Information Extraction

Brooke Cowan

October 29, 2012

Table of contents

Outline

Introduction to Information Extraction

Named Entity Recognition

Relation Extraction
- NELL: Never-Ending Language Learner
- Open IE and TextRunner

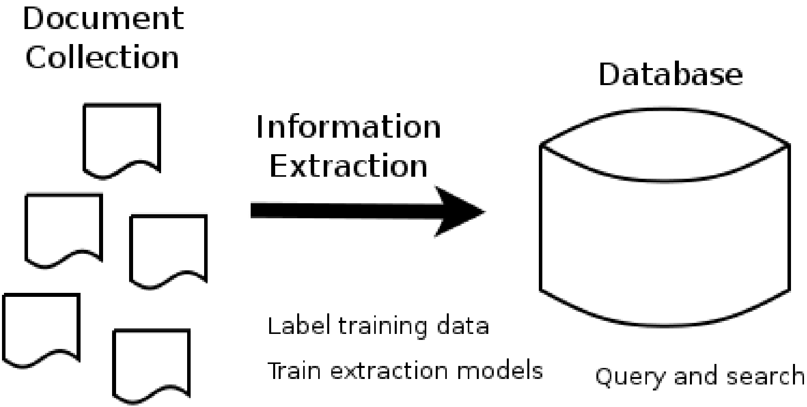
Event Extraction
- Event Extraction as Dependency Parsing
- Unsupervised Learning of Narrative Event Chains

Discussion and Questions

What is Information Extraction?

- **Goal:**
  Extract structured information from unstructured (or loosely formatted) text
- Typical description of task:
  - Identify named entities
  - Identify relations between entities
  - Populate a database
- May also include:
  - Event extraction
  - Resolution of temporal expressions
  - Wrapper induction (automatic construction of templates)
- Applications: natural language understanding, question-answering, summarization, etc.

What is Information Extraction?



Label training data

Train extraction models

Query and search

Data mine

Create ontology

Crawl

Filter by relevance

Example

Target relation: HeadquarteredIn(⟨company⟩, ⟨city⟩)

Headquartered in Portland, OR, AFMS was founded by Mike Erickson in 1992.	→	HeadquarteredIn(AFMS, Portland)
The files, kept at Boy Scout headquarters in Irving, Texas, consist of....	→	HeadquarteredIn(Boy Scout, Irving)
Yes to both, at the Nike HQ in Beaverton.	→	HeadquarteredIn(Nike, Beaverton)
AAPH is moving our headquarters and member fulfillment services to Portland, Oregon!	→	HeadquarteredIn(AAPH, Portland)

Outline

Introduction to Information Extraction

Named Entity Recognition

Relation Extraction

- NELL: Never-Ending Language Learner
- Open IE and TextRunner

Event Extraction

- Event Extraction as Dependency Parsing
- Unsupervised Learning of Narrative Event Chains

Discussion and Questions

IE Subtask: Named Entity Recognition

- Detect and classify all proper names mentioned in text
- What is a proper name? Depends on application.
  - People, places, organizations, times, amounts, etc.
  - Names of genes and proteins (Settles 05)
  - Names of college courses (McCallum 05)

NER Examples

Celebrate the [$_{NAME}$Chinese Moon Festival] at [$_{LOC}$100th Monkey Studios] on [$_{DATE}$Friday] from [$_{TIME}$10 to 12]. Paint with washable ink and brushes, make a votive candle holder, and take home a recipe for a mooncake.

Please join us at [$_{ORG}$The Peanut Gallery Playschool] for our open house on [$_{DATE}$February 26th], from [$_{TIME}$10am-12pm]. Located in [$_{LOC}$NE Portland], [$_{ORG}$The Peanut Gallery] is a unique certified in-home program accepting toddlers and young preschoolers age [$_{AGE}$18 months] through [$_{AGE}$3.5 years] old.

- Find extent of each mention
- Classify each mention
- Sources of ambiguity
    - Different strings that map to the same entity
    - Equivalent strings that map to different entities (e.g., `U.S. Grant`)

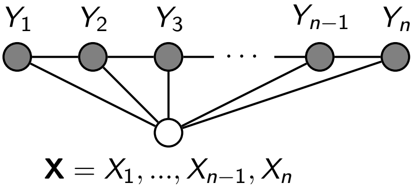
Approaches to NER

- Early systems: hand-written rules
- Statistical systems
  - Supervised learning (HMMs, Decision Trees, MaxEnt, SVMs, CRFs)
  - Semi-supervised learning (bootstrapping)
  - Unsupervised learning (rely on lexical resources, lexical patterns, and corpus statistics)

A Sequence-Labeling Approach using CRFs

- **Input:** Sequence of observations (tokens/words/text)
- **Output:** Sequence of states (labels/classes)
    - B: Begin
    - I: Inside
    - O: Outside
    - Some evidence that including L (Last) and U (Unit length) is advantageous (Ratinov and Roth 09)
- CRFs define a conditional probability $p(\mathbf{Y}|\mathbf{X})$ over label sequences $\mathbf{Y}$ given an observation sequence $\mathbf{X}$
    - No effort wasted modeling the observations (in contrast to joint models like HMMs)
    - Arbitrary features of the observations may be captured by the model

Linear Chain CRFs



- Simplest and most common graph structure, used for sequence modeling
- Inference can be done efficiently using dynamic programming $O(|\mathbf{X}||\mathbf{Y}|^2)$

Linear Chain CRFs

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})$$

- $\mathbf{Y}$ is a sequence of labels (e.g., `ORG`, `PER`, etc)
- $\mathbf{X}$ is a sequence of observed tokens
- f_k is a feature function mapping y_t, y_{t-1}, and x_t to $\mathbb{R}$
- λ_k is a weight associated with feature function f_k
- $Z(x)$ is a normalization factor

NER Features

Several feature families used, all time-shifted by -2, -1, 0, 1, 2:

- the word itself
- capitalization and digit patterns (shape patterns)
- 8 lexicons entered by hand (e.g., honorifics, days, months)
- 15 lexicons obtained from web sites (e.g., countries, publicly-traded companies, surnames, stopwords, universities)
- 25 lexicons automatically induced from the web (people names, organizations, NGOs, nationalities)

McCallum and Li, CONLL 2003

NER Performance

	Prec	Recall	F1
LOC	87.23	87.65	87.44
MISC	74.44	71.37	72.87
ORG	79.52	78.33	78.92
PER	91.05	89.98	90.51
overall	84.52	83.55	84.04

- CoNLL-2003 English shared task (news articles)
- Stanford CRF NER, same data: 87.94–92.99 F1
- Ratinov and Roth 2009, same data: 90.57 F1

McCallum and Li, CONLL 2003

Limitations of Conventional NER (and IE)

- Supervised learning
  - Expensive
  - Inconsistent: *...the data included sentences such as "Hear, O Israel, the Lord our God, the Lord is one." We could not agree on whether "O Israel" should be labeled as ORG, LOC, or PER. Similarly, we could not agree on whether "God" and "Lord" is an ORG or PER.* (Ratinov and Roth 09)
  - Worse for relations and events!
- Fixed, narrow, pre-specified sets of entity types
- Small, homogeneous corpora (newswire, seminar announcements)

Outline

Introduction to Information Extraction

Named Entity Recognition

Relation Extraction

- NELL: Never-Ending Language Learner
- Open IE and TextRunner

Event Extraction

- Event Extraction as Dependency Parsing
- Unsupervised Learning of Narrative Event Chains

Discussion and Questions

What is Relation Extraction?

- Typically defined as identifying relations between two entities

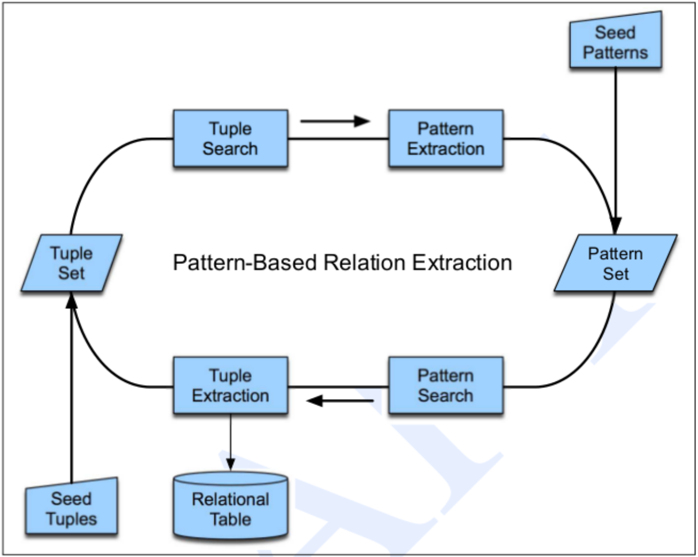
Relations	Subtypes	Examples
Affiliations		
	Personal	married to, mother of
	Organizational	spokesman for, president of
	Artifactual	owns, invented, produces
Geospatial		
	Proximity	near, on outskirts
	Directional	southeast of
Part-of		
	Organizational	a unit of, parent of
	Political	annexed, acquired

Typical (Supervised) Approach

```
function FindRelations(words) returns relations
    relations ← nil
    entities ← FindEntities(words)
    forall entity pairs ⟨e1, e2⟩ in entities do
        if Related?(e1,e2)
            relations ← relations + ClassifyRelation(e1,e2)
```

- FindEntities(): Named entity recognizer
- Related?(): Binary classifier that says whether two entities are involved in a relation
- ClassifyRelation(): Classifier that labels relations discovered by Related?()

Typical (Semi-Supervised) Approach



NELL: Can computers learn to read?

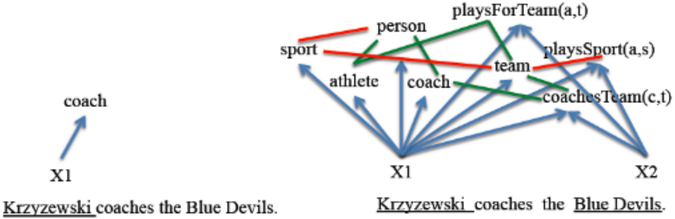
- **Goal**: create a system that learns to read the web
  - **Reading task**: Extract facts from text found on the web
  - **Learning task**: Iteratively improve reading competence.
- Running since Jan 2010
- To date, 15M candidate beliefs – 9% high confidence
- `http://rtw.ml.cmu.edu/rtw/`

Approach

- **Inputs**
  - Ontology with target categories and relations (i.e., *predicates*)
  - Small number of seed examples for each
  - Set of constraints that couple the predicates
  - Large corpus of unlabeled documents
- **Output**: new predicate instances
- Semi-supervised bootstrap learning methods
- Couple the learning of functions to constrain the problem
- Exploit redundancy of information on the web.

Carlson et al., WSDM 2010, AAAI 2010

Coupled Semi-Supervised Learning



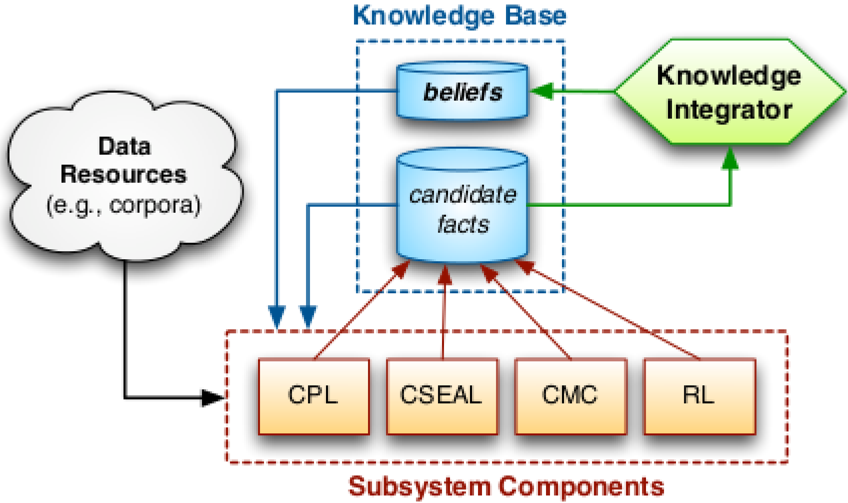
(A) A difficult semi-supervised learning problem

(B) An easier semi-supervised learning problem

Types of Coupling

1. Mutual Exclusion (output constraint)
    - Mutually exclusive predicates can't both be satisfied by the same input x
    - E.g., x cannot be a `Person` and a `Sport`
2. Relation Argument Type-Checking (compositional constraint)
    - Arguments of relations declared to be of certain categories
    - E.g., `CompanyIsInEconomicSector(Company, EconomicSector)`
3. Unstructured and Semi-Structured Text Features (multi-view-agreement constraint)
    - Look at different views (like co-training)
    - Require classifiers agree
    - E.g., freeform textual contexts and semi-structured contexts

System Architecture



Coupled Pattern Learner (CPL)

Predicate	Pattern
emotion	hearts full of X
beverage	cup of aromatic X
newspaper	op-ed page of X
teamPlaysInLeague	X ranks second in Y
bookAuthor	Y classic X

- Free-text extractor that learns contextual patterns to extract predicate instances
- Use mutual exclusion and type-checking constraints to filter candidates instances
- Rank instances and patterns by leveraging redundancy: if an instance or pattern occurs more frequently, it's ranked higher

Coupled SEAL (CSEAL)

- SEAL (Set Expander for Any Language) is a wrapper induction algorithm
- Operates over semi-structured text such as web pages
- Constructs page-specific extraction rules (*wrappers*) that are human- and markup-language independent
- CSEAL adds mutual-exclusion and type-checking constraints

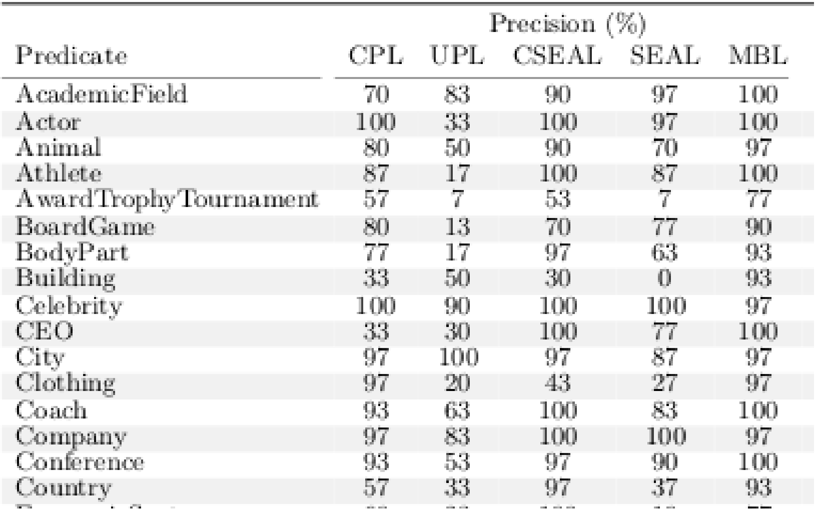
Wang and Cohen, Language-independent set expansion of named entities using the web, ICDM 07

CSEAL Wrappers

URL:	http://www.shopcarparts.com/
Wrapper:	.html” CLASS=”shopcp”>*arg1* Parts</A>
Content:	acura, audi, bmw, buick, cadillac, chevrolet, chevy, ...
URL:	http://www.allautoreviews.com/
Wrapper:	</a></br> <a href=”auto_reviews/*arg1*/
Content:	acura, audi, bmw, buick, cadillac, chevrolet, chrysler, ...
URL:	http://www.hertrichs.com/
Wrapper:	<li class=”franchise *arg1*” <h4><a href=”#”>
Content:	buick, chevrolet, chrysler, dodge, ford, gmc, isuzu, ...

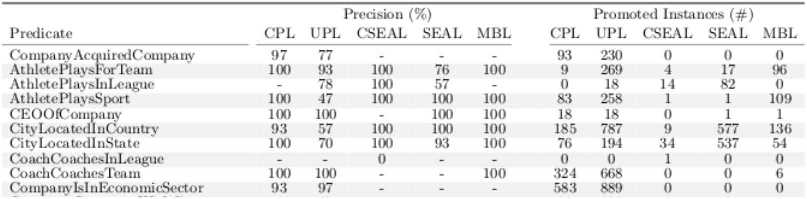
- Seeds: Ford, Nissan, Toyota
- *arg1* is a placeholder for extracting instances

Experimental Results



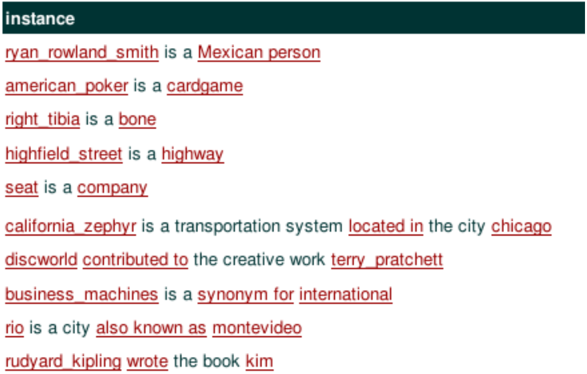
	Precision (%)				
Predicate	CPL	UPL	CSEAL	SEAL	MBL
AcademicField	70	83	90	97	100
Actor	100	33	100	97	100
Animal	80	50	90	70	97
Athlete	87	17	100	87	100
AwardTrophyTournament	57	7	53	7	77
BoardGame	80	13	70	77	90
BodyPart	77	17	97	63	93
Building	33	50	30	0	93
Celebrity	100	90	100	100	97
CEO	33	30	100	77	100
City	97	100	97	87	97
Clothing	97	20	43	27	97
Coach	93	63	100	83	100
Company	97	83	100	100	97
Conference	93	53	97	90	100
Country	57	33	97	37	93

Experimental Results



	Precision (%)					Promoted Instances (#)				
Predicate	CPL	UPL	CSEAL	SEAL	MBL	CPL	UPL	CSEAL	SEAL	MBL
CompanyAcquiredCompany	97	77	-	-	-	93	230	0	0	0
AthletePlaysForTeam	100	93	100	76	100	9	269	4	17	96
AthletePlaysInLeague	-	78	100	57	-	0	18	14	82	0
AthletePlaysSport	100	47	100	100	100	83	258	1	1	109
CEOOfCompany	100	100	-	100	100	18	18	0	1	1
CityLocatedInCountry	93	57	100	100	100	185	787	9	577	136
CityLocatedInState	100	70	100	93	100	76	194	34	537	54
CoachCoachesInLeague	-	-	0	-	-	0	0	1	0	0
CoachCoachesTeam	100	100	-	-	100	324	668	0	0	6
CompanyIsInEconomicSector	93	97	-	-	-	583	889	0	0	0

Recently-learned facts



instance
ryan_rowland_smith is a Mexican person
american_poker is a cardgame
right_tibia is a bone
highfield_street is a highway
seat is a company
california_zephyr is a transportation system located in the city chicago
discworld contributed to the creative work terry_pratchett
business_machines is a synonym for international
rio is a city also known as montevideo
rudyard_kipling wrote the book kim

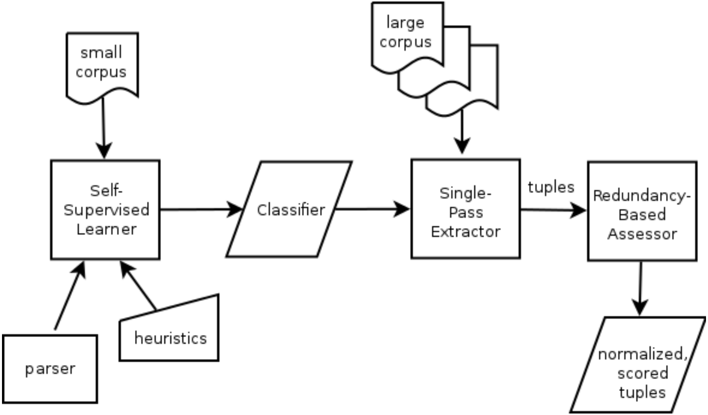
- Learned in Sept 2012
- Confidence >92
- How many of these facts are correct?

Open IE and TextRunner

- Motivations:
  - Web corpora are massive, introducing scalability concerns
  - Relations of interest are unanticipated, diverse, and abundant
  - Use of "heavy" linguistic technology (NERs and parsers) don't work well
- **Input:** a large, heterogeneous Web corpus
  - 9M web pages, 133M sentences
  - No pre-specified set of relations
- **Output:** huge set of extracted relations
  - 60.5M tuples, 11.3M high-probability tuples
  - Tuples are indexed for searching

Banko et al, IJCAI 07

TextRunner Architecture



- **Learner** outputs a classifier that labels trustworthy extractions
- **Extractor** finds and outputs trustworthy extractions
- **Assessor** normalizes and scores the extractions

Architecture: Self-Supervised Learner

1. **Automatically labels training data**
   - Uses a parser to induce dependency structures
   - Parses a small corpus of several thousand sentences
   - Identifies and labels a set of positive and negative extractions using relation-independent heuristics
     - An extraction is a tuple $t = (e_i, r_{i,j}, e_j)$
     - Entities are base noun phrases
     - Uses parse to identify potential relations
2. **Trains a classifier**
   - Domain-independent, simple non-parse features
   - E.g., POS tags, phrase chunks, regexes, stopwords, etc.

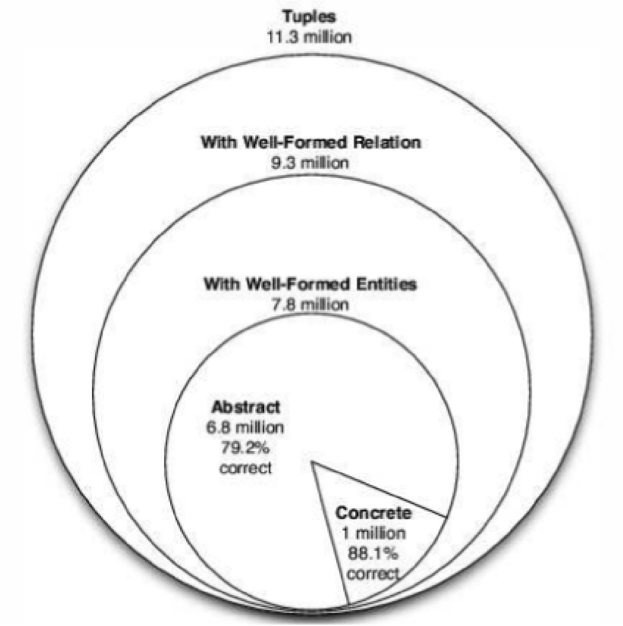
Architecture: Single-Pass Extractor

1. POS tag each word
2. Identify entities using lightweight NP chunker
3. Identify relations
4. Classify them

Architecture: Redundancy-Based Assessor

- Take the tuples and perform
  - Normalization, deduplication, synonym resolution
  - Assessment
- Number of distinct sentences from which each extraction was found serves as a measure of confidence
- Entities and relations indexed using Lucene

Estimating Correctness



Outline

Introduction to Information Extraction

Named Entity Recognition

Relation Extraction

- NELL: Never-Ending Language Learner
- Open IE and TextRunner

Event Extraction

- Event Extraction as Dependency Parsing
- Unsupervised Learning of Narrative Event Chains

Discussion and Questions

Problem Definition

- Corresponds most closely to *template filling* in J&M
- Fixed set of event types; each type has a set of attributes
  - Types: `attack`, `killing`, `injury`, `election`, `protest`, ...
  - Attributes: `location`, `time`, `attacker`, `victim`, `speaker`, ...
  - Example:

attack:	`instrument:`	`car bomb`
	`attacker:`	*NULL*
	`target:`	`Shiite shrine`
	`time:`	`9am today`
	`location:`	`Baghdad`
	`fatalities:`	`95 people`

Example

A powerful car bomb exploded today in Baghdad inside the holiest Shiite shrine. As many as 95 people were killed in the event, according to sources in Washington. The blast came only two days after another car bomb exploded in a crowded street in Mosul in the northern part of Iraq, killing 13 pedestrians, in an attack carried out by Al Qaeda. Together with the previous attack by Al Qaeda, the shooting in Najaf three weeks ago that killed 15 American soldiers, violence seemed to spike to its highest level. The bombing today happened around 9am, when the roads are crowded with people...

Question: How many distinct (unique) events are there?

Events Anchors/Triggers

A powerful car bomb exploded today in Baghdad inside the holiest Shiite shrine. As many as 95 people were killed in the event, according to sources in Washington. The blast came only two days after another car bomb exploded in a crowded street in Mosul in the northern part of Iraq, killing 13 pedestrians, in an attack carried out by Al Qaeda. Together with the previous attack by Al Qaeda, the shooting in Najaf three weeks ago that killed 15 American soldiers, violence seemed to spike to its highest level. The bombing today happened around 9am, when the roads are crowded with people...

Events Fields/Attributes

A powerful car bomb exploded today in Baghdad inside the holiest Shiite shrine. As many as 95 people were killed in the event, according to sources in Washington. The blast came only two days after another car bomb exploded in a crowded street in Mosul in the northern part of Iraq, killing 13 pedestrians, in an attack carried out by Al Qaeda. Together with the previous attack by Al Qaeda, the shooting in Najaf three weeks ago that killed 15 American soldiers, violence seemed to spike to its highest level. The bombing today happened around 9am, when the roads are crowded with people...

Desired Output

attack :	instrument:	car bomb
	attacker:	*NULL*
	target:	Shiite shrine
	time:	9am today
	location:	Baghdad
	fatalities:	95 people

attack :	instrument:	car bomb
	attacker:	Al Qaeda
	target:	*NULL*
	time:	two days ago
	location:	Mosul, Iraq
	fatalities:	13 pedestrians

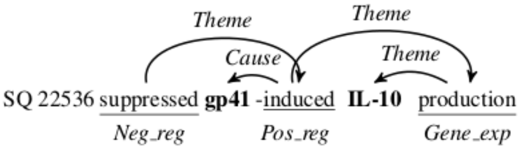
attack :	instrument:	shooting
	attacker:	Al Qaeda
	target:	Shiite shrine
	time:	three weeks ago
	location:	Najaf
	fatalities:	15 American soldiers

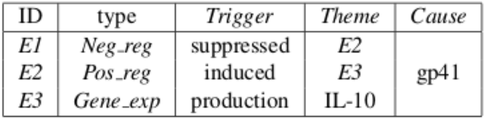
Recent Activity

- Government programs and shared tasks
  - DARPA Machine Reading program (earlier: MUC and ACE)
  - BioNLP 2009 and 2011
- A lot of interesting work including
  - Unsupervised Learning of Narrative Schemas and their Participants (Chambers and Jurafsky, ACL 2009)
  - Joint Inference for Knowledge Extraction from Biomedical Literature (Poon and Vanderwende, NAACL 2010)
  - Event Extraction as Dependency Parsing (McClosky et al., ACL 2011)
  - Fast and Robust Joint Models for Biomedical Event Extraction (Reidel and McCallum, EMNLP 2011)
  - Multi Event Extraction Guided by Global Constraints (Reichart and Barzilay, NAACL 2012)

A Different Example

"SQ 22536 suppressed **gp41**-induced **IL-10** production in monocytes."





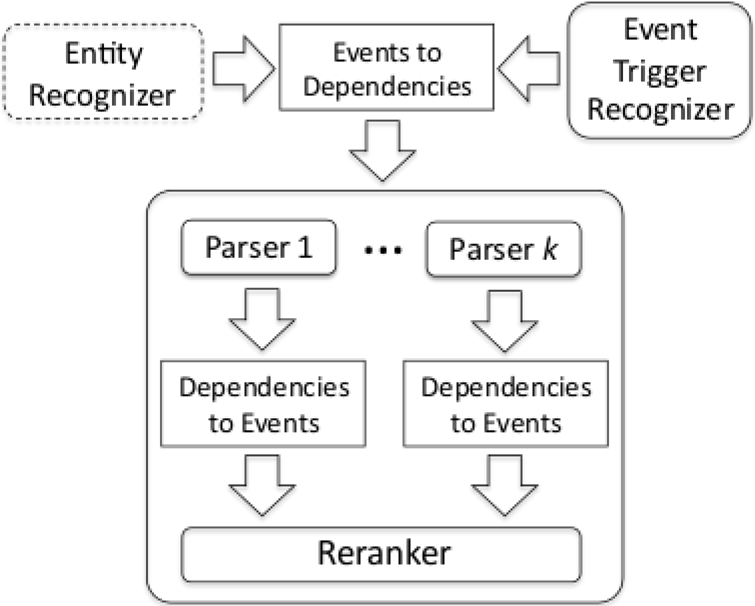
ID	type	*Trigger*	*Theme*	*Cause*
E1	*Neg_reg*	suppressed	*E2*	gp41
E2	*Pos_reg*	induced	*E3*	
E3	*Gene_exp*	production	IL-10	

Event Extraction as Dependency Parsing

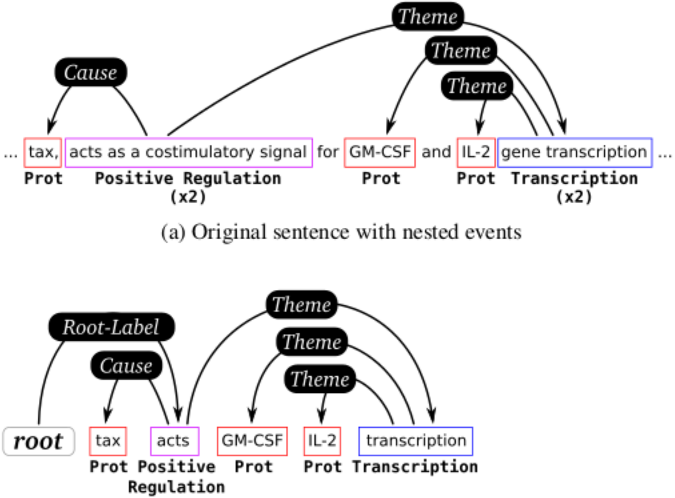
- Main ideas:
    - Train dependency parser(s) to recognize event structures
    - Handle nested events
    - Use reranker to incorporate arbitrary global features
    - Largely domain-independent approach
    - Evaluated on biomedical domain (BioNLP'09)
    - History of similar approaches, starting with Miller et al. 1997

McClosky et al., ACL 11

System Architecture



Converting Events to Dependency Structures



(a) Original sentence with nested events

(b) After conversion to event dependencies

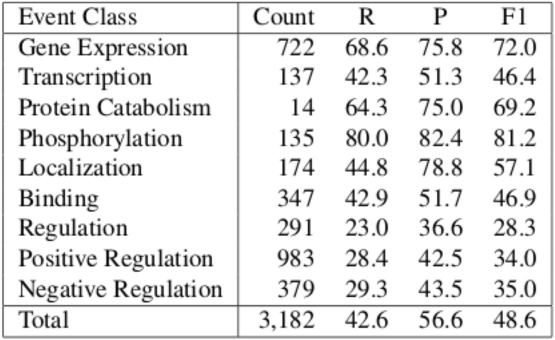
Three Step Process

1. Predict event anchors
    - Multiclass classifier that labels each anchor independently
    - Standard set of NLP features (form, lemma, gazetteer, contextual, dependency paths, counts)
2. Parse event structures
    - Use MST parser to find maximum weight spanning tree
    - Each graph edge is a possible dependency and has a score
    - Features derived from
        - words and syntax of original sentence
        - constraints based on domain knowledge
        - ontology used to capture generalizations like *simple event anchors can't take other events as arguments*
3. Rerank
    - Dependencies converted back to event structures first
    - Features derived from
        - source: score, rank, number of different parsers produced by
        - the event frames: paths, the structure itself
        - constraints based on domain knowledge: e.g., incorrect number or types of arguments

Experimental Setup

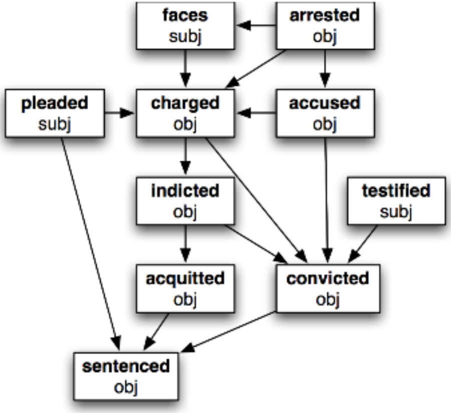
- BioNLP 09 shared task corpus
  - Train: 800 abstracts (7449 sentences, 8597 events)
  - Dev: 150 abstracts (1450 sentences, 1809 events)
  - Test: 260 abstracts (2447 sentences, 3182 events)
- Metric: BioNLP F1 scores with approximate span and recursive event matching
  - Gold text-entities (i.e., anchors and arguments) are extended by one word to the left and right
  - An event can match even if a nested event is only partially correct (i.e., if only Theme arguments match)

Results



Event Class	Count	R	P	F1
Gene Expression	722	68.6	75.8	72.0
Transcription	137	42.3	51.3	46.4
Protein Catabolism	14	64.3	75.0	69.2
Phosphorylation	135	80.0	82.4	81.2
Localization	174	44.8	78.8	57.1
Binding	347	42.9	51.7	46.9
Regulation	291	23.0	36.6	28.3
Positive Regulation	983	28.4	42.5	34.0
Negative Regulation	379	29.3	43.5	35.0
Total	3,182	42.6	56.6	48.6

Unsupervised Learning of Narrative Event Chains



- Learn narrative chains from text
- Arrows represent the *before* relation
- Events in the chain share a common protagonist

Chambers and Jurafsky, ACL 2008

Narrative Chain Model

- A **narrative event** is a tuple of ⟨event, dependency⟩
- A **narrative chain** is a partially-ordered set of n narrative events $e_1, e_2, ..., e_n$ with...
  - a common actor (the protagonist)
  - a relation $B(e_i, e_j)$ that is true if e_i occurs strictly before e_j
- Assumption of **narrative coherence**: verbs sharing coreferring arguments are semantically connected by virtue of narrative discourse structure

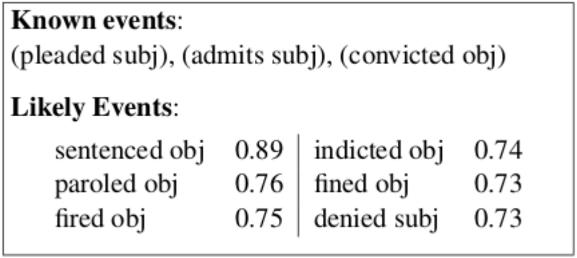
Overview

1. **Event Relatedness**: Learn the degree to which events with coreferring arguments are related
2. **Temporal Ordering**: Order narrative events using scores learned in step 1
3. **Chain Extraction**: Extract discrete narrative chains

Step 1: Learn Narrative Relations

- Use unsupervised distributional methods to learn narrative relations between events sharing coreferring arguments
  1. Parse training text (e.g., Gigaword corpus)
  2. Extract events: all verbs with subj, obj, or prep typed dependencies
  3. Resolve coreferring entity mentions
  4. For each document, record events (verb/dependency pairs) in which the verbs share coreferent entities
  5. Compute a score for each pair based on pointwise mutual information (PMI)
     - Sensitive to number of times two events had a coreferring entity filling the values of the dependencies

Step 1: Learn Narrative Relations



Known events:

(pleaded subj), (admits subj), (convicted obj)

Likely Events:

sentenced obj	0.89	indicted obj	0.74
paroled obj	0.76	fined obj	0.73
fired obj	0.75	denied subj	0.73

$$pmi(e(w,d),e(v,g)) = log \frac{P(e(w,d),e(v,g))}{P(e(w,d))P(e(v,g))}$$

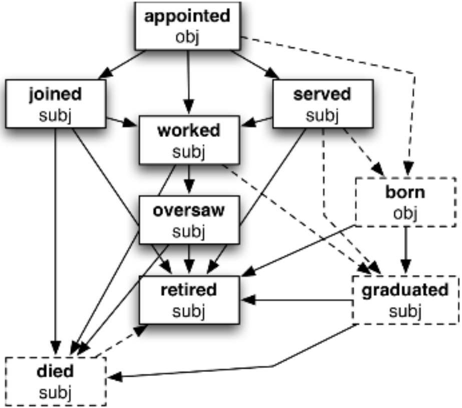
$$P(e(w,d)e(v,g)) = \frac{C(e(w,d),e(v,g))}{\sum_{x,y}\sum_{d,f} C(e(x,d),e(y,f))}$$

$$max_{j:0<j<m} \sum_{i=0}^{n} pmi(e_i, f_j)$$

Step 2: Order Narrative Events

- Apply a temporal classifier to partially order connected events
- Use Timebank corpus as supervised training data to train two classifiers
  1. Label temporal attributes of events (e.g., tense, aspect)
  2. Use labels from first stage to classify the temporal relationship between two events that share arguments
- Classify Gigaword corpus in two stages, and then
  - Count number of *before* relations between each event pair
  - Confidence between events A and B measured by comparing how many times A seen before B, vs B seen before A

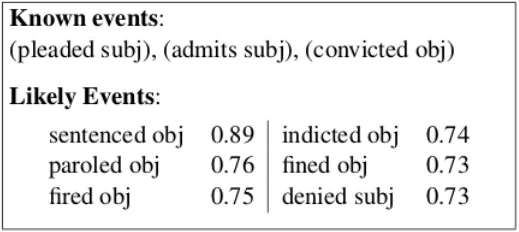
Step 3: Extract Discrete Narrative Chains



- Prune and cluster self-contained chains
- Inspired by Schank and Abelson's *scripts* of the late 70s
- Use the PMI scores in an agglomerative clustering algorithm
- Then apply ordering relations to produce a directed graph

Narrative Cloze Task

- Evaluate how well the model captures event relatedness
- Given a sequence of narrative events from which one event has been removed, predict the missing event
- Generate a set of ranked guesses using PMI
- Assign a score based on the rank of the missing event

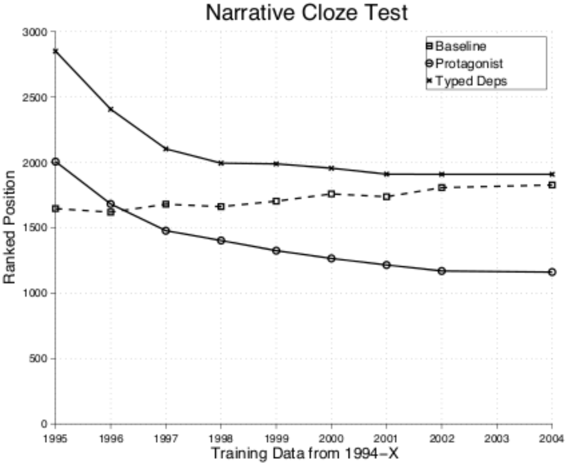


Known events:
(pleaded subj), (admits subj), (convicted obj)

Likely Events:

sentenced obj	0.89	indicted obj	0.74
paroled obj	0.76	fined obj	0.73
fired obj	0.75	denied subj	0.73

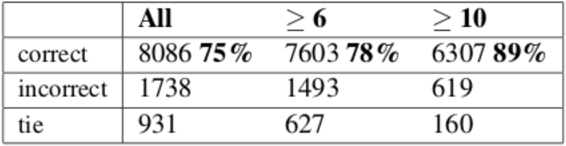
Narrative Cloze: Results



- *Baseline* is a verb-only model
- Compare to *protagonist* model
    - Compute PMI over verb pairs that share arguments

Order Coherence Task

- Present trained model with the task of deciding which is better:
  1. a narrative chain hand-labeled with partial ordering, or
  2. a narrative chain with random ordering
- Coherence score is sum of all relations that classified corpus agrees with, weighted by confidence



	All	$\geq$ **6**	$\geq$ **10**
correct	8086 **75%**	7603 **78%**	6307 **89%**
incorrect	1738	1493	619
tie	931	627	160

Outline

Introduction to Information Extraction

Named Entity Recognition

Relation Extraction

- NELL: Never-Ending Language Learner
- Open IE and TEXTRUNNER

Event Extraction

- Event Extraction as Dependency Parsing
- Unsupervised Learning of Narrative Event Chains

Discussion and Questions

What We've Seen

- Named Entity Recognition
- Relation Extraction
  1. NELL: Model for learning pre-specified set of relations using multiple views on a large amount of web data
  2. TextRunner: Model that extracts any available relations from a large amount of web data
- Event Extraction
  1. Extraction of nested events using a dependency parser and supervision
  2. Unsupervised extraction of narrative event chains