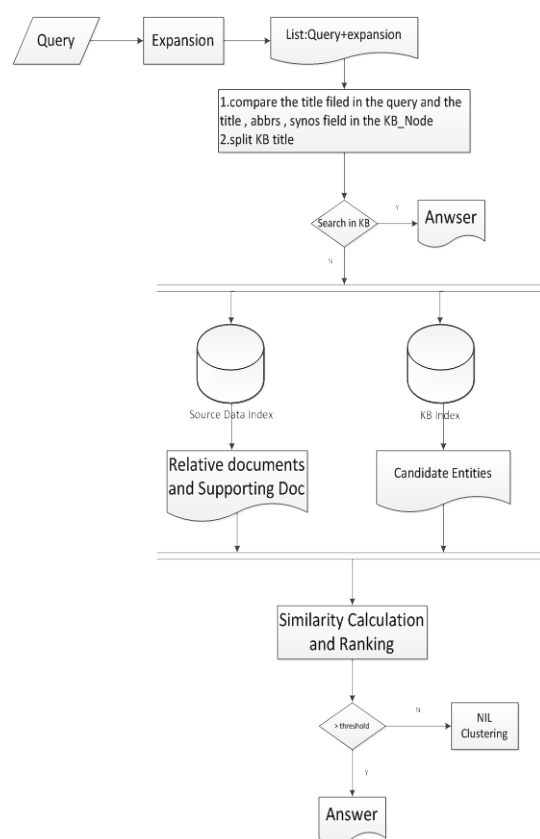


Figure 3. Framework of Entity Linking system

As Figure 3 shows, our system begins with query expansion so that we can get candidate entities. The next step is to calculate similarities between candidate entities and the query.

3.1 Query Expansion

We need to find possible candidates for the given query using the information extracted from the support documents. In general, the given query is only part of the real entity. Query expansion can effectively reduce the ambiguity of the query.

In this part, our missions are as follows:

A) Finding full person names for PER entities from source documents. For example, *Lucy* is the query, but *Lucy Walsh* is also mentioned in the support document.

B) If the query is a city name, we try to find the state where it locates.

C) For abbreviatory queries, we need to find their full names. In particular, we first employ Stanford NER to extract named entities from support documents. If the document contains the query, we extract contiguous token sequence ranging from the acronym's first letter to the last letter.

D) For GPE queries, we use dictionary containing state names, their abbreviations and aliases. If a query is followed by words in the dictionary, we take them as expansion.

3.2 Similarity Calculation

The key part in our Entity Linking system is to calculate similarities between candidate entities and the query. The candidate most similar to the query is regarded as answer. If the largest similarity is below a threshold, NIL will be given. The strategy considers the following four factors.

A) String match using Levenshtein

This factor is to check whether the title of a candidate matches the query. If there is great difference between the title and the query, they certainly are not the same entity.

Here we use the Levenshtein distance as follows:

$$S_1 = 1 - \frac{L'}{\max(L_1, L_2)}$$

where L' represents the Levenshtein distance, L_1 and L_2 stand for the length of two words.

B) The special fields of candidates

Considering answers may include nickname, full name, official name and so on, we use the special field of the KB text to carry out similarity calculation. This information is useful for determining whether the candidate entity is the same as the query. It is helpful to improve the recall. Therefore, the similarity between the special field of candidate and query is account for a large weight. The formula is the same as the one used in the first step.

C) The fact fields of candidates

Thirdly, we can use the fact field of the KB text to calculate the similarity, this step is applied when there are many candidates sharing the same similarity using the above two methods. First, there are some figures which should be filtered in the fact field of the KB text. On the other hand, we use the Stanford NER tools to help to identify the named entities in supporting documents, and then we use the following formula to calculate the third similarity factor:

$$S_3 = 1 - \frac{\log(\max(|C_1|, |C_2|)) - \log(|C_1 \cap C_2|)}{\log(|W|) - \log(\min(|C_1|, |C_2|))}$$

where C_1 represents a set of named entities in the supporting documents, C_2 indicates the word set in the fact field and W stands for the sum of these two entities.

D) Entity category

This factor is to check whether the category of candidates belong to our consideration. According to the task definition, the entity types contain location, organization and person. But there are many candidates whose titles may be movies, songs, novels and so on. Therefore we need to filter out these entities. If the category of a candidate entity is one of the three concerned categories, we set the similarity to 1

otherwise 0.

4 Sentiment Slot Filling

4.1 Preprocessing

As a whole, our team conducts the same steps for data preprocessing in Slot Filling and Sentiment Slot Filling. In addition, in Sentiment Slot Filling, we took special care of the new source data Meta, because the authors in forums are often exact sentiment holders.

4.2 Training

In the training set applied by the authority, we extracted the holder, target and the whole sentence according to the annotation. We also have conducted pronoun resolution, because we believe that the holder or the target can be a pronoun in the training data.

We have trained two models based on Conditional Random Fields (CRFs) to identify sentiment holders and targets respectively.

There are four types of features:

A) POS tagging

B) NE: whether the token is a named entity.

C) SubW: whether the token is a subjective word.

D) SenW: whether the token is a sentiment word.

As for identifying sentiment orientations, we just simply use the dominant orientation of the opinion words in the sentence to determine the orientation of the sentence. The general opinion lexicon we used is the popular MPQA Subjectivity Lexicon.

4.3 Extracting Candidate Fillers

Figure 3 shows the process of extracting candidate fillers in our Sentiment Slot Filling system. For holder detection, we first check the source of the related sentence. If

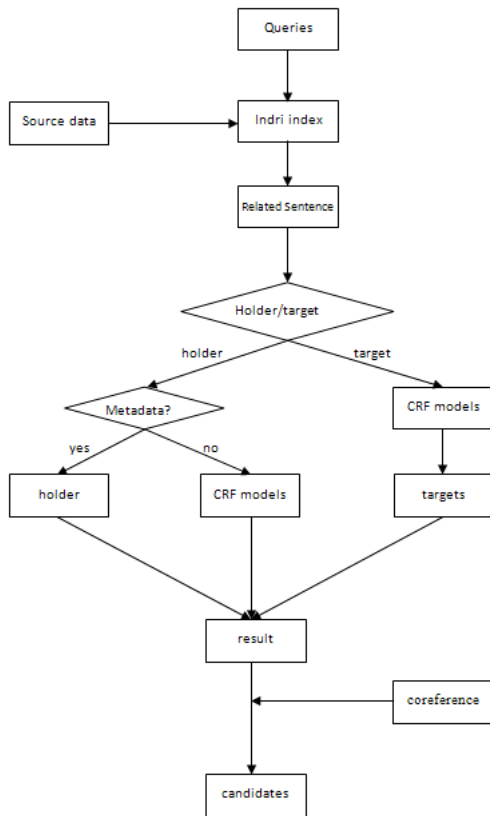


Figure 3. The process of extracting candidate fillers

it comes from forums, we just regard the author in the Metadata as the holder. As to the target, we just view the CRF result as a candidate.

For every sentence with a candidate, if its sentiment orientation confirms to the query request and its strength is bigger than a threshold, we took the candidate as the final filler.

5 Evaluation Results

5.1 Evaluation Results of English Regular Slot Filling

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

5.2 Evaluation Results of Entity Linking

Two runs were submitted for the Entity Linking task this year, and Table 2 shows

the evaluation results on B3+ F1 measure.

5.3 Evaluation Results of Sentiment Slot Filling

Table 3 shows the Sentiment Slot Filling evaluation results for the 3 submitted runs.

We can see that the run using the metadata information performs much better than the single CRF model. The main reason may be the limited features of the CRF input.

References

- J. Guo, H. Guo and Z. Wang. An Activation Force-based Affinity Measure for Analyzing Complex Networks. *Sci. Rep.* 1, 113; DOI:10.1038/srep00113 (2011).
- T. McIntosh and J. R. Curran. Weighted Mutual Exclusion Bootstrapping for Domain Independent Lexicon and Template Acquisition. In proceedings of the Australasian Language Technology Association Workshop. 2008: 97-105.
- J. R. Curran, T. Murphy and B. Scholz. Minimising semantic drift with Mutual Exclusion Bootstrapping. In proceedings of the 10th Conference of the Pacific Association for Computational Linguistics, pages 172-180, 2007.
- S. Tamang, Z. Chen and H. Ji. CUNY BLENDER TAC-KBP2012 Entity Linking System and Slot Filling Validation System. In proceedings of TAC-2012.
- J. Clarke, Y. Merhav, G. Suleiman, S. Zheng and D. Murgatroyd. Basis Technology at TAC 2012 Entity Linking. In proceedings of TAC-2012.
- Y. Choi, C. CardieIdentifying, E. Riloff and S. Patwardhan. Sources of Opinions with Conditional Random Fields and Extraction Patterns. In Proceedings of HLT/EMNLP 2005.

Number of filled slots in key that are not in reference KB: 1251	
Number of filled slots in key that are in reference KB: 217	
Number of filled slots in responses: 1042	
Number Correct (not in reference KB): 348	
Number Redundant with reference KB: 57	
Number redundant with another response: 28	
Number inexact: 80	
Number incorrect / spurious: 529	
Precision	0.38867563 ((348+57)/1042)
Recall	0.27588555 ((348+57)/(1251+217))
F1	0.32270917

Table 1. Slot Filling Evaluation Results

	All	In KB	Not in KB	NW docs	WB docs	DF docs	PER	ORG	GPE
PRIS20131	0.467	0.428	0.503	0.539	0.476	0.344	0.544	0.503	0.368
PRIS20132	0.475	0.437	0.510	0.546	0.486	0.353	0.550	0.510	0.379

Table 2: Entity Linking Evaluation Results

Runid	Precision	Recall	F1
PRIS20131	0.0915	0.20243363	0.12603307
PRIS20132	0.0955	0.21128319	0.1315427
PRIS20133	0.055	0.121681415	0.07575758

Table 3: Sentiment Slot Filling Evaluation Results