Ghent University - IBCN Participation in the TAC-KBP 2015 Cold Start Slot Filling task

Team UGENT_IBCN

Lucas Sterckx, Thomas Demeester, Johannes Deleu, Chris Develder

Ghent University - iMinds

lucas.sterckx@intec.ugent.be

Abstract

This poster describes the system of team UGENT_IBCN for the TAC KBP 2015 Cold Start (slot filling variant) task. The slot filling system uses distant supervision to generate training data for feature-based relation classifiers, combined with feature labeling and pattern based extractions. An overall performance 18.9% in $F_1$was obtained, which is an increase of 5% compared to the team's 2014 system.

Introduction

This was the second participation of team UGENT_IBCN in the Knowledge Base Population - Cold Start Slot Filling variant, the successor of the English Slot Filling track. Our system builds upon last year's system [1] and uses techniques described in [2]. The main relation extractor is based on distant supervision combined with minimal amounts of supervision.

System Overview

Figure 1 shows an overview of the slot filling system. Interactions between different components of the system and the different sources of data are visualized by arrows.

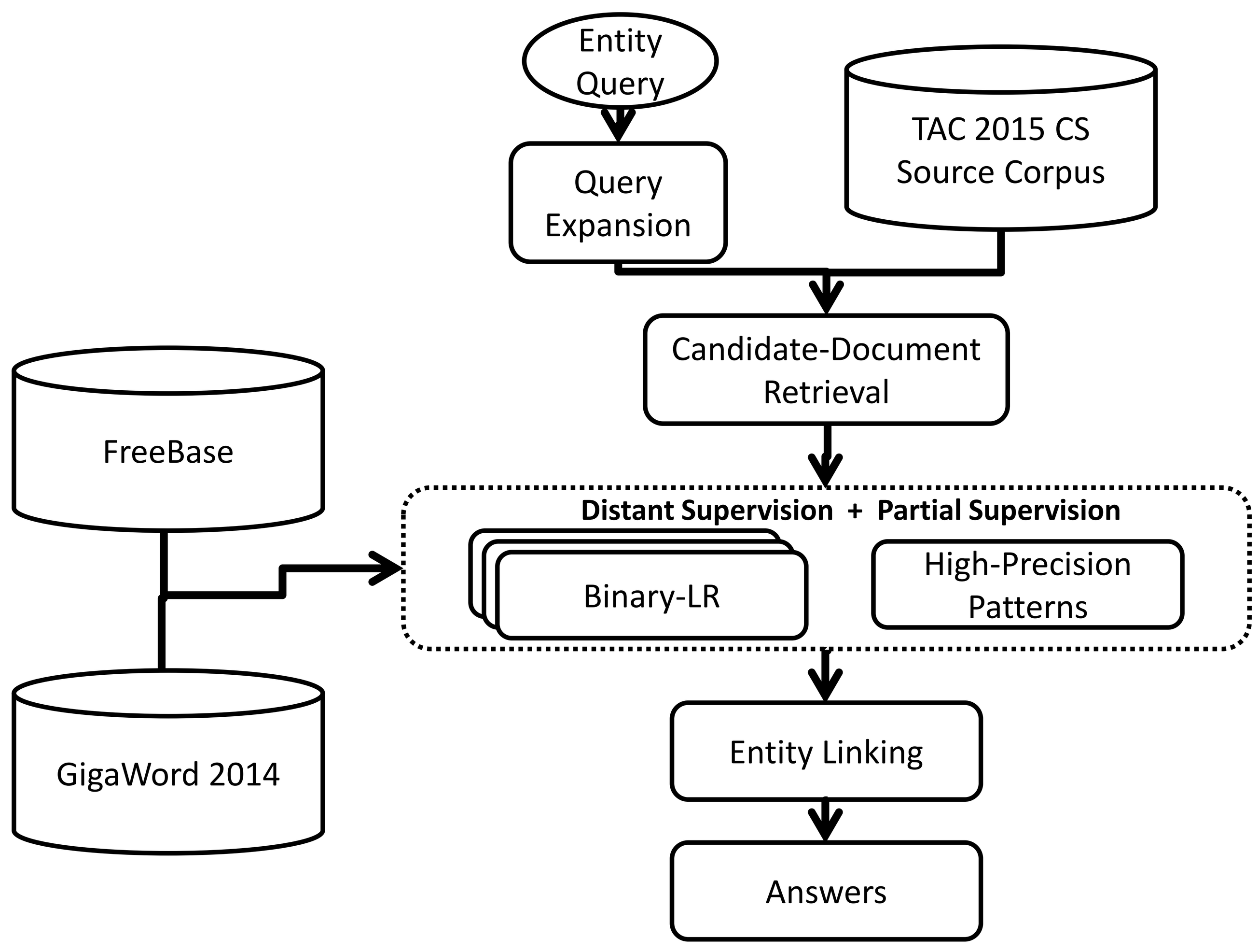

Figure 1: System Overview

Query Expansion and Document Retrieval

First we retrieve all documents containing entity queries (person or organization) from the TAC Cold Start 2015 source document collection. We expand the query by including all **alternate names** obtained from Freebase and Wiki-redirects for the given query. This year, **no Named Entity Disambiguation** was included, which resulted in wrong slot fillers for ambiguous entities, e.g., Gotham (New-York City), Blues (Everton FC).

Named Entity Tagging

In each retrieved document we identify relevant sentences by searching for any of the entities from the expanded set of query entity names. This year we include a **co-reference module** and resolve all synonymous noun phrases to a single entity. Noun phrases linked to any of the queries are used as subject entities for possible filler extractions. Next, we assign all slot candidates from the relevant sentences with a type (e.g., title, state-or-province). Slot candidates are extracted using the **Stanford 7-class Named Entity Recognizer** and assigned a type using **lists of known candidates** for each type. Lists were expanded this year with those from the *RelationFactory* system.

Relation Extraction Classifiers

For each combination of tagged entities with a query entity, we perform a classification of a type-matching relation from the TAC Cold Start schema. For classification we extract features from each candidate phrase and use **binary Logistic Regression** (LR) classifiers together with a small selection of **High-Precision patterns**.

Binary LR classifiers detect the presence or absence of a relation in the sentence for the query entity and a possible slot filler. All LR classifiers use the same set of features, which is a **combination of dependency tree features, token sequence features, entity features, semantic features and an order feature**. Next to feature-based classification, a small selection of high precision patterns was used, some obtained from feature labeling and others from the *Relation Factory* KBP system. If an exact match in the surface text between entities and a pattern is detected, the probability of the classifier is set to 1.

Whereas in [1] instance labeling was used to self-train relation classifiers and reduce noisy mentions, we focus on learned features from an initial distantly supervised classifier. In a second stage, most **confident positive features** learned by the initial classifier are presented to an annotator with knowledge of the semantics of the relation and labeled as true positive, false positive (noise) or ambiguous. The collection of training instances is then **filtered** by only including mentions with one of the true positive labeled features present, after which a second classifier is trained. To represent features, we use shortest dependency paths as shown in Table 1.

Relation	Top Feature	Assessment
top_members_employees	PER $\xleftarrow{appos}$ $executive$ $\xrightarrow{prep\_of}$ ORG	✓
	PER $\xleftarrow{appos}$ $chairman$ $\xrightarrow{appos}$ ORG	✓
	ORG $\xleftarrow{nn}$ $founder$ $\xrightarrow{prep\_of}$ PER	✗
children	PER-2 $\xleftarrow{appos}$ son $\xrightarrow{prep\_of}$ PER-1	✓
	PER-1 $\xleftarrow{appos}$ $father$ $\xrightarrow{prep\_of}$ PER-2	✓
	PER-2 $\xleftarrow{nn}$ $grandson$ $\xrightarrow{prep\_of}$ PER-1	✗
city_of_birth	PER $\xleftarrow{rcmod}$ $born$ $\xrightarrow{prep\_in}$ LOC	✓
	PER $\xleftarrow{nsubj}$ $mayor$ $\xrightarrow{prep\_of}$ LOC	✗
	PER $\xleftarrow{appos}$ $historian$ $\xrightarrow{prep\_from}$ LOC	✗
schools_attended	PER $\xleftarrow{nsubj}$ $graduated$ $\xrightarrow{prep\_from}$ ORG	✓
	PER $\xleftarrow{dep}$ $student$ $\xrightarrow{prep\_at}$ ORG	✓
	PER $\xleftarrow{appos}$ $teacher$ $\xrightarrow{prep\_at}$ ORG	✗
(org:)parents	ORG-2 $\xleftarrow{appos}$ $subsidiary$ $\xrightarrow{prep\_of}$ ORG-1	✓
	ORG-1 $\xleftarrow{appos}$ $division$ $\xrightarrow{prep\_of}$ ORG-2	✓
	ORG-2 $\xleftarrow{prep\_to}$ $shareholder$ $\xrightarrow{dep}$ ORG-1	✗

Table 1: Example of feature annotation.

Entity Linking

In a final stage, the slot fillers extracted from the different documents are combined in an Entity Linking step. We link the entities from different documents and combine the extracted relation-tuples to obtain a final set of extracted relations. The output of this step consists of a list of all possible relation-tuples, if the relation can have multiple tuples, e.g., for person_cities_of_residence. If only one relation instance is allowed, e.g., for city_of_birth, we choose the relation-tuple with the highest evidence. The evidence score for each relation-tuple is obtained by choosing the maximum evidence of all relation instances of this relation-tuple, i.e., the highest evidence-score given by the classifier of all sentences that express the relation-tuple.

Results

System Development

The system was developed on data from the 2013 and 2014 English Slot Filling task. We found that important parameters to fine-tune, in order optimize F_1-scores, are **classifier regularization, the ratio of true and false examples and the classification threshold**. The highest micro-F_1scores obtained for these development sets are shown in Table 2. Compared to classifiers used in 2014 participation in the English Slot Filling Task (ESF), large increases in performance ($+10\%$) were attained.

	2013 ESF			2014 ESF		
System	P	R	F_1	P	R	F_1
2014 Classifiers	42.8	19.7	27.0	28.0	18.6	22.4
2015 Classifiers	37.7	**37.2**	37.5	35.7	33.7	34.7
Patterns	**60.6**	12.1	20.2	**53.0**	8.7	14.9
Classifiers+Patterns	40.2	36.6	**38.6**	36.9	**35.9**	**36.4**

Table 2: Results on development sets.

Cold Start Results

Four runs were generated using the same set of classifiers. Submissions differ in the selection of thresholds put on the maximum amount of fillers and confidence values. In each of the runs, at most, **10 fillers with highest confidence** were used to generate the second hop queries to reduce the generation second-hop fillers for wrong first-hop fillers. The micro-averaged P/R/F1 at each hop level for the different runs of the slot filling variant of the Cold Start task are shown in Table 3. Compared to last year's participation an increase of 5% in $F_1$was obtained, placing 5th among 20 KBP systems from all variants and second out of 12 systems participating in the slot filling variant.

	Hop 0			Hop 1			All Hops		
Run	P	R	F_1	P	R	F_1	P	R	F_1
2014 - Best Run	24.7	16.6	19.9	7.5	4.9	5.9	16.7	11.1	13.3
2015 - 1 (High Precision)	**33.5**	19.9	**25.0**	**11.5**	6.0	7.9	**26.0**	14.8	**18.9**
2015 - 2 (Higher Recall)	31.8	20.1	24.6	9.1	6.5	7.5	22.8	15.3	18.1
2015 - 3 (Highest Recall)	26.5	**22.5**	24.3	9.8	**7.3**	**8.4**	20.8	**16.9**	18.6

Table 3: Results of the different hops and the aggregate in the slot filling variant of the 2015 Cold Start task.

References

[1] Matthias Feys, Lucas Sterckx, Laurent Mertens, Johannes Deleu, Thomas Demeester, and Chris Develder. Ghent University-IBCN participation in TAC-KBP 2014 slot filling and cold start tasks. In *7th Text Analysis Conference, Proceedings*, pages 1–10, 2014.

[2] Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. Using active learning and semantic clustering for noise reduction in distant supervision. In *4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014)*, pages 1–6, 2014.