Extraction system for Personal Attributes Extraction of CLP2014

Zhen Wang
ERTIM-INALCO / 2, rue de Lille, 75007, Paris, France
wangzhen1027@gmail.com

Abstract

This paper presents the design and implementation of our extraction system for Personal Attributes Extraction in Chinese Text (task 4 of CLP2014). The objective of this task is to extract attribute values of the given personal name. Our extraction system employs a linguistic analysis following by a dependency patterns matching technique.

1 Introduction

This is the first year that we take part of in CLP's Personal Attributes Extraction in Chinese Text task. The goal of this task is to extract specific attributes values of given personals names, such as, birth_date, birth_city, children, title etc. from the collections of unstructured Chinese texts. We are required to fill an extracted result into a single attribute slot.

Our approach is based on dependency patterns matching process, which is similar to the works of Xu et al. (2013).

2 System Architecture

In order to accomplish the task, we have proceeded in four steps :

- a pre-processing module;
- an extraction treatment and alignment;
- an ontology alignment;
- a result generation.

Pre-processing module consists of a morphsyntactic analysis and a parsing. Morphosyntactic analysis is used for word segmentation and part of speech tagging. Operations are based on dictionnaries and linguistics rules. Unknown words, especially proper nouns are detected in this step. A type, like "person", "location", "organization" or "unknown" for each proper noun is attributed. The other unknown words received several hypothetical categories, such as "noun", "verb", etc.. A statistical n-gram part of speech model is used for disambiguation. As a result, we only keep one analysis solution among whole solutions. This solution includes lemmas, POS tag, semantic properties and words positions. Our parsing uses dependency grammar. Based on words postions and categories, we build relations between two words and associate with a type, like SV for Suject-Verb,VO for Verb-Object, etc.. Negation and anaphora problems are treated after parsing. All segmentation and parsing results are reported into an XML file.

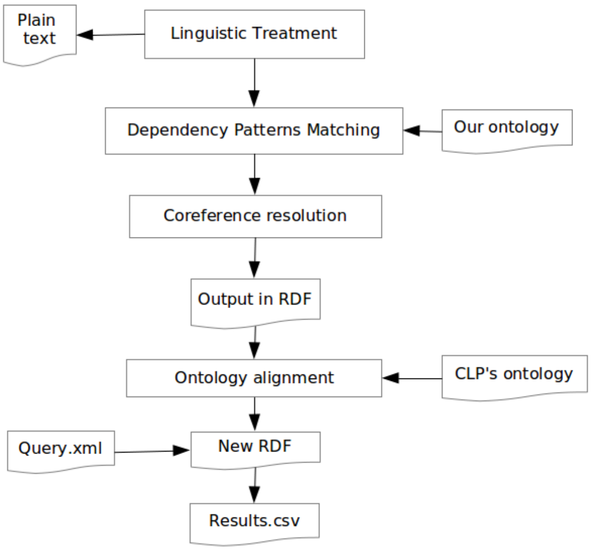

Figure 1: Process for task 4

Extraction treatment uses reported patterns to match dependency relations in the XML file. The extracted informations are saved into an RDF format file. Alignment process is used to group same classes and to remove duplicates in RDF file. The RDF file has to be conform to our ontology.

Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 189–191,
Wuhan, China, 20-21 October 2014

We created a software to align our ontology to CLP's. The idea is to generate a new RDF file by collecting personal name classes and personal attributes classes from all classes. Given person names is used to question the new RDF file. When a person name is matched to one of them in query, each attribut is generated as a line and saved into a CSV file.

2.1 Dependency patterns matching

Dependency patterns are used to extract information from the parsing results. A dependency pattern is composed of dependency relations elements and of a class of our ontology (see example in figure 2).

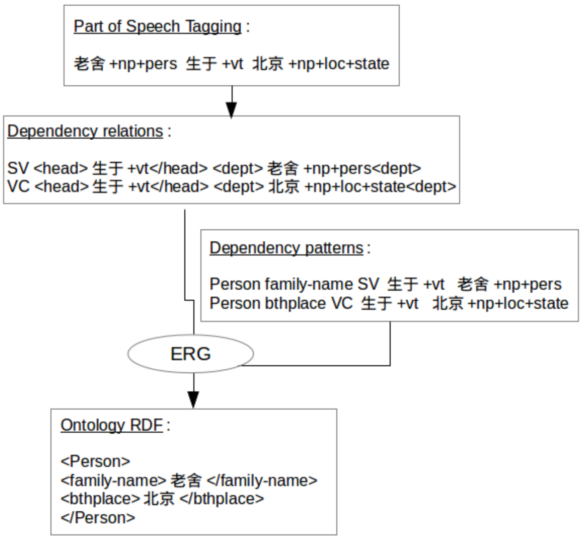
Part of Speech Tagging :
老舍 +np+pers 生于 +vt 北京 +np+loc+state
Dependency relations :
SV <head> 生于 +vt</head> <dept> 老舍 +np+pers<dept>
VC <head> 生于 +vt</head> <dept> 北京 +np+loc+state<dept>
Dependency patterns :
Person family-name SV 生于 +vt 老舍 +np+pers
Person bthplace VC 生于 +vt 北京 +np+loc+state
ERG
Ontology RDF :
<Person>
<family-name> 老舍 </family-name>
<bthplace> 北京 </bthplace>
</Person>

Figure 2: Process of extraction

ERG (Extraction Rule Generator) begins by getting a list of relations, then based on these relations, ERG selects the corresponding patterns. By using these patterns, ERG generates triplets RDF to represent the extracted informations. One matching between a relation and a pattern is enough to generate one triplet. The position of head or dependancy is assigned to be the triplet's ID. ERG repeats this process sentence by sentence. All triplets with same ID are grouped together in the end of process.

2.2 Coreference resolution

Coreference resolution is used to group equal elements, such as events, actions and named entities (persons, organizations, locations, objets, etc.). We make some attributes as decisive elements for equal elements identification. They can be personal family name, organization name or location name. For the equal elements, we change their ID to be the same.

2.3 Ontology alignment

In order to fill the slot, we have to transform our ontology(see example in figure 3) to CLP's. A software was created for this interest (see examples in table 1). After getting a personal named entity and its id, we search all classes containing this id and make these classes as sub-classes of the personal named entity. By aligning the classes with those in CLP's ontology, we transform our RDF result.

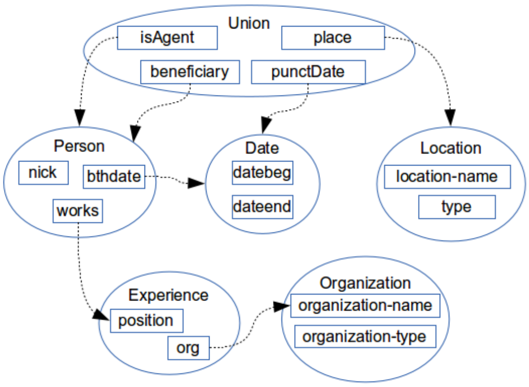

Figure 3: Example of our ontology and their links

CLP's ontology	Our ontology
PER:Alternate Names	Person:nick
PER:Age	Person:age
PER:Date of Birth	Person:bthdate
PER:City of Birth	Person:bthplace + Location:location-name + Location:type=city
PER:Spouses	Union:beneficiary=PER1 + Union:beneficiary=PER2

Table 1: Examples of ontology alignment

For some basic personal attributes, we have equal classes, so the alignment is easy. But for some others, we have to take two or more classes to align with one class of CLP's ontology. For instance, in order to fill the slot *PER:City of Birth*, we have to find in our RDF result that a *Person:familyname* is equal to a given name in query, and that it has a *bthplace* which is pointed to a *Location*. We have to ensure that the *type* of this *Location* is equal to "city". When all these conditions are fulfilled, the mentioned slot can be filled. Another example, in order to generate *PER:Spouses*, we have to find *Union* where there is two and only two *beneficiairy*.

The principal advantage of this step is to merge the named entities of different texts/files. Before the entity creation step, we check if it already exists in reported file.

2.4 Result generation

The objective of this step is to parse queries, create slots for each given personal name and to interrogate ontology in order to verify if it has a corresponding entity request and set all informations which are already integrated during the transformation step.

3 Results and error analysis

A lot of slots haven't been filled in this bake-off. Our single score is 0.0043 and SF value 0.004311. Here are the main dysfunctions : some personal names weren't identified because of morphsyntactic analysis: given name without family name, family name without given name, these are the cases that we have not treated yet; some relations between personal name and attribut haven't been established because of parsing. The main reason of a bad parsing is that the two elements (like personal name and attribute) are located in two differents clauses. Another reason is that anaphora between two sentences, omission of suject or possessive suject, are not solved yet. Some attributs haven't been extracted because extraction rules weren't created. Some slots have not been filled because of name matching between query and ontology, that did not work correctly. All foreign personal names with a " **dot** " were extracted in CSV because the matching between foreign personal names in query and in ontology did not work. The name is writen as "*given name* **dot** *family name*" in query but in ontology it is writen as "family name given name" which is the order used for chinese names but without the " **dot** ".

4 Conclusions

The paper presents our submission to the Personal Attributes Extraction in Chinese Text. Our system uses a linguistic analysis as pre-processing and an extration rule generation which employs a dependency patterns matching. In the future, we will improve our extraction rules and treat the relations between clauses. We will find a solution for anaphora problems between sentences. We also plan to expand the queries (see Xu et al. (2013)) and register the names similarity.

Acknowledgments

We would linke to thank the National Research Agency, the project for reference ANR-09-CSOSG-08-01, for their help in producing this work.

References

Sheng Xu, Chunxia Zhang, Zhendong Niu, Rongyue Mei, Junpeng Chen, Junjiang Zhang, and Hongping Fu. 2013. Bits slot-filling method for tac-kbp 2013. Technical report.