



IE Systems

Contents

- Open Information Extraction
 - TextRunner System 2007
 - WOE system 2010
 - Reverb 2011
 - ARGlearner 2011
 - OLLIE 2012

Disadvantages of KnowItAll

<http://www.cs.washington.edu/research/knowitall>

- ❖ Bottleneck to **scalability**: issuing **search engines** queries has limitations.
- ❖ PMI-based assessor is effective, however it needs **search engines**. 搜索引擎是关键。
- ❖ Relation-specific, require **a laborious bootstrapping process** for each relation of interest.
- ❖ Due to the templates, extractions are limited. 由于**模板的约束**，抽取的内容有限。

Research Background

- Aim : scale IE methods to the **size** and **diversity** of the Web corpus.
- Problems:
 - An unbound number of relations
 - They are not known in advance
- Solution:
 - Learn **a general model** of how relations are expressed in a particular language.
 - **A relation-independent** extractor.

Open Information Extraction

--- Development

- **Knowledge-based encoding:** a human enters regular expressions or rules.
- **Supervised learning:** a human provides labeled training examples.
- **Self-supervised learning:** the system automatically finds and labels its own examples:
 1. KnowItAll --- 2003
 2. TextRunner --- 2007
 3. The second generation of Open IE –2011

What is open IE?

- Open IE: no hand-labeled training examples, and avoids domain-specific verbs and nouns, to develop *unlexicalized, domain-independent* extractors that scale to the Web corpus.
- Unlexicalized**: formulated only in terms of syntactic tokens (e.g., part-of-speech tags) and closed-word classes (e.g., of, in, such as).

What is open IE? (cont.)

Patterns of open IE vs. patterns of KnowItAll

Relative Frequency	Category	Simplified Lexico-Syntactic Pattern
37.8	Verb	E_1 Verb E_2 <i>X established Y</i>
22.8	Noun + Prep	E_1 NP Prep E_2 <i>X settlement with Y</i>
16.0	Verb + Prep	E_1 Verb Prep E_2 <i>X moved to Y</i>
9.4	Infinitive	E_1 to Verb E_2 <i>X plans to acquire Y</i>
5.2	Modifier	E_1 Verb E_2 Noun <i>X is Y winner</i>
1.8	Coordinate _n	E_1 (and , - :) E_2 NP <i>X-Y deal</i>
1.0	Coordinate _v	E_1 (and ,) E_2 Verb <i>X, Y merge</i>
0.8	Appositive	E_1 NP (: ,)? E_2 <i>X hometown : Y</i>

(*<proper noun>*, acquired, *<proper noun>*)
(*<proper noun>*, graduated from, *<proper noun>*)
(*<proper noun>*, is author of, *<proper noun>*)
(*<proper noun>*, is based in, *<proper noun>*)
(*<proper noun>*, studied, *<noun phrase>*)
(*<proper noun>*, studied at, *<proper noun>*)
(*<proper noun>*, was developed by, *<proper noun>*)
(*<proper noun>*, was formed in, *<year>*)
(*<proper noun>*, was founded by, *<proper noun>*)
(*<proper noun>*, worked with, *<proper noun>*)

Traditional IE v.s Open IE

	Traditional IE	Open IE
Input	Corpus + labeled seeds	Corpus+ Domain independent Knowledge
Relations	Specified in advance	Discovered automatically
Complexity	$O(D \cdot R)$ D:documents R:relations	$O(D)$ D: documents
Text analysis	Parser + named entity tagger	Parser + NP chunker

Task of Open IE

- **Input:** "McCain fought hard against Obama, but finally lost the election,"
- **Output:** two tuples,
(McCain, fought against, Obama)
(McCain, lost, the election).

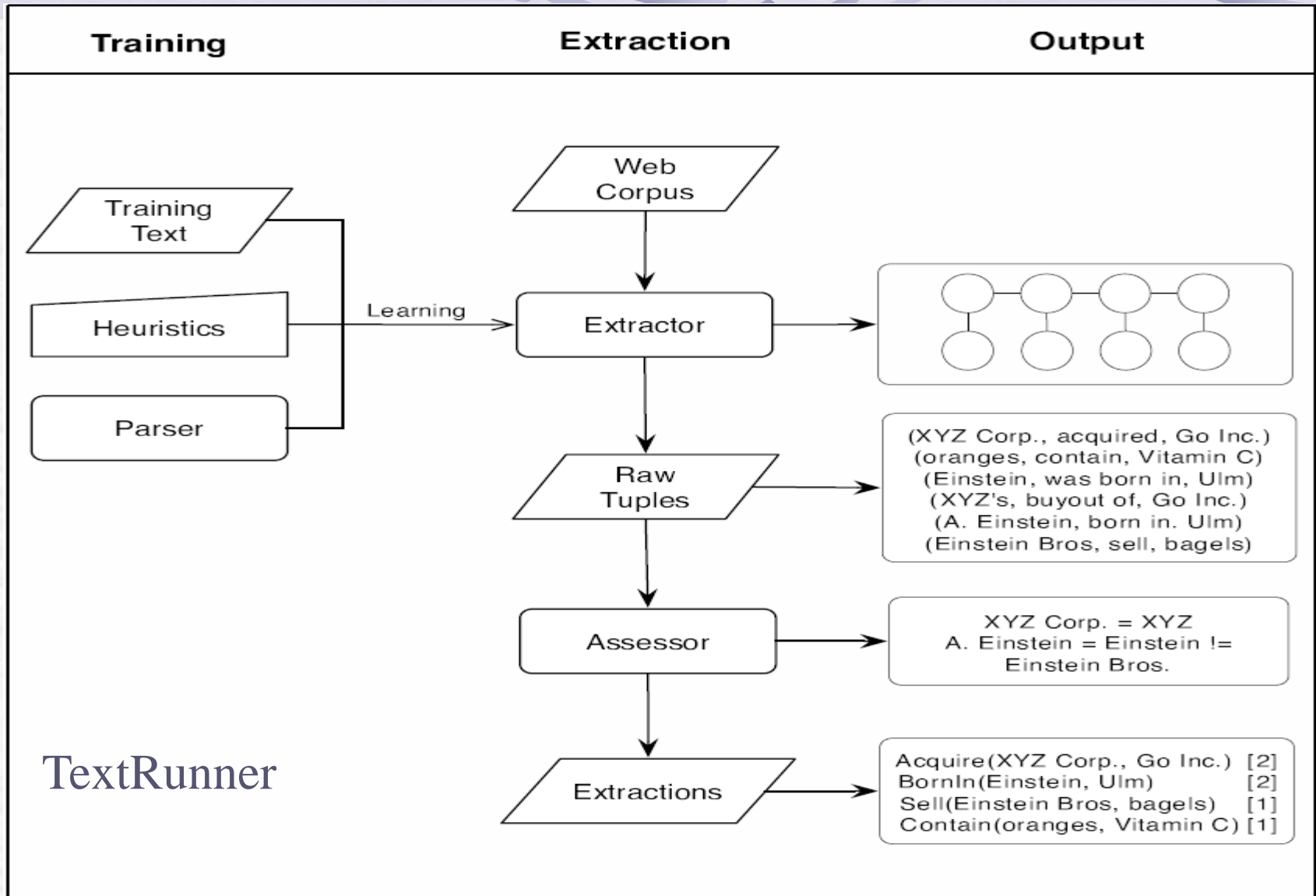
Features:

- a single pass over a corpus (**size** → fast)
- any relation extraction (**diversity** → domain independent)

Open IE: Representation

- Any relation can be represented as $t=(e1,r,e2)$ where $e1,e2$ are strings meant to denote **entities**, while r is a string meant to denote **a relationship** between them. Such as:
 - (McCain, fought against, Obama)
 - (McCain, lost, the election).

Open IE: Implementation



Open IE: Features

- **Self-Supervised Learner:** Given a small corpus sample as input, the Learner outputs a **classifier** that labels candidate extractions as “trustworthy” or not. The Learner requires no hand-tagged data.
- **Single-Pass Extractor:**
 - Extract tuples for all possible relations from the entire corpus.
 - Use **classifier** to identify, and retain the ones labeled as trustworthy.
- **Redundancy-Based Assessor:** assign a probability to each retained tuple based on a probabilistic model of redundancy in text.

Self-Supervised Learner

- the Learner uses a parser to automatically identify and label a set of trustworthy (and untrustworthy) extractions. -- *seeds generation*
- It maps each tuple to a domain independent feature vector representation, such as, *the presence of part-of-speech tag sequences in the relation r , the number of tokens in r , whether e is a proper noun, ...* -- *feature generation*
- These extractions are used as positive (or negative) training examples to a Naive Bayes classifier and a CRF model. *classifier generation*

Self supervised Learner: Seeds Generation

Some constraints:

- Dependency chain between e1 and e2
- Path(e1,e2) does not cross sentence-like boundary. E.g.

"George Bush, president of the US..."

(George Bush, president of, the US) can not be found from the above sentence.

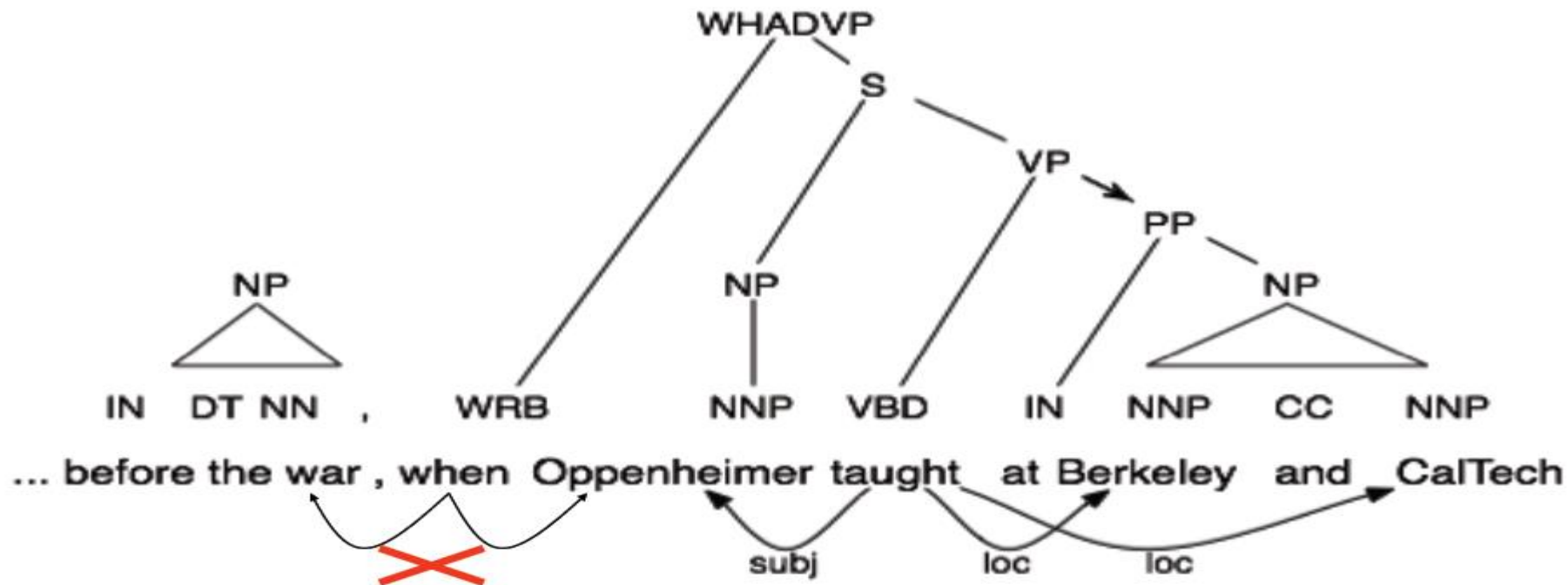
- Neither e1 nor e2 is a pronoun.

Self Supervised Learner: Naïve Bayes Classifier

Features for a classifier

- ✓ POS sequence on relation r
 - ✓ Number of tokens r
 - ✓ Number of stop words r
 - ✓ POS tag of e
 - ✓ Left/right POS of e
- Classifier is **language specific**, but does not contain relation-specific or lexical features → **domain independent**.

Example of Extractor Training



Instances
t1 (+) (Oppenheimer, taught at, Berkeley)
t2 (+) (Oppenheimer, taught at, CaTech)
t3 (-) (war, , when, Oppenheimer)

Features(t1)
IsProperNoun(NP1) = true
LeftContext(NP1) = WRB
Rel-Length = 2
Rel-NumStopWords = 1
Rel-StartsWith = VB
Rel-Contains(VB) = true
Rel-Contains(VB IN) = true
...

Single-Pass Extractor: use Naïve Bayes Classifier

- Automatically tagging each word by POS, entities are found by identifying noun phrases.
- Relations are found by examining the text between the noun phrases. -- ***candidates generation***
- If the classifier labels t as trustworthy, it is extracted and stored by TEXTRUUNER – ***result generation***

e.g.: ...After the war, when Heiker taught at Berkeley and CalTech. ...

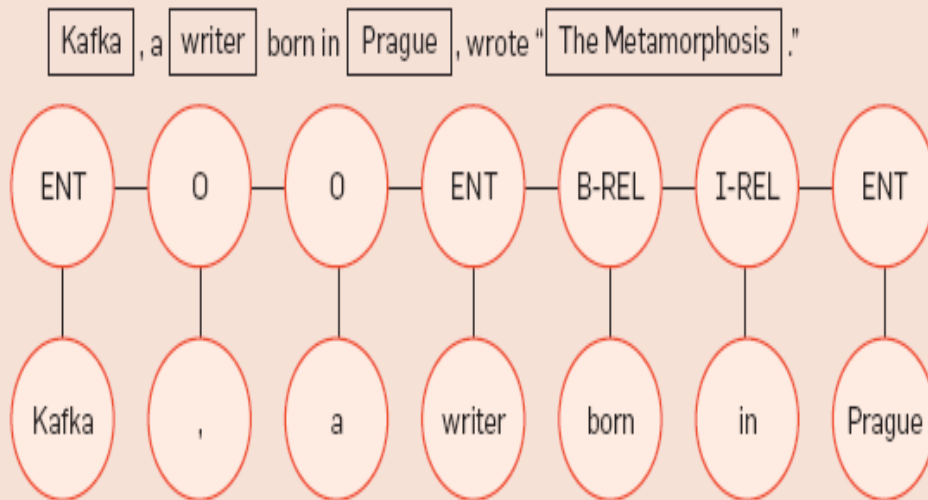
→ t1: (war, when, Heiker) ×

→ t2: (Heiker, taught at, Berkeley) ✓

→ t3: (Heiker, taught at, CalTech) ✓

Single-Pass Extractor: use CRF

Figure 3: Information extraction as sequence labeling. A CRF is used to identify the relationship, *born in*, between *Kafka* and *Prague*. Entities are labeled as ENT. The B-REL label indicates the start of a relation, with I-REL indicating the continuation of the sequence.



- ☛ *Entities are found.*
- ☛ *A CRF is used to identify the relationship.*

Redundancy-Based Assessor

- **Normalize** the relation by omitting non-essential modifiers to verbs and nouns.
“was originally developed by” → “was developed by”
- **Merge** tuples *where both entities and normalized relation are identical* and **counts** the number of distinct sentences from which each extraction are found.

0.97	41	Alexander Graham Bell	invented	the telephone
0.97	36	Thomas Edison	invented	light bulbs
0.97	29	Eli Whitney	invented	the cotton gin

- Use these counts to **assign a probability** (used in KnowItAll system) **---result evaluation**

Comparison with KnowItAll

- KnowItAll extraction is based on 10 focuses. →
- Open IE operates **without knowing the relations a priori** and extracts information from all relations at once.

(<proper noun>, acquired, <proper noun>)
(<proper noun>, graduated from, <proper noun>)
(<proper noun>, is author of, <proper noun>)
(<proper noun>, is based in, <proper noun>)
(<proper noun>, studied, <noun phrase>)
(<proper noun>, studied at, <proper noun>)
(<proper noun>, was developed by, <proper noun>)
(<proper noun>, was formed in, <year>)
(<proper noun>, was founded by, <proper noun>)
(<proper noun>, worked with, <proper noun>)

	Average Error rate	Correct Extractions
TEXTRUNNER	12%	11,476
KNOWITALL	18%	11,631

Table 1: Over a set of ten relations, TEXTRUNNER achieved a **33% lower error rate** than KNOWITALL, while finding approximately as many correct extractions.

Run Time:

KnowItAll: 10 relations → ~3 days

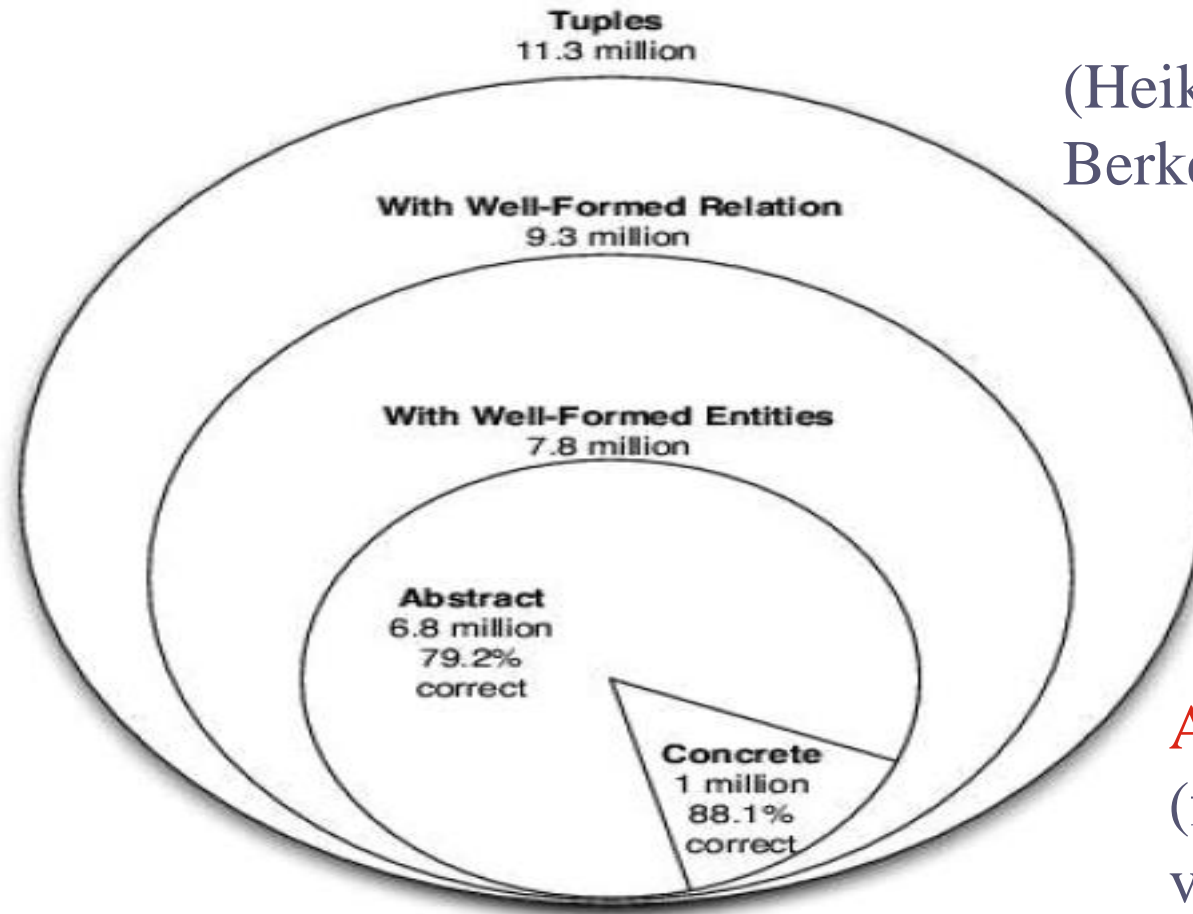
TextRunner: 10^3 - 10^5 → ~3 days

Two to four orders of magnitude boost!

TextRunner results

Concrete facts:

(Heike, taught at,
Berkeley)



Abstract facts:
(fruit, contain,
vitamins)

Figure 1: Overview of the tuples extracted from 9 million Web page corpus. 7.8 million well-formed tuples are found having probability ≥ 0.8 . Of those, TEXTRUNNER finds 1 million concrete tuples with arguments grounded in particular real-world entities, 88.1% of which are correct, and 6.8 million tuples reflecting abstract assertions, 79.2% of which are correct.

Some Problems of TextRunner

- **Incoherent extractions:** the extracted relation phrase has no meaningful interpretation.

Sentence	Incoherent Relation
The guide <i>contains</i> dead links and <i>omits</i> sites.	contains omits
The Mark 14 <i>was central</i> to the <i>torpedo</i> scandal of the fleet.	was central torpedo
They <i>recalled</i> that Nungesser <i>began</i> 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.

• **Uninformative results,** occurs when extractions omit critical information

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.

Second generation of Open IE

TO improve both precision and recall:

- **Reverb**: implements a novel relation phrase identifier based on generic syntactic and lexical constraints.
- **R2A2**: adds an argument identifier to better extract the arguments for these relation phrases.

Reverb

- verb-based relation phrases, expressed as two simple constraints:
- Syntactic constraint:**

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

- Lexical constraint:** relation with many argument pairs will not be extracted.

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

Reverb (cont.)

- Input: a POS-tagged and NP chunked sentence (s)
- Output: a set of (x,r,y) extraction triples.

- 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.
- 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.

R2A2 motivation

- ↪ Input a sentence
“The cost of the war against Iraq has risen above 500 billion dollars,”
 - ↪ Output:
(Iraq, has risen above, 500 billion dollars)
 - ↪ Reason: REVERB’s argument heuristics truncate Arg1:
 - ↪ Input: a sentence “The plan would reduce the number of teenagers who begin smoking,” Arg2 gets truncated:
 - ↪ Output:
(The plan, would reduce the number of, teenagers)
- **argument learning component, that reduces such errors.**

R2A2

☛ Aim:

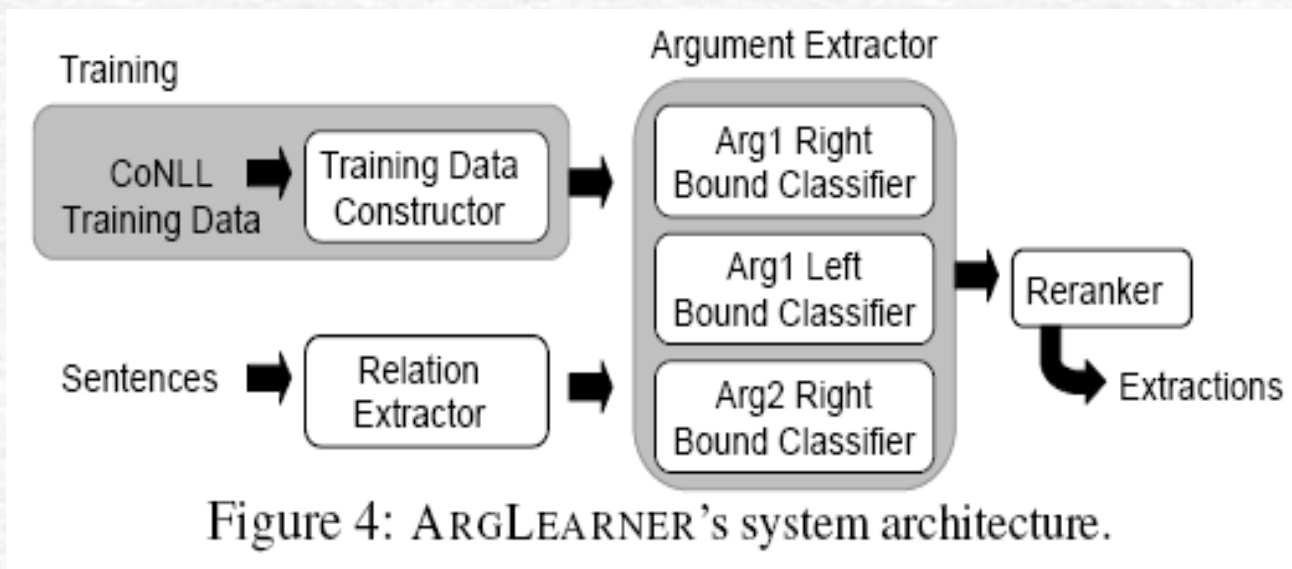
to identify the **relation arguments** (**ARG1,ARG2**), given a sentence and relation phrase pair.

☛ Problems to be solved:

identifying the **left bound** and the **right bound** of each argument.

R2A2 (cont.)

- Methods: **three classifiers.**
- Features include the noun phrase in question, context around it, POS-tags, capitalization, punctuation and ...



Frequent Argument Categories, both for Arg1 and Arg2

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

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.

Experiment Results

- Evaluate R2A2 against Reverb

		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.

Two Weaknesses

- Extract relations that are mediated only by verbs.

E.g: “there are plenty of taxis available at Bali airport”

- Ignore context, extracting tuples are not facts.

E.g: “if he wins five key states, Romney will be elected President”

OLLIE

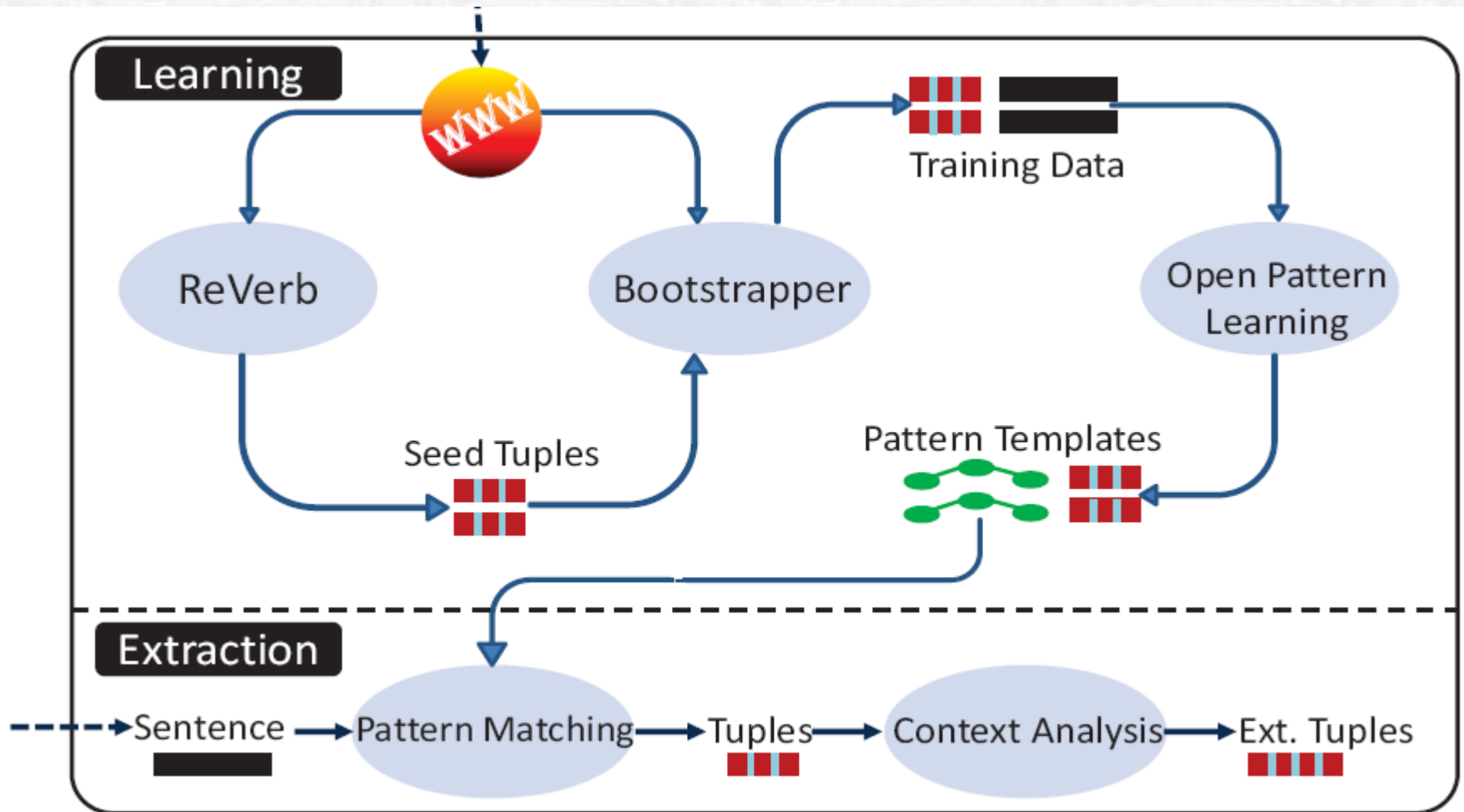
- Extract relations **mediated by nouns, adjectives and more.**

“there are plenty of taxis available at Bali airport” → (taxis, be available at, Bali airport)

- Add context –analysis**

“if he wins five key states, Romney will be elected President” → ((Romney, will be elected, president) ClausalModifier If, he wins five key states) instead of (Romney, will be elected, President)

Architecture of OLLIE



Bootstrapper

- **Aim:** uses a set of high precision seed tuples from REVERB to bootstrap a **large training set.**
- **Size:** 110,000 seed tuples and corresponding 18 million sentences.
- **Tools:** use Malt Dependency Parser for dependency parsing. Post-process the parses using Stanford's Ccprocessed algorithm.

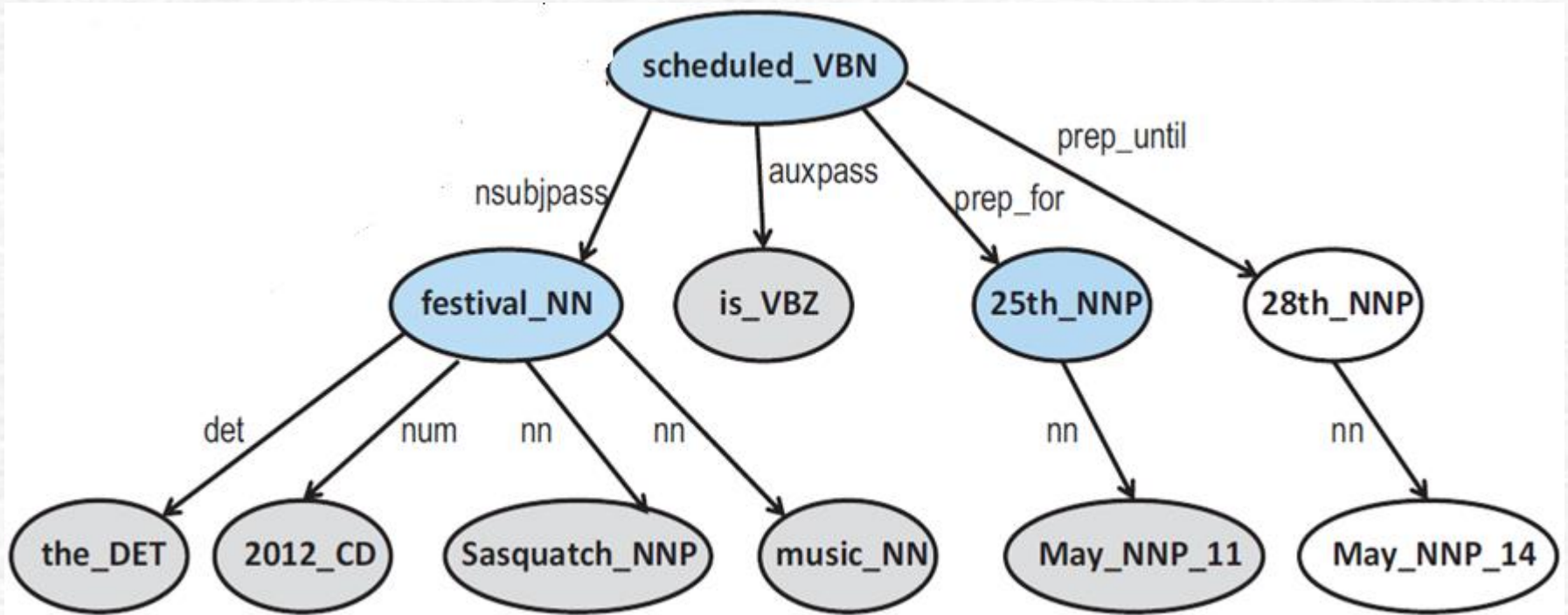
Open pattern Learning

- Input: Training corpus (includes dependency path, relation words and the associated sentence)
- Output: open extraction patterns, such as

Extraction Template	Open Pattern
1. (arg1; be {rel} {prep}; arg2)	{arg1} ↑nsubjpass↑ {rel:postag=VBN} ↓{prep_*}↓ {arg2}
2. (arg1; {rel}; arg2)	{arg1} ↑nsubj↑ {rel:postag=VBD} ↓dobj↓ {arg2}
3. (arg1; be {rel} by; arg2)	{arg1} ↑nsubjpass↑ {rel:postag=VBN} ↓agent↓ {arg2}
4. (arg1; be {rel} of; arg2)	{rel:postag=NN;type=Person} ↑nn↑ {arg1} ↓nn↓ {arg2}
5. (arg1; be {rel} {prep}; arg2)	{arg1} ↑nsubjpass↑ {slot:postag=VBN;lex ∈announce} ↓dobj↓ {rel:postag=NN} ↓{prep_*}↓ {arg2}

Dependency Tree

- The 2012 Sasquatch music festival is scheduled for May 25th until May 28th
- (*the 2012 Sasquatch Music Festival; is scheduled for; May 25th*)



→ the pattern $\{arg1\} \uparrow nsubjpass \uparrow \{rel:postag=VBN\}$
 $\downarrow \{prep *\} \downarrow \{arg2\}$.

Purely Syntactic Patterns

Extraction Template	Open Pattern
1. (arg1; be {rel} {prep}; arg2)	{arg1} ↑nsubjpass↑ {rel:postag=VBN} ↓{prep_*}↓ {arg2}
2. (arg1; {rel}; arg2)	{arg1} ↑nsubj↑ {rel:postag=VBD} ↓dobj↓ {arg2}
3. (arg1; be {rel} by; arg2)	{arg1} ↑nsubjpass↑ {rel:postag=VBN} ↓agent↓ {arg2}


In order to **generalize** to unseen relations and prepositions:

- Remove all lexical restrictions from the relation nodes.
- Convert all preposition edges to an abstract {prep_*}
- Replace the specific preposition with {prep}.

Semantic/lexical Patterns

4. (arg1; be {rel} of; arg2)	{rel:postag=NN;type=Person} ↑nn↑ {arg1} ↓nn↓ {arg2}
5. (arg1; be {rel} {prep}; arg2)	{arg1} ↑nsubjpass↑ {slot:postag=VBN;lex ∈ announce name choose...} ↓dobj↓ {rel:postag=NN} ↓{prep-*}↓ {arg2}

Motivation

- Microsoft co-founder Bill Gates... → (Microsoft, co-founder, Bill Gates)
- Chicago Symphony Orchestra.. → 
(Chicago, symphony, orchestra)

Methods:

- Convert **lexical constrains** into a list of words (pattern 5)
- Generalize to some types based on **Wordnet class** (pattern 4)

Pattern Matching

- **Use** these open patterns to extract binary relations from a new sentence.
- **Match** the patterns with the dependency parse of the sentence.
- **Identify** the base nodes for arguments and relations.

Context Analysis

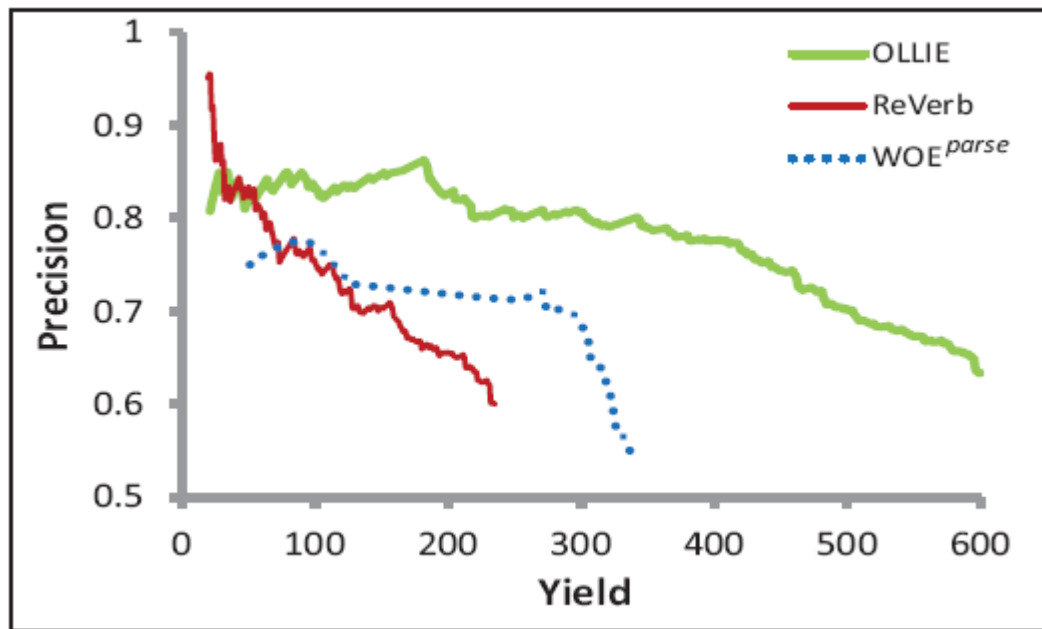
- OLLIE **adds** an *AttributedTo* field to represent who said, suggested, believes, hopes or doubts the information.
(the earth, be the center of, the universe)
attributedTo believe, early astronomers)
- OLLIE **adds** a *ClausalModifier* field to extract those only conditionally true.
- How to do them?
 1. Make use of the **dependency parse structure** with a **list of keywords**, such as “believe, and other cognition verb”, or “if, when, although,...” conditional words.

Context Analysis (cont.)

2. Use a supervised **logistic regression classifier** for the confidence function.
 - Features: **frequency** of the extraction patterns, the **presence** of *AttributedTo* or *ClausalModifier* fields, the **position** of certain words in the extraction's context.

Experiments of OLLIE

- OLLIE performance vs existing state-of-the-art OIE?



Relation	OLLIE	REVERB	incr.
<i>is capital of</i>	8,566	146	59x
<i>is president of</i>	21,306	1,970	11x
<i>is professor at</i>	8,334	400	21x
<i>is scientist of</i>	730	5	146x

Conclusions: 1) OLLIE finds **4.4 times** more correct extractions than REVERB at a **precision of about 0.75**. 2) OLLIE found up to **146 times** as many extractions for these relations than REVERB.

Experiments of OLLIE

- Patterns without semantic / lexical restriction (blue line) produces lower precision.

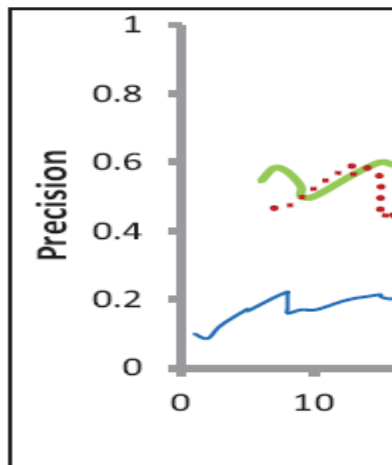


Figure 7: Results on terms with semantic/lexical restriction on patterns with semantic/lexical restriction. Type generalization on patterns with only lexical restriction.

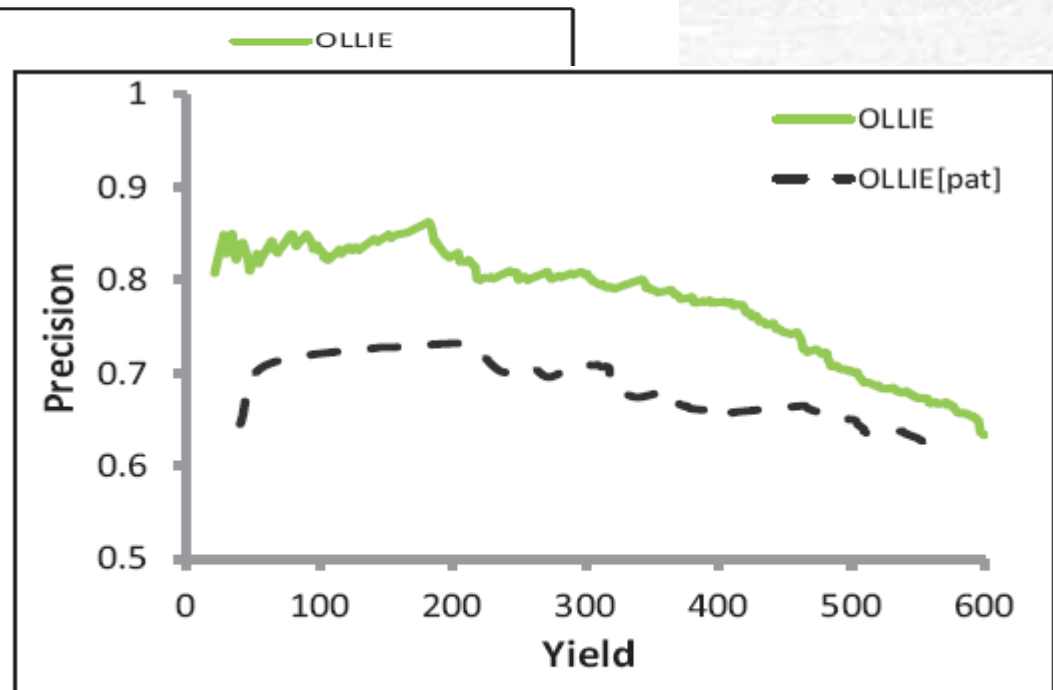


Figure 8: Context analysis increases precision, raising the area under the curve by 19%.

Errors Analysis in OLLIE

- **Parser errors** account for a large part of OLLIE's errors (32%)
- 18% of the errors are due to **aggressive generalization** of a pattern to all unseen relations.
- 12% due to incorrect application of **lexically annotated patterns**.
- 14% of the errors are due to important context **missed by OLLIE**.
- 12% of the errors are because of the limitations of binary representation, which misses important information that can only be expressed in **n-ary tuples**.

Conclusions of OLLIE

- ☞ Increase the number of correct extractions **by two orders of magnitude.**
- ☞ Identify the relation is not asserted as factual, but is **hypothetical** or **conditionally** true.


Applications of Open IE

Question answering: e.g. what kills bacteria?

TextRunner Search Results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Back Forward Reload Stop Home Go Search

 **TextRunner Search**

Retrieved **2849** results for **kills** in the predicate and **bacteria** in argument 2.
Grouping results by predicate. Group by: [argument 1](#) | [argument 2](#)

kills - 31 results

the new antibiotics (69), Benzoyl peroxide (47), Chlorine (36), [119 more...](#) **kills bacteria**

<http://turingc.cs.washington.edu:7125 - TextRunner Search Result>

- Heat (27) **kills** the beneficial **bacteria**
- Amoxicillin (26) **kills bacteria**
- ozone (24) **kills** any **bacteria**
- Penicillin (23) **kills** the pneumococcal **bacteria**
- Oxygen (16) **kills** anaerobic **bacteria**
- Honey (12) **kills** the **bacteria**
- Cooking (12) **kills** these **bacteria**
- The process (11) **kills** other **bacteria**
- Irradiation (11) **kills** most harmful **bacteria**
- Garlic (9) **kills** the bad **bacteria**
- Alcohol (9) **kills** the **bacteria**

Read turingc.cs.washington.edu

Search again:

Argument 1

Predicate

Argument 2

examples of "bacteria":

- e. coli (13)
- salmonella (12)

Jump to:

- [kills \(31\)](#)
- [will kill \(7\)](#)

Figure 4: TextRunner aggregates answers to the query "What kills bacteria?"

Applications of Open IE (cont.)

- **Opinion mining:** extract opinion information about objects that are contained in blog posts, reviews and other texts.
- **Fact checking:** identify assertions that directly or indirectly conflict with the body of knowledge extracted from the web.
- ...

Conclusion for first generation of open IE (TextRunner)

- No focus input.
- Any relations.
- Much more extractions than KnowItAll.
- Much fast than KnowItAll.

Conclusion for second generation of open IE

- a **linguistic analysis** which guides the design of three systems.
- **Reverb** focuses on identifying a more meaningful and informative relation phrase.
- **R2A2** adds an argument learning component.
- **OLLIN** solves two other problems.

Discussion

- ☞ Open IE :
 - ✓ domain independent.
 - ✓ No prior knowledge.
- ☞ Discussion:
 - ✓ What do you think about the Open IE?
 - ✓ What are the bottlenecks of open IE?

Challenges (1)

- Handle **n-ary** and even **nested** extractions.
- **Noun-based extractions** are challenging to get at high precision.
- Language dependent, general open IE will extend to other languages.
- The extraction of **temporally changing facts** (president of USA is a function of time)
- the distinction between **facts**, **opinions** and **misinformation** on the Web.

Challenges (2)

- (X, born in, 1970)
- (M, born in, 1970) → $P(X=M) \sim$ share relation ?
- (1, R, 2)
- (1, R', 2) → $P(R=R') \sim$ share argument pairs
- Truly synonymous relations are rare to find and mostly needs domain-specific type checking. E.g. “developed” could mean “invented”, “created”

Challenge (3)

- Reason based on the facts and generalizations it extracts from text.

References

- Oren etzioni et al. Open Information extraction from the Web. Communications of the ACM Dec.2008, vol 51.no.12
- Oren etzioni, et al. Open Information Etraction: the second generation. IJCAI 2011.
- Mausam,et al. Open language learning for Information extraction EMNLP2012