Open Information Extraction to KBP Relations in 3 Hours

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld
Turing Center, Department of Computer Science and Engineering
Box 352350
University of Washington
Seattle, WA 98195, USA
{soderlan, jgilme1, rbart, etzioni, weld}@cs.washington.edu

Abstract

We participated in both the English Slot Filling and Entity Linking in the 2013 TAC-KBP evaluation. Our Slot Filling system provides an answer to the following conjectures: Can Open Information Extraction (Open IE) form the basis of a high precision extractor for a set of target relations in an ontology? And, just as importantly, can this be done with a minimum of human knowledge engineering?

We built rules to map Open IE extractions to KBP Slot Filling relations and found that just three hours of rule creation gave extractor precision of 0.79. Spending a total of 12 hours refining the rules increased precision slightly to 0.80 with recall near the median of other KBP systems.

1 Introduction

The University of Washington built upon its research in Open Information Extraction (Open IE) for both the English Slot Filling (SF) and the English Entity Linking (EL) tracks in this year's TAC-KBP evaluation (Mausam et al., 2012; Fader et al., 2011; Lin et al., 2012a).

We used our SF system to explore a novel approach to high-precision information extraction. Previous approaches have required either a tagged training corpus for supervised or semi-supervised learning of the relations of interest, or have required a large knowledge engineering effort. Open IE can avoid both types of expense, as it provides extractions out of the box with no domain tuning and no pre-specified relations. However, these extractions express relations textually as shown in Figure 1.

Open IE tuples	KBP relations
(Steve Jobs, **died of**, cancer) (Steve Jobs, **succumbed to**, cancer) (Steve Jobs, **lost his battle** to, cancer)	per:cause_of_death
(Nasrallah, **is leader of**, Hezbollah) (Hezbollah, **headed by**, Nasrallah) (Nasrallah, **is Secretary-General of**, Hezbollah)	org:top_members_employees

Figure 1: 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.

The question arises: can Open IE support extraction for a target set of relations in a specific ontology? Furthermore, can this be done with minimal training examples and minimal knowledge engineering effort?

We demonstrate that an end user can map Open IE output to relations in an ontology with high precision with as little as a few hours of knowledge engineering effort. One of the runs we submitted used a set of rules that took only three hours to create and had extraction precision of 0.79. Another run in which a total of 12 hours were spent designing the rules had precision of 0.80 at recall of approximately the median of other KBP systems.

Another advantage is the simple rule language that makes our approach accessible to end users who lack expertise in linguistics or machine learning. Details about the rule creation process are found in Section 3.1 and results in Section 3.3.

We also report on our EL system, which has a pipeline that begins by looking for the most informative reference to the query entity in the document, uses the Google CrossWikis database of anchor text phrases to find candidate links in the KB, and then ranks the links based on cosine similarity. Our EL system is described in Section 4.

2 Open IE

Researchers at the University of Washington have pioneered a new paradigm for massively scalable, domain-independent information extraction. Open IE systems extract tuples consisting of argument phrases from the input sentence and a phrase from the sentence that expresses a relation between the arguments, in the format (arg1, rel, arg2). This is done without a pre-specified set of relations and with no domain-specific knowledge engineering.

Figure 1 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 2.

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 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").

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 require the relations to be mapped to the relations in a particular ontology. Figure 1 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.

Input sentence:
"Steve Jobs, the co-founder of Apple, died of cancer in his Palo Alto home."

Open IE tuples:
1. (Steve Jobs, died of, cancer)
2. (Steve Jobs, died in, his Palo Alto home)
3. (Steve Jobs, is co-founder of, Apple)

Figure 2: Open IE tuples from a sample sentence.

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.

This will not handle all the variations in how a relation may be expressed, but will have good coverage for relations that are expressed in a straightforward way. In addition, it will take advantage of redundancy in a large text corpus – it will extract a relation between entities E_1 and E_2 if the relation is expressed at least once in a lexical-syntactic pattern that the rules cover.

3 Mapping Open IE to a Target Ontology

Our goal is to create a method to map Open IE tuples to target relations that is accessible to end-users, who may not have the expertise to set up a machine learning system and may not have the computational linguistics background to deal with syntactic parses and other language resources.

We chose to create rules manually rather than adopt a machine learning approach, since an end user will often lack large training sets tagged for their target relations. In prior research for the DARPA Machine Reading Project, we found that the limited training available was not sufficient for high precision rule learning even when we incorporated active learning (Soderland et al., 2010).

3.1 Creating Rules for KBP Relations

We designed a simple rule language, shown in Figure 3, that specifies the target relation, which tuple element contains the entity and the slotfill, and any

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

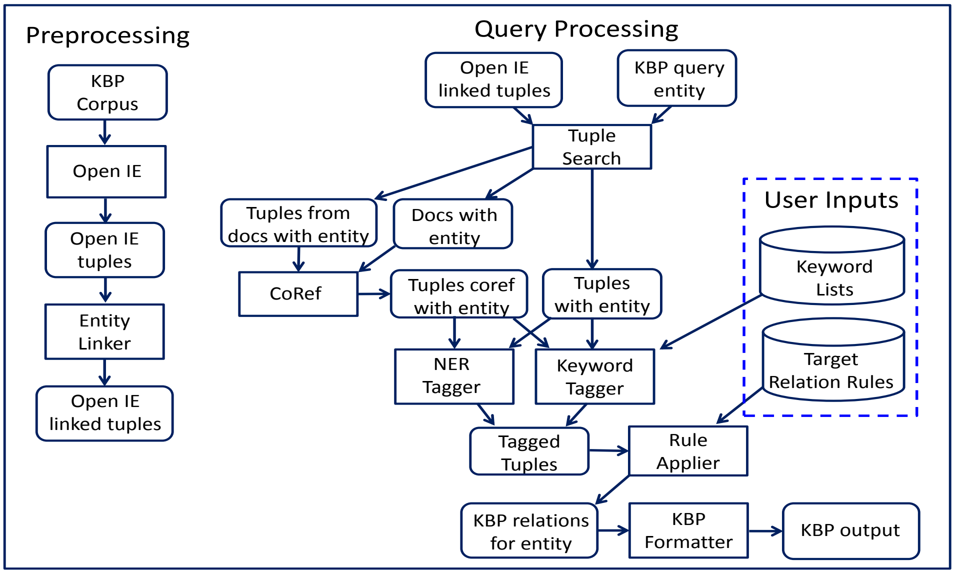

Figure 4: System architecture for KBP Slot Filling. The only user-specified inputs are the Keyword Lists for semantic tagging and the Target Relation Rules used during query processing.

Terms in Rule	Example
Target relation:	per:employee_or_member_of
Functional?:	No
Query entity in:	Arg1
Slotfill in:	Arg2
Slotfill type:	Organization
Arg1 terms:	-
Relation terms:	appointed
Arg2 terms:	<JobTitle> of

(Smith, was appointed, Acting Director of Acme Corporation)

per:employee_or_member_of (Smith, Acme Corporation)

Figure 3: A simple rule language specifies the target relation, which tuple elements have the entity and slotfill, and a combination of lexical and semantic constraints.

lexical or semantic constraints on tuple elements. When the rule in this example is applied to the tuple (Smith, was appointed, Acting Director of Acme Corporation), all constraints in the rule are met – the relation phrase has the term "appointed", arg2 has a *JobTitle* followed by "of", the query entity is in arg1 and arg2 includes a phrase of type *Organization* that is extracted as the slotfill.

The Rule Applier can extract a sub-phrase from a tuple element if that phrase has been tagged with the semantic class for the slotfill. Thus, the phrase "Acme Corporation" is extracted from the tuple argument "Acting Director of Acme Corporation" in Figure 3.

This rule language relies on semantic type constraints to maintain high precision while allowing generalized rules. Yet, these semantic types may be quite domain specific, such as *JobTitle* in Figure 3, along with more general types such as *Organization*. We devised a method for semantic tagging that is easily extensible by an end user.

The tagging combines Named Entity Recognizers (NER) for *Organization*, *Person*, *Location*, and *Date*, with a Keyword Tagger that we had developed earlier[2]. The Keyword Tagger takes a file with a list of terms and then walks through a tokenized sentence to find the longest matches to keywords in the list.

In our KBP Slot Filling system, we used lists for JobTitle, Religion, Nationality, School, City, StateOrProvince, and Country. We used lists provided by CMU's NELL system (Carlson et al., 2010), although any source of keyword lists can be used. In addition, we manually determined a subset of the 4K JobTitles to create a list of *HeadJobTitles* for the relation *org:top_members_employees*.

The inputs from an end user are the lists of key-

[2] Available at github/knowitall/taggers

words and a tab-delimited file with a set of rules for each target relation. We created the rules for KBP Slot Filling using a spreadsheet.

We saved a copy of the initial rules and keywords files that were created in just three hours, in order to measure the impact of user effort. For the *3 Hour* rule set, the first author wrote a small set of the most obvious rules for each KBP relation. For example, *per:cause_of_death* has four rules, one for "died of", "died from", "died due to", and "died as a result of".

These rules and keyword lists were expanded and refined in approximately 12 hours over the course of two weeks, testing them against a benchmark from the 2012 Slot Filling answer key. The final *12 Hour* rule set had an average of 16 rules per relation, about five times as many rules as the 3 Hour rule set.

3.2 Our KBP Slot Filling System

Figure 4 shows our architecture for our KBP Slot Filling system. A preprocessing step uses Open IE v4.0 to extract tuples from the KBP corpus, and then does Entity Linking, NER tagging, and tagging using the Keyword Lists. For Entity Linking, we used a modification of Tom Lin's linker (Lin et al., 2012a; Lin et al., 2012b) in which the Google Crosswikis dataset was used to find candidate Freebase entities for linking.

At query time, we search these tagged and linked tuples for tuples where either arg1 or arg2 match the query entity string argument or is linked to the query entity. We applied the Target Relation Rules to these tuples, producing a set of KBP relation extractions for the entity.

We also incorporate coreference by retrieving documents that either have the entity string in a tuple argument or have arguments linked to the query entity. We then ran the Coreference module of the Stanford NLP Pipeline on these documents. Any tuples where either arg1 or arg2 was in the same coreference set as the query entity were then passed to the Target Relation Rules. This produced additional KBP relation extractions.

A final step was to format the extractions as required for the KBP evaluation, normalizing dates, and eliminating redundant extractions. We post-processed extracted locations with the Tipster Gazetteer to help disambiguate between City, StateOrProvince, and Country.

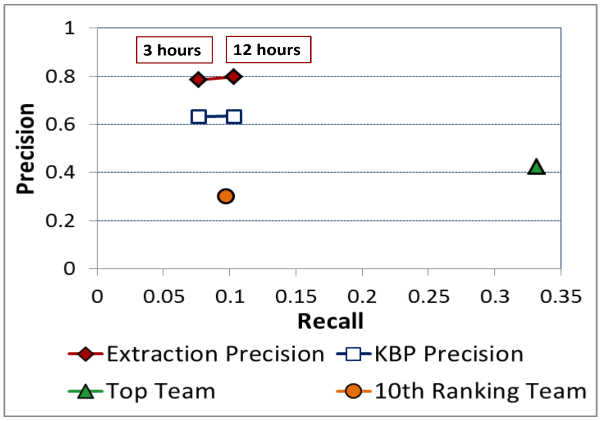

Figure 5: Our systems based on 3 hours and on 12 hours of rule creation achieved *Extraction Precision* of 0.79 and 0.80. The stricter *KBP Precision* requires entity disambiguation as well as correct extraction.

Disambiguating query entities has an impact on the official KBP scores, but we put in only a cursory effort on this. If the set of extractions for an entity string included arguments linked to multiple Freebase entities, then we discarded all extractions for that entity string as unreliable.

3.3 KBP Slot Filling Results

In order to evaluate our approach to extraction, we teased apart the factors in the KBP Slot Filling evaluation. The top line in Figure 5 shows *Extraction Precision*. For this, an extraction is considered correct if the KBP relation holds between the entity and slotfill. Thus a *per:title* extraction of "president" is correct if the document states that Paul Gray was president of a company. These precision figures are based on our own tagging and are not the official KBP results.

The lower line in Figure 5 is for the official *KBP Precision*, which also requires entity disambiguation. For this, "president" is an incorrect title when the intended Paul Gray is a rock musician.

We chose not to tackle entity disambiguation and spent our energy building the infrastructure needed to participate in the KBP evaluation. Entity disambiguation is orthogonal to the research questions we wanted to explore (see Section 1). Accordingly, we looked for any tuple where an argument matched the query entity string. We used all resulting tuples unless the tuple set was linked to multiple KB entries, in which case we discarded the entire tuple set.

We submitted three runs for KBP SF, the data point labeled "3 hours" is for a system based on the 3 Hour rule set and the data point labeled "12 hours" is the system with the 12 Hour rule set. Both of those runs included the coreference step. A third run used the 12 Hour rule set, but omitted coreference. This run had recall between the other two runs at comparable precision, and is not shown on the graph.

We were pleased with the high precision, between 0.79 and 0.80, of each of our runs. The recall of the 12 Hour run was just over 0.10. While not high, this recall is about the median of other KBP systems, which indicates the difficulty of the KBP Slot Filling task, and was at considerably higher precision than the other systems. Somewhat surprisingly, the 12 Hour run had recall that was only 35% higher than the 3 hour run, which leads to the question of whether our approach was reaching a ceiling effect.

3.4 Discussion of Slot Filling Results

Our goal was to explore Open IE as a practical option for extraction of target relations in an ontology. We have certainly demonstrated that high precision extraction is possible from a small effort in knowledge engineering, although at modest recall.

3.4.1 Error Analysis

We analyzed the primary source of errors for extractions that were incorrect with respect to the information in the sentence.

31% of the errors seemed to be correct according to the understanding of KBP guidelines by the author who created the rule sets. Examples are extracting *per:title(Tantawi, sheik)* from "Tantawi was the grand sheik" and extracting *org:subsidiary(ETA, Batasuna)* from "ETA's political wing Batasuna." Apparently "sheik" is not considered a person title, and the wing of a political party is not a subsidiary. Most of these errors would not occur with rules that aligned better with the KBP guidelines.

23% were caused by rules that overgeneralized. For example, a "critic" is often a person title (*e.g.* a drama critic), but not in the case of "Ginzburg was an outspoken critic of the policy". Similarly, a person who leads an organization is often an employee_or_member_of that organization, but not in the case of "Meredith led the NFL in scoring." We may need to extend the rule language to allow it to recognize exceptions to more general rules. Even then, there will be a recall-precision trade-off in finding the right level of generality to the rules.

19% of the errors occurred when the rules matched a non-head term. For example, our system extracted *per:spouse(Kahn, Shankar)* from "Kahn's younger sister married Shankar" rather than applying to the head of the first noun phrase "sister". Tightening the rule applier to only match head nouns would eliminate these errors. In some cases, it would also discard correct extractions such as a city_of_death as "Baltimore" when someone dies in "a Baltimore hospital" or in his "Baltimore home."

15% were due to a variety of errors by the Open IE extractor, some of which are caused by parsing errors and some by tuple argument identification errors.

12% were due to coreference errors. These may be unavoidable, short of eliminating the coreference module entirely, which would reduce recall.

From this error analysis, it appears that precision could be raised considerably higher than 80% from a better understanding of the KBP guidelines and a minor tightening of the rule applier: allowing rules to require a match on head terms and to include explicit exceptions. We now turn to the question of recall.

3.4.2 Limits to Recall

Is the low recall or our system due to a fundamental limit to the recall of Open IE, upon which it is based? We analyzed a random sample of sentences where at least one KBP participant found a correct extraction. We ran our Open IE extractor on these sentences and examined the resulting tuples to see whether the tuples had sufficient information to produce the extraction, given ideal rules.

42% of the time the tuple contained all the necessary information for the KBP extraction, given an appropriate rule set.

16% of the time the extractor truncated an argument, for example omitting an appositive or parenthetical phrase. The sentence "Sheikh Tantawi, the top Egyptian cleric who died on Wednesday ..." has a tuple (the top Egyptian cleric, died on, Wednesday). This omits the cleric's name, and thus cannot support an extraction of date_of_death for Sheikh Tantawi.

10% of the time the Open IE system fails to recognize a noun-based relation. It relies on a learned lexicon of relation nouns that may not include less frequent terms. For example, it extracts tuples with relations such as "is CEO of" or "is son of", but misses the relation in "Tantawi, the Grand Imam of Al-Azhar".

10% of the time syntactic complexity results in no extraction from the part of a sentence containing information needed for a KBP relation.

22%: a variety of other causes of the tuple not having sufficient information to support the KBP relation.

Clearly, there is a potential for much higher recall than our system had in the KBP Slot Filling. With some improvements to handling of appositives and parenthetical phrases, and better coverage of relational nouns, Open IE tuples provide sufficient information for KBP extractions over half the time.

On the whole, our approach is an attractive one in practice, where the cost of designing and training a system for higher recall may outweigh the benefit of diminishing returns as it becomes more and more difficult for any relation extractor to increase recall while maintaining precision.

4 Entity Linking Approach

Our KBP Entity Linking system is an adaptation of Tom Lin's entity linker (Lin et al., 2012a; Lin et al., 2012b). We use a four step approach.

1. Find the most specific mention of the query entity in the document.

2. Find candidates for linking from the Google Crosswikis table of anchor text that is linked to Wikipedia articles.

3. Evaluate the candidate links with a logistic regression classifier that combines the Crosswikis probability and the cosine similarity between context in the document and in the knowledge base entry.

4. Assign NIL to any entities with classifier score below a threshold. Merge two entities with NIL links if a more specific mention was found for at least one of them and they shared the same most specific mention.

We created sequences of rules manually to find the most specific entity mention in the document, with distinct rule sets for each NER type, *Person*, *Organization*, or *Location*. This proved to be higher precision than using Stanford's CoreNLP Coreference module.

We used Stanford's Coreference module to gather context for evaluating candidate links, using all sentences that contain a term in the coreference set for the entity.

We used benchmark sets from the 2011 and 2012 KBP evaluations to tune our procedures and set classifier thresholds empirically.

Our EL system had B-cubed score of 0.588, with the highest score for the Newswire documents (0.673) and the lowest for Discussion Forums (0.453). These results are slightly above the median scores for all 2013 EL systems.

5 Conclusions and Future Work

We have presented our systems for KBP 2013 English Slot Filling and Entity Linking, introducing a novel method for Slot Filling that leverages Open IE extractions. We have shown that a simple rule language can map Open IE tuples to a set of KBP relations at high precision with as little as three hours of knowledge engineering.

Future work includes improvements to Open IE recall and to rule precision – we identified some easily implemented improvements in both areas. Another avenue for future work is to generalize our method to apply to any ontology of relations and to operate without the input of KBP queries that specify a target entity.

6 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

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*.

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*.

Thomas Lin, Mausam, and Oren Etzioni. 2012a. Entity linking at Web scale. In *The Knowledge Extraction Workshop (AKBC-WEKEX) at NAACL*.

Thomas Lin, Mausam, and Oren Etzioni. 2012b. No noun phrase left behind: Detecting and typing unlinkable entities. In *EMNLP-CoNLL*, pages 893–903.

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

Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. 2010. Adapting open information extraction to domain-specific relations. *AI Magazine*, 31(3):93–102.