CS 4650/7650
Information Extraction and Question Answering[1]

Jacob Eisenstein

November 12, 2013

[1]Some material from Brendan O'Connor, Claire Cardie, and the FASTUS team.

Course roadmap

- Words (WSD, classification and morphology)
- Sequences (tagging)
- Trees (parsing)
- Semantics and discourse
- **Applications**
  - Today: knowledge from text
  - Next week: machine translation

Knowledge from text

- **Information extraction**
  - input: schema of desired knowledge base
  - output: populate schema from text resources
- **Question answering**
  - input: natural language questions
  - output: natural language answers
  - intermediate representation usually includes structured knowledge base

(see wolfram alpha video)

WolframAlpha computational... knowledge engine

what are the three lightest metals?

Examples Random

Input interpretation:

3 lightest metallic elements

Results:

By atomic weight: More

1	lithium	6.941 u	
2	beryllium	9.012182 u	
3	sodium	22.98976928 u	

⋮

By density: More

1	lithium	0.535 g/cm^3	
2	potassium	0.856 g/cm^3	
3	sodium	0.968 g/cm^3	

⋮

Units »

Computed by Wolfram *Mathematica* Source information » Download page

WolframAlpha computational knowledge engine

how expensive are the three lightest metals?

Examples Random

Using closest Wolfram|Alpha interpretation: **three lightest metals**

More interpretations: expensive

Input interpretation:

3 lightest metallic elements

Results:

By atomic weight: More

1	lithium	6.941 u	
2	beryllium	9.012182 u	
3	sodium	22.98976928 u	

⋮

By density: More

1	lithium	0.535 g/cm^3	
2	potassium	0.856 g/cm^3	
3	sodium	0.968 g/cm^3	

WolframAlpha computational knowledge engine

what is the lightest radioactive element?

Examples Random

Input interpretation:

lightest radioactive elements

Results:

By atomic weight: More

1	technetium	98 u	
2	promethium	145 u	
3	polonium	209 u	
4	astatine	210 u	
5	radon	222 u	

By density: More

1	radon	0.00973 g/cm^3	
2	radium	5 g/cm^3	
3	promethium	7.264 g/cm^3	
4	polonium	9.196 g/cm^3	
5	actinium	10.07 g/cm^3	

WolframAlpha computational knowledge engine

what is the lightest element which is radioactive?

Examples Random

Assuming "element" is elements | Use as a class of elements instead

Input interpretation:

lightest radioactive elements

Results:

By atomic weight: More

1	technetium	98 u
2	promethium	145 u
3	polonium	209 u
4	astatine	210 u
5	radon	222 u

By density: More

1	radon	0.00973 g/cm^3
2	radium	5 g/cm^3

WolframAlpha computational... knowledge engine

of all of the radioactive elements, which is the lightest?

Examples Random

Using closest Wolfram|Alpha interpretation: **the radioactive elements**

More interpretations: elements which

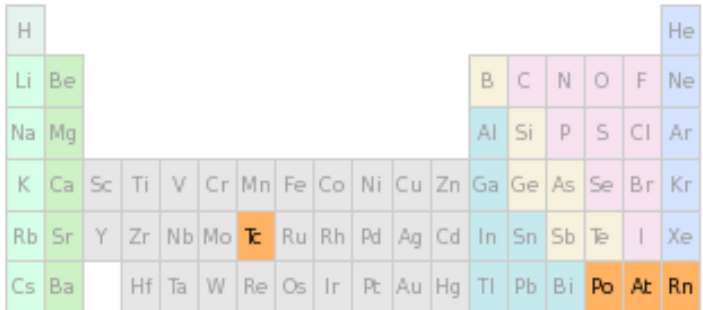
Input interpretation:

radioactive elements

Members: More

actinium | americium | astatine | berkelium | bohrium | californium | copernicium | curium | darmstadtium | dubnium | einsteinium | fermium | francium | hassium | lawrencium | meitnerium | mendelevium | neptunium | nobelium | plutonium | ... (total: 37)

Periodic table location:



Knowledge extraction in NLP

Knowledge extraction requires solving lots of NLP problems

- understand the source data (syntax, discourse, semantics)
- understand the query
- reason about how they fit together

Outline

Information Extraction
- Finding entities
- Entity linking
- Relation extraction
- Event detection

The six Ws

- Who, what, where, when, why, how?
- IE is mostly concerned with the first four.

The six Ws

- Who, what, where, when, why, how?
- IE is mostly concerned with the first four.
  - **who/where**: named entity extraction and coreference (we've already talked about this)
  - **what**: usually defined in terms of *relations* between entities
  - **when**
    - parsing time expressions, finding the temporal order of events
    - this is a big part of IE, but I'm not going to talk about it today

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

Named entity recognition

Find and tag **mentions** of **entities** in text.

At a meeting of <ORG>the Thirteen</ORG>,
<PER>Pyat Pree</PER> tells <PER>Daenerys</PER>
that he has <OBJ>her dragons</OBJ> in the
<PER>House of the Undying</PER>.

The Information Extraction pipeline

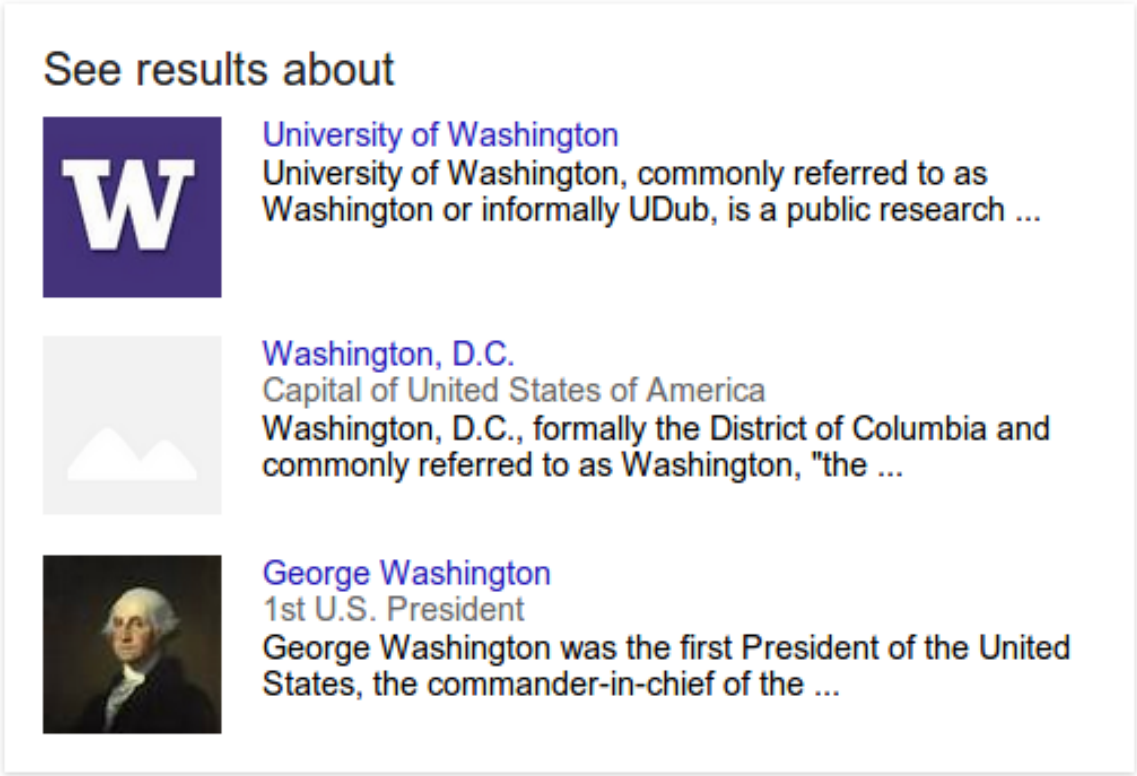
- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

Entity linking

Goal: link entity mentions to knowledge base entries.
Like multi-document coreference resolution, but must ultimately resolve to KB entry.



Entity linking: challenges

From Rao et al (2010)

1. Name variations: Boston Symphony Orchestra vs BSO, Qaddafi vs Gadaffi, etc
2. Name polysemy: Washington (person, place, football team, US Government, ...)
3. Absence: many entities do not appear in the KB.

In combination, these challenges are especially pernicious: William Clinton is a variation of Bill Clinton, but appears in Wikipedia as two other individuals.

Entity linking: steps

Candidate identification

- Brute force: check Google Knowedge Graph for all strings that could link to an entity
- Add source document coreference resolution

Entity linking: steps

Candidate identification

- Brute force: check Google Knowedge Graph for all strings that could link to an entity
- Add source document coreference resolution

Ranking Supervised formulation (Dredze et al 2010):

$$\begin{aligned} \min_{\mathbf{w}} \quad & ||\mathbf{w}||_2^2 \\ s.t. \quad & \mathbf{w}^\mathsf{T} f(\mathbf{x}_i, y_i) > \max_{\hat{y} \neq y_i} \mathbf{w}^\mathsf{T} f(\mathbf{x}_i, \hat{y}) \end{aligned}$$

Features:

- String match
- Popularity
- Local context and entity type
- Document context (similar entities)

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

Relations

A relation is a *predication* about a pair of entities.

- Davos **works for** Stannis
- King's Landing **is in** Westeros
- Joffrey's **father is** Jaime

Relations are typically permanent.

Example relations

From the Automatic Content Extraction (ACE) 2004 Task:

relation type	*subtypes*
physical	located, near, part-whole
personal-social	business, family, other
employment/membership/ subsidiary	employ-executive, employ-staff, employ-undetermined, member-of-group, partner, subsidiary, other
agent-artifact	user-or-owner, inventor-or-manufacturer, other
person-org affiliation	ethnic, ideology, other
GPE affiliation	citizen-or-resident, based-in, other
discourse	-

Relations in text

- Typically we focus on cases in which the relation and the two entities are all mentioned in the same sentence. (Exception: Robert and Cersei were married. A son was born the next year)

Relations in text

- Typically we focus on cases in which the relation and the two entities are all mentioned in the same sentence. (Exception: Robert and Cersei were married. A son was born the next year)
- Relation extraction requires coreference resolution.
  - He has her dragons:
    Pyat Pree <HAS> Daenerys' dragons

Relations in text

- Typically we focus on cases in which the relation and the two entities are all mentioned in the same sentence. (Exception: Robert and Cersei were married. A son was born the next year)
- Relation extraction requires coreference resolution.
  - He has her dragons:
    PYAT PREE <HAS> DAENERYS' DRAGONS
  - Eddard died. His daughter, Sansa, said the Eulogy.
    - Must resolve his to EDDARD and his daughter to SANSA
    - Then we can recover SANSA <DAUGHTER-OF> EDDARD

Relations in text

- Typically we focus on cases in which the relation and the two entities are all mentioned in the same sentence. (Exception: Robert and Cersei were married. A son was born the next year)
- Relation extraction requires coreference resolution.
  - He has her dragons:
    Pyat Pree <HAS> Daenerys' dragons
  - Eddard died. His daughter, Sansa, said the Eulogy.
    - Must resolve his to Eddard and his daughter to Sansa
    - Then we can recover Sansa <Daughter-of> Eddard
- **Micro-reading**: correctly identify every relation *mention*
- **Macro-reading**: correctly identify every relation in the text

Knowledge-base population (KBP)

Extract attributes for each named person or organization:

entity	house	father	mother	position
Arya	Stark	Eddard	Catelyn	
Daenerys	Targaryen	Aerys		Mother-of-dragons
Qhorin	commoner			Knight-of-the-Watch

KBP is similar to relation extraction.

- Columns define relation types
- Rows define the left entity
- Cells define the right entity

Relations from patterns

- Early idea: lexical patterns, like regular expressions
- Possible patterns for PERSON lives-in LOCATION:
  - PERSON lives in LOCATION
  - PERSON lived in LOCATION
  - PERSON has lived in LOCATION
  - PERSON resides in LOCATION

Relations from patterns

- Early idea: lexical patterns, like regular expressions
- Possible patterns for PERSON lives-in LOCATION:
  - PERSON lives in LOCATION
  - PERSON lived in LOCATION
  - PERSON has lived in LOCATION
  - PERSON resides in LOCATION
- Can we generalize beyond lexical patterns?

Relations from patterns

- Early idea: lexical patterns, like regular expressions
- Possible patterns for PERSON LIVES-IN LOCATION:
  - PERSON <V BASE=LIVE> in LOCATION
  - PERSON has lived in LOCATION
  - PERSON resides in LOCATION
- Can we generalize beyond lexical patterns?
  - morphological analysis

Relations from patterns

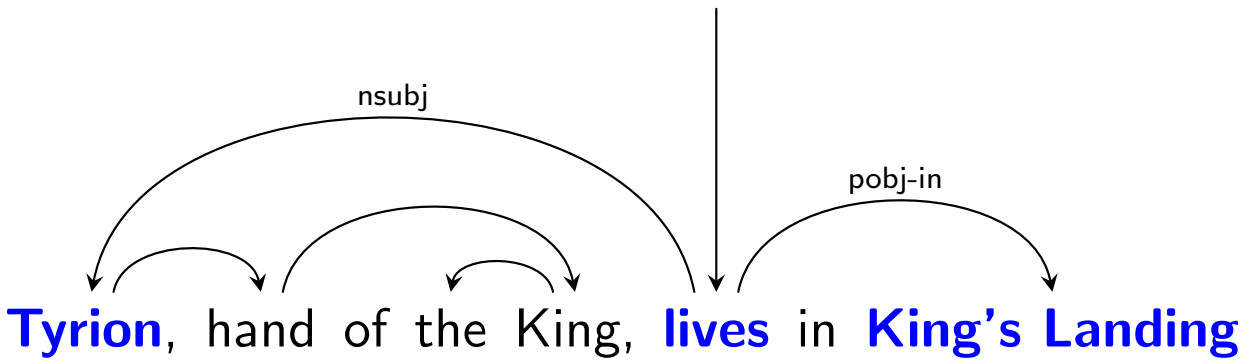
- Early idea: lexical patterns, like regular expressions
- Possible patterns for PERSON LIVES-IN LOCATION:
  - PERSON <VGROUP BASE=LIVE> in LOCATION
  - PERSON resides in LOCATION
- Can we generalize beyond lexical patterns?
  - morphological analysis
  - phrase chunking

Relations from patterns

- Early idea: lexical patterns, like regular expressions
- Possible patterns for PERSON LIVES-IN LOCATION:
  - PERSON <VGROUP SYNSET=LIVE#1> in LOCATION
- Can we generalize beyond lexical patterns?
  - morphological analysis
  - phrase chunking
  - lexical semantics

Syntactic patterns

Given a dependency parse, we can define more flexible patterns:



Supervised relation extraction

We can develop a classifier for each relation type, or a general classifier for detecting relations of any type.

- Feature-based classification (**Features**: heads of each entity, types of each entity, distance between entities, words between entities, dependency path between entities, ...)
- Kernel-based classification
  - Kind of like K-nearest-neighbors classification
  - The label for a test instance should be based on similar training instances

Whirlwind tour of kernel-based classification

- The kernel function maps from pairs of instances to a non-negative real value.

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$$

  - $K(x_1, x_2)$ is large for similar x_1, x_2, small if they are different
  - K can count number of shared words, etc.

Whirlwind tour of kernel-based classification

- The kernel function maps from pairs of instances to a non-negative real value.

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$$

  - $K(x_1, x_2)$ is large for similar x_1, x_2, small if they are different
  - K can count number of shared words, etc.

- Classification rule:

$$\hat{y}(\mathbf{x}) = \sum_i y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b$$

  - Each $\alpha_i \geq 0$ is a parameter, which must be learned.
  - $\#|\alpha| = \#|\mathbf{x}|$, regardless of number of features

Other training paradigms: bootstrapping

- Start with a few seed patterns
- Extract some high-confidence relations
- Induce more patterns
- Extract more relations
- ...

Other training paradigms: distant learning

- Start with a large set of known relations (e.g. from Freebase)
- Collect all sentences that include both entities in the relation. These are positive training instances.
- Sample negative training instances (for example, sentences that contain one entity in a relation but not both).
- a.k.a. multi-instance learning

The Information Extraction pipeline

- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

The Information Extraction pipeline

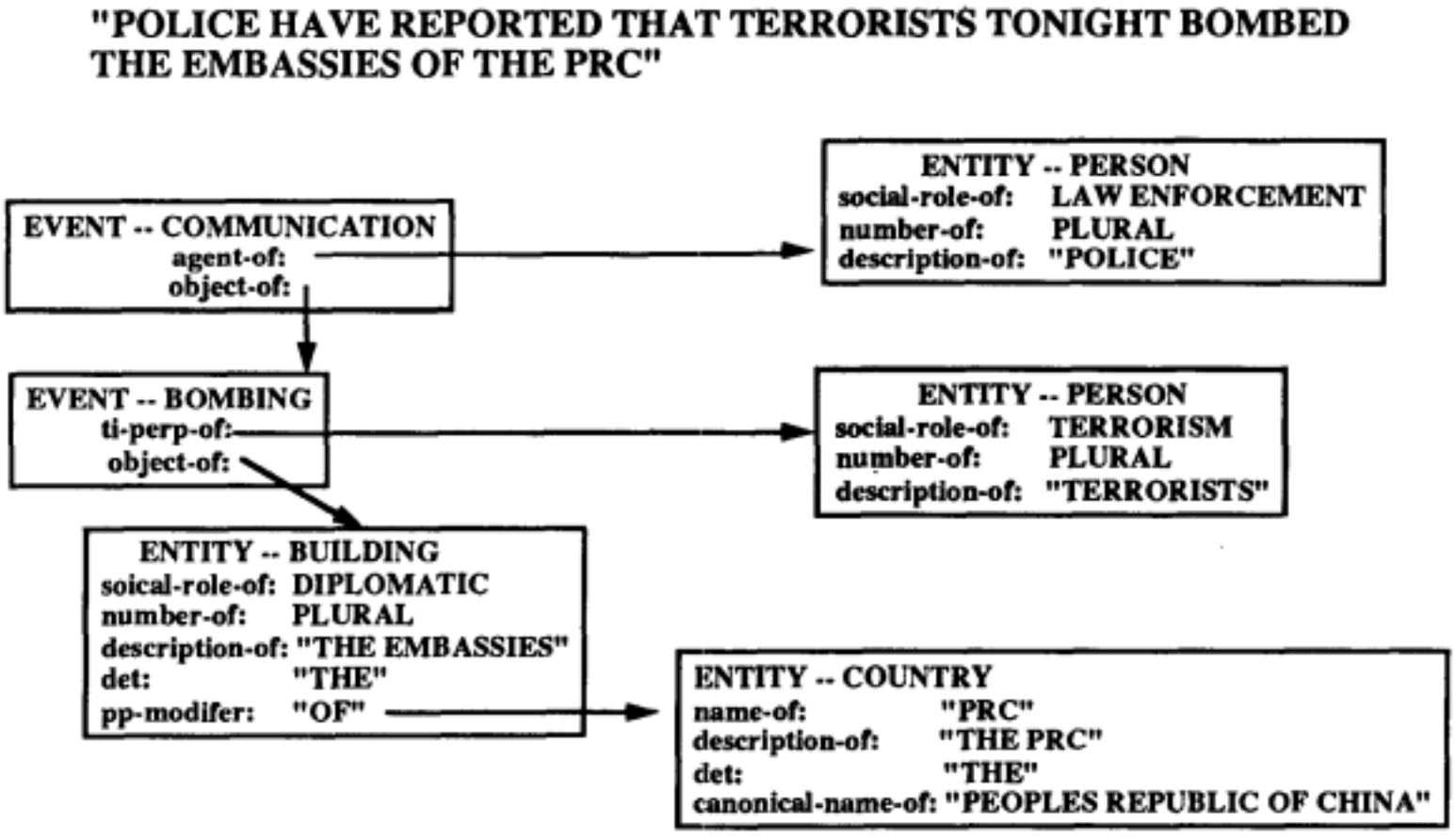
- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying</PER>.
- Linked entities:
  - <PER>Pyat Pree</PER> → Pyat Pree
  - <PER>Daenerys</PER> → Daenerys Targaryen
- Relations:
  - Pyat Pree <HAS> Dragons
  - Dragons <LOCATED-IN> House of the Undying
- Events:
  Possession: [Object: dragons; Location: House of the Undying; Possessor: Pyat Pree]

Event extraction

- **Relations** are predications involving two arguments.
- **Events** are predications involving arbitrary numbers of arguments.

Event type	Subtypes
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-ownership, Transfer-money
Business	Start-org, Merge-org, Declare-bankruptcy, End-org
Conflict	Attack, Demonstrate
Personnel	Start-position, End-position, Nominate, Elect
Justice	Arrest-jail, Release-parole, Trial-hearing Charge-indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Representing events



Event templates

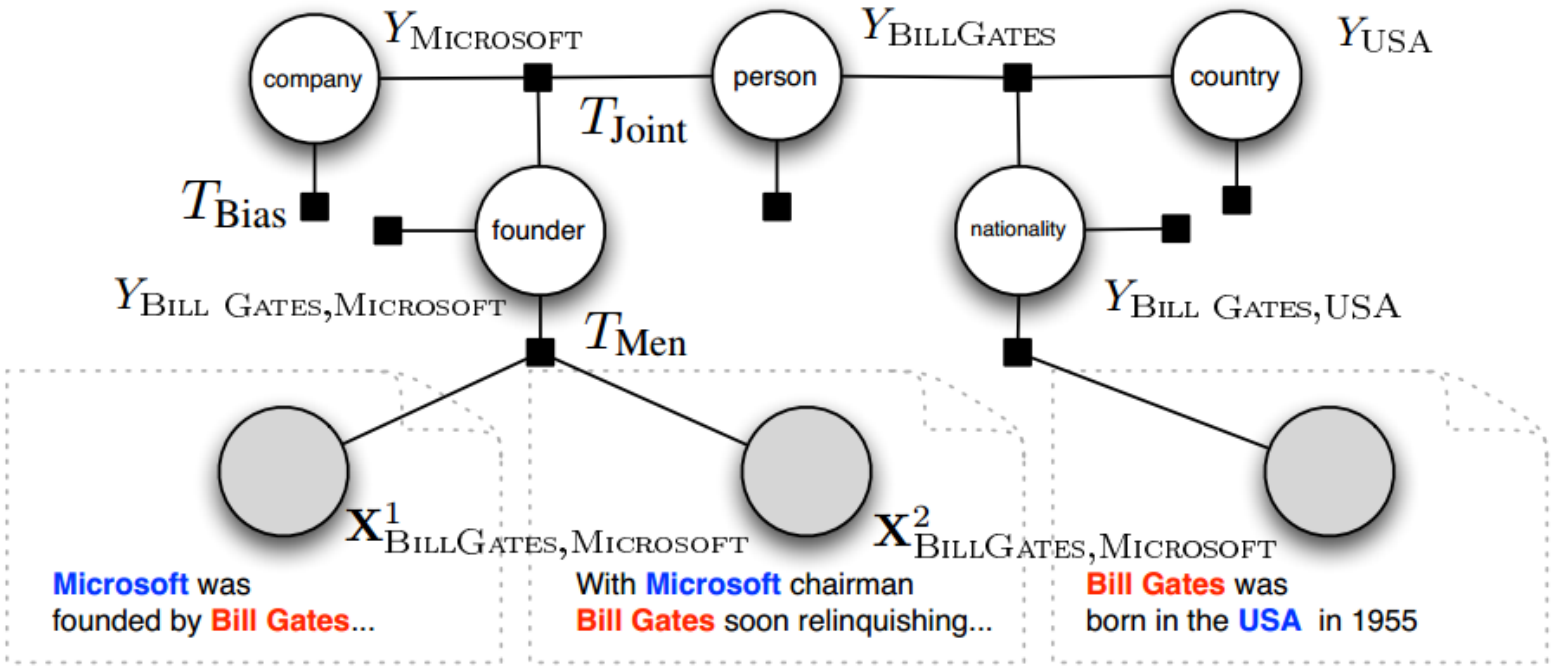
In supervised event extraction, each event type has a template of relevant attributes.

0.	MESSAGE ID	TST1-MUC3-0099
1.	TEMPLATE ID	1
2.	DATE OF INCIDENT	- 25 OCT 89
3.	TYPE OF INCIDENT	BOMBING
4.	CATEGORY OF INCIDENT	TERRORIST ACT
5.	PERPETRATOR: ID OF INDIV(S)	"TERRORISTS"
6.	PERPETRATOR: ID OR ORG(S)	-
7.	PERPETRATOR CONFIDENCE	-
8.	PHYSICAL TARGET: ID(S)	"THE EMBASSIES"
9.	PHYSICAL TARGET: TOTAL NUM	PLURAL
10.	PHYSICAL TARGET: TYPE(S)	DIPLOMAT OFFICE OR RESIDENCE: "THE EMBASSIES"
11.	HUMAN TARGET: ID(S)	-
12.	HUMAN TARGET: TOTAL NUM	-
13.	HUMAN TARGET: TYPE(S)	-
14.	TARGET: FOREIGN NATIONS	-
15.	INSTRUMENT: TYPE(S)	*
16.	LOCATION OF INCIDENT	EL SALVADOR: SAN ISIDRO (TOWN)
17.	EFFECT ON PHYSICAL TARGET	SOME DAMAGE: "THE EMBASSIES"
18.	EFFECT ON HUMAN TARGET	NO INJURY: "-"

Typical approach: train classifiers for each slot in the template

Collective relation extraction

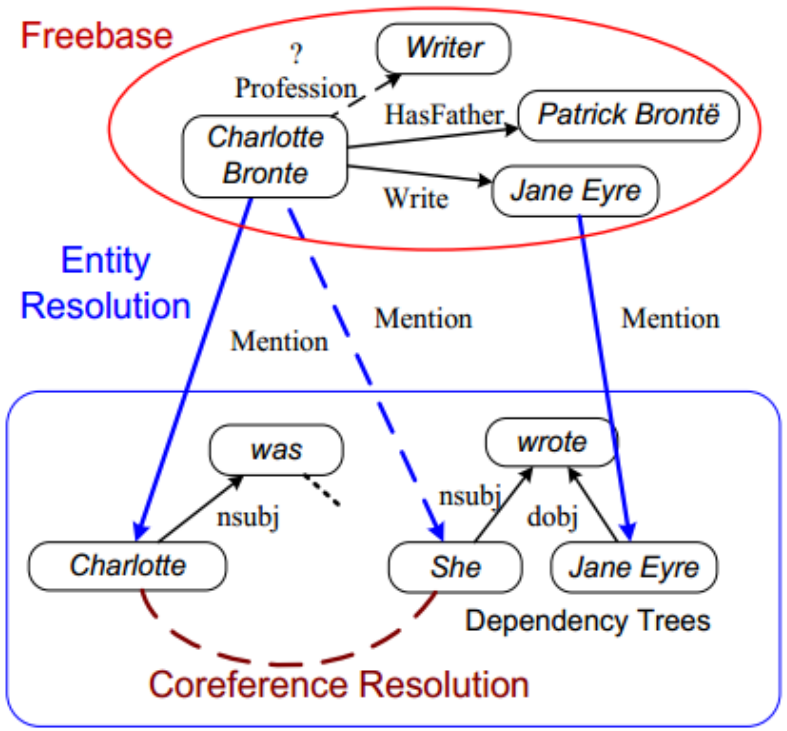
Joint reasoning about both language understanding and the underlying semantics.



(Yao, Riedel, and McCallum, 2010)

Collective relation extraction

Joint reasoning about both language understanding and the underlying semantics.



(Lao, Subramanya, Pereira, and Cohen, 2012)

Special cases: time

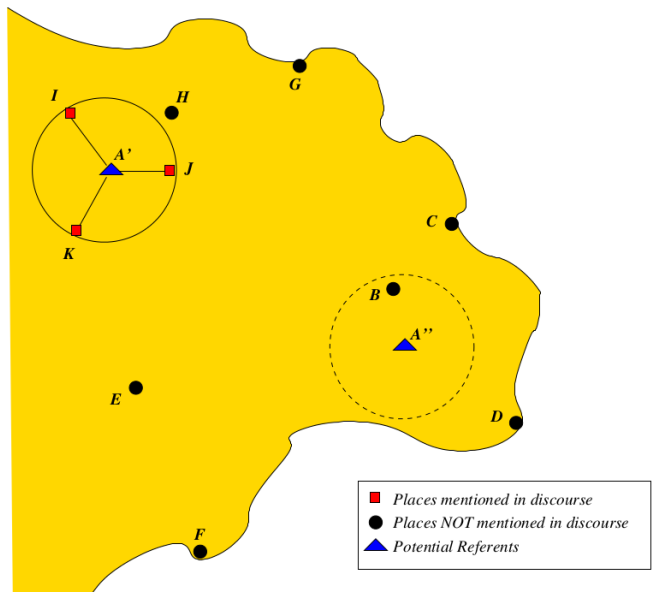
Expression	*Type*	*Value*
October of 1963	DATE	1963-10
October	DATE	2011-10
last Friday	DATE	2011-09-16
next weekend	DATE	2011-W39-WE
the day after tomorrow	DATE	2011-09-21
the nineties	DATE	199X
winter of 2000	DATE	2000-WI
5th century B.C.	DATE	-05XX
now	DATE	PRESENT_REF
Saturday morning	TIME	2011-09-24TMO
4 p.m. Tuesday	TIME	2011-09-20T16:00

- SUTIME is a rule-based system for parsing time expressions.
- Recent work (Agneli et al) has focused on statistical parsing.

Special cases: space

Location descriptions also have a structure that is hierarchical and complex, yet arguably tractable.

- the first gas station after you cross under the 85 on Peachtree Road in Vinings
- on the left after you come up stairs in TSRB
- in the bottom of a drawer in the cabinet opposite the green bookcase



The problem of resolving place names is **toponym resolution**.

Next steps: beliefs and evidence

- Possibly factual: United States may extend its naval quarantine to Jordans Red Sea port of Aqaba.
- Possibly counter-factual: They may not have enthused him for their particular brand of political idealism.
- Source-specific facuality: Izvestiya **said** that the G-7 leaders **pretended** everything was OK in Russia's economy.
- Epistemic marking: He saw the gunman, The WSJ editorial page speculated ...

FactBank is a corpus of factuality annotations (Saurí and Pustejovsky 2009).

	Positive (+)	Negative (−)	Underspecified (u)
Certain (CT)	Fact: <CT,+>	Counterfact: <CT,−>	Certain but unknown output: <CT,u>
Probable (PR)	Probable: <PR,+>	Not probable: <PR,−>	(NA)
Possible (PS)	Possible: <PS,+>	Not certain: <PS,−>	(NA)
Underspecified (U)	(NA)	(NA)	Unknown or uncommitted: <U,u>