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

Lucas Sterckx, Thomas Demeester, Johannes Deleu, Chris Develder
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
{firstname.lastname@intec.ugent.be}

Abstract

This paper 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 23.3% in micro-mean F_1 was obtained, which is an increase of 10% compared to the team's 2014 participation.

1 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 is based on the team's 2014 system (Feys et al., 2014) and uses techniques described in (Sterckx et al., 2014). The relation extractor is based on Distant Supervision together with minimal amounts of supervision.

In the following Sections, we give a brief overview of the system and describe different components of the Knowledge Base Population system. A more elaborate discussion of the training with Distant Supervision is given in Section 3. Finally, results and a conclusion are given in Sections 4 and 5.

2 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 using arrows. We discuss those parts of the system which act at run time for the generation of slot fillers.

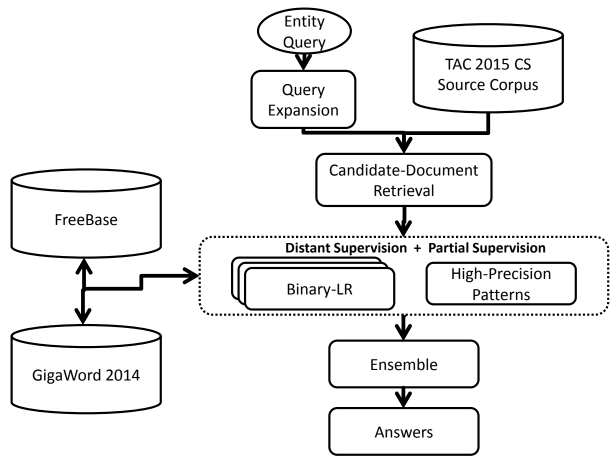

Figure 1: System Overview

2.1 Query Expansion and Document Retrieval

We first 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. When we do not retrieve any alternate names, we clean the query, e.g., remove middle initials for persons, remove any company suffixes (LLC, Inc., Corp.) and repeat the search for alternate names using this filtered query. For indexing and search of the source collection we use the

Whoosh[1] module for Python. This year no Named Entity Disambiguation was included, which resulted in wrong slot fillers for ambiguous entities, e.g., Gotham (New-York), Blues (Everton FC).

2.2 Named Entity Tagging

Each document was preprocessed using components of the Stanford CoreNLP toolkit (Manning et al., 2014). 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 (Finkel et al., 2005) and assigned a type using lists of known candidates for each type. Lists were expanded this year with those from the *RelationFactory* system (Roth et al., 2014).

2.3 Relation 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. These correspond for the most part to the features used in (Sun et al., 2011). A complete overview of the used features is given in Table 1 using an illustration of the features for example relation-tuple <Ray Young, General Motors> and the sentence "Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi"[2].

[1] `http://pythonhosted.org/Whoosh/`

[2] The same example sentence as used in (Sun et al., 2011)

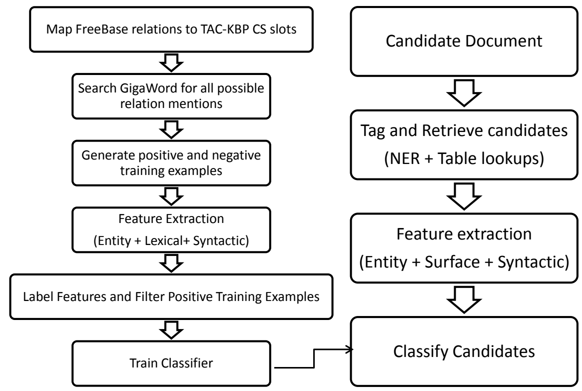

Figure 2: Classifier Overview

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 (Roth et al., 2014). If an exact match in the surface text between entities and a pattern is detected, the probability of the classifier is set to 1.

2.4 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 is allowed to 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 probability assigned by the classifier. 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 probability given by the classifier of all sentences that express the relation-tuple.

3 Distant Supervision with Feature Labeling

Distant supervision (DS) has become an effective way for generating training data in the slot filling task, as proven in many top-performing submissions in previous years (Surdeanu and Ji, 2014). In this year's competition we looked into ways of combin-

Table 1: Overview of different features used for classification for the sentence “Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi”.

Feature	**Description**	**Example Feature Value**
Dependency tree	Shortest path connecting the two names in the dependency parsing tree coupled with entity types of the two names	PERSON←appos←officer → prep_of→ ORGANIZATION
	The head word for name one	said
	The head word for name two	officer
	Whether *1dh* is the same as *e2dh*	false
	The dependent word for name one	officer
	The dependent word for name two	nil
Token sequence features	The middle token sequence pattern	, the chief financial officer of
	Number of tokens between the two names	6
	First token in between	,
	Last token in between	of
	Other tokens in between	{the, chief, financial, officer}
	First token before the first name	nil
	Second token before the first name	nil
	First token after the second name	,
	Second token after the second name	said
Entity features	String of name one	Ray_Young
	String of name two	General_Motors
	Conjunction of *e1* and *e2*	Ray_Young–General_Motors
	Entity type of name one	PERSON
	Entity type of name two	ORGANIZATION
	Conjunction of *et1* and *et2*	PERSON–ORGANIZATION
Semantic feature	Title in between	True
Order feature	1 if name one comes before name two; 2 otherwise.	1
Parse Tree	POS-tags on the path connecting the two names	NNP→DT→JJ→JJ →NN→IN→NNP

	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.

	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 - Run 1 (High Precision)	**34.5**	25.0	**29.0**	**14.4**	9.9	11.7	**27.9**	19.8	**23.2**
2015 - Run 2 (Higher Recall)	33.0	25.2	28.6	12.5	10.6	11.5	25.4	20.2	22.15
2015 - Run 3 (Highest Recall)	28.0	**27.4**	27.8	13.1	**13.7**	**13.36**	22.7	**22.7**	22.7
2015 - Run 1 (Macro Mean)	-	-	34.29	-	-	13.3	-	-	27.0

Table 3: Results of the different hops and the aggregate in the slot filling variant of the 2015 Cold Start task[1]

ing DS with minimal amounts of supervision.

The left side of Figure 2 shows the different steps for the generation of training data. We start by mapping FreeBase relations to KBP slots and subsequently search the full GigaWord corpus for possible mentions of these relations, i.e., two entities from a fact tuple co-occurring in sentences. Negative examples are all phrases with co-occurring entities for relations which are not present in FreeBase.

Whereas in (Feys et al., 2014) instance labeling was used to self-train relation classifiers and reduce noisy mentions, we focus on learned features from an initial DS 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.

Our strategy is related to the *guidelines* strategy from Pershina et al. (Pershina et al., 2014), but instead of extracting guidelines using a fully annotated corpus, we label features entirely based on distant supervision. We then use a strategy from active learning literature, feature certainty (Attenberg et al., 2010) to rank and present features to the annotator, in order to further reduce the labeling effort. Feature Certainty is intuitively an attractive choice, as the goal is to reduce most influential sources of noise as quickly as possible e.g., for the relation *founded_by* there are many persons that founded the company which are also *top_members*, leading to many instances that we wish to remove when cleaning up the training data for the relation *founded_by*.

In the final set of classifiers an ensemble of two classifiers was chosen and confidences for relation extraction were averaged.

4 Results

4.1 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, large increases in performance (+10%) were attained.

4.2 Cold Start Results

Four runs were generated using the same set of classifiers. Submissions differ in the selection of thresholds set on the amount of fillers and confidence values. For each of the runs, at most, 10 fillers with the highest confidences were used to generate the second hop queries, this 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 almost 10% in $F_1$was obtained, placing fourth among 20 KBP systems from all variants and second out of twelve systems participating in the slot filling variant.

5 Conclusion

This paper described our second setup for the slot filling variant of the Cold Start task. We significantly increased the performance of our previous relation extraction classifiers by incorporating noise reduction of the distantly supervised training data using feature labeling and high-precision patterns.

References

Josh Attenberg, Prem Melville, and Foster Provost. 2010. A unified approach to active dual supervision for labeling features and examples. In *In European conference on Machine learning and knowledge discovery in databases*, pages 40–55.

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

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. 2014. Infusion of labeled data into distant supervision for relation extraction.

Benjamin Roth, Tassilo Barth, Michael Wiegand, Mittul Singh, and Dietrich Klakow. 2014. Effective slot filling based on shallow distant supervision methods. *arXiv preprint arXiv:1401.1158*.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2014. 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.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. New York university 2011 system for KBP slot filling. In *Proceedings of the Text Analytics Conference*.

Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. *Proc. Text Analysis Conference (TAC2014)*.