Open Information Extraction: The Second Generation

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam
Turing Center
Department of Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA 98195, USA
{etzioni,afader,janara,soderlan,mausam}@cs.washington.edu

Abstract

How do we scale information extraction to the massive size and unprecedented heterogeneity of the Web corpus? Beginning in 2003, our KnowItAll project has sought to extract high-quality knowledge from the Web.

In 2007, we introduced the *Open Information Extraction* (Open IE) paradigm which eschews hand-labeled training examples, and avoids domain-specific verbs and nouns, to develop *unlexicalized*, domain-independent extractors that scale to the Web corpus. Open IE systems have extracted billions of assertions as the basis for both common-sense knowledge and novel question-answering systems.

This paper describes the second generation of Open IE systems, which rely on a novel model of how relations and their arguments are expressed in English sentences to double precision/recall compared with previous systems such as TEXTRUNNER and WOE.

1 Introduction

Ever since its invention, text has been the fundamental repository of human knowledge and understanding. With the invention of the printing press, the computer, and the explosive growth of the Web, we find that the amount of readily accessible text has long surpassed the ability of humans to read it. This challenge has only become worse with the explosive popularity of new text production engines such as Twitter where hundreds of millions of short "texts" are created daily [Ritter *et al.*, 2011]. Even finding relevant text has become increasingly challenging. Clearly, automatic text understanding has the potential to help, but the relevant technologies have to scale to the Web.

Starting in 2003, the KnowItAll project at the University of Washington has sought to extract high-quality collections of assertions from massive Web corpora. In 2006, we wrote: "The time is ripe for the AI community to set its sights on **Machine Reading**—the automatic, unsupervised understanding of text." [Etzioni *et al.*, 2006]. In response to the challenge of Machine Reading, we have investigated the *Open Information Extraction* (Open IE) paradigm, which aims to scale IE methods to the size and diversity of the Web corpus [Banko *et al.*, 2007].

Typically, Information Extraction (IE) systems learn an extractor for each target relation from labeled training examples [Kim and Moldovan, 1993; Riloff, 1996; Soderland, 1999]. This approach to IE does not scale to corpora where the number of target relations is very large, or where the target relations cannot be specified in advance. Open IE solves this problem by identifying *relation phrases*—phrases that denote relations in English sentences [Banko *et al.*, 2007]. The automatic identification of relation phrases enables the extraction of *arbitrary* relations from sentences, obviating the restriction to a pre-specified vocabulary.

Open IE systems avoid specific nouns and verbs at all costs. The extractors are *unlexicalized*—formulated only in terms of syntactic tokens (*e.g.*, part-of-speech tags) and closed-word classes (*e.g.*, of, in, such as). Thus, Open IE extractors focus on generic ways in which relationships are expressed in English—naturally generalizing across domains.

Open IE systems have achieved a notable measure of success on massive, open-domain corpora drawn from the Web, Wikipedia, and elsewhere. [Banko *et al.*, 2007; Wu and Weld, 2010; Zhu *et al.*, 2009]. The output of Open IE systems has been used to support tasks like learning selectional preferences [Ritter *et al.*, 2010], acquiring common-sense knowledge [Lin *et al.*, 2010], and recognizing entailment rules [Schoenmackers *et al.*, 2010; Berant *et al.*, 2011]. In addition, Open IE extractions have been mapped onto existing ontologies [Soderland *et al.*, 2010].

This paper outlines our recent efforts to develop the second generation systems for Open Information Extraction. An important aspect of our methodology is a thorough linguistic analysis of randomly sampled sentences. Our analysis exposed the simple canonical ways in which verbs express relationships in English. This analysis guided the design of these Open IE systems, resulting in a substantially higher performance over previous work.

Specifically, we describe two novel Open IE systems: RE-

VERB[1] and R2A2 [Fader *et al.*, 2011; Christensen *et al.*, 2011a], which substantially improve both precision and recall when compared to previous extractors such as TEXTRUNNER and WOE. In particular, REVERB implements a novel relation phrase identifier based on generic syntactic and lexical constraints. R2A2 adds an argument identifier, ARGLEARNER, to better extract the arguments for these relation phrases. Both systems are based on almost five years of experience with Open IE systems, including TEXTRUNNER, WOE, and a careful analysis of their errors.

The remainder of the paper is organized as follows. We first define the Open IE task and briefly describe previous Open IE systems in Section 2. Section 3 outlines the architecture of REVERB and Section 4 compares this to the existing Open IE extractors. We present R2A2's argument learning component in Section 5. We compare R2A2 and REVERB in Section 6. Section 7 describes some recent research related to large scale IE. We conclude with directions for future research in Section 8.

2 Open Information Extraction

Open IE systems make a single (or constant number of) pass(es) over a corpus and extract a large number of relational tuples *(Arg1, Pred, Arg2)* without requiring any relation-specific training data. For instance, given the sentence, "McCain fought hard against Obama, but finally lost the election," an Open IE system should extract two tuples, (*McCain, fought against, Obama*), and (*McCain, lost, the election*). The strength of Open IE systems is in their efficient processing as well as ability to extract an unbounded number of relations.

Several Open IE systems have been proposed before now, including TEXTRUNNER [Banko *et al.*, 2007], WOE [Wu and Weld, 2010], and StatSnowBall [Zhu *et al.*, 2009]. All these systems use the following three-step method:

1. **Label:** Sentences are automatically labeled with extractions using heuristics or distant supervision.
2. **Learn:** A relation phrase extractor is learned using a sequence-labeling graphical model (*e.g.*, CRF).
3. **Extract:** the system takes a sentence as input, identifies a candidate pair of NP arguments (Arg1, Arg2) from the sentence, and then uses the learned extractor to label each word between the two arguments as part of the relation phrase or not.

The extractor is applied to the successive sentences in the corpus, and the resulting extractions are collected.

The first Open IE system was TEXTRUNNER [Banko *et al.*, 2007], which used a Naive Bayes model with unlexicalized POS and NP-chunk features, trained using examples heuristically generated from the Penn Treebank. Subsequent work showed that utilizing a linear-chain CRF [Banko and Etzioni, 2008] or Markov Logic Network [Zhu *et al.*, 2009] can lead to improved extractions. The WOE systems made use of Wikipedia as a source of training data for their extractors, which leads to further improvements over TEXTRUNNER [Wu and Weld, 2010]. They also show that dependency parse features result in a dramatic increase in precision and recall over shallow linguistic features, but at the cost of extraction speed.

[1]Downloadable at *http://reverb.cs.washington.edu*

Sentence	Incoherent Relation
The guide *contains* dead links and *omits* sites.	contains omits
The Mark 14 *was central* to the *torpedo* scandal of the fleet.	was central torpedo
They *recalled* that Nungesser *began* his career as a precinct leader.	recalled began

Table 1: Examples of incoherent extractions. Incoherent extractions make up approximately 13% of TEXTRUNNER's output, 15% of WOE^{pos}'s output, and 30% of WOE^{parse}'s output.

is	is an album by, is the author of, is a city in
has	has a population of, has a Ph.D. in, has a cameo in
made	made a deal with, made a promise to
took	took place in, took control over, took advantage of
gave	gave birth to, gave a talk at, gave new meaning to
got	got tickets to see, got a deal on, got funding from

Table 2: Examples of uninformative relations (left) and their completions (right). Uninformative extractions account for approximately 4% of WOE^{parse}'s output, 6% of WOE^{pos}'s output, and 7% of TEXTRUNNER's output.

2.1 Limitations in Previous Open IE Systems

We identify two significant problems in all prior Open IE systems: *incoherent extractions* and *uninformative extractions*. Incoherent extractions are cases where the extracted relation phrase has no meaningful interpretation (see Table 1 for examples). Incoherent extractions arise because the learned extractor makes a sequence of decisions about whether to include each word in the relation phrase, often resulting in incomprehensible relation phrases.

The second problem, uninformative extractions, occurs when extractions omit critical information. For example, consider the sentence "Hamas claimed responsibility for the Gaza attack". Previous Open IE systems return the uninformative: *(Hamas, claimed, responsibility)* instead of *(Hamas, claimed responsibility for, the Gaza attack)*. This type of error is caused by improper handling of light verb constructions (LVCs). An LVC is a multi-word predicate composed of a verb and a noun, with the noun carrying the semantic content of the predicate [Grefenstette and Teufel, 1995; Stevenson *et al.*, 2004; Allerton, 2002]. Table 2 illustrates the wide range of relations expressed with LVCs, which are not captured by previous open extractors.

3 ReVerb Extractor for Verb-based Relations

In response to these limitations, we introduce REVERB, which implements a general model of verb-based relation phrases, expressed as two simple constraints. We first describe the constraints and later, the REVERB architecture.

3.1 Syntactic Constraint

The syntactic constraint serves two purposes. First, it eliminates incoherent extractions, and second, it reduces uninfor-

$V \mid VP \mid VW^*P$
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

Figure 1: A simple part-of-speech-based regular expression reduces the number of incoherent extractions like *was central torpedo* and covers relations expressed via light verb constructions like *made a deal with*.

mative extractions by capturing relation phrases expressed via light verb constructions.

The syntactic constraint requires relation phrase to match the POS tag pattern shown in Figure 1. The pattern limits relation phrases to be either a simple verb phrase (*e.g.*, *invented*), a verb phrase followed immediately by a preposition or particle (*e.g.*, *located in*), or a verb phrase followed by a simple noun phrase and ending in a preposition or particle (*e.g.*, *has atomic weight of*). If there are multiple possible matches in a sentence for a single verb, the longest possible match is chosen.

Finally, if the pattern matches multiple adjacent sequences, we merge them into a single relation phrase (*e.g.*, *wants to extend*). This refinement enables the model to readily handle relation phrases containing multiple verbs. A consequence of this pattern is that the relation phrase must be a contiguous span of words in the sentence.

While this syntactic pattern identifies relation phrases with high precision, to what extent does it limit recall? To answer this, we analyzed Wu and Weld's set of 300 Web sentences, manually identifying all verb-based relationships between noun phrase pairs. This resulted in a set of 327 relation phrases.

For each relation phrase, we checked whether it satisfies the REVERB syntactic constraint. We found that 85% of the relation phrases do satisfy the constraints. Of the remaining 15%, we identified some of the common cases where the constraints were violated, summarized in Table 3. Many of these cases involve long-range dependencies between words in the sentence. As we show in Section 4, attempting to cover these harder cases using a dependency parser can actually reduce recall as well as precision.

3.2 Lexical Constraint

While the syntactic constraint greatly reduces uninformative extractions, it can sometimes match relation phrases that are so specific that they have only a few possible instances, even in a Web-scale corpus. Consider the sentence

> The Obama administration is offering only modest greenhouse gas reduction targets at the conference.

The POS pattern will match the phrase:

is offering only modest greenhouse gas reduction targets at (1)

Thus, there are phrases that satisfy the syntactic constraint, but are not useful relations.

To overcome this limitation, we introduce a lexical constraint that is used to separate valid relation phrases from over-specified relation phrases, like the example in (1). The constraint is based on the intuition that a valid relation phrase should take many distinct arguments in a large corpus. The phrase in (1) will not be extracted with many argument pairs, so it is unlikely to represent a *bona fide* relation. We describe the implementation details of the lexical constraint in Section 3.3.

	Binary Verbal Relation Phrases
85%	Satisfy Constraints
8%	Non-Contiguous Phrase Structure Coordination: X is produced and maintained by Y Multiple Args: X was founded in 1995 by Y Phrasal Verbs: X turned Y off
4%	Relation Phrase Not Between Arguments Intro. Phrases: Discovered by Y, X ... Relative Clauses: ... the Y that X discovered
3%	Do Not Match POS Pattern Interrupting Modifiers: X has a lot of faith in Y Infinitives: X to attack Y

Table 3: Approximately 85% of the binary verbal relation phrases in a sample of Web sentences satisfy our constraints.

3.3 The ReVerb Architecture

This section introduces REVERB, a novel open extractor based on the constraints defined in the previous sections. REVERB first identifies relation phrases that satisfy the syntactic and lexical constraints, and then finds a pair of NP arguments for each identified relation phrase. The resulting extractions are then assigned a confidence score using a logistic regression classifier trained on 1,000 random Web sentences with shallow syntactic features.

This algorithm differs in three important ways from previous methods. First, the relation phrase is identified "holistically" rather than word-by-word. Second, potential phrases are filtered based on statistics over a large corpus (the implementation of our lexical constraint). Finally, REVERB is "relation first" rather than "arguments first", which enables it to avoid a common error made by previous methods—confusing a noun in the relation phrase for an argument, *e.g.* the noun *responsibility* in *claimed responsibility for*.

REVERB takes as input a POS-tagged and NP-chunked sentence and returns a set of (x, r, y) extraction triples.[2] Given an input sentence s, REVERB uses the following extraction algorithm:

1. **Relation Extraction:** For each verb v in s, find the longest sequence of words r_v such that (1) r_v starts at v, (2) r_v satisfies the syntactic constraint, and (3) r_v satisfies the lexical constraint. If any pair of matches are adjacent or overlap in s, merge them into a single match.
2. **Argument Extraction:** For each relation phrase r identified in Step 1, find the nearest noun phrase x to the left of r in s such that x is not a relative pronoun, WH-term, or existential "there". Find the nearest noun phrase y to the right of r in s. If such an (x, y) pair could be found, return (x, r, y) as an extraction.

[2] REVERB uses OpenNLP for POS tagging and NP chunking: `http://opennlp.sourceforge.net/`

We check whether a candidate relation phrase r_v satisfies the syntactic constraint by matching it against the regular expression in Figure 1.

To determine whether r_v satisfies the lexical constraint, we use a large dictionary D of relation phrases that are known to take many distinct arguments. In an off-line step, we construct D by finding all matches of the POS pattern in a corpus of 500 million Web sentences. For each matching relation phrase, we heuristically identify its arguments (as in Step 2 above). We set D to be the set of all relation phrases that take at least k distinct argument pairs in the set of extractions. In order to allow for minor variations in relation phrases, we normalize each relation phrase by removing inflection, auxiliary verbs, adjectives, and adverbs. Based on experiments on a held-out set of sentences, we find that a value of $k = 20$ works well for filtering out over-specified relations. This results in a set of approximately 1.7 million distinct normalized relation phrases, which are stored in memory at extraction time.

4 ReVerb Experimental Results

We compare REVERB to the following systems:

- REVERB$^{\neg lex}$ - The REVERB system described in the previous section, but without the lexical constraint. REVERB$^{\neg lex}$ uses the same confidence function as REVERB.
- TEXTRUNNER - Banko and Etzioni's 2008 extractor, which uses a second order linear-chain CRF trained on extractions heuristically generated from the Penn Treebank. TEXTRUNNER uses shallow linguistic features in its CRF, which come from the same POS tagger and NP-chunker that REVERB uses.
- WOE^{pos} - Wu and Weld's modification to TEXTRUNNER, which uses a model of relations learned from extractions heuristically generated from Wikipedia.
- WOE^{parse} - Wu and Weld's parser-based extractor, which uses a large dictionary of dependency path patterns learned from extractions heuristically generated from Wikipedia.

Each system is given a set of sentences as input, and returns a set of binary extractions as output. We created a test set of 500 sentences sampled from the Web, using Yahoo's random link service.[3] After running each extractor over the input sentences, two human judges independently evaluated each extraction as correct or incorrect. The judges reached agreement on 86% of the extractions, with an agreement score of $\kappa = 0.68$. We report results on the subset of the data where the two judges concur. The judges labeled uninformative extractions (where critical information was dropped from the extraction) as incorrect. This is a stricter standard than was used in previous Open IE evaluations.

Each system returns confidence scores for its extractions. For a given threshold, we can measure the precision and recall of the output. Precision is the fraction of returned extractions that are correct. Recall is the fraction of correct extractions in the corpus that are returned. We use the total number of extractions from all systems labeled as correct by the judges as our measure of recall for the corpus.

[3] http://random.yahoo.com/bin/ryl

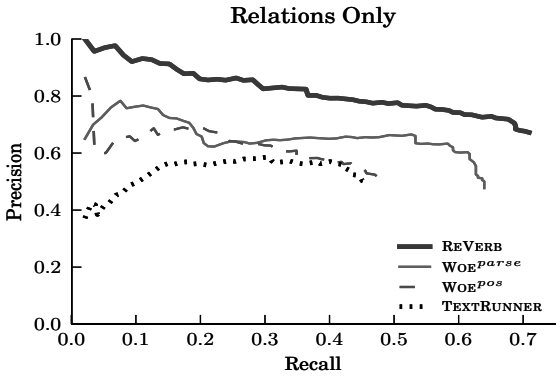

Figure 2: REVERB identifies correct relation phrases with substantially higher precision and recall than state-of-the-art open extractors, including WOE^{parse} that uses patterns learned over dependency parse paths.

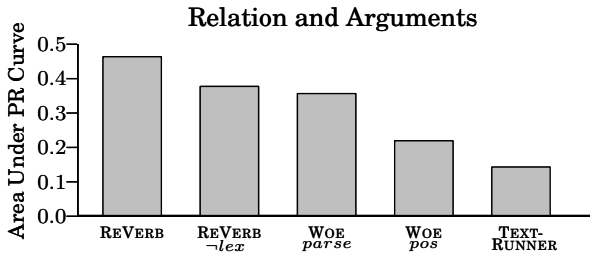

Figure 3: REVERB has AUC more than twice that of TEXTRUNNER or WOE^{pos}, and 30% higher than WOE^{parse} for extracting both relation and arguments.

We begin by evaluating how well REVERB and other Open IE systems identify correct relation phrases. As Figure 2 shows, REVERB has high precision in finding relation phrases, well above precision for the comparison systems.

The full extraction task, identifying both a relation and its arguments, produced relatively similar precision-recall curves for each system, but with a lower precision. Figure 3 shows the area under the curve (AUC) for each system. REVERB achieves an AUC that is 30% higher than WOE^{parse} and is more than double the AUC of WOE^{pos} or TEXTRUNNER. The lexical constraint provides a significant boost in performance, with REVERB achieving an AUC 23% higher than REVERB$^{\neg lex}$.

4.1 ReVerb Error Analysis

To better understand the limitations of REVERB, we performed a detailed analysis of its errors in precision (incorrect extractions returned) and its errors in recall (correct extractions that it missed). We found that 65% of the incorrect extractions returned by REVERB were cases where a relation phrase was correctly identified, but the argument-finding heuristics failed. Of the remaining errors, a common problem was to mistake an n-ary relation as a binary relation. For

example, extracting *(I, gave, him)* from the sentence "I gave him 15 photographs".

As with the false positive extractions, the majority of false negatives (52%) were due to the argument-finding heuristics choosing the wrong arguments, or failing to extract all possible arguments (in the case of coordinating conjunctions). We now turn to a system that is able to identify arguments with much higher precision than REVERB's simple heuristics.

5 Learning Arguments

In addition to the relation phrases, the Open IE task also requires identifying the proper arguments for these relations. Previous research and REVERB use simple heuristics such as extracting simple noun phrases or Wikipedia entities as arguments. Unfortunately, these heuristics are unable to capture the complexity of language. A large majority of extraction errors by Open IE systems are from incorrect or improperly-scoped arguments. Recall from previous section that 65% of REVERB's errors had a correct relation phrase but incorrect arguments.

For example, from the sentence "The cost of the war against Iraq has risen above 500 billion dollars," REVERB's argument heuristics truncate Arg1:

(Iraq, has risen above, 500 billion dollars)

On the other hand, in the sentence "The plan would reduce the number of teenagers who begin smoking," Arg2 gets truncated:

(The plan, would reduce the number of, teenagers)

In this section, we describe an argument learning component, ARGLEARNER, that reduces such errors.

5.1 Linguistic-Statistical Analysis of Extractions

Our goal is to find the largest subset of language from which we can extract reliably and efficiently. To this cause, we first analyze a sample of 250 random Web sentences to understand the frequent argument classes. We hope to answer questions such as: What fraction of arguments are simple noun phrases? Are Arg1s structurally different from Arg2s? Is there typical context around an argument that can help us detect its boundaries?

Table 4 reports our observations for frequent argument categories, both for Arg1 and Arg2. By far the most common patterns for arguments are simple noun phrases such as "Obama," "vegetable seeds," and "antibiotic use." This explains the success of previous open extractors that use simple NPs. However, we found that simple NPs account for only 65% of Arg1s and about 60% of Arg2s. This naturally dictates an upper bound on recall for systems that do not handle more complex arguments. Fortunately, there are only a handful of other prominent categories – for Arg1: prepositional phrases and lists, and for Arg2: prepositional phrases, lists, Arg2s with independent clauses and relative clauses. These categories cover over 90% of the extractions, suggesting that handling these well will boost the precision significantly.

We also explored arguments' position in the overall sentence. We found that 85% of Arg1s are adjacent to the relation phrase. Nearly all of the remaining cases are due to either compound verbs (10%) or intervening relative clauses (5%). These three cases account for 99% of the relations in our sample.

An example of compound verbs is from the sentence "Mozart was born in Salzburg, but moved to Vienna in 1781", which results in an extraction with a non-adjacent Arg1:

(Mozart, moved to, Vienna)

An example with an intervening relative clause is from the sentence "Starbucks, which was founded in Seattle, has a new logo". This also results in an extractions with non-adjacent Arg1:

(Starbucks, has, a new logo)

Arg2s almost always immediately follow the relation phrase. However, their end delimiters are trickier. There are several end delimiters of Arg2 making this a more difficult problem. In 58% of our extractions, Arg2 extends to the end of the sentence. In 17% of the cases, Arg2 is followed by a conjunction or function word such as "if", "while", or "although" and then followed by an independent clause or VP. Harder to detect are the 9% where Arg2 is directly followed by an independent clause or VP. Hardest of all is the 11% where Arg2 is followed by a preposition, since prepositional phrases could also be part of Arg2. This leads to the well-studied but difficult prepositional phrase attachment problem. For now, we use limited syntactic evidence (POS-tagging, NP-chunking) to identify arguments, though more semantic knowledge to disambiguate prepositional phrases could come in handy for our task.

5.2 Design of ARGLEARNER

Our analysis of syntactic patterns reveals that the majority of arguments fit into a small number of syntactic categories. Similarly, there are common delimiters that could aid in detecting argument boundaries. This analysis inspires us to develop ARGLEARNER, a learning-based system that uses these patterns as features to identify the arguments, given a sentence and relation phrase pair.

ARGLEARNER divides this task into two subtasks - finding Arg1 and Arg2 - and then subdivides each of these subtasks again into identifying the left bound and the right bound of each argument. ARGLEARNER employs three classifiers to this aim (Figure 4). Two classifiers identify the left and right bounds for Arg1 and the last classifier identifies the right bound of Arg2. Since Arg2 almost always follows the relation phrase, we do not need a separate Arg2 left bound classifier.

We use Weka's REPTree [Hall *et al.*, 2009] for identifying the right boundary of Arg1 and sequence labeling CRF classifier implemented in Mallet [McCallum, 2002] for other classifiers. Our standard set of features include those that describe the noun phrase in question, context around it as well as the whole sentence, such as sentence length, POS-tags, capitalization and punctuation. In addition, for each classifier we use features suggested by our analysis above. For example, for right bound of Arg1 we create regular expression indicators to detect whether the relation phrase is a compound verb and whether the noun phrase in question is a subject of the compound verb. For Arg2 we create regular expression indicators to detect patterns such as Arg2 followed by an independent clause or verb phrase. Note that these indicators

Category	Patterns	Frequency Arg1	Frequency Arg2
Basic NP	NN, JJ NN, etc	65% **Chicago** *was founded in* 1833.	60% Calcium *prevents* **osteoporosis**.
Prepositional Attachments	NP PP^+	19% **The forest in Brazil** *is threatened by* ranching.	18% Lake Michigan *is one of* **the five Great Lakes of North America**.
List	NP (,NP)* ,? and/or NP	15% **Google and Apple** *are headquartered in* Silicon Valley.	15% A galaxy *consists of* **stars and stellar remnants**.
Independent Clause	(that|WP|WDT)? NP VP NP	0% **Google will acquire YouTube,** *announced* the New York Times.	8% Scientists *estimate* **that 80% of oil remains a threat**.
Relative Clause	NP (that|WP|WDT) VP NP?	<1% **Chicago, which is located in Illinois**, *has* three million residents.	6% Most galaxies *appear to be* **dwarf galaxies, which are small**.

Table 4: Taxonomy of arguments for binary relationships. In each sentence, the argument is bolded and the relational phrase is italicized. Multiple patterns can appear in a single argument so percentages do not need to add to 100. In the interest of space, we omit argument structures that appear in less than 5% of extractions. Upper case abbreviations represent noun phrase chunk abbreviations and part-of-speech abbreviations.

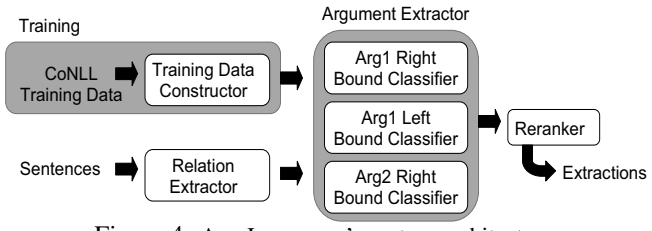

Figure 4: ARGLEARNER's system architecture.

		REVERB	R2A2
Web	Arg1	0.69	**0.81**
	Arg2	0.53	**0.72**
News	Arg1	0.75	**0.86**
	Arg2	0.58	**0.74**

Table 5: R2A2 has substantially higher F1 score than REVERB for both Argument 1 and Argument 2.

will not match all possible sentence structures, but act as useful features to help the classifier identify the categories. We design several features specific to these different classifiers.

The other key challenge for a learning system is training data. Unfortunately, there is no large training set available for Open IE. We build a novel training set by adapting data available for semantic role labeling (SRL), which is shown to be closely related to Open IE [Christensen *et al.*, 2011b]. We found that a set of post-processing heuristics over SRL data can easily convert it into a form meaningful for Open IE training.

We used a subset of the training data adapted from the CoNLL 2005 Shared Task [Carreras and Marquez, 2005]. Our dataset consists of 20,000 sentences and generates about 29,000 Open IE tuples. The cross-validation accuracies of the classifiers on the CoNLL data are 96% for Arg1 right bound, 92% for Arg1 left bound and 73% for Arg2 right bound. The low accuracy for Arg2 right bound is primarily due to Arg2's more complex categories such as relative clauses and independent clauses and the difficulty associated with prepositional attachment in Arg2.

Additionally, we train a confidence metric on a hand-labeled development set of random Web sentences. We use Weka's implementation of logistic regression, and use the classifier's weight to order the extractions.

We name our final system that combines REVERB relation phrases with ARGLEARNER's arguments as R2A2. We evaluate R2A2 against REVERB next.

6 R2A2 Experimental Results

We conducted experiments to answer the following questions. (1) Does R2A2 improve argument detection compared to arguments returned by REVERB's simple heuristics? and (2) What kind of errors does R2A2 reduce and which errors require more research?

We tested the systems on two datasets. The first dataset consists of 200 random sentences from the New York Times. The second dataset is made up of 200 random Web sentences. Three judges with linguistic backgrounds evaluated the output of the systems, labeling whether the relation phrase was correct, and, if so, whether arguments 1 and 2 were the correct arguments for the relation.

We used a stricter criterion for correct arguments than previous Open IE evaluations, which counted an extraction as correct if its arguments were reasonable, even if they omitted relevant information. The annotators were instructed to mark an argument correct only if the argument was as informative as possible, but did not include extraneous information.

For example, "the President of Brazil" is a correct Arg2 for the relation "spoke to" in the sentence "President Obama spoke to the President of Brazil on Thursday", but the less informative "the President" is considered incorrect, as is "the President of Brazil on Thursday". The inter-annotator agreement for the three judges is 95%.

In this evaluation, we are primarily concerned with the effect of ARGLEARNER, hence do not consider possibly correct extractions missed by REVERB, since neither system has

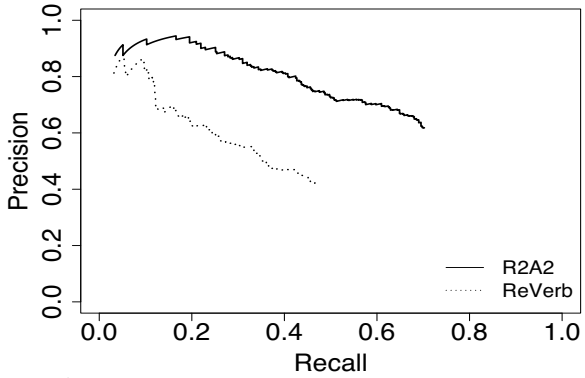

Figure 5: R2A2 has substantially higher recall and precision than REVERB.

a chance of extracting them. In other words, our recall calculations use the total number of correct extractions possible using REVERB relation phrases as the denominator.

R2A2 has both precision and recall substantially above that of REVERB. Table 5 compares F1 for each argument on both data sets at the confidence values that produce the highest F1 on a development set. R2A2 increases the F1 by 11 - 19 points. Figure 5 shows recall and precision for the entire extraction on the combined set of Web and newswire sentences. R2A2 has over 0.20 higher precision than REVERB over nearly all the precision-recall curve as well as higher recall.

We also analyzed how R2A2 performs on the argument types and context patterns identified in Section 5.1. R2A2 had high F1 (from 0.82 to 0.95) on the major patterns of Arg1: simple NP, prepositional attachment, and list. For the example sentence given at the beginning of Section 5, R2A2 correctly extracts "The cost of war against Iraq" as Arg1 instead of "Iraq."

Its performance on adjacent Arg1 is high (0.93), but could be improved on compound verbs (0.69) and intervening relative clauses (0.73). R2A2 also has difficulty recognizing when the given relation has no valid Arg1, such as in, "If interested in this offer, please contact us," with relation phrase "contact." Future systems could make better use of negative training data to correct this issue.

For Arg2, R2A2 performs well on simple noun phrases, with an F1 of 0.87. However, F1 is between 0.62 and 0.71 for all other syntactic patterns for Arg2. This is considerably above REVERB's F1 of 0.0 to 0.19 on these patterns, but still leaves considerable room for improvement. In contrast to REVERB, R2A2 also gets the second example sentence from the previous section extracting *(The plan, would reduce the number of, teenagers who begin smoking)*.

7 Related Work

Web-scale information extraction has received considerable attention in the last few years. Pre-emptive Information Extraction and Open Information Extraction are the first paradigms that relax the restriction of a given vocabulary of relations and scale to all relation phrases expressed in text [Shinyama and Sekine, 2006; Banko *et al.*, 2007; Banko and Etzioni, 2008; Wu and Weld, 2010]. Preemptive IE relies on document and entity clustering, which is too costly for Web-scale IE. Open IE favors speed over deeper processing, which aids in scaling to Web-scale corpora.

There is recent work on incorporating distant supervision from manually constructed knowledge bases such as FreeBase or Wikipedia to automatically learn extractors for the large number of relations in the KB [Mintz *et al.*, 2009; Hoffmann *et al.*, 2010; 2011]. These approaches use heuristic methods to generate training data by mapping the entities of the KB to sentences mentioning them in text. This reduces the relation extraction problem to a multi-class supervised learning problem.

The Never Ending Language Learner (NELL) project aims to learn a macro-reading agent that gets better and better at reading as it reads the same text multiple times [Carlson *et al.*, 2010a; 2010b]. Essentially, NELL learns newer extraction patterns using previous system extraction instances as training data. Such pattern learning systems in the past have been prone to concept drift. NELL largely overcomes concept drift by employing coupled-training, which generates negative training for one concept based on the positive example of another and known mutual exclusions between types. There is also a sophisticated mechanism to advance a hypothesized extraction to an accepted extraction.

NELL and the distant supervision approaches differ from our Open IE paradigm in an important way – they all learn extractors for a known set of relations. Distant supervision approaches have scaled to a few thousand relations, whereas NELL knowledge base is much smaller, extracting around a hundred relations. In contrast, our recent-most run of Open IE on a Web-scale corpus returned about 1.5 million distinct relation phrases. In other words, Open IE can be applied as is to any domain and any corpora of English text and it will extract meaningful information. The flip side to Open IE is its unnormalized output. Open IE has key challenges due to polysemous and synonymous relation phrases. Our follow-up work attempts to learn synonymous relations [Yates and Etzioni, 2009] as well as proposes a first solution to normalize open relation phrases to a domain ontology with minimal supervision [Soderland *et al.*, 2010].

8 Conclusions

We have described the second generation Open Information Extraction systems, REVERB, and R2A2. The key differentiating characteristic of these systems is a linguistic analysis that guides the design of the constraints in REVERB and features in R2A2. REVERB focuses on identifying a more meaningful and informative relation phrase and outperforms the previous Open IE systems by significant margins. R2A2 adds an argument learning component, ARGLEARNER, and almost doubles the area under precision-recall curve compared to REVERB. Both these systems are amazingly scalable, since they require only shallow syntactic features.

There are three key directions to pursue this work further. First, while Open IE systems have primarily focused on binary extractions, not all relationships are binary. Events have time and locations. Numerous verbs naturally take three arguments (*e.g.*, *"Singh gifted the French President a tra-*

ditional painting."). We need to extend Open IE to handle n-ary and even nested extractions. Secondly, not all important relationships are expressed in verbs. For example, the sentence *"Seattle Mayor Bloomberg said that..."* expresses *(Bloomberg, is the Mayor of, Seattle)*. However, such noun-based extractions are challenging to get at high precision, since exposing the semantics hidden in these compound nouns is tricky. For example, "Seattle Symphony Orchestra" does not imply that *(Orchestra, is the Symphony of, Seattle)*. Finally, as our current Open IE systems learn general characteristics of the English language, we recognize that the techniques to handle another language, say Chinese, will likely be quite different. However, we believe that the general Open IE paradigm will extend to other languages.

Acknowledgments: This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-08-1-0431, and DARPA contract FA8750-09-C-0179.

References

[Allerton, 2002] David J. Allerton. *Stretched Verb Constructions in English*. Routledge Studies in Germanic Linguistics. Routledge (Taylor and Francis), New York, 2002.

[Banko and Etzioni, 2008] Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *ACL'08*, 2008.

[Banko *et al.*, 2007] Michele Banko, Michael J. 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'11*, 2011.

[Carlson *et al.*, 2010a] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI'10*, 2010.

[Carlson *et al.*, 2010b] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM 2010*, 2010.

[Carreras and Marquez, 2005] Xavier Carreras and Lluis Marquez. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling, 2005.

[Christensen *et al.*, 2011a] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Learning Arguments for Open Information Extraction. Submitted, 2011.

[Christensen *et al.*, 2011b] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. The tradeoffs between syntactic features and semantic roles for open information extraction. In *Knowledge Capture (KCAP)*, 2011.

[Etzioni *et al.*, 2006] Oren Etzioni, Michele Banko, and Michael J. Cafarella. Machine reading. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[Fader *et al.*, 2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying Relations for Open Information Extraction. Submitted, 2011.

[Grefenstette and Teufel, 1995] Gregory Grefenstette and Simone Teufel. Corpus-based method for automatic identification of support verbs for nominalizations. In *EACL'95*, 1995.

[Hall *et al.*, 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 1(1), 2009.

[Hoffmann *et al.*, 2010] Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. Learning 5000 relational extractors. In *ACL '10*, 2010.

[Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Distant supervision for information extraction of overlapping relations. In *ACL '11*, 2011.

[Kim and Moldovan, 1993] J. Kim and D. Moldovan. Acquisition of semantic patterns for information extraction from corpora. In *Procs. of Ninth IEEE Conference on Artificial Intelligence for Applications*, pages 171–176, 1993.

[Lin *et al.*, 2010] Thomas Lin, Mausam, and Oren Etzioni. Identifying Functional Relations in Web Text. In *EMNLP'10*, 2010.

[McCallum, 2002] Andres McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP'09*, 2009.

[Riloff, 1996] E. Riloff. Automatically constructing extraction patterns from untagged text. In *AAAI'96*, 1996.

[Ritter *et al.*, 2010] Alan Ritter, Mausam, and Oren Etzioni. A Latent Dirichlet Allocation Method for Selectional Preferences. In *ACL*, 2010.

[Ritter *et al.*, 2011] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named Entity Recognition in Tweets: An Experimental Study. Submitted, 2011.

[Schoenmackers *et al.*, 2010] Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. Learning first-order horn clauses from web text. In *EMNLP'10*, 2010.

[Shinyama and Sekine, 2006] Yusuke Shinyama and Satoshi Sekine. Preemptive Information Extraction using Unrestricted Relation Discovery. In *NAACL'06*, 2006.

[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, 1999] S. Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1-3):233–272, 1999.

[Stevenson *et al.*, 2004] Suzanne Stevenson, Afsaneh Fazly, and Ryan North. Statistical measures of the semi-productivity of light verb constructions. In *2nd ACL Workshop on Multiword Expressions*, pages 1–8, 2004.

[Wu and Weld, 2010] Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 118–127, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

[Yates and Etzioni, 2009] A. Yates and O. Etzioni. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34(1):255–296, 2009.

[Zhu *et al.*, 2009] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. StatSnowball: a statistical approach to extracting entity relationships. In *WWW'09*, 2009.