CMUML System for KBP 2015 Cold Start Slot Filling

Bryan Kisiel, Bill McDowell, Matt Gardner,
Ndapandula Nakashole, Emmanouil A. Platanios,
Abulhair Saparov, Shashank Srivastava, Derry Wijaya, Tom Mitchell
School of Computer Science
Carnegie Mellon University, Pittsburgh, PA
{bkisiel, ndapa}@cs.cmu.edu

Abstract

In this paper, we present an overview of the CMUML system for KBP 2015 English Cold Start Slot Filling (SF) task.The CMUML 2015 SF system aggregates the output of several semantic analysis sub-components that read natural language at the sentence level. These sub-components, which we refer to as **micro-readers**, have different reading capabilities. In addition, we ran a fraction of the queries on our 2014 CRF-based system. We also used our 2014 rule-based system

1 Introduction

In this paper, we describe the CMUML system for KBP 2015 Cold Start Slot Filling (SF) task organized by NIST. The system is different from our 2013 and 2014 approaches. In 2013, we used a combination of distant supervision (Mintz et al., 2009), stacking (Wolpert, 1992), and CRF-based structured prediction. In 2014, we added the following to the 2013 system: an inference mechanism, a rule-based predictor, and an Open-IE-based predictor. In 2015, we added components based on a completely different approach. We developed micro-reading components that perform sentence level analysis of documents. This type of reading is conceptually different from the type of machine reading embodied in our prior systems. In prior systems, the type of reading we implemented was based on pattern detection to identify mentions of various relevant relations. In contrast, this year our methods have the goal of achieving a more encompassing type of machine reading trying to understanding text at a sentence level rather than selective relation extraction. Given that this is a much more difficult task then pattern detection, our methods are still in the very early stages and therefore their performance is inferior to our prior methods. However, it is our belief that this is approach is on the right path towards deep language understanding. Our intention was to combine the predictions of the 2015 micro-readers with those of our prior systems. However, we could only run our prior systems on a small fraction of the 2015 questions due to a last minute system failure. Our overall 2015 system architecture is shown in Figure 1, 2015 micro-readers are highlighted in red. Below, we provide brief discussions of the new micro-readers, and the other components.

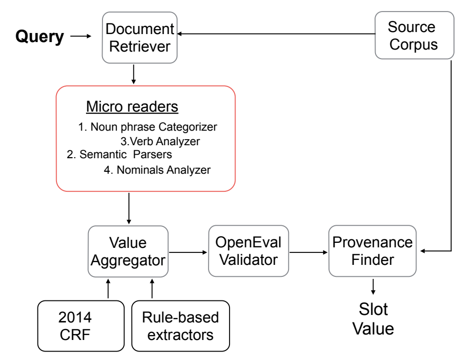

Figure 1: Overview of the CMUML 2015 Slot Filling (SF) system. The micro-readers, newly introduced in 2015 are highlighted in red. The OpenEval validator was not used in the official submission (CMUML1) as it requires web access.

2 Document Retrieval and Entity Matching

The KBP source documents were indexed using Lucene[1]. Now, given a query, this index was used to retrieve relevant documents. In order to identify relevant sentences in the document, we perform matching between the arguments in the query and the retrieved document.

The Entity Matcher aims to correctly map surface strings in documents to query entities, even when the strings are syntactically different from the text in the entity name. We therefore, leveraged the Freebase Annotations of the ClueWeb Corpora (FACC)[2] dataset recently released by Google. This corpus enabled us to generate synonym sets containing all surface strings that can refer to the same Freebase entity. During entity matching, we perform lookups against an indexed version of this synonym sets data. This significantly improved entity matching recall.

3 2015 Micro Readers

Once relevant documents were identified, we applied our micro-readers, described below.

StanfordCoreNLP: Before applying our own micro-readers, each document is processed using the StanfordCoreNLP pipeline: (`http://nlp.stanford.edu/software/corenlp.shtml`. The *tokenize, ssplit, pos, lemma, ner, parse, dcoref*, and *SUTime* modules were used.

Noun phrase categorizer: Predicts fine-grained semantic types for noun phrases. Used for type checking by other micro-readers that predict relations. This micro-reader categorizes noun phrases in context, it uses the context of a noun phrase to determine its type. The context is treated as features for a logistic regression classifier.

Semantic parser: We had two semantic parsers for predicting relations. One is an unpublished generative semantic parser. The other is our 2014 joint syntax and semantic CCG parser (Krishnamurthy and Mitchell, 2012; Krishnamurthy and Mitchell, 2014) which was trained to extract NELL(Carlson et al., 2010; Mitchell et al., 2015) relations from text using distant supervision.

[1] Lucene: http://lucene.apache.org/

[2] http://lemurproject.org/clueweb09/FACC1/

Verb analyzer: A reader based on learning which verbs express which relations. This micro reader uses an approach based on ontology alignment (Wijaya et al., 2013).

Nominal noun analyzer: A micro reader for extracting relationships from nominal nouns such as "British journalist John Murray" from which the nationality and job title of John Murray are extracted. This micro reader uses background knowledge about type sequences to learn which compound noun sequences express which relations.

3.1 Regular Expression Slot Filler

In 2015, we introduced an updated rule-based system, which specifies regular expressions for specific relations.

4 2014 System

4.1 CRF-based Extractor and Aggregator

This extractor takes tokenized sentences found to contain the query text, these are then converted into CRF instances, with several semantic annotations as features. When training, the tokens corresponding to the known slot filler value would be labeled with the relation being expressed; when making predictions, the CRF would be responsible for identifying the span of tokens representing a slot filler value along with the relation being expressed.

At prediction time, this process yields a set of sentences potentially expressing a variety of slot fillers, typically with significant redundancy. Redundant predictions were eliminated by way of identifying sentences expressing the same (relation, filler) pair, or where two filler values were deemed to be synonymous. Each filler is then assigned a confidence score based on the number of times it was found to be be expressed in the corpus.

4.2 Rule-based Extractor

This 2014 slot filler uses manually specified relation-specific rules to make predictions.

5 Slot Value Validation using OpenEval

Slot filler values were filtered using OpenEval (Samadi et al., 2013) to determine whether or not sufficient evidence for them could be found by

Run Id	Measure	Precision	Recall	F1
CMUML1	CSSF	0.6250	0.0040	0.0080
CMUML1	CSLDC	0.7500	0.0058	0.0116
CMUML2	CSSF	0.6098	0.0040	0.0080
CMUML2	CSLDC	0.6923	0.0058	0.0116
CMUML3	CSSF	0	0	0
CMUML3	CSLDC	0	0	0

Table 1: Official evaluation scores of various CMUML submissions.

querying the live web. Please note that this component was not used in CMUML1, the official submission, as web access was not allowed in the main submission.

OpenEval is an online information validation technique, which uses information on the web to automatically evaluate the truth of queries that are stated as multi-argument predicate instances (e.g., *drugHasSideEffect(Aspirin, GI Bleeding)*). It trains a classifier by taking a small number of instances of the predicate as an input and converting them into a set of positive and negative Context-Based Instances (CBI), which are used as training data. Each CBI consists of a set of features constructed by querying the open Web and processing the retrieved unstructured web pages. To evaluate a new predicate instance, OpenEval follows a similar process but then gives the extracted CBIs to the trained classifier to compute the correctness probability of the input predicate instance. To navigate the diversity of information that exists on the Web, it uses a novel exploration/exploitation search approach, which enables formulating effective search queries and increases the accuracy of its responses.

6 Provenance Finder

It was necessary to locate the spans of text expressing filler values in the original source documents so that character offsets could be provided for provenance information. We again used the Apache Lucene index over source documents along with a series of heuristic string similarity metrics to identify the span of characters in the original documents that sufficiently matched the post-processed version of the text seen by the CRF and the micro-readers. While not perfect, we did not find during system development that this approach ever failed to locate the correct span of text.

7 Evaluation

7.1 Additional Training Data

As training data, we used the data from NIST. Additionally, in 2014 we carried out a few rounds of internal manual evaluations of our system output on queries from 2013 and earlier years.

7.2 Submissions

We have submitted three entries (CMUML1-3) for the KBP cold start slot filling evaluation.

- CMUML1 Our main run using our 2015 micro readers. We did not use OpenEval for this run, so that the system did not access the live web. (Please note that we had a last minute problem running our 2014 system on the 2015 queries, therefore, even though the plan was to use the predictions of the 2015 micro-readers and the 2014 system, this ended up not being the case and our performance was negatively affected.)

- CMUML2 This is the same as CMUML1, but with OpenEval in use for an expected precision boost by attempting to vet predictions via web access.

- CMUML3 The 2014 system used on a small fraction of the queries(about 20 % of the queries).

Experimental results of these systems are shown in Table 1.

8 Conclusion

In this paper, we presented an overview of the CMUML system for the KBP 2015 English Cold

Start Slot Filling (SF) task. The system used a combination of micro-readers added in 2015, and a 2014 distant supervision CRF-based structured prediction system. We made no attempt to perform entity disambiguation. For future submissions, this would be an obvious area to address, in addition to our ongoing work to improve our micro-readers.

Acknowledgments

This work was supported in part by DARPA (award number FA87501320005), and Google. Any opinions, findings, conclusions and recommendations expressed in this papers are the authors' and do not necessarily reflect those of the sponsors.

References

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI 2010*.

Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of EMNLP 2012*.

Jayant Krishnamurthy and Tom M Mitchell. 2014. Joint syntactic and semantic parsing with combinatory categorial grammar. In *Proceedings of ACL 2014*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL 2009*.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (AAAI)*, pages 2302–2310.

Mehdi Samadi, Manuela Veloso, and Manuel Blum. 2013. Openeval: Web information query evaluation. In *Proceedings of AAAI 2013*.

Derry Tanti Wijaya, Partha Pratim Talukdar, and Tom M. Mitchell. 2013. PIDGIN: ontology alignment using web text as interlingua. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 589–598.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.