Open Information Extraction Systems and Downstream Applications

Mausam
Computer Science and Engineering
Indian Institute of Technology Delhi
New Delhi, India
mausam@cse.iitd.ac.in

Abstract

Open Information Extraction (Open IE) extracts textual tuples comprising relation phrases and argument phrases from within a sentence, without requiring a pre-specified relation vocabulary. In this paper we first describe a decade of our progress on building Open IE extractors, which results in our latest extractor, OPENIE4, which is computationally efficient, outputs n-ary and nested relations, and also outputs relations mediated by nouns in addition to verbs. We also identify several strengths of the Open IE paradigm, which enable it to be a useful intermediate structure for end tasks. We survey its use in both human-facing applications and downstream NLP tasks, including event schema induction, sentence similarity, text comprehension, learning word vector embeddings, and more.

1 Introduction

Supervised learning is the dominant approach to solve information extraction (IE) problems, however, it relies heavily on manually annotated training data. The required annotation effort per relation is significant, and does not scale to the huge number of relations commonly found in Web text. Distant supervision (e.g., [Mintz *et al.*, 2009]) approaches provide a partial solution to the problem, but are applicable only to the relations present in existing knowledge-bases (KBs). How to extract the vast number of relations for which annotated data isn't available?

Moreover, IE systems populate a given ontology – they only learn to extract the relations that are already defined in the ontology. However, this assumes enormous foresight on the part of an ontologist to have defined all relations of interest ahead of time, and also eschews the ability of the system to *discover* new relations that are prominent in a given dataset.

In response, Open Information Extraction [Banko *et al.*, 2007] forgoes per-relation training data and is not bound by a fixed relation vocabulary. Its key insight is to not only extract arguments, but also extract *relation phrases* from text itself. For e.g., our latest Open IE system would extract (Modi, be Prime Minister of, India) from the sentence *"Indian Prime Minister Modi signs an executive order..."* irrespective of whether the ontology defines a PresidentOf relation or not, and whether there is training data for that relation or not.

In this paper we first describe a series of Open IE systems developed by us, which differ in their performance characteristics (precision, recall, speed), linguistic assumptions (relations expressed via verbs, nouns) as well as output representations (binary extractions, n-ary extractions, nested extractions). All our extractors are publicly available and free to use for research purposes.

Secondly, we identify the specific strengths of Open IE systems and KBs composed of Open IE tuples. In particular, Open IE systems are fairly robust and often apply out-of-the-box on most kinds of textual corpora written in English, even when they are from an unseen domain. They are also computationally efficient and thus can be easily run on large corpora. Open KBs are easy to interpret by humans – they form natural intermediate representations for end users interested in interacting with information in the text. Open IE also makes different design choices in their predicate-argument structures compared to other NLP tasks such as dependency parsing, and semantic role labeling (SRL), and can be used as useful intermediate representation also for downstream NLP tasks.

We discuss the various end tasks where Open IE representation has been applied. These end tasks may be human-facing (such as fact finding, entity summarization) or NLP applications solved by machines. For the latter, we have used Open IE representation for a wide variety of semantic tasks. These include creating a large repository of event schemas, classifying if two sentences mention redundant information, text comprehension and learning word vector embeddings evaluated on lexical similarity and lexical analogy tasks. We can also rapidly program extractors for ontology relations using regular expressions over open tuples.

2 Open IE: Representation and Systems

Open IE's goal is to read a sentence and extract tuples with a relation phrase and arguments that are related by that relation phrase. Originally, Open IE extracted binary tuples, i.e., *two* arguments connected by one relation phrase. E.g., from *"IJCAI 2016 took place in New York."*, an Open IE system will extract (IJCAI 2016, took place in, New York). It will identify that the appropriate relation phrase is 'took place in' and not 'took'. It will also identify appropriate argument boundaries. An Open IE system needs to classify whether to extract, and,

Extractor	Output	Linguistics
REVERB	binary	verb-based rels, NP args
OLLIE	binary, nested*	verb/noun rels, phrasal args
SRLIE	binary, n-ary, nested	verb rels, phrasal args
RELNOUN	binary	noun rels, NP args

Table 1: Comparison of prominent Open IE extractors based on their output representation and linguistic assumptions regarding relation phrases and arguments.

if so, what should the arguments and relation boundaries be. The key technical challenge is lack of training data at scale.

The first generation extractor named TEXTRUNNER [Banko *et al.*, 2007] learns a CRF to label the intermediate words between each potential pair of arguments as part of the relation phrase or not. This sequence labeling formulation is learned by self-supervised training data constructed using heuristics over Penn Tree Bank. The CRF uses only unlexicalized features, so that it can work on lexical items not seen at all in training data. REVERB [Fader *et al.*, 2011; Etzioni *et al.*, 2011] improves over TEXTRUNNER via a careful linguistic analysis of patterns expressing relation phrases in English text. REVERB identifies that a simple regular expression (verb | verb particle | verb word* particle) covers about 85% of verb-based relation phrases in English. A natural extension, ARGLEARNER performs similar analyses for argument phrases [Etzioni *et al.*, 2011]. They learn simple rules/classifiers to detect relation and argument boundaries.

R2A2, combines REVERB and ARGLEARNER. Its strength is this semantically tractable subset of English language, which can be compactly represented using regular expressions, covers a large fraction of the language and can be effectively used to construct high precision open extractors.

Further analysis reveals that R2A2 misses important recall from verb-based relations that have long-range dependencies or where both arguments are on one side of the verb, from noun-based relations and more. To improve recall, one must identify long tail of patterns, necessitating machine learning approaches; unfortunately, data annotation is expensive at scale. We develop bootstrapping methods where REVERB's high confidence extractions (seed tuples) can act as a source of distant supervision – we match a seed tuples' content words with sentences that use similar words, and hypothesize that these sentences are likely expressing the seed tuple.

Our 3^{rd} generation extractor, OLLIE, learns (mostly) unlexicalized pattern templates on top of this bootstrapped training data [Mausam *et al.*, 2012]. By operating over dependency parses, OLLIE can naturally incorporate long-range dependencies. With machine learning OLLIE can move down the long tail of textual patterns. OLLIE's methods learn verb-based, noun-based, and even some inferential relation patterns. For example, OLLIE can extract (Ahmadinejad, is the President of, Iran) from the sentence *"Ahmadinejad is elected the President of Iran"*, because its patterns learn that 'elected' in such constructions can be dropped in extraction process.

We also develop RELNOUN, a rule-based Open IE system [Pal and Mausam, 2016] for extracting noun-mediated relations such as extracting (Collins, be director of, NIH) from sentences like *"Collins, the director of NIH,..."* or *"NIH director Collins..."*. It encodes various nominal patterns, and pays special attention to demonyms and compound relational nouns. For example, it extracts (Modi, be the Prime Minister of, India) instead of (Minister Modi, be Prime of, India) or (Modi, be Prime Minister of, Indian) from the sentence *"Indian Prime Minister Modi..."*.

Beyond Binary Tuples: A binary representation isn't always sufficient for expressing information. For instance, from *"Yakub flew from London to Seattle."* it is better to extract one *n-ary* tuple (Yakub, flew, from London, to Seattle)[1] instead of two independent binary tuples (Yakub, flew from, London) and (Yakub, flew to, Seattle). The n-ary tuple retains the relationship between London and Seattle, whereas multiple binary extractions lose it.

Moreover, early systems did not distinguish between information asserted in the sentence versus not asserted. For example, from the sentence *"Early scientists believed that earth is the center of the universe"*, extracting that (earth, is the center of, the universe) will be inappropriate, since the sentence isn't asserting it; it is merely noting that this tuple was believed by early scientists. This requires two modifications to Open IE systems – (1) extracting a *nested* tuple (Early scientists, believed, (earth, is the center of, the universe)), and (2) filtering out (earth, is the center of, the universe).

OLLIE attempts a first solution to this by additionally extracting an attribution context with a tuple, when available (e.g., 'early scientists believed'). SRLIE, our latest Open IE extractor (based on ideas in [Christensen *et al.*, 2011]), solves this problem by analyzing the output of a state-of-the-art PropBank-trained SRL system. It analyzes the hierarchical structure between semantic frames to construct multi-verb open relation phrases, and nested relational tuples. This also identifies when a tuple should be filtered out. For example, for the sentence *"John refused to visit a Vegas casino"*, SRL provides two frames, a 'refuse' frame and a 'visit' frame. For the 'visit' frame, the A0 is 'John' and A1 is 'a Vegas casino'. SRLIE's post-processing recognizes that 'visit' frame is nested within the 'refuse' frame and so (John, visit, a Vegas casino) should not be extracted; SRLIE extracts (John, refused to visit, a Vegas casino) in addition to (John, refused, to visit a Vegas casino). Because semantic frames are n-ary, SRLIE is also able to output n-ary extractions.

We combine SRLIE and RELNOUN in our latest Open IE system called OPENIE4.[2] It uses ClearNLP's SRL system, which runs fast and has good SRL performance. OPENIE4 is, therefore, quite efficient itself and obtains a good balance of precision and yield.[3] In evaluation on out-of-domain sentences, OPENIE4 obtains a speed of 52 sentences/sec, which is a little slower than REVERB's 167 sentences/sec, but OPENIE4 has better precision and enormously better yield compared to REVERB (over 4 times AUC, area under precision-yield curve). Moreover, OPENIE4 obtains 1.32 times AUC

[1] Note that, for n-ary open extractions, prepositions are written as part of the arguments for accurate reading.

[2] Available at https://github.com/knowitall/openie

[3] Computing recall is very difficult for Open IE. We compute yield, the no. of correct extractions, which is proportional to recall.

compared to OLLIE run with the fast MaltParser [Nivre *et al.*, 2007], while maintaining the same speed.

3 Open IE for End User Tasks

Open IE systems has several strengths. They extract many kinds of relations without requiring any per-relation annotation. Their implementations are robust and accept a variety of English language inputs. They have been developed with an eye on speed. Overall, our systems can often run on a new dataset out-of-the-box and create an Open KB, irrespective of the dataset's domain or size, and without much fine-tuning.

A key advantage of Open IE representation is human readability – open tuples can (almost) be read like small sentences, which makes it very convenient for an end user interacting with an Open KB. This is especially in contrast to parsing or SRL, which output annotations that Open IE does not (like edge labels in dependency tree, or semantic roles), but are harder to interpret for a non-linguist end user.

This readability allows Open IE's direct use in human-facing applications. For instance, our Open IE demo[4] allows an end user to pose novel queries like "(?, kill, bacteria)" or "(Bill Gates, ?, Microsoft)", which are natural query forms of "what kills bacteria" and "what are the relationships between Bill Gates and Microsoft". The Open IE demo provides a list of answers for each such query, for example, identifying 'antibiotics', 'chlorine', 'heat' for the former and 'founded', 'is the chairman of', 'retired from', for the latter. It is also an alternative way to perform corpus exploration, since the extractions are linked to the source documents from where they were extracted. A user may choose to read an article based on a tuple extracted from it.

Open IE also provides useful data compression (compared to search snippets or reading original document) while still retaining important information. This can help an end-user in obtaining a summary view of a concept. For example, if a user wishes to learn from their text dataset about a specific person named 'Jahangir', running a query for "(Jahangir, ?, ?)" in Open KB enables her to quickly browse all the information extracted from a variety of sentences from possibly different documents, all describing Jahangir. This may let her rapidly learn various facts about this entity – that (Jahangir, was a Mughal ruler in, India), (Jahangir, died in, 1627), (Jahangir, succeeded, Akbar), and (Jahangir, was succeeded by, ShahJahan). We believe such interactions are quite valuable for intelligence analysts, and news reporters, who are drowning in information and need an interactive method for information summarization.

4 Open IE for NLP End Tasks

Semantic applications typically derive features from some intermediate representation of text, such as bag of words (BoW), dependency parses, semantic role labels, verb-subject/verb-object structures, etc. In several recent works, we have found Open IE to be a useful intermediate representation for downstream NLP tasks. Compared to parsing and SRL, Open IE does not offer any dependency/semantic annotations, but it offers a lightweight representation to crisply express structural predicate-argument information. Compared to verb-subject/object structures and BoW, Open IE retains important facts expressed in the sentence, which can lead to better downstream performance. We now list NLP tasks in which Open IE representation has been effective.

Traditional IE: For populating a new ontology, Open IE provides a useful domain-independent representation of facts in text; domain-specific rules can operate over open tuples to populate a given ontology. Since Open IE representation is human-interpretable, these rules are fairly easy to write by a domain expert. For example, for TACKBP'2013 an NLP expert spent three hours writing such rules and was able to create a working extractor for 41 relations of interest obtaining a precision of about 0.8 but low recall [Soderland *et al.*, 2013]. Another nine hours of work achieved, at a similar precision, the median recall among all the competitors of the competition. We have released our software OREO, a traditional IE system that can be rapidly retargeted to a new domain.[5] We have also investigated rule learning and active learning approaches to automate the process of creating rules from labeled data [Soderland *et al.*, 2010].

Event Schema Induction: Inducing open-domain event schemas at scale first requires compiling sets of relations that appear in concert. We run Open IE on a large news corpora (1.8 million articles) and release a novel *Relgrams* dataset, which lists pairs of relation phrases that frequently co-occur in news [Balasubramanian *et al.*, 2012].[6] We further run graph clustering over Relgrams to induce a set of common event schemas. We post-process the schemas to infer argument types, and their roles in a schema. Careful evaluation over Mechanical Turk reveals that our Open IE-based schemas are much more coherent and accurate compared to earlier work that uses pairs of verb-subject and verb-object as the base representation [Balasubramanian *et al.*, 2013].

Sentence Similarity: Summarization systems require a model of redundancy so that summaries don't repeat information. We find that Open IE-based tuple overlap is an effective unsupervised measure of redundancy between two sentences [Christensen *et al.*, 2013]. When we improved this model to create a supervised redundancy model, Open IE-based overlap continued to be a useful feature [Christensen *et al.*, 2014].

Text Comprehension: In this study, we convert a sentence into different base representations (Open IE, SRL, dependency parsing, BoW) from which natural features are extracted. We then create a text comprehension system, in which machine reads a passage of text and then answers questions. We use a single Q/A algorithm operating over different feature sets. We find that Open IE comprehensively outperforms BoW and SRL; its improvements over dependency parsing are marginal [Stanovsky *et al.*, 2015].

Lexical Similarity and Analogy: Training of vector embeddings for words has become an important intermediate

[4] Available at http://openie.allenai.org/

[5] Available at https://github.com/abhishekyadav43/OREO-OpenIE-based-Relation-Extraction-for-Ontology

[6] Available at http://relgrams.cs.washington.edu

task, since word vectors have been widely useful in a variety of downstream semantic applications. Recent work [Levy and Goldberg, 2014] shows that vector embeddings can be trained not only over BoW contexts, but also over dependency parse and other contexts. In this experiment we compare embeddings learned by Open IE with those by BoW, SRL and dependency parses. Comparing across seven lexical similarity datasets we find Open IE to outperform all others in six. Comparing across two lexical analogy datasets, Open IE again outperforms all other representations by wide margins. A key observation is that Open IE skilfully combines both syntactically related as well as topically relevant words in its tuples, whereas other representations usually can get to one or the other, but not both [Stanovsky *et al.*, 2015].

5 Conclusions and Future Work

Open Information Extraction is an influential paradigm for extracting relational tuples from text in a scalable, domain-independent fashion. This paper describes the progress of Open IE systems over the last nine years, focusing on their performance characteristics and output representations, leading to the publicly available OPENIE4 system. We also observe the various strengths of Open IE, which make it a useful intermediate representation for a variety of end user, as well as downstream NLP applications. We list several applications where Open IE has been used and found effective.

While current Open IE systems have good performance, their yield could be improved further, e.g., by extracting information present in numerical quantities [Anand *et al.*, 2016], lists of entities, and implicit information easily inferred by humans. A fundamental drawback of Open IE is its lack of relation normalization. We believe that the most significant challenge in this line of research is to learn a large corpus of high-precision inference rules between relation phrases, which will enable information expressed using one phrase to be connected to other synonymous or inferrable phrases [Berant *et al.*, 2011; Jain and Mausam, 2016].

Acknowledgments: Most of this research was carried out at Univ. of Washington Turing Center and supported by numerous funding agencies. This paper is supported by Google's language focused grants, a Bloomberg award, and Visvesvaraya faculty award by Govt. of India. This research is carried out in collaboration with Aman Anand, Niranjan Balasubramanian, Gagan Bansal, Robert Bart, Janara Christensen, Anthony Fader, Prachi Jain, Ashish Mittal, Harinder Pal, Brendan Roof, Bo Qin, Gabriel Stanvosky, Shi Xu, Abhishek Yadav, Ido Dagan, Ganesh Ramakrishnan, Sunita Sarawagi, Michael Schmitz, Stephen Soderland and Oren Etzioni.

References

[Anand *et al.*, 2016] Aman Anand, Ashish Mittal, Mausam, Ganesh Ramakrishnan, and Sunita Sarawagi. Numerical relation extraction with minimal supervision. In *AAAI*, 2016.

[Balasubramanian *et al.*, 2012] Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. Rel-grams: A probabilistic model of relations in text. In *AKBC-WEKEX*, 2012.

[Balasubramanian *et al.*, 2013] Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. Generating coherent event schemas at scale. In *EMNLP*, 2013.

[Banko *et al.*, 2007] Michele Banko, Michael Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, 2007.

[Berant *et al.*, 2011] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *ACL*, 2011.

[Christensen *et al.*, 2011] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *KCAP*, 2011.

[Christensen *et al.*, 2013] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Towards coherent multi-document summarization. In *NAACL*, 2013.

[Christensen *et al.*, 2014] Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. Hierarchical summarization: Scaling up multi-document summarization. In *ACL*, 2014.

[Etzioni *et al.*, 2011] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In *IJCAI*, 2011.

[Fader *et al.*, 2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *EMNLP*, 2011.

[Jain and Mausam, 2016] Prachi Jain and Mausam. Knowledge-guided linguistic rewrites for inference rule verification. In *NAACL*, 2016.

[Levy and Goldberg, 2014] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *ACL*, 2014.

[Mausam *et al.*, 2012] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *EMNLP*, 2012.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP*, 2009.

[Nivre *et al.*, 2007] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[Pal and Mausam, 2016] Harinder Pal and Mausam. Demonyms and compound relational nouns in nominal Open IE. In *AKBC*, 2016.

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

[Soderland *et al.*, 2013] Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld. Open information extraction to KBP relations in 3 hours. In *TAC*, 2013.

[Stanovsky *et al.*, 2015] Gabriel Stanovsky, Ido Dagan, and Mausam. Open IE as an intermediate structure for semantic tasks. In *ACL*, 2015.