Open Information Extraction at Web Scale

Oren Etzioni
KnowItAll Group (2003 - ?)

- Rob Bart
- Janara Christensen
- Tony Fader
- Tom Lin
- Prof. Mausam
- Alan Ritter
- Michael Schmitz
- Dr. Stephen Soderland
- Prof. Dan Weld

- PhD alumni: Michele Banko, Prof. Michael Cafarella, Prof. Doug Downey, Ana-Maria Popescu, Stefan Schoenmackers, and Prof. Alex Yates.
“The most critical choice for a scientist is what problems to work on.”

Les Valiant (Turing Award 2011)
Knowledge Acquisition Bottleneck

1. Massive knowledge is necessary for AI
   a) Cyc? (Doug Lenat)
   b) Games? (Luis von Ahn)
   c) Volunteers? (OpenMind)

2. Knowledge acquisition has to be automatic

3. Machine Reading of the Web!
   (Etzioni et. al, AAAI ’06)
   a) 2009 DARPA MR Program
   b) NELL (Mitchell, AAAI ‘10)
   c) Watson (IBM, ‘11)
What is Machine Reading?

Text ➔ Assertions ➔ Inferences

Micro versus Macro

Etzioni, University of Washington
More Pragmatic Motivation: Information Overload

Today a person is subjected to more new information in a day than a person in the middle ages in his entire life!
Paradigm Shift: from retrieval to reading

KnowItAll

World Wide Web

Google™

Information Food Chain

How is the iPad 2?

Found 28,900 reviews; 87% positive.

Key features include…

Etzioni, University of Washington
RevMiner (Huang, Etzioni, Zettlemoyer)

- Extracts key attributes + opinions
- Applied to 400,000 Yelp reviews (Seattle)
- Based on Opine (Popescu & Etzioni ‘05)

Extractive UI versus search UI
(Yatani et. al, HCI ‘11)
RevMiner is a University of Washington KnowItAll project by Jeff Huang, Oren Etzioni, Luke Zettlemoyer

RevMiner is an unsupervised extractor for user reviews about places in Seattle, like restaurants and stores

Try these examples:
good dim sum
agua verde cafe
cheap indian food
great view clean rooms comfortable bed
great location
elliott bay book
mexican
free parking friendly staff fresh fish
Umi Sake House
Seattle

sushi
freshest (2), incredible, superb (2), amazing (16), fantastic (3), awesome (7), perfect, favorite (5), reasonable, excellent (7), best (6), delicious (16), affordable (2), fancy, worth (2), warm, inexpensive, fresh (40), traditional (7), not solid, tasty (4), huge (2), top (2), prepared, quick, real (3), korean, different (4), full, japanese (6), not bad, decent (3), expensive (4), not great (2), not good, raw, average (2), bad (3), okay (2), alright (4), ok (2), not fresh, mediocre (3), poor

place
freshest, amazing (2), superb (2), awesome (7), glad, favorite (11), perfect (5), intimate, delicious (2), gorgeous, excellent (3), best (7), yummy, classy, great (42), enormous, simple, worth (2), chic, everyday, authentic (2), fresh, impressive, fast, top (2), fun (4), tasty, modern, not loud, clean, entertaining, solid (2), fancy, sophisticated, cute (3), full (2), not small, crazy (3), different (2), packed (13),Sol, latest, central, hot, good (31), big (3), early, bigger, small (3), nice (8), japanese (3), stylish, pricey, tiny (2), open (4), long (2), popular (2), specific, close, fired (2), late (4), busy (4), cheap, not top, last, trendy (7), typical, not damn, giant, higher, not free, bad, decent, complete, closed (3), loud (7), else, noisy (2), total (2), excited, empty, overpriced, poor, rude, not nasty, not stuffy

rolls
wonderful (2), amazing (14), awesome (5), fantastic (5), perfect, delicious (12), best (3), excellent (2), creative (18), yummy (5), great (13), beautiful (2), superb (3), unique (3), not unique, huge (13), light, tasty (7), large (9), extra (4), japanese (3), good (23), different (11), interesting (3), hot (4), full (3), big (6), fried (11), small, nice (4), not better (5), not standard, standard (6), decent (2), fine, not tasty, raw (5), not good (2), average, okay, bad (6), not mediocre

food
fabulous (2), outstanding, amazing (11), wonderful, fantastic (2), awesome (4), pure, favorite (2), professional, excellent (11), affordable, delicious (7), best (7), yummy, friendly (2), fresh (7), not sweet, inexperienced, worth, generous, classic, authentic, baked, consistent, not delicious, expensive, damn, fast, top, tasty (6), spicy, fast (6), regular (2), spicy, timely, full (2), japanese (13), smaller, cheap, late (2), enough, whole (2), pricey (5), usual, memorable, typical, not hard, not familiar, standard, bad, hungry, half, par, fine (3), slow (2), disappointed, cold, average (2), bad (2), okay (3), alright, not impressed, mediocre, passable, overpriced (2), superb, worst, terrible (2)

service
incredible (2), impeccable (4), amazing (2), wonderful (2), superb, fabulous, outstanding (2), awesome (5), fantastic (5), reasonable, helpful (3), excellent (11), excellent (12), efficient (2), stellar (3), warm, speedy, attentive (8), top (2), fast (8), prompt (2), consistent, quick (8), large, solid (2), polite (2), full, above (2), good (3), best, usual, not old, standard (4), better (4), not consistent, spootty (3), decent (9), not great (3), fine, par (2), not hot, not attentive, slow (12), average, pretentious, bad, so-so, not acceptable, terrible (2), sucked, not mediocre, indifferent, poor, awful, terrible (2), horrible (7), worse (2)

happy hour
incredible, fabulous, amazing (14), awesome (11), fantastic (6), favorite (2), unbeatable, excellent (2), delicious (3), best (19), reasonable, impressed, great (43), huge (2), different, low, good (13), big, regular (4), nice (5), late (14), long (4), busy, cheap (2), last (2), not bad, better (3), decent (3), not good (2), bad (3), ridiculous, not mediocre, not satisfying, not enjoyable, not outstanding, not impressive, not tasteless, not frightening, not sinister, not disappointing, not frustrating, not horrifying, not horror (2)

atmosphere
incredible, fabulous, amazing (2), wonderful, fantastic (3), perfect (2), awesome (2), excellent, relaxed, best, romantic, lovely, unique, classy (2), comfortable (4), lively, modern (2), fun (3), contemporary, attentive (2), cute, large, cool (13), exciting, pleasant, fancy, swanky (3), good (7), different (2), low, nice (20), not small, not terrible, sucks (2)

prices
incredible, awesome, reasonable (17), not crunchy, great (20), worth, fair (4), huge (2), moderate, steeper, sweet, large, small, regular, full (4), spendy, not cheap, steep (3), high (7), higher (3), hard, typical (2), sticky, due, better (2), not bad (4), lower (2), standard, decent (5), expensive, not sticky, average, acceptable, dry, overcooked (2), wet

fish
freshest (3), amazing, awesome, delicious (3), best (2), excellent, reasonable, flavorful, super (2), great (5), fresh (44), generous, prepared, huge, ideal, top, tasty, high (2), better, not fishy (2), not nice, fine, fishy, average, bad, not fresh (4), mushy

server
knowledgeable (2), wonderful, amazing (2), best (2), adorable, excellent, helpful (4), friendly (11), patient (2), super (3), gracious (2), great (4), efficient, pleasant, quick (3), different, nice (8), good (6), busy (3), long (3), whole, not nice, not bad (2), not knowledgeable, fine, slow (2), not friendly, not helpful, terrible (2), unreliable, not local

menu
incredible, amazing (3), fantastic, massive, delicious, creative (2), unique, super, great (2), inventive, diverse (2), enormous, wide, impressive, fresh (3), tasty, huge, large (3), robust, able, nice (2), full, big, regular (8), good (3), crazy, interesting (2), small, long, normal (3), not traditional, typical, special, standard (3), decent, fine, delicious (2), good (2), short, small (2), dark, whole, open, hard, not free, better (2), front (2), not packed, loud (2), fine, average, empty (3), not busy

restaurant
incredible, amazing, favorite (2), best, classy, romantic, great (3), airy, beautiful, worth, authentic, top, spacious, clean, tasty, huge (2), large (3), larger, pleasant, good (2), short, small (2), dark, whole, open, hard, not free, better (2), front (2), not packed, loud (2), fine, average, empty (3), not busy
Outline

I. Twin Motivations for Information Extraction (IE)
   1) Knowledge acquisition bottleneck
   2) New paradigm for search (Extractive UI)

II. Machine Reading = IE + inference
   1) Overview of IE
   2) Open IE
   3) Demo of Open IE
   4) Inference over extractions

III. Lessons and Future Work
1. Information Extraction (IE)

$IE(\text{sentence}) = \text{Relation instance, probability}$

“Edison was the inventor of the light bulb.”

invented(Edison, light bulb), 0.9

“You shall know a word by the company it keeps” (Firth, 1957)
Context ➔ clues

• ...Barcelona mayor...
• ...Downtown Barcelona...
• Spanish cities such as Madrid, Barcelona, and..

Where do clues come from?
How to Scale IE?

1970s-1980s: heuristic, hand-crafted clues
• Facts from earnings announcements
• **Narrow** genres; **brittle** clues

1990s: IE as supervised learning
“Mary was named to the post of CFO, succeeding Joe who retired abruptly.”
Learned Extraction Clues

“Mary was named to the post of CFO, succeeding Joe who retired abruptly.”

- <New> was named to
- Post of <post>

Does “IE as supervised learning” scale to reading the Web?
No.
Critique of IE=supervised learning

• Relation specific
• Genre specific
• Hand-craft clues ➔
• Hand-craft training examples

Does not scale to the Web!
Semi-Supervised Learning

• Few hand-labeled examples per relation!
• \( \rightarrow \) Limit on the number of relations
• \( \rightarrow \) relations are pre-specified

• \( \rightarrow \) Still does not scale to the Web
Outline

I. Twin Motivations
   1) Knowledge acquisition bottleneck
   2) New paradigm for search

II. Machine Reading
   1) Overview of Information Extraction (IE)
   2) Open IE
   3) Demo

III. Inference over Extractions

IV. Lessons and Future Work
2. Open IE (Banko, IJCAI ’07; ACL ’08)

• Avoid hand-labeling sentences

• Single pass over corpus

• No pre-specified vocabulary (cf. Sekine ’06)
  – Challenge: map relation *phrase* to canonical relation
  – E.g., “was the inventor of” ➔ invented
Open versus Traditional IE

**Input:**
- Traditional IE: Corpus + Hand-labeled Data
- Open IE: Corpus + Existing resources

**Relations:**
- Traditional IE: Specified in Advance
- Open IE: Discovered Automatically

**Complexity:**
- Traditional IE: $O(D \times R)$
- Open IE: $O(D)$

**Output:**
- Traditional IE: relation-specific
- Open IE: Relation-independent
TextRunner

First Web-scale, Open IE system (Banko, IJCAI ‘07)

1,000,000,000 distinct extractions

Peak of 0.9 precision (but low recall)
Relation Extraction in TextRunner

“Tim Berners-Lee is credited with having invented the WWW”

• Which words denote the relation?
• Mechanism: learn via linear CRF
TextRunner Architecture

Distant supervision ➞ 180,000 training examples

Unlexicalized model of relations

Count tuples; identify synonyms

Index in Lucene; relational queries
Two Types of Extraction Errors

“Al Gore invented the Internet.”
Invented(Al Gore, Internet)

Sound extraction of incorrect fact.

“The cost of the war against Iraq has risen above 500 billion dollars”
above(Iraq, 500 billion dollars)

Unsound extraction.
How to Filter Unsound Extractions?

Leverage redundancy:

• More **distinct** clues $\rightarrow$ more confidence

• Higher **proportion** of clues $\rightarrow$ confidence
  – proportion = clues/mentions

Caveat: count over **independent** sentences!
Combinatorial Model (Downey, IJCAI ’05, AIJ ‘10)

If an extraction \( x \) appears \( k \) times in a set of \( n \) distinct sentences matching a clue, what is the probability that \( x \in \text{class } C \)?

\[
P(x \in C | x \text{ appears } k \text{ times in } n \text{ draws}) = \frac{\sum_{r \in \text{num}(C)} \left( \frac{r}{s} \right)^k \left( 1 - \frac{r}{s} \right)^{n-k}}{\sum_{r' \in \text{num}(C \cup E)} \left( \frac{r'}{s} \right)^k \left( 1 - \frac{r'}{s} \right)^{n-k}}
\]

15x more accurate than previous work.
Key Ideas in TextRunner

- Open IE on the Web is possible!
- Identified tractable subset of English
- Used “macro reading” to filter errors
Error Analysis of TextRunner Relations

Incoherent relations: 13% of the time

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Incoherent Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The guide <em>contains</em> dead links and <em>omits</em> sites.</td>
<td>contains omits</td>
</tr>
<tr>
<td>The Mark 14 <em>was central</em> to the <em>torpedo</em> scandal of the fleet.</td>
<td>was central torpedo</td>
</tr>
<tr>
<td>They <em>recalled</em> that Nungesser <em>began</em> his career as a precinct leader.</td>
<td>recalled began</td>
</tr>
</tbody>
</table>

Uninformative relations: 7% of the time

| 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 |
ReVerb  (Fader, EMNLP ’11; Etzioni et al., IJCAI ‘11)

Identify Relations from Verbs.

1. Find longest phrase matching a simple syntactic constraint:

\[
V | VP | VW^*P
\]

\[
V = \text{verb particle? adv?}
\]

\[
W = (\text{noun | adj | adv | pron | det})
\]

\[
P = (\text{prep | particle | inf. marker})
\]
ReVerb Refinement

Overly-specific Relation phrase: “is offering only modest greenhouse gas reductions at”

2. Constraint: $|\text{args(Relation)}| > k$

ReVerb $\approx$ two simple constraints!
Sample of ReVerb Relations

- inhibits tumor growth in
- is a person who studies
- has a maximum speed of
- gained fame as
- was the first person to
- has a PhD in
- voted in favor of
- died from complications of
- granted political asylum to
- identified the cause of
- joined forces with
- won an Oscar for
- mastered the art of
- is the patron saint of
- wrote the book on

Etzioni, University of Washington
## Number of Relations

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yago</td>
<td>92</td>
</tr>
<tr>
<td>NELL</td>
<td>~500</td>
</tr>
<tr>
<td>DBpedia 3.2</td>
<td>940</td>
</tr>
<tr>
<td>PropBank</td>
<td>3,600</td>
</tr>
<tr>
<td>VerbNet</td>
<td>5,000</td>
</tr>
<tr>
<td>Wikipedia InfoBoxes, $f &gt; 10$</td>
<td>~5,000</td>
</tr>
<tr>
<td>TextRunner</td>
<td>100,000+</td>
</tr>
<tr>
<td>ReVerb</td>
<td>1,500,000+</td>
</tr>
</tbody>
</table>

Etzioni, University of Washington
ReVerb versus TextRunner

Surprise: “overlearning”
3. **Demo**

- Note: open source ReVerb extractor + sample of data publically available at
  [reverb.cs.washington.edu](http://reverb.cs.washington.edu)
Retrieved 363 results for Predicate containing "kills" and Argument 2 containing "bacteria"
Retrieved 363 results for Predicate containing "kills" and Argument 2 containing "bacteria"

killing (211 results)

- antibiotics (67), Antibiotics (33), Chlorine (31), 162 more... kills bacteria
- UV-lights (3), antibiotics (5), chlorine (4), 14 more... kills most bacteria
- UV technology (3), Cooking food (2), Iodine (2), 2 more... kills bacteria and viruses
- Antibiotics (5) can kill both beneficial and harmful bacteria
- Antibiotics (2), Antibiotics (2) kill the gonorrhea bacteria
- Benzoyl peroxide (4) kills the acne-causing bacteria
- Low-level disinfection (5), UV-C light (2) kills some viruses and bacteria
- Antibiotics (2) kill chlamydia bacteria
- Home-care technique (2) kills deep gum disease bacteria
- Antibiotics (3) kill good gut bacteria
- Chlorine (2) kills iron bacteria
- Powerful and effective sanitizer (2) kills algae and bacteria
- Antibiotics (3) kill ALL bacteria
- Antibiotics (2) kill disease bacteria
- Heat (2) kills food poisoning bacteria
- Pasteurization (2) kills harmful levels of bacteria
- Benzoyl peroxide (2) kills the p-acnes bacteria
- Freezing (3) kills all parasites and bacteria
- Chlorination (2) kills many pathogenic bacteria
- High temperatures (2) kill Salmonella bacteria
- Active ingredient Triclosan (2) kills a broad spectrum of bacteria and yeasts
- Razor Rinse (3) instantly kills harmful staph bacteria
Have we made progress towards Machine Reading?
III. Extractions as basis for Inference

1,000,000,000 Extractions

Identify synonyms
(Yates & Etzioni, JAIR ‘09)

1st order Horn-clause inference
(Schoenmackers, EMNLP ‘08)

Learn argument types
(Ritter, ACL ‘10)

Transitive Inference
(Berant, ACL ‘11)
Synonyms (Mars = Red Planet)

Resolver (Yates & Etzioni, HLT ’07, JAIR ‘09): determines synonymy based on relations found by TextRunner

- born in(X, 1961) born in(Y, 1961)
- citizen(X, US) citizen(Y, US)
- Married to(X, Obama) Married to(Y, Obama)

\[ P(X = Y) \sim \text{shared relations} \]

\[ P(R1 = R2) \sim \text{shared argument pairs} \]
Argument Typing

Example: \( P \) was born in \( Y \)

\( -P \) is a person
\( -Y \) is location or date
Text $\rightarrow$ Argument Types
(Ritter et. al, ACL ‘10)

• Previous work (Resnick, Pantel, etc.)
• Utilize generative topic models
• Topics $\rightarrow$ Terms $\rightarrow$ document

relation + args = “document”
born_in(Sergey Brin, Moscow)
headquartered_in(Microsoft, Redmond)

born_in(Bill Gates, Seattle)

born_in(Einstein, March)
founded_in(Google, 1998)

headquartered_in(Google, Mountain View)
born_in(Sergey Brin, 1973)
founded_in(Microsoft, Albuquerque)
born_in(Einstein, Ulm)
founded_in(Microsoft, 1973)
Pick a topic for \( \text{arg2} \)

For each relation, randomly pick a distribution over types

\[
\begin{align*}
X &\, \text{born_in} \, Y \\
P(\text{Topic1}|\text{born_in}) &= 0.5 \\
P(\text{Topic2}|\text{born_in}) &= 0.3 \\
\ldots
\end{align*}
\]

\textbf{Person born_in Location}

Sergey Brin \, \textit{born_in} \, Moscow

Two separate sets of type distributions

For each extraction, pick type for \( a_1, a_2 \)

Then pick arguments based on types

LinkLDA

[Erosheva et. al. 2004]
Demo of LDA-SP
(data publically available)

Argument types for relations
IV. Open IE Lessons

• Open IE is simple and highly scalable (download at reverb.cs.washington.edu)

• Open IE is basis for “extractive interfaces”

• Open IE is basis for inference!
Conclusions/Speculations

Machine Reading = platform for NLP and AI (VLSAI)

Keyword Search ➔ Question answering

Machine Reading ≠ human reading (Remember Computer Chess!)
<table>
<thead>
<tr>
<th>Percentage</th>
<th>Phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td>Satisfy Constraints</td>
</tr>
<tr>
<td>8%</td>
<td>Non-Contiguous Phrase Structure</td>
</tr>
<tr>
<td></td>
<td>Coordination: X is produced and maintained by Y</td>
</tr>
<tr>
<td></td>
<td>Multiple Args: X was founded in 1995 by Y</td>
</tr>
<tr>
<td></td>
<td>Phrasal Verbs: X turned Y off</td>
</tr>
<tr>
<td>4%</td>
<td>Relation Phrase Not Between Arguments</td>
</tr>
<tr>
<td></td>
<td>Intro. Phrases: Discovered by Y, X …</td>
</tr>
<tr>
<td></td>
<td>Relative Clauses: … the Y that X discovered</td>
</tr>
<tr>
<td>3%</td>
<td>Do Not Match POS Pattern</td>
</tr>
<tr>
<td></td>
<td>Interrupting Modifiers: X has a lot of faith in Y</td>
</tr>
<tr>
<td></td>
<td>Infinitives: X to attack Y</td>
</tr>
</tbody>
</table>
Locating Arguments for Relations

ReVerb, TextRunner: arguments are the two nearest NPs.

“The cost of the war against Iraq has risen above 500 billion dollars”

(Iraq, has risen above, 500 billion dollars)
ArgLearner (*Etzioni et al.*, *Ijcai* ‘11)

- Learn independent extractors for left and right boundaries of each arg.

\[ R2A2 = \text{ReVerb} + \text{ArgLearner} \]
Combinatorial Model  (Yates, JAIR ‘09)

**Theorem:** If two strings, $s_i$ and $s_j$, have $P_i$ and $P_j$ potential properties, and they appear in extracted assertions $D_i$ and $D_j$ such that $|D_i| = n_i$ and $|D_j| = n_j$, and they share $k$ extracted properties, the probability that $s_i$ and $s_j$ co-refer is:

$$P(R_{i,j}^t | D_i, D_j, P_i, P_j) = \frac{P(k | n_i, n_j, P_i, P_j, S_{i,j} = \min(P_i, P_j))}{\min(P_i, P_j)}$$

$$= \sum_{S_{i,j} = k} P(k | n_i, n_j, P_i, P_j, S_{i,j})$$

where:

$$P(k | n_i, n_j, P_i, P_j, S_{i,j}) = \binom{S_{i,j}}{k} \sum_{r,s \geq 0} \binom{S_{i,j} - k}{r + s} \binom{r + s}{r} \binom{P_i - S_{i,j}}{n_i - (k + r)} \frac{P_i - S_{i,j}}{n_i - (k + s)} \frac{P_j}{n_j}$$