University of Washington System for 2015 KBP Cold Start Slot Filling

Stephen Soderland, Natalie Hawkins, Gene L. Kim, and Daniel S. Weld
Turing Center, Department of Computer Science and Engineering
Box 352350
University of Washington
Seattle, WA 98195, USA
soderlan@cs.washington.edu, nhawkins@cs.washington.edu,
gkim21@cs.rochester.edu, weld@cs.washington.edu

Abstract

The University of Washington participated in Cold Start Slot Filling for TAC-KBP 2015 with a system that combines three methods: 1) its 2013 OPENIE-KBP system (Soderland et al., 2013); 2) a novel Implicit Relation Information Extractor (IMPLIE); and 3) MULTIR extractor (Hoffmann et al., 2011), trained on a combination of distant supervision and crowd-sourced training instances.

These three methods are complementary with little overlap in the resulting extractions. In particular, IMPLIE finds extractions that are beyond the scope of OPENIE-KBP.

1 Overview

This year, the University of Washington participated in the Cold Start Slot Filling evaluation with a combination of three systems: 1) OPENIE-KBP is based on Open Information Extraction (Soderland et al., 2013; Mausam et al., 2012); 2) IMPLIE is a novel Implicit Relation Information Extractor that detects relations without an explicit relation phrase in the sentence, such as is often the case with *nationality*, *job title*, *religion*, and *lived in* relations; and 3) MULTIR-KBP which is based on MULTIR (Hoffmann et al., 2011), and in this case is trained on a combination of distant supervision and crowd-sourced training instances.

Combining these three systems gave good synergy that increased recall, since there is little overlap in the extractions found by the three systems: the sum of recall for each method is only 16% greater than recall of the combined system for hop 0.

Our combined system ranked #9 out of 21 sites. As shown in Figure 1, overall precision was 75% and recall 38% that of the top ranked site; overall precision was 7% higher than the second rate site and recall 34% that of the second ranked site.

	Hop 0			Hop 1			All		
	P	R	F1	P	R	F1	P	R	F1
Combined	0.42	0.10	0.16	0.17	0.07	0.10	0.30	0.09	0.13
OpenIE	0.51	0.03	0.05	0.71	0.01	0.01	0.53	0.02	0.04
ImplIE	0.49	0.05	0.10	0.41	0.02	0.03	0.48	0.04	0.07
MultiR	0.36	0.03	0.06	0.51	0.02	0.04	0.39	0.03	0.05
#1 ranked	0.49	0.28	0.36	0.23	0.13	0.16	0.40	0.22	0.29
#2 ranked	0.34	0.26	0.29	0.21	0.25	0.23	0.28	0.26	0.27

Figure 1: Performance at hop 0 (from query entity), hop 1 (from results of hop 0), and overall for combined system and each component system.

We present the OPENIE-KBP system in Section 2, the IMPLIE system in Section 3, and MULTIR-KBP in Section 4. Section 5 discusses how we combined the three systems, and Section 6 gives conclusions.

2 Mapping Open IE to a Target Ontology

OPENIE-KBP begins by running an Open Information Extractor over the TAC-KBP corpus, which produces tuples of the form *(arg1, rel, arg2)* where *rel* is a phrase from the input sentence that expresses an arbitrary relation between *arg1* and *arg2*.

Our first Open IE system was TextRunner (Etzioni et al., 2006; Banko et al., 2007; Banko and Etzioni, 2008), followed by ReVerb (Fader et al., 2011; Etzioni et al., 2011) and OLLIE (Mausam et

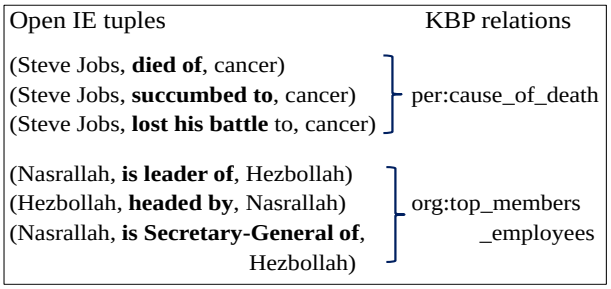

Figure 2: Open IE finds textual relations with no tuning required for a domain or set of target relations. The challenge is to map these extractions to relations in an ontology.

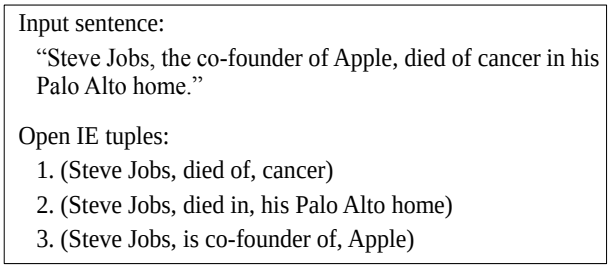

Figure 3: Open IE tuples from a sample sentence. OPENIE 4.0 is more robust in identifying verb-based relations, but also handles noun-based relations such as "(is) co-founder".

al., 2012). The most recent Open IE v4.0[1] handles both verb-mediated relations (e.g. "died at","lost his battle to") and noun-mediated relations (e.g. "is co-founder of", "is leader of"). These extractions express relations textually as shown in Figure 2.

An advantage of Open IE over previous information extraction systems is that it works out of the box, requiring no training or tuning for a new domain. The relations it extracts are represented as text strings rather than as relations in an ontology. This is not a problem if the tuples are for human use, for example searching a database of Open IE tuples extracted from a text corpus.

However, some applications such as the KBP Slot Filling and KBP Cold Start require the relations to be mapped to the relations in a particular ontology. Figure 2 shows just a few of the textual relations that correspond to *per:cause_of_death* or *org:top_members_employees*. In general, there are a few high frequency surface forms used to express a relation such as "died of" or "died from", and a long tail of other surface forms with diminishing frequency.

It is this Zipfian distribution of surface forms that gives us the possibility to create a mapping from target relations in an ontology to Open IE tuples with minimal knowledge engineering effort. A simple rule language built on Open IE is sufficient to identify the most common surface forms with high precision.

Figure 2 illustrates several Open IE extractions. The first tuple (Steve Jobs, died of, cancer) is one of the extractions from "Steve Jobs, the co-founder of Apple, died of cancer in his Palo Alto home." Other tuples from this sentence are shown in Figure 3.

As was the case in the 2013 and 2014 KBP Slot Filling evaluation, OPENIE-KBP gives high precision for this year's Cold Start Slot Filling with relatively low recall. As seen in Figure 1, OPENIE-KBP has high precision compared to the top performing systems: 0.51 for the hop 0 and 0.73 for hop 1, but recall of only 0.30 and 0.10 respectively. This is due to an inherent limitation of Open IE for the KBP Slot Filling task as described in the following section.

2.1 Limits to Open IE recall

We analyzed the correct extractions from from our OPENIE-KBP and from all runs submitted by 2014 KBP participants. As Table 1 shows, most of the correct OPENIE-KBP extractions were from noun-based constructions, either appositives or slot fills that were noun modifiers to the entity.

Correct slot fills in responses from all KBP participants shows a similar trend. A large proportion are found in noun phrases, often with no explicit relation phrase to create an Open IE tuple. This was particularly true of per:origin and per:title relations, two of the most common slot fills. We examined all

Table 1: Only a small percentage of the correct KBP extractions from OPENIE-KBP were from verb-based relation phrases. The great majority were from noun-based patterns.

Syntactic structure	percent
apppositive	0.38
noun modifier	0.26
verb phrase	0.26
other	0.09

[1] Available at github.com/knowitall/openie

correct per:origin and a sample of 100 of the per:title slot fills. Only 9% of the per:title slot fills were in a context that had a verb predictive of the relation (*e.g.* "worked as" or "served as"); 29% were in a light verb construction (*e.g.* "was" or "became"); and 62% had the slot fill in the same NP as the entity. For per:origin, none were in a context with a verb that indicated nationality; 9% were found in light verb constructions; and 91% were in the same NP as the entity.

Another limit of Open IE is that it forms tuples only for *binary* relations, where there is both an Arg1 and Arg2 for the relation phrase. Consider the example, "Jean DuPuis is a journalist at Le Monde" and a noun-based variant "Jean DuPuis, a journalist at Le Monde, reported that ...". Each of these produces the same tuple, (Jean DuPuis, is a journalist at, Le Monde).

In many cases, however, a sentence expresses an attribute of an entity, but there is no Arg2. Take for example "French journalist Jean DuPuis reported that ...". There is no second argument for a "journalist" relation – we don't know a place, date, or newspaper name to serve as Arg2. What we would like is a tuple with an *implicit relation* such as "has job title": (Jean DuPuis, [has job title], journalist). Such implicit relations, with no relation phrase in the sentence, is beyond the scope of current Open IE systems.

3 The IMPLIE System

The IMPLIE system (Implicit relation Information Extraction) is designed to find extractions that are beyond the scope of Open IE – those where there is typically no explicit relation phrase in the sentence.

IMPLIE extracts binary relations *(Arg1, has Class, Arg2)*, where Arg2 is a term of a target Class. Our KBP system with IMPLIE used the following implicit relations: *has nationality*, *has jobTitle*, *has religion*, *has city*, and *has province*. These were mapped to KBP relations in a straightforward manner, assuming that a person who has nationality, city, or province resides in that location.

IMPLIE begins with user-supplied semantic taggers for a set of target classes and then applies dependency parse rules to find noun phrase that are modified by terms of a target class.

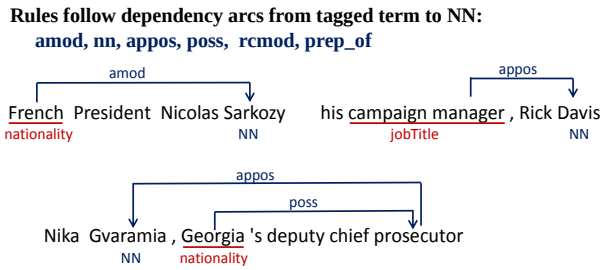

Figure 4: Examples of IMPLIE following dependency arcs from a term that has been tagged with a target class to a noun that the term modifies.

3.1 Tagging Terms for a Target Class

We approach implicit relation extraction by first selecting a class of interest, and tagging phrases for the class. This semantic tagging may be done with techniques similar to NER taggers. Our implementation of IMPLIE simply used keyword lists for each class. We used lists of keywords from CMU's NELL (Carlson et al., 2010), from Freebase, and from tables found on the Web.

Tagger selection is important, since the tagged terms form the pool of candidates for extractions in the following steps.

3.2 Dependency Path Rules for IMPLIE

In this step, IMPLIE starts with the tagged class term t and a set of dependency parse sequences S, or rules, that indicate the existence of an implicit relation. IMPLIE searches for a path $p \in S$, starting from t. IMPLIE parses the sentence using the Stanford Parser (de Marneffe et al., 2006).

Starting with term t of class c, IMPLIE searches through the dependency parse for any path p to a noun n, where $p \in S$. The dependency parse sequences S were constructed from a combination of linguistic interpretation of the parse dependencies and from tuning on a development set of sentences. Path p is stored for the extraction step.

$S \subset \tilde{S}$, where $\tilde{S}$ is a set of all possible combinations (with repetition) of up to three elements from the following list of dependency arcs: *amod*, *nn*, *appos*, *poss*, *rcmod*, and *prep_of*. A few examples of following a path $p \in S$ from a tagged term to the modified noun are shown in Figure 4.

3.3 Extraction

For the extraction step, IMPLIE identifies an extraction substring of the sentence, which contains both Arg1 and Arg2 of the relation, then performs a set of checks to ensure that the extraction is a well-formed implicit relation noun phrase. The extraction of the substring is performed by taking the maximum and minimum indices in p as the substring endpoints. Then, the substring is extended so that all parentheses are closed. This method of extraction results in noun phrases by construction of S.

Finally, IMPLIE runs the extraction through a series of filters to eliminate three types of mistakes: parser failures, parse ambiguity, and noun phrases where the terminal noun n is not the head noun.

Parser failures occur when an incorrect dependency arc is placed between two words. IMPLIE identifies commonly incorrectly marked arcs in the extraction path p, and throws away the extraction if it finds any syntactic indicators of a badly placed arc. An example of such a filter is the arc *appos*, and the indicator of having the word "and" in between Arg1 and Arg2. This eliminates *appos* arcs that should have been marked *conj*.

IMPLIE also identifies common syntactic patterns of incorrect extractions, where the incorrectness of the extraction cannot be explained by the dependency parse and eliminates those extractions. In essence this is identifying when there is ambiguity in the rules in S. That is, a rule in S may in some instances, legitimately relate t to n in some way other than an implicit relation.

IMPLIE eliminates instances where the terminal noun n is not the head noun of the extraction by running the extraction through a head finder and checking that the head found matches the terminal noun n. We use the head finding algorithm found in Michael Collins' thesis (Collins, 1999)

As seen in Figure 1, IMPLIE has high precision for hop 0 and recall higher than OPENIE-KBP. IMPLIE by itself can handle the relations involving nationality, job title, residence, and religion, which gives high recall when starting from the original query entity. It is relatively uncommon for a two hop query to be composed only of these relations, which gives IMPLIE low recall for hop 1.

4 MultiR System

In adapting our MULTIR distant supervision system to the KBP Cold Start Slot Filling task, we were especially interested in assessing how crowdsourced training could boost performance over distant supervision alone.

Distant supervision (DS) has been explored in recent years as a way to provide abundant training for relation extraction at little cost. The earliest work on DS was by Mintz et al. (2009) and by Riedel et al. (2010). Distant supervision provides relation labels by consulting a knowledge base (KB) such as Wikipedia[2] or Freebase[3], to find pairs of entities $E1$ and $E2$ for which a relation R holds.

Distant supervision then makes that assumption that any sentence that contains $E1$ and $E2$ is a positive training instance for R. This leads to a large proportion of false positive training instances for many relations. For example, Freebase asserts that Nicolas Sarkozy was born in Paris, but nearly all sentences in a news corpus that mention Sarkozy and Paris do not give evidence for a $born\_in$ relation.

To address this shortcoming of distant supervision, there have been attempts to model the relation dependencies as multi-instance multi-object with graphical models, in particular MultiR (Hoffmann et al., 2011), which we used for our KBP system, and MIML-RE (Surdeanu et al., 2012).

We found that adding high-quality training from crowdsourcing is effective in increasing both the precision and he recall for MultiR. The key is careful training of the crowdsource workers and filtering to retain only the highest precision workers.

4.1 Crowdsourcing for MultiR Training

We used Amazon Mechanical Turk for our crowdsourcing, but designed our own website rather than use the platform Amazon provides directly. This allowed us great flexibility in providing a tutorial for workers, giving them feedback as they went, and rejecting workers who failed a proportion of gold-standard questions.

We required workers to complete an interactive tutorial to learn the criteria for the following relations: $nationality$, $born\_in$, $lived\_in$, $died\_in$, and $traveled\_to$. These are all relations between a

[2]https://www.wikipedia.org/

[3]https://www.freebase.com

phrase highlighted as a $person$ and another phrase highlighed as a $location$. All but $traveled\_to$ are based on TAC-KBP relations.

We also selected a set of gold questions that workers are likely to get wrong if they don't clearly understand the annotation criteria. The first five questions are weed-out questions used to eliminate spammers and careless workers early on. These questions look no different than normal questions, but we give feedback to workers with the right answers if workers give wrong answers to any of the weed-out questions. If a worker failed a majority of such questions, the worker will be disqualified from the task.

We then place gold questions among real test questions in order to spot-check workers' response. Since we have spotted spammers early on, the number of gold questions we place decreases exponentially with the index of the question batch the gold questions are in. Workers must maintain at least 80% accuracy on the most recent 10 gold questions to continue working on the task. In our experience, workers who started out with high accuracy maintained that accuracy throughout the entire tagging.

In separate experiments, we found that our crowdsourcing protocol produced high precision training, which results in high precision extractors. We had our workers tag the same 10K sentences as used by Angeli et al. (2014). This raised precision of the training examples from 0.50 to 0.77, and raised F1 of the extractor from 0.31 to 0.40.

As shown in Figure 1, MULTIR has precision 0.36 for hop 0 and 0.51 for hop 1. The recall is limited to relations between persons and locations, because we did our crowdsourcing on only these relations. Recall for our KBP system is 0.03 for hop 0 and 0.02 for hop 1.

5 Combined Systems

We first preprocessed the cold start corpus using the Stanford NLP pipeline and cached the results. Then we ran OPENIE-KBP, IMPLIE, and MULTIR systems separately, taking the union of the results as our main submission. If the two systems had different output for a functional relation, we used only the OPENIE-KBP output, if any. If no OPENIE-KBP output, we used only IMPLIE, and lacking that used MULTIR output.

For the combined system, we used the combined output from hop 0 as input to hop 1. For each of the other systems, we used only hop 0 output from that system as input to hop 1.

There was surprisingly little overlap of responses by the three systems. The sum of recall for each method is only 16% greater than recall of the combined system for hop 0.

6 Conclusions

We participated in the 2015 KBP Cold Start Slot Filling evaluation with a combination of three systems, one based on Open IE, another based on a novel Implicit Relation IE system (IMPLIE), and a third using MULTIR trained on a combination of distant supervision and crowdsourced training. The resulting system had high precision: 75% that of the top ranking system and 7% higher than the second ranking system. This was at recall 38% that of the top system and 34% of the second ranking system.

Our combination of methods had good synergy. In particular, IMPLIE found extractions that are beyond the scope of Open IE, which requires an explicit relation phrase in the sentence.

7 Acknowledgements

This research was supported in part by ONR grant N00014-11-1-0294, DARPA contract FA8750-13-2-0019, and ARO grant W911NF-13-1-0246, and was carried out at the University of Washington's Turing Center.

References

Gabor Angeli, Julie Tibshirani, Jean Y. Wu, and Christopher D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *EMNLP*.

M. Banko and O. Etzioni. 2008. The tradeoffs between traditional and open relation extraction. In *Proceedings of ACL*.

M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the Web. In *Procs. of IJCAI*.

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 *Procs. of AAAI*.

Michael John Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, Philadelphia, PA, USA. AAI9926110.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Language Resources and Evaluation (LREC 2006)*.

O. Etzioni, M. Banko, and M. Cafarella. 2006. Machine Reading. In *AAAI*.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: the second generation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '11)*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of EMNLP*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 1003–1011.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld. 2013. Open information extraction to KBP relations in 3 hours. In *Proceedings of TAC-KBP 2013*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.