CMUML System for KBP 2013 Slot Filling

Bryan Kisiel, Justin Betteridge, Matt Gardner, Jayant Krishnamurthy,
Ndapa Nakashole, Mehdi Samadi, Partha Talukdar, Derry Wijaya, Tom Mitchell
School of Computer Science
Carnegie Mellon University, Pittsburgh, PA
{bkisiel,jbetter,mg1,jayantk,ndapa,msamadi,ppt,dwijaya,tom}@cs.cmu.edu

Abstract

In this paper, we present an overview of the CMUML system for KBP 2013 English Slot Filling (SF) task. The system used a combination of distant supervision, stacked generalization and CRF-based structured prediction. Recently available anchor text data was also used for better entity matching. The system takes a modular approach so that independently developed semantic annotators can be effectively integrated without needing target ontology-specific retraining. While precision can of course be improved, the system turned out to be particularly conservative in its predictions resulting in lower recall. In addition to the main submission, we also made publicly available[1] automatically tagged semantic categories of about 13 million noun phrases extracted from the KBP 2013 source corpus.

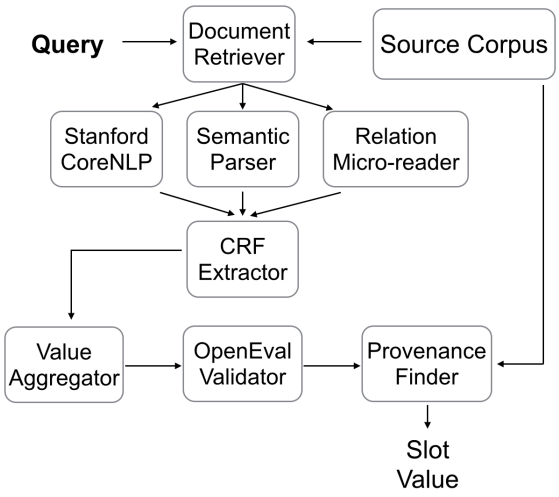

Figure 1: Overview of the CMUML Slot Filling (SF) system. The OpenEval validator was not used in the official submission (CMUML1) as it requires web access.

1 Introduction

In this paper, we describe the CMUML system for KBP 2013 Slot Filling (SF) task organized by NIST. The system used a combination of distant supervision (Mintz et al., 2009), stacking (Wolpert, 1992), and CRF-based structured prediction. The driving philosophy behind this system was to keep components modular so that researchers working with different ontologies could contribute without having to conform their tools to one common ontology. Given a candidate sentence, this system first used multiple semantic and syntactic annotators with heterogeneous schemas to produce different layers of annotations. These layers of annotations were integrated and mapped to the KBP ontology using a Conditional Random Fields (CRF)-based structured predictor. Finally, extractions from the CRF were integrated and validated before producing the final answers with provenance. The overall system architecture is shown in Figure 1. We provide brief discussion of each component below.

2 Document Retrieval and Entity Matching

The KBP 2013 source documents were indexed using Lucene[2]. Now, given a query, this index was

[1] Semantic categories of 13 million noun phrases from the KBP 2013 corpus: http://rtw.ml.cmu.edu/rtw/nps

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

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)[3] 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 Semantic Annotations

Once relevant sentences were identified, multiple syntactic and semantic analyses of each sentence was generated using different components as described below.

StanfordCoreNLP: We processed each document in the KBP source document collection using the StanfordCoreNLP pipeline[4]. The *tokenize, ssplit, pos, lemma, ner, parse, dcoref*, and *SUTime* modules were used.

Semantic Parsing: We trained a CCG semantic parser (Krishnamurthy and Mitchell, 2012) to extract NELL(Carlson et al., 2010) relations from text using distant supervision. This parser was run to annotate NELL relations in selected sentences. This and other annotations were used as features in the CRF.

Relation Micro-reader using Factor Graphs: A factor graph-based relation micro-reader (extractor) (Betteridge et al., 2014) was used to identify NELL relation instances in the selected sentence. This micro-reader was trained using a new variant of distance-supervision which is described in the paper cited above.

Please note that both the CCG-based semantic parser and the Relation Micro-reader were trained in schemas other than the KBP ontology. As mentioned earlier, this decoupling from the target ontology is highly desirable as it allows reuse of the same components for many different target ontologies without any retraining. This makes the overall architecture highly modular.

3.1 CRF-based Extractor and Aggregator

Tokenized sentences found to contain the query text were then converted into CRF instances, with the semantic annotations (generated as described above) 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 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.

3.2 Slot Value Validation using OpenEval

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

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

[4] Stanford CoreNLP: http://nlp.stanford.edu/software/corenlp.shtml

Run Id	Recall	Precision	F1
LDC (Manual)	57.08	85.60	68.49
Top KBP 2013 SF System	33.17	42.53	32.28
CMUML1	9.67	30.34	14.67
CMUML2	10.69	32.3	16.07
CMUML4	5.31	44.31	9.49

Table 1: Official evaluation scores of various CMUML submissions. For reference, scores of the top performing KBP 2013 SF system and a manual submission from LDC are also shown.

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.

3.3 Provenance Finder

Finally, 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. While not perfect, we did not find during system development that this approach ever failed to locate the correct span of text.

4 Evaluation

For training data, we used the queries, answers, and assessments from past KBP years to retrieve training examples from the 2012 KBP corpora.

We have submitted three entries (CMUML1, CMUML2, and CMUML4) for the KBP slot filling evaluation. CMUML2 builds on CMUML1 by including OpenEval, which queries the live web as a slot value validator. We observed this to greatly improve precision without much cost to recall. CMUML4 builds on CMUML2 by using two CRFs, one for PER queries and the other for ORG queries. Using different CRFs to model these two categorically different cases seemed to result in better fitting models during cross-validation, but this approach was under-explored at the time of submission. Experimental results comparing these systems to the best overall team and a manual submission from LDC (the upper bound) are shown in Table 1.

5 Conclusion

In this paper, we presented an overview of the CMUML system for the KBP 2013 English Slot Filling (SF) task. The system used a combination of distant supervision, stacked generalization, and CRF-based structured prediction. Low recall is one of the primary limiting factors of the system which we hope to improve in future iterations. In addition to participating in the official evaluation, we also made publicly available automatically tagged semantic type information spanning 271 categories of about 13 million noun phrases from the KBP 2013 source corpus. Please see `http://rtw.ml.cmu.edu/rtw/nps` for more details.

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

Justin Betteridge, Alan Ritter, and Tom Mitchell. 2014. Assuming facts are expressed more than once. In *Proceedings of the 27th FLAIRS Conference*.

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*.

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

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

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