GDUFS at Slot Filling TAC-KBP 2012

Xin Ying Qiu, Xiaoting Li, Weijian Mo, Manli Zheng, Zhuhe Zheng

CISCO School of Informatics
Guangdong University of Foreign Studies
Guangzhou, China
xinying.qiu@gmail.com

Abstract

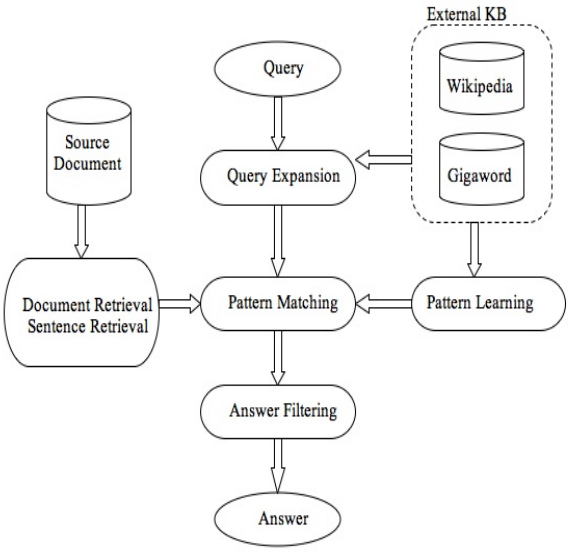
This document describes our first attempt at Slot Filling task at TAC-KBP 2012. We construct a baseline system consisting of four components, i.e. query expansion, document retrieval, pattern learning and matching, and answer selection and filtering. We are pleased to identify areas for improvement and directions for exploring new approaches based on the groundwork prepared for this task.

1 Introduction

The GDUFS team participated for the first time in the Regular Slot Filling task at TAC Knowledge Base Population 2012 track. We construct a baseline answer extraction pipeline based on CUNY Blender tools generously provided by City University of New York (Chen et al., 2010). Our main methodologies include query expansion, document retrieval, pattern learning, and answer selection and filtering. We submitted five runs that differ in terms of the document used for sentence retrieval, and the patterns that we generated from multiple corpora. While this is our first attempt at Slot Filling task, we are pleased to have established groundwork for future development and identify directions for further improvement.

2 System Architecture

The Slot Filling task defines a total of 26 attributes for the person (PER) and organization (ORG) entities together. The goal of the task is to extract from source documents the value(s) for each slot (i.e. attribute) of each entity query. For example, per:children defines the "children" attribute of a person entity. Given the knowledge base entry of the person instance and the background document describing the person instance, the task requires the children's names of this person instance to be extracted from the source documents. Answer should be presented along with document id and the position of the answer in the document. Our main approach to this task is pattern learning and matching. Figure 1 describes the major component of our slot value extraction pipeline.



We begin by taking each query and expand it into a set of variants (Section 3.1). We retrieve relevant

documents from source collection using the expanded query list and the query's background document (Section 3.2). Our main approach is pattern learning and matching. We learn offline patterns that describe the answer sentences for each slot and apply them to select sentences from relevant documents (Section 3.3). Answers identified in the selected sentences are ranked and filtered to provide final answers (Section 3.4). Details of these components are provided below.

3 Methodology

3.1 Query expansion

As many entity names have variants, query expansion is applied to increase recall in slot filling task. For example, the query "Barnes Foundation" refers to "Barnes Foundations of Philadelphia". The latter will also be used to retrieve relevant document about this entity. While there are different methods for query expansion (Jian et al., 2011; Li et al., 2011), we depend on Wikipedia redirect page database to extract additional query variation according to the following heuristics:

1. We query Wikipedia snapshot database of 2009 for page title that is the same as the query term. If the title page is redirected to another title, we use the referred title term as a variation to add to the query expansion list.
2. We use the query variation from step 1 to find out if there are other titles that are redirected to this query variation. If redirection is found, we add these other titles to the query expansion list.
3. We add the person entity's last name to the query expansion list.
4. If the query in the expansion list has a edit distance of over 5, the query is dropped from the expansion list.

3.2 Document retrieval

We use Lucene to index source documents. In order to select candidate sentences, we first need to identify relevant documents. We first use the expanded query list to query the source document index and retrieve the top 1000 documents for each query. Then we use the query's background document to retrieve from the source index the top N documents. We vary different number of N for each run. We use the union set of these retrievals as the corpus from which to extract candidate sentences for each query entity during task evaluation.

3.3 Pattern Learning and Matching

Our main approach is to learn sentence patterns for each entity slot, and then match the pattern to the sentences in the retrieved documents for answer selection. To learn patterns, we construct a list of entity query and their corresponding slot value pairs from the past KBP slot filling task data set of 2009, 2010, and 2011. We also extracted from Wikipedia knowledge base info box the entity-slot value pairs that match KBP slot filling types. These correct query and slot answer pairs are used to retrieve candidate sentences from the corpus provided, including the source documents and the English gigaword documents. We search for sentences that contain both the query term and the slot value. These sentences are categorized as learning data sets for different slots. For example, we collected about 600 sentences that contains the query-answer pairs for per:employee_of slot. We assume that these are positive examples for the sentence patterns describing the entity slot, following the assumption of the distant supervised learning method (Mintz et al., 2009).

With the learning set of sentences ready for each slot, we apply Stanford NER to recognize and tag the named entities in each sentence. Then we automatically generate regular expressions for the sentence segment between the query term and the slot value in each sentence. For example, a query-answer pair of "Raul Castro : Fidel Castro" for the slot "per:other_family", we identify the following sentence as a positive sentence because of the occurrences of the entity and slot value:

"Now begins the second stage that will also be triumphant," said Raul Castrol, Fidel Castro's brother and minister of the armed forces.

We then learn a regular expression pattern such that:

[Target_Person], [Slot_Person]'s

We store such candidate patterns for each slot. Using a separate set of query-slot answer pairs and their correct sentences, we evaluate the validity (or confidence score) for each pattern, by counting the number of sentences that match a certain patterns

as a percentage of the total assessment sentences. We keep only patterns with a confidence score over 0.5, as a heuristics.

During task evaluation, we perform a similar NER tagging to each sentence in the document retrieved as in Section 3.2. We then apply the patterns learned above to find matching sentences and extract candidate answers from the sentences. The candidate answers are assigned the score of the matching pattern used as the confidence score of the answers.

3.4 Answer Selection and Filtering

The candidate answers for each query-slot pair need to be filtered before generating final answer set. We apply several simple filtering methods such as removing redundant answers for the same slot, and removing answers with score under a threshold by heuristics. The scores for each final answer are further normalized to a range of [0,1].

4 Results and Analysis

We submitted five runs that are different in the number of retrieved documents for entity query, and the types of patterns used for answer selection. Given our choice of simple pattern learning algorithm and a pattern-matching baseline design, our results leave a lot to be desired.

Submission	F1	Recall	Precision
Run1	0.0326	0.0479	0.0247
Run2	0.029	0.0428	0.022
Run3	0.0255	0.0376	0.0193
Run4	0.0247	0.0363	0.0187
Run5	0.0111	0.0104	0.0119

These runs differ in the number top retrieved documents using query's background document and the patterns used. We have two sets of patterns, one smaller set but are validated as in Section 3.3, one much bigger but not validated due to lack of time. The order of the runs generally follow the trend of increasing number of documents retrieved and using smaller set of patterns. From the analysis, we learn that our larger set of patterns and using smaller top ranking documents (but not as small as in Run5) are the most effective.

The more instructive lesson learned is that there is much room for improvement in refining the patterns. Out of the 42 slots for evaluation, we are only able to correctly extract answers for 16 of them. Out of the 16 correct slots, 12 are for person entity. We will look more into improving our patterns by applying methodologies such as including cue words and more surface features.

5 Conclusion and Future Work

As our first attempt at the KBP Slot Filling task, we approach it with a baseline system and a simple pattern learning and matching algorithm. We are pleased to identify the weaknesses of our methodology. We plan on improving our system with more sophisticated methods including improving pattern learning rules, and applying supervised learning methods for slot value extraction.

Acknowledgments

We thank Dr. Chen and Professor Ji at City University of New York for making their KBP tools available for new participants at this task.

References

Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, Heng Ji. 2010. CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description, Proc. TAC 2010 Workshop

Xu Jian, Zhengzhong Liu, Qin Lu, Yu-Lan Liu, Chenchen, Wang. 2011. PolyUCOMP in TAC 2011 Entity Linking and Slot Filling, Proc. TAC 2011 Workshop

Yan Li, Xiaoning Li, Hanying Huang, Yang Song, Cheng Chang, Liaoming Zhou, Jing Xiao, Dian Yu, Weiran Xu, Guang Chen, Jun Guo. 2011. PRIS at TAC2011 KBP Track, Proc. TAC 2011 Workshop

Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky. 2009. Distant Supervision of Relation Extraction without Labeled Data. Proc. Of the Joint ACL-IJCNLP Conference, 1003-1011