OpenKN at TAC KBP 2015

Xinlei Chen, Yantao Jia, Yuanzhuo Wang, Yuxiao Chang, Manling Li
CAS Key Laboratory of Network Data Science and Technology
Institute of Computing Technology
Chinese Academy of Science
jiayantao@ict.ac.cn

Abstract

This paper describes the system OpenKN which we established in the TAC KBP 2015. In TAC KBP 2015, we participated in one track: Cold Start Track. In order to complete the task, we developed a five-step system which solves the problem of building knowledge base from a document collection of unstructured text. These five steps are pre-processing, relation extraction, cross-document co-reference resolution, inference, and post-processing. In relation extraction step, we use CRF model and rule-based pattern extractor; in cross-document co-reference resolution step, we use hierarchical clustering method to merge similar entities.

1 Introduction

The goal of TAC KBP 2015 is to develop and evaluate technologies for building and populating knowledge bases (KBs) from unstructured text. It contains several tracks, and we participate in Cold Start Track this year.

The Cold Start KBP track builds a knowledge base from scratch using a given document collection and a predefined schema for the entities and relations that will compose the KB. Given a collection of documents, the Cold Start KB Construction system must find all PER, ORG, and GPE entities mentioned in the collection, create a KB node for each entity, and link each name mention to the correct entity node in the KB.

For this purpose, we propose a system which includes five parts to finish this task: pre-processing for preparing entities, relation extraction (regarded as slot filling, aims to extract relations by both CRF-based (Conditional Random Field) and rule-based methods), cross-document co-reference resolution to merge similar entities, inferring relations from already-known relations, and post-processing.

The paper is organized as follows. Section 2 describes our system. Preparation of the training data is explained in Section 3. Section 4 lists the evaluation results of the task. Finally we conclude the paper in Section 5 and present related references.

2 System description

Our proposed system contains five steps, i.e., pre-processing, relation extraction, cross-document co-reference resolution, inference, and post-processing, as illustrated in Figure 1.

More specifically, in the pre-processing step we extract all named entities from the corpus and save their type and offset, intra-document co-reference resolution is also included. Then we extract entities' relations from corpus (i.e. slot filling) in two ways: CRF-based and rule-based. Now the task turns into co-reference resolution of the cross-document entities. It is implemented by name similarity. After this step is inference, which supplies the existing relations. Finally, we utilize the post-processing component to remove wrong results and format final results. In the following, we explain each step in detail.

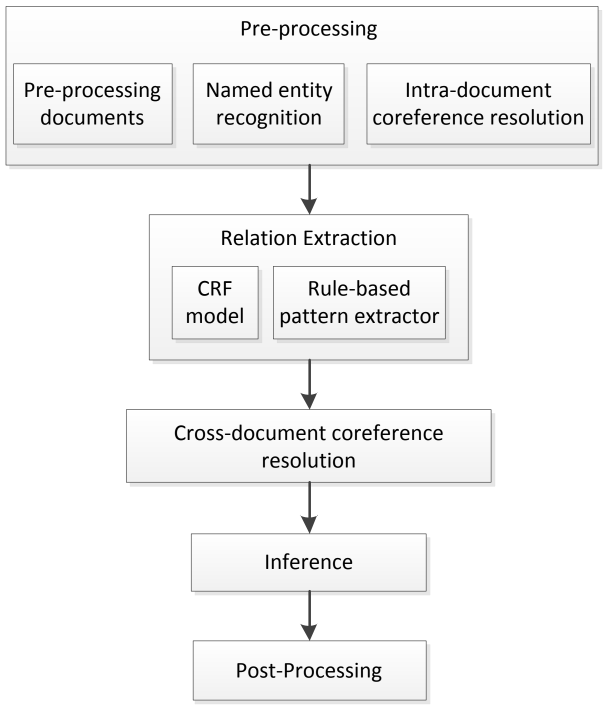

Figure 1 Architecture of the system

2.1 Pre-processing

First, the documents are pre-processed in three aspects:

1) Delete the quote contents which appear in forum posts, such as"<quote orig_author=…>", to avoid reprocess these pieces of content. For fear of offset's change, we simply replace them by whitespaces.

2) Get documents' publish dates which are used to infer date about entities (in most cases is death date, see details in section 2.4).

3) Get the start position of text.

We use the Stanford NER tool (Finkel et al., 2005) for named entity recognition among the corpus. Offsets and types need to be saved. After entities are extracted, entities' dates are added. For newswires, the date is article's publish date; For forums, the date is post's date.

Intra-document co-reference resolution is also included in this step. For entities which aim to same thing, there is only one parent, whose children are the rest. There are some standards about intra-document co-reference resolution:

a) Entities in titles shouldn't be parent for their lack of contextual information.

b) Child is a part of parent and has the same type with parent.

c) If there are more than one candidate parent to child, we choose the closest one.

d) For entities whose type is person, we deal with them solely by dividing into first name, middle name and family name.

2.2 Relation Extraction

The relation extraction step consists of two modules, namely, the CRF (Conditional Random Field) module and the rule-based pattern extractor module. In the CRF module, we use the training data (see Section 3 for preparation of the training data) to train a CRF model for each slot, then use the model to extract the corresponding slots for each entity in the relevant documents. In the rule-based pattern extractor module, for each entity, we use the patterns deduced from the training data to extract its slots. The choice of these two modules depends on the characteristic of the slots. For those with enough training data (e.g., per:country_of_birth, per:date_of_birth), CRF module is preferred. Otherwise, the rule-based pattern extractor module takes charge of this step.

More precisely, we use the CRF part of Stanford NER tools to learn and extract slots of entities via cross validation. The training data comes from previous KBP tracks and the reference KB by segmenting documents into sentences. For those sentences, we divide them into positive and negative examples, according to whether the sentence contains the slots of the given entity. As for the features of the CRF module, many were used in the literature, such as the first token of the sentence, the Part-Of-Speech tags (Wu and Weld, 2007). In our system, we find that for each sentence, the combination of four features performs the best, namely, the words in the sentences, the Part-Of-Speech tags of the sentence, the entity type of words, and the root (single headed node) of the dependency parsing tree of the sentence. The Part-Of-Speech tags analysis utilizes the Stanford Log-linear Part-Of-Speech Tagger (Toutanova et al. 2003), and the dependency parsing works by using the Stanford parser (Marneffe and Manning, 2008).

The rule-based pattern extractor module works as follows. It firstly describes the rule-based patterns in terms of the context free grammar with respect to the target slot of entities. Then these grammars are compiled under the cascaded finite-state transducers (CFT). With the aid of CFT, the slots are extracted under the collision detection phrase. More specifically, the rule-based patterns are firstly represented in the form of triplets (Heads, Arguments, Body). For example, to extract the slot per:age, we can define the pattern as follows:

AGE(person,age):(DIST_5,"_person{NAME}", "_age{AGE}","year's old"),

where the parameter DIST_n is used to define the distance (i.e., n words) between the followed two arguments. In particular, when the sentences match two or more rules, in order to boost the extraction efficiency, we adopt the inversed index to store the mapping from arguments to their locations in the corresponding patterns.

After defining these patterns, we also consider the bootstrapping technique for the purpose of recursive extraction. This can be accomplished by training the data sets and tuning the parameters appearing in the body of the rules. For example, these parameters may include the distances between arguments, the order of arguments, the distances between arguments by analyzing the results of dependency parsing.

2.3 Cross-document Co-reference Resolution

To address the tagging of entities, the system employs two steps to cluster the cross-document entities across target documents. Firstly, it employs the hierarchical clustering method to cluster the entities in terms of the context similarity and name similarity between entity mentions. Secondly, the heuristic rule aiming to augment the cluster results is proposed.

We use acronym expansion matching in the document text. For example, PA is expanded to “Pennsylvania” state in several different documents in the training data set. Thus, they are mapped to the same identifier.

2.4 Inference

In inference module, we apply these rules:

1) For an entity which has value about slot “city”, we can infer corresponding "stateorprovince" and "country" by gazetteer, i.e., the geographical dictionary such as GeoNames. Similarly, "country" can be inferred from "stateorprovince".

2) For per:date_of_birth, per:date_of_death, per:age, given two of these three slots, the third one can be inferred (except birth && age -> death, because someone who has birthdate and age may not die yet). For example, if A died in 2010 at age 78, so we can infer that A was born in 1932.

3) Rules for family relationships, which is illustrated in Table 1.

Table 1 Rules for inferring family relationships

A --- B	B --- C	A --- C
children	siblings	children
children	spouse	other_family
children	children	other_family
spouse	children	children
spouse	parents	other_family
spouse	siblings	other_family
parents	parents	other_family
parents	siblings	other_family
parents	spouse	parents

4) For results for the slots describing date which doesn’t express year/month/day explicitly, such as “died in Tuesday”, we transform it into standard date format according to the calendar.

5) For a person entity whose title is CEO, president, vice-president, or other titles which represent top employees, and this person entity has slot “per:employee_or_member_of”, we can infer slot “org:top_members_employees”.

6) For every slot which has inverse slot, we add the inverse relation of this slot.

2.5 Post-processing

To correct the errors in the slots extracted by the Filler component, we introduce the post-processing step. Specifically, we mainly use some rules, which are listed below.

1) The values of some slots must be of certain particular type. For example, when slots describe relations between people (e.g. spouse, children), the type of the results must be person (PER). This can be examined by means of the Stanford NER tool (Finkel et al., 2005).

2) Standardization of dates. Convert all answers which represent dates to standard date format ”XXXX-XX-XX”.

3) Delete unreasonable answers. For example, the results for the slots describing age should be a number usually larger than 0 and smaller than 130 respectively.

3 Preparation of the training data

In order to use CRF model, we must train CRF model first. The source of training data is the main problem. KBP provided true answers of previous years, and what we use are these true answers. After extracting the answers from standard answer, given an entity name and its slot-filling answer, we take these two words as well as the query and retrieve a collection of sentences. In order to provide train files for CRF Classifier, we convert these to a five-column file, see in Table 2:

Table 2 Explanation of columns in train file

column	explanation
1	Original words
2	Part-of-speech
3	Entity type (PER or ORG or GPE)
4	Root of the dependency tree
5	Labels provided for manually annotation

After the manually annotation of these files, we can use them to train CRF models (one model per slot) and subsequently generate relation extraction results.

4 Evaluation results

The results for Cold Start track are divided into two sub-tasks (Slot Filling and Entity Discovery) and are illustrated in Table 3 and Table 4.

For Cold Start Task, we submitted four runs. These four runs are operated as follows:

Run1: Use the pattern based extractors to fulfill slot filling and consider co-reference resolution and post-processing process.

Run2: Use the pattern based extractors to fulfill slot filling, but do not consider co-reference resolution.

Run3: Considering both Wikipedia linking result and cross-document co-reference resolution.

Run4: Considering Wikipedia linking result and discarding cross-document co-reference resolution.

Table 3 and Table 4 are the results (Precision, Recall and F1-measure) of Entity Discovery and Slot Filling, respectively.

Table 3 Entity Discovery Result

	strong_mention_match			strong_typed_metion_match			mention_ceaf			b_cubed		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	0.229	0.175	0.198	0.210	0.161	0.182	0.196	0.150	0.170	0.176	0.103	0.130
2	0.229	0.175	0.198	0.211	0.162	0.183	0.173	0.133	0.150	0.219	0.094	0.131
3	0.229	0.175	0.198	0.210	0.161	0.182	0.199	0.152	0.172	0.176	0.105	0.132
4	0.229	0.175	0.198	0.210	0.161	0.182	0.173	0.132	0.150	0.222	0.092	0.130

Table 4 Slot Filling Result

	hop0_P	hop0_R	hop0_F	hop1_P	hop1_R	hop1_F	All_P	All_R	All_F
1	0.4889	0.0056	0.0111	0	0	0	0.4889	0.0035	0.0070
2	0.7500	0.0008	0.0015	0	0	0	0.7500	0.0005	0.0010
3	0.4384	0.0081	0.0160	0	0	0	0.3951	0.0051	0.0101
4	0.8000	0.0010	0.0020	0	0	0	0.8000	0.0006	0.0013

From these results we can conclude that cross-document co-reference resolution can help improve the recall at the cost of precision, and linking with Wikipedia seems has little effect.

It should be mentioned that we do not carry out the hop-1 typed cold start knowledge base construction because we do not keep the extraction of knowledge from the obtained facts.

5 Conclusion

In this paper, we present the OpenKN system for the Cold Start Track of the KBP 2015. The proposed system contains pre-processing, relation extraction, cross-document co-reference resolution, inference, post-processing five steps corresponding to the task. The official evaluation results are also provided.

References

Cucerzan S. 2011. TAC Entity Linking by Performing Full-document Entity Extraction and disambiguation. TAC KBP.

Finkel J. Rose, Grenager T. and Manning C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 363-370.

Marneffe M. D. and Manning C. 2008. Stanford Dependencies manual.

Toutanova K., Klein D., Manning C., and Singer Y. 2003. Feature-rich Part-of-Speech Tagging with a cyclic dependency network. Proceedings of HLT-NAACL 2003, 252-259.

Wu F. and Weld D. 2007. Autonomously semantifying Wikipedia. Proceedings of the sixteenth ACM conference on information and knowledge management, 41-50.

Lin H, Zhao Z, Jia Y, et.al. OpenKN at TAC KBP 2014. TAC KBP.