A Language Model for Extracting Implicit Relations

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

Abstract

Open Information Extraction has shown promise of overcoming a knowledge engineering bottleneck, but has a fundamental limitation. It is unable to extract *implicit* relations, where the sentence lacks an explicit relation phrase. We present IMPLIE (Implicit relation Information Extraction) that uses an open-domain syntactic language model and user-supplied semantic taggers to overcome this limitation. IMPLIE can extract an implicit *has nationality* relation, *has job title*, and *has city* relation from "French journalist Paul Legall reported ... at Athens International Airport.".

Formal evaluations of IMPLIE show high precision, over 0.90 for nationality and job title on newswire text. IMPLIE nearly doubles recall for 2013 KBP Slot Filling queries and more than doubles it for 2014 when combined with an Open IE-based system, maintaining precision of 0.58 and 0.59 respectively.

1 Introduction

Open Information Extraction (Open IE) helps overcome a knowledge engineering bottleneck of traditional domain-specific IE that requires either hand-coded knowledge or hand-tagged training examples, or both. Open IE automatically extracts tuples of the form *(Arg1, Rel_phrase, Arg2)*, using a domain-independent language model (Etzioni et al., 2006; Banko et al., 2007; Fader et al., 2011; Mausam et al., 2012).

However, Open IE has a fundamental limitation. Much of the information in free text is expressed without explicit relation phrases. This is particularly true for relations such as *has nationality*, *has job title*, and *has religion*, which are rarely expressed with an explicit relation phrase.

"French journalist Paul Legall reported that all three hostages arrived safely at Athens International Airport."

IMPLIE extractions:
(French journalist Paul Legall; has nationality; French)
(French journalist Paul Legall; has jobTitle; journalist)
(Athens International Airport; has city; Athens)

Figure 1: IMPLIE finds three extractions from this sentence, which are not found by Open IE since the sentence lacks explicit relation phrases.

Consider the example in Figure 1, "French journalist Paul Legall reported ... at Athens International Airport". This has implicit *has nationality*, *has jobTitle*, and *has city* relations, but no explicit relation phrases in the sentence. These implicit relations are beyond the reach of current Open IE.

We present IMPLIE, which extracts implicit relations in the form *(Arg1, has Class, Arg2)*, where Arg2 is a word or phrase of type Class. In the spirit of Open IE, IMPLIE uses a syntactic language model that is independent of any pre-defined set of relations. However, IMPLIE requires user-supplied semantic taggers for a set of target classes. There is a one-to-one mapping of the target classes and target relations – given a tagger for class c, IMPLIE automatically builds an extractor for the relation *has c*.

Our contributions are the following:

- We demonstrate a limitation of Open IE that it cannot extract implicit relations.
- We present IMPLIE, an implicit relation extrac-

tor that overcomes this limitation[1].

- Formal evaluations of IMPLIE show high precision and good coverage on newswire text, with precision over 0.90 for the relations, *has nationality* and *has jobTitle*.
- Evaluation on 2013 and 2014 KBP Slot Filling shows that IMPLIE nearly doubles recall for 2013 and more than doubles it for 2014 when combined with an Open IE-based system, while maintaining precision of 0.58 and 0.59.

The remainder of the paper presents background and related work in Section 2, followed by a description of IMPLIE in Section 3. We present experimental results on newswire text and on the KBP Slot Filling task in Section 4, and conclusions in Section 5.

2 Background and Related Work

Several approaches to information extraction (IE) in recent years have been proposed that avoid the requirement of large sets of manually labeled training data. These include distant supervision, supervision from existing annotated data sets such as ACE-2004, and Open Information Extraction (Open IE).

2.1 Distant Supervision

Distant supervision has been used to generate positive training data for IE (Hoffmann et al., 2010; Mintz et al., 2009). Any sentence with a pair of entities that have a relation in a knowledge base (*e.g.* Wikipedia) are considered positive for that relation – which produces extremely noisy training. Sophisticated probabilistic models have been developed to alleviate the effect of noisy data (Surdeanu et al., 2012; Riedel et al., 2010; Hoffmann et al., 2010), including Universal Schemas that uses matrix factorization (Riedel et al., 2013). None of these systems can achieve high precision except at low recall.

2.2 Supervised Learning

One system that most closely resembles IMPLIE is (Chan and Roth, 2011), which uses supervised learning to identify implicit relations covered by four syntactic patterns: premodifier, possessive, preposition, and formulaic (*e.g.* addresses with "City, State"). They observe that in the ACE-2004 training set, 80% of the instances are not verb-based, but fit one of those four patterns.

They then use these patterns to guide learning of ACE relations from the ACE-2004 training set. Given a pair of mentions m_i and m_j from a sentence, their system predicts whether one of the ACE relations holds between the mention pair.

Chan and Roth's syntactic patterns with the exception of their formulaic patterns are essentially IMPLIE's dependency path rules limited to a single dependency arc (See Section 3.2). Both systems rely on the input of a knowledge source – for Chan and Roth this is an annotated training data set, while for IMPLIE it is a set of semantic taggers for target classes. From these inputs, Chan and Roth do supervised learning, while IMPLIE applies a syntactic language model that is independent of the input set of classes.

2.3 Open Information Extraction

Another paradigm is Open Information Extraction, which allows an extractor to scale to an arbitrary number of relations, without requiring the user to provide either training examples or a pre-specified set of relations. Open IE uses a syntactic language model to extract 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*.

The 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). Preemptive IE (Shinyama and Sekine, 2006) is a variant of Open IE that first groups documents based on pairwise vector clustering, then applies additional clustering to group entities based on document clusters. The clustering steps make it difficult to scale to large corpora.

The most recent Open IE v4.0[2] handles both verb-mediated relations (*e.g.* "died at","was appointed as") and uses a learned vocabulary of relational nouns to detect noun-mediated relations (*e.g.* "co-founder of", "leader of").

Another system that detects noun-mediated relations is ReNoun (Yahya et al., 2014), which starts

[1] IMPLIE download is available at implie.cs.washington.edu

[2] ReVerb, OLLIE, and Open IE v4.0 can be downloaded at openie.allenai.org

Table 1: For some high-frequency KBP relations, only a small percentage of correct extractions from all KBP sites were from verb-based phrases. The great majority were from complex noun phrases.

Syntactic structure	per:title	per:origin
verb-based	0.09	0.00
light verb	0.29	0.09
complex NP	0.62	0.91

with a vocabulary of relational nouns as input, *e.g.* wife, protege, chief-economist. Like other Open IE systems, it extracts binary relations with a relation phrase found in the sentence, in this case a noun phrase.

OPENIE-KBP, an Open IE-based system (Soderland et al., 2013) participated in the 2013 KBP Slot Filling evaluation[3], a formal evaluation in which participants are given a test set of 100 query entities, 50 persons and 50 organizations. Participants must find all distinct mentions of relations such as per:origin, per:title, per:city_of_birth, org:city_of_headquarters, org:top_members_employees, etc. from a text corpus. OPENIE-KBP used a set of hand coded rules based on a set of semantic taggers to map Open IE tuples to KBP relations.

OPENIE-KBP achieved the highest precision (0.634) of all participants for 2013, but had recall of only 0.103. As a comparison, the highest performing system, whch used a combination of distant supervision and hand coded rules had precision 0.425 at recall 0.332 and the system that was #10 of 18 teams had precision 0.303 at recall 0.097.

2.4 Limits to Open IE recall

We analyzed a sample of correct extractions from all runs submitted by 2013 KBP participants. As Table 1 shows, most of the correct extractions for per:title (*i.e.* job title) and per:origin (*i.e.* nationality), two of the most common slot fills, were from complex noun phrases, either appositives or noun modifiers of the entity.

We examined all 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.

Open IE systems can only find an extraction when the sentence contains an explicit relation phrase, which generally begin with a verb. Open IE is also limited to extractions where the sentence expresses a *binary* relation with both Arg1 and Arg2 for the relation phrase. Consider the example given earlier, "French journalist Paul Legall reported ...". Open IE not only cannot find a relation phrase, it cannot find an Arg2. If the sentence had been "Paul Legall is a journalist at Le Monde, who reported ..." or "Paul Legall, a journalist at Le Monde, reported ..." then Open IE v4.0 would be able to extract a binary relation (Paul Legall; is a journalist at; Le Monde).

3 The IMPLIE System

IMPLIE extracts binary relations *(Arg1, has Class, Arg2)*, where Arg2 is a term of a target Class. Our experiments with IMPLIE used the following implicit relations: *has nationality*, *has jobTitle*, *has religion*, *has school*, *has city*, and *has province*.

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.

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

[3] http://www.nist.gov/tac/2013/KBP/

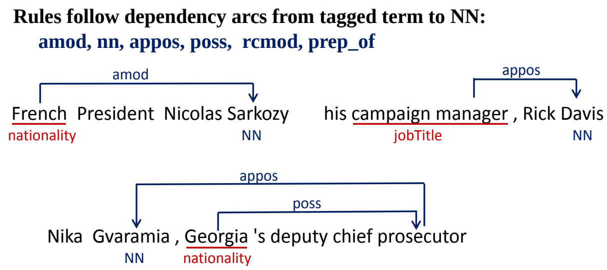

Figure 2: Examples of IMPLIE following dependency arcs from a term that has been tagged with a target class to a noun that the term modifies.

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 2.

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)

3.4 High Precision and High Recall Versions

We created two variants of IMPLIE for varying degrees of precision and recall. The high precision system uses case-sensitive keyword taggers and all of the filters we developed to handle extraction mistakes. In the high recall system, we relax some of the constraints. It uses case-insensitive taggers and eliminates many of the filters. We removed filters where number of correct extractions eliminated is at least half of the number of incorrect extractions eliminated on a development set.

3.5 Future Enhancements

Our current implementation of IMPLIE uses a small number of hand-coded rules to find the noun n modified by a term of class c, and a small number of hand-coded rules to expand n to the most informative, well-formed noun phrase headed by n. A machine learning approach could do a more nuanced job of each of these tasks, combining weaker evidence and producing a probability that the extraction is correct with a properly delimited noun phrase.

This implementation always expands on the rules until exhaustion and retains all of the expansion steps during the noun phrase extraction phase. Using a computable measure of noun phrase quality to prune the noun phrases may result in higher quality noun phrase boundaries.

4 Experimental Results

We conducted two sets of experiments to evaluate IMPLIE. The first experiments evaluate precision and coverage of IMPLIE on sentences sampled from the newswire section of the 2010 TAC KBP corpus. We explored two variants of IMPLIE, the full system that uses only the syntactic language model and a type-constrained variant where output is filtered by expected argument types. For example, *has nationality* and *has jobTitle* are restricted to entities that are persons or groups of persons. For each of these variants, we explored high-precision configurations and configurations that increased recall at the expense of precision. See sections 4.1.1 and 4.1.2 for details.

The second set of experiments assesses IMPLIE's performance on the KBP Slot Filling task. In particular, we evaluated the boost in recall IMPLIE gives when combined with the OpenIE-KBP system that the University of Washington fielded in the 2013 and 2014 KBP Slot Filling evaluation (Soderland et al., 2013), and whether it maintains the high precision that their system achieved.

4.1 Target Classes for Experiments

We evaluated IMPLIE on a set of six relations that are often expressed without explicit relation phrases: nationality, jobTitle, religion, school, city, province. These relations were also selected because they naturally map to relations used in the TAC KBP Slot Filling evaluation (http://www.nist.gov/tac).

We constructed keyword lists for each of these relations, using a combination of lists from Freebase, from United Nations Web pages, and from lists learned automatically by CMU's NELL system (Carlson et al., 2010). The list for nationality included country names (*e.g.* US or America) and adjective forms (*e.g.* American) plus some ethnic groups (*e.g.* Basque).

These keyword lists are inherently incomplete, particularly the list of job titles. Even a list of several thousand job titles from Academic coordinator and Access Management Specialist to Zen master and Zonal Underwriting Manager may not cover all job titles in a test set. We did some manual editing of these keyword lists and then found it necessary to create stop lists of terms that hurt precision on a development set.

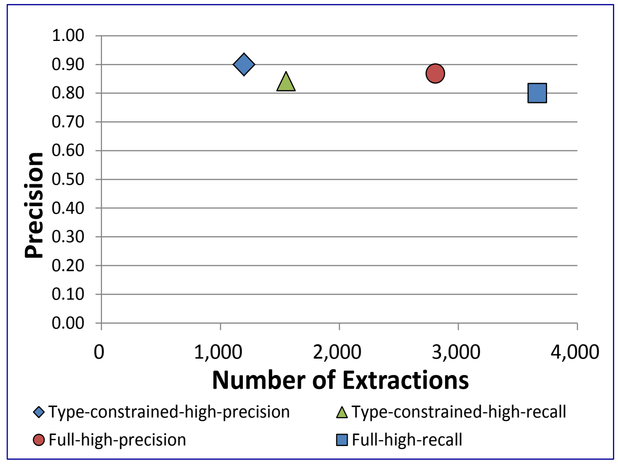

Figure 3: Four versions of IMPLIE each have high precision. The full versions use only syntactic patterns, while the type-constrained versions restrict the entity types (e.g. only persons can have job title). For each of these we have a high precision and a high recall setting. The test set has 9,000 sentences, so 3,663 extractions for full-high-recall means about one extraction for every 2.5 sentences.

4.1.1 Syntactic-only Version

This version of IMPLIE finds all extractions where a noun phrase is modified by a term of a target class. The extracted tuple *(Arg1, has Class, Arg2)* has the following interpretation: the entity Arg1 is associated in some way with the Arg2, which is a term of the Class.

This is similar to the interpretation of an Open IE tuple *(Arg1, Rel, Arg2)*, which has the interpretation that the phrase Rel represents some kind of relation between the two arguments. Thus "Brazil's airway infrastructure" is associated with the nationality Brazil and "Madrid train bombings" are associated with the city Madrid, giving the IMPLIE extractions (Brazil's airway infrastructure; has nationality; Brazil) and (Madrid train bombings; has city; Madrid).

For each version of the system, a sample of 1,000 extractions was tagged independently by two taggers, and any disagreements reconciled. Inter-tagger agreement was 0.92 for the high recall version and 0.94 for the high precision version.

Table 2: IMPLIE achieves high precision, 0.90 or higher for the highest frequency relations, on a test set of 9,000 newswire sentences.

	Full IMPLIE				Type Constrained			
	high recall		high precision		high recall		high precision	
relation	extr.	precision	extr.	precision	extr.	precision	extr.	precision
nationality	1,960	0.89	1,787	**0.90**	604	0.87	568	**0.90**
jobTitle	880	0.78	477	**0.90**	588	0.88	388	**0.95**
city	512	0.60	356	0.71	186	0.77	130	0.82
province	199	0.66	132	0.77	98	0.73	72	0.86
religion	59	**0.93**	42	0.80	50	0.84	32	0.84
school	53	0.28	11	0.75	26	0.29	9	0.44
total	3,663	0.80	2,806	0.87	1,552	0.84	1,199	**0.90**

4.1.2 Type Constrained Versions

We also created a type-constrained version that has a more well-defined interpretation of the extracted relations than the syntactic-only version. This version is closer to an end use of IMPLIE. For example, entities for nationality and for jobTitle are constrained to be of type person using NER tags and WordNet classes. Extractions are only considered correct if Arg1 is a person or group of people who has that nationality or job title. For example, the extraction (Japanese yen; has nationality; Japanese), which is correct for the syntactic-only version is filtered out in the type-constrained extractor. Intertagger agreement was 0.97 for the high recall version and 0.94 for the high precision version.

The type constraints and interpretation of each relation is as follows:

- Nationality: entity is a person or group of persons with that nationality
- JobTitle: entity is a person with that job title
- City: entity is a person, group of persons, or organization located in that city
- Province: entity is a person, group of persons, or organization located in that state or province
- Religion: entity is a person, group of persons, or organization with that religion
- School: entity is a person who attended that school or is an organization that is part of that school

4.1.3 Results

Table 2 gives results for each version of IMPLIE, broken down by relation and Figure 3 displays the aggregate results graphically. The full, syntactic-only versions had good coverage of the test sentences with high precision. The high recall version found 3,663 extractions from 9,000 sentences (an extraction every 2.5 sentences) with precision 0.80; the high precision version found 77% as many extractions but raised the precision to 0.87.

By design, the type constrained versions found fewer extractions, but with higher precision. The high recall version found 1,552 extractions, 42% as many as the full high recall extractor, and had precision 0.84, while the high precision version found 33% as many and achieved precision of 0.90.

4.2 Evaluation on KBP Slot Filling

We created a KBP Slot Filling system based on the full high recall version of IMPLIE, and evaluated it on both the 2013 and 2014 queries. No modifications were made to the IMPLIE extractor for this set of experiments, although we used a random set of 10% of the queries as a development set for adapting IMPLIE as a KBP Slot Filling system as described in Section 4.2.1.

In particular, we were interested to see how much of a boost to recall IMPLIE gives when its output is combined with that of the University of Washington's 2013 and 2014 KBP Slot Filling system, which is based on Open IE.

4.2.1 Adapting IMPLIE for KBP

Our IMPLIE-KBP system begins by searching the KBP corpus for documents that contain the query entity, using an exact string match. For queries of type Person where the query has first, middle, and last name or has a suffix such as “Jr.”, we create aliases for the query string, leaving out the middle name or dropping the suffix. This gives us a set of relevant documents, although it may miss some that refer to the query entity only by an alternate name.

We then process each relevant document with the

Stanford CoreNLP pipeline and run the high-recall version of IMPLIE on each sentence. We filter the IMPLIE extractions to those where Arg1 includes either the query entity string, an alias for the entity string, or a proper noun that is in the coreference set for the query string or alias.

All that remains is to map the extractions to a KBP relation, which was done in a straightforward manner. For queries of type Person, we mapped nationality to both per:origin and per:countries_of_residence; jobTitle to per:title; religion to per:religion; city to per:cities_of_residence; province to per:stateorprovince_of_residence; and produced no output for school extractions.

For queries of type Organization, we mapped city to org:city_of_headquarters; province to org:stateorprovince_of_headquarters; nationality to org:country_of_headquarters; religion to org:political_religious_affiliation; and ignored extractions for school. If a jobTitle extraction includes not only the query organization, but also a person name in Arg1, and has a job title on a list of top employee job titles, we create an extraction for org:top_members_employees.

4.2.2 Results

We evaluated IMPLIE-KBP by comparing its output to an answer key composed of all slot fills judged correct by the KBP organizers from all KBP participants for 2013 and 2014. If our system's output was included in the answer key, it was considered to be correct, otherwise to be an error.

IMPLIE-KBP gives a large boost to recall when added to OPENIE-KBP output, nearly doubling the recall for 2013 and more than doubling it for 2014 as shown in Figure 4. Precision is somewhat lower for the combined system than for OPENIE-KBP, but the higher recall dominates and gives a substantial boost to F1 score for each year. F1 for 2013 was raised from 0.183 to 0.305, and for 2014 F1 was raised from 0.109 to 0.252.

The precision of IMPLIE-KBP is in the mid 0.50's for each year, compared to precision of 0.80 for the full high recall configuration of IMPLIE that was used. We did an error analysis to determine the cause of this drop in precision.

We found that 24% of the "errors" were extractions that were arguably correct, making plausible inferences. IMPLIE-KBP extracted a headquarters

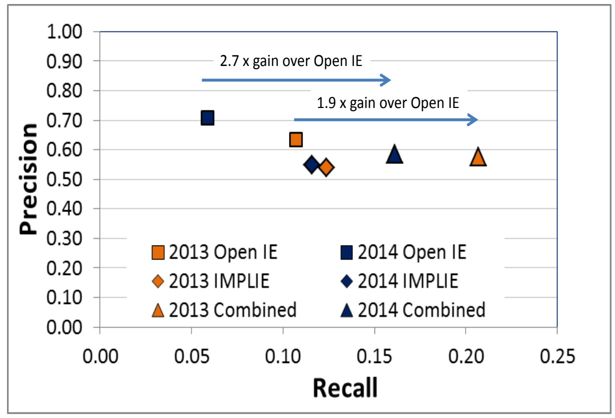

Figure 4: Combining IMPLIE with Open IE nearly doubles recall over Open IE alone for 2013 KBP Slot Filling queries and more than doubles it for 2014.

relation for companies identified as being from a city, province, or country or where a country name was in the organization name. Similarly a city, province, or country of residence was inferred for persons from that location. Another 7% of the errors were cases where the extraction was correct in terms of the sentence, but the sentence did not refer to the correct query entity.

21% of the errors were from incorrect tagging of terms that did not refer to the target class, for example tagging a person's name Sydney as a city, or tagging a reference to American Indians as the nationality of India. 16% of the errors were from bad parses, including a dependency arc from a city or country in the news article's by-line.

While the org:top_members_employees relation had high precision 0.91, we were surprised that per:title had precision of only 0.64. One third of the errors for per:title were from sentences where it was the brother, cousin, or wife that had the job title and another third were where the title was truncated. These two types of errors for per:title accounted for 14% of the errors.

Overall, we find the relatively high precision of IMPLIE-KBP encouraging, since it was achieved with no KBP-specific tuning of the output. We simply mapped the IMPLIE extraction that a person "is associated with" a city, province, or country to a KBP city/province/country of residence relation, and similarly made the inference that any organization associated with a city, province, or country has headquarters there. Applying post-processing rules to the output of these IMPLIE extractions and those for nationality, religion, and jobTitle could raise the KBP precision while maintaining most of the gain in recall.

5 Conclusions and Future Work

We have identified a fundamental limitation of existing Open IE systems – they can only identify relations where there is an explicit relation phrase in the sentence. Our novel relational extractor IMPLIE addresses this limitation and identifies *implicit* relations, which are the most typical expression of relations such as *nationality* and *job title*, among others.

IMPLIE uses an open-domain syntactic language model, but requires input of semantic taggers supplied by the user for a set of target classes. It then extracts the relation *has c* for each target class *c*.

IMPLIE has high precision and good coverage on news text, and gives a large boost to recall on the KBP Slot Filling task when combined with an Open IE-based system. IMPLIE nearly doubles recall for the 2013 KBP Slot Filling queries and more than doubles it for 2014 queries, with only a modest drop in precision.

Our implementation of IMPLIE uses a small set of manually coded dependency path rules. A next step is to transform these rules into features for a machine learning classifier. This would not only be able to combine weak evidence for an implicit relation, but give a confidence for the extraction.

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*.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of ACL*.

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, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 286–295.

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 *ECML/PKDD (3)*, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL*.

Y. Shinyama and S. Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Procs. of HLT/NAACL*.

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 EMNLP*.

Mohamed Yahya, Steven Euijong Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of EMNLP*.