CS 4650/7650
Information Extraction

Jacob Eisenstein

April 10, 2017

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

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>.

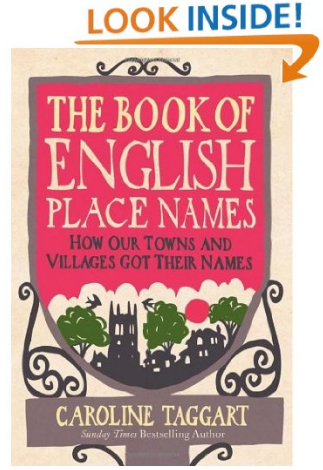
NER with rules

Entity recognition can be performed with rules.

Rule: TheGazOrganization
Priority: 50
// Matches "The <in list of company names>"
({Part of speech = DT | Part of speech = RB} {DictionaryLookup = organization})
→ Organization

Rule: LocOrganization
Priority: 50
// Matches "London Police"
({DictionaryLookup = location | DictionaryLookup = country} {DictionaryLookup
= organization} {DictionaryLookup = organization}?) → Organization

- These rules are from GATE (General Arch. for Text Engineering), `http://gate.ac.uk/`
- Rules may leverage POS tags and dictionaries.



NER with rules

Rule: INOrgXandY
Priority: 200
// Matches “in Bradford & Bingley”, or “in Bradford & Bingley Ltd”
({Token string = “in”})
({Part of speech = NNP}+ {Token string = “&”} {Orthography type = upperInitial}+ {DictionaryLookup = organization end}?):orgName → Organization=:orgName

Rule: OrgDept
Priority: 25
// Matches “Department of Pure Mathematics and Physics”
({Token.string = “Department”} {Token.string = “of”} {Orthography type = upperInitial}+ ({Token.string = ”and”} {Orthography type = upperInitial}+)?) → Organization

- Rules may overlap or disagree; the better the coverage, the more likely this is.
- Arbitrating disagreements is a complex engineering task.
- One solution: order rules by precision on training data.

NER as Sequence Labeling

Pyat/B-PER Pree/I-PER tells/O Daenerys/B-PER that/O he/O has/O her/B-OBJ dragons/I-OBJ ...

- **Tags**: B,I,O for each entity type
- **Features**: bag-of-words, word shape (characters), dictionary (list of known names), part-of-speech...
- **Method**: sequence labeling

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} \sum_i \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_i, y_{i-1}, i)$$

  - Hidden Markov Model: $\boldsymbol{\theta} = \operatorname{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\theta})$
  - Conditional Random Field: $\boldsymbol{\theta} = \operatorname{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{y} \mid \boldsymbol{w}; \boldsymbol{\theta})$
  - Structured Perceptron: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) - \boldsymbol{f}(\boldsymbol{w}, \hat{\boldsymbol{y}})$

Features on spans of text

- Dictionaries may contain **multitoken spans** (e.g. The House of the Undying)
- We want features that fire when a **span** matches a dictionary entry. $\boldsymbol{f}(\boldsymbol{w}, y_i, y_{i'}, i', i)$: set of features for the span from $i' + 1$ to i
- Can we still use Viterbi?

Features on spans of text

- Dictionaries may contain **multitoken spans** (e.g. The House of the Undying)
- We want features that fire when a **span** matches a dictionary entry. $\boldsymbol{f}(\boldsymbol{w}, y_i, y_{i'}, i', i)$: set of features for the span from $i' + 1$ to i
- Can we still use Viterbi?
- Can we still use dynamic programming?

Features on spans of text

- Dictionaries may contain **multitoken spans** (e.g. The House of the Undying)
- We want features that fire when a **span** matches a dictionary entry. $\boldsymbol{f}(\boldsymbol{w}, y_i, y_{i'}, i', i)$: set of features for the span from $i' + 1$ to i
- Can we still use Viterbi?
- Can we still use dynamic programming?

$$V(i, y) = \begin{cases} \max_{y'} \max_{i' \in i-L,\ldots,i-1} V(i', y') + \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_i, y_{i'}, i', i), & i > 0 \\ 0, & i = 0 \\ -\infty, & i < 0 \end{cases}$$

Features on spans of text

- Dictionaries may contain **multitoken spans** (e.g. The House of the Undying)
- We want features that fire when a **span** matches a dictionary entry. $\boldsymbol{f}(\boldsymbol{w}, y_i, y_{i'}, i', i)$: set of features for the span from $i' + 1$ to i
- Can we still use Viterbi?
- Can we still use dynamic programming?

$$V(i, y) = \begin{cases} \max_{y'} \max_{i' \in i-L,\ldots,i-1} V(i', y') + \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_i, y_{i'}, i', i), & i > 0 \\ 0, & i = 0 \\ -\infty, & i < 0 \end{cases}$$

- Complexity: $\mathcal{O}(MLK^2)$, with $M = \#|\boldsymbol{w}|, K = \#|\mathcal{Y}|, L =$ max span

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.

These challenges are especially tough in combination: 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):

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

Features:

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

Shea Serrano
@SheaSerrano

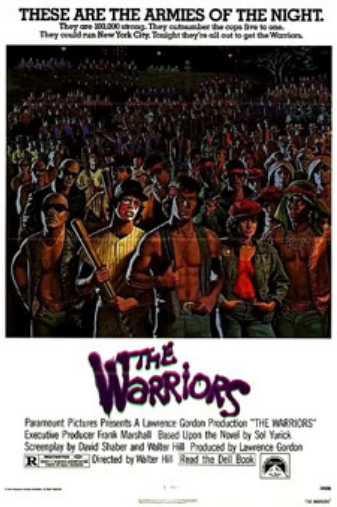
Follow

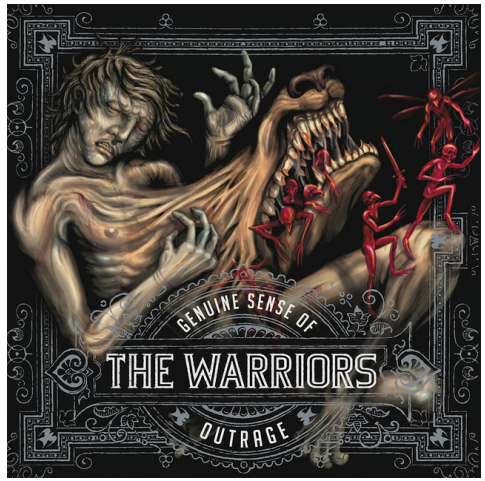
an absolutely perfect response by the warriors

Shea Serrano
@SheaSerrano

Follow

an absolutely perfect response by the warriors

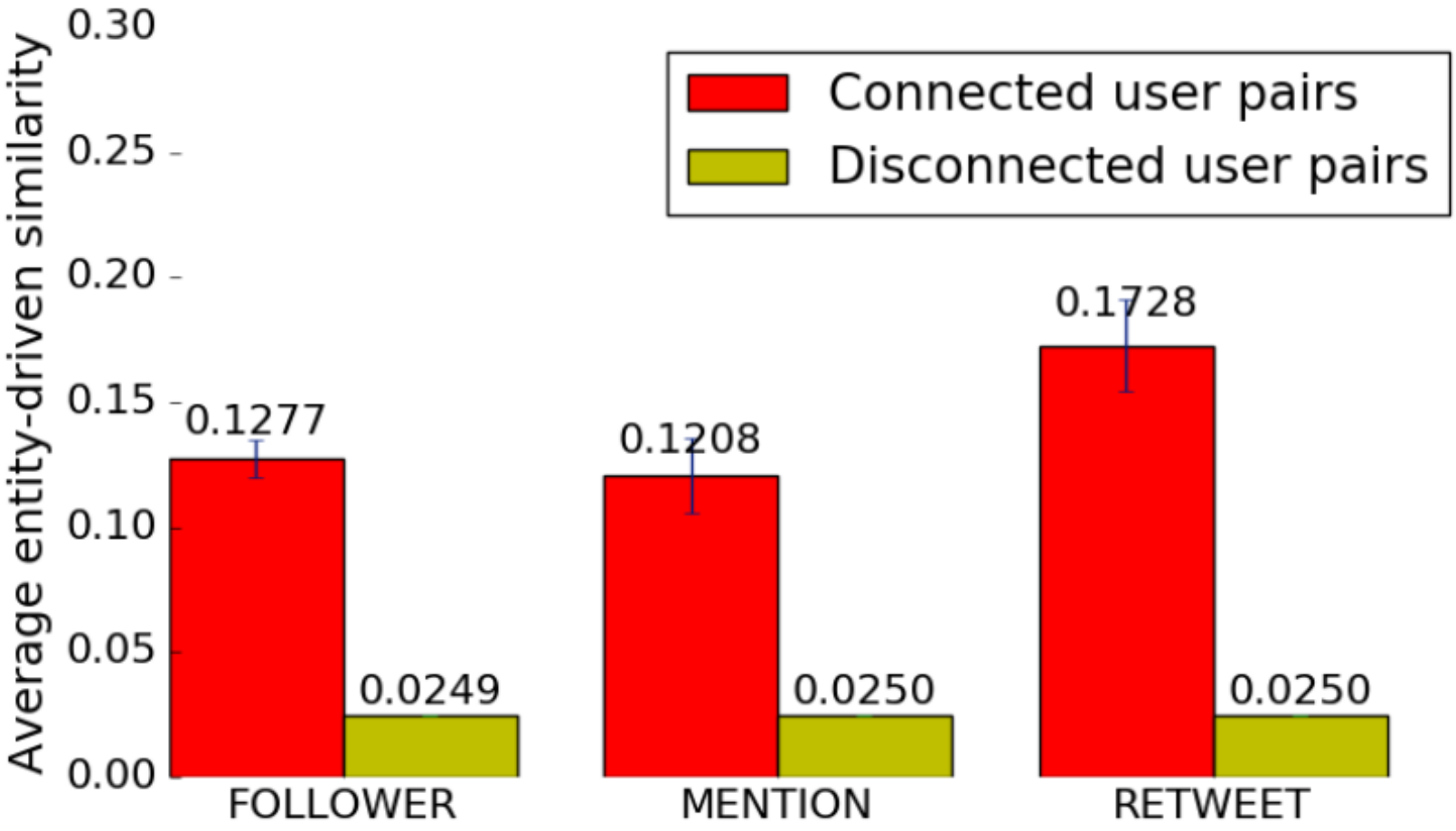




Finding tacit context in the social network

- Social media texts lack context, because it is implicit between the writer and the reader.
- **Homophily**: socially connected individuals tend to share traits.

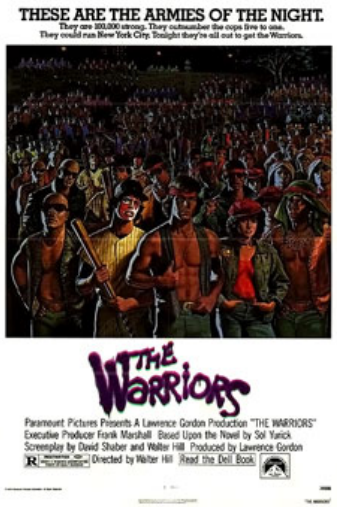
Evidence for linguistic homophily

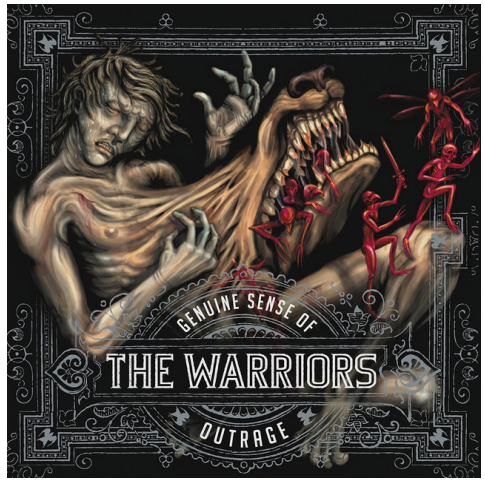


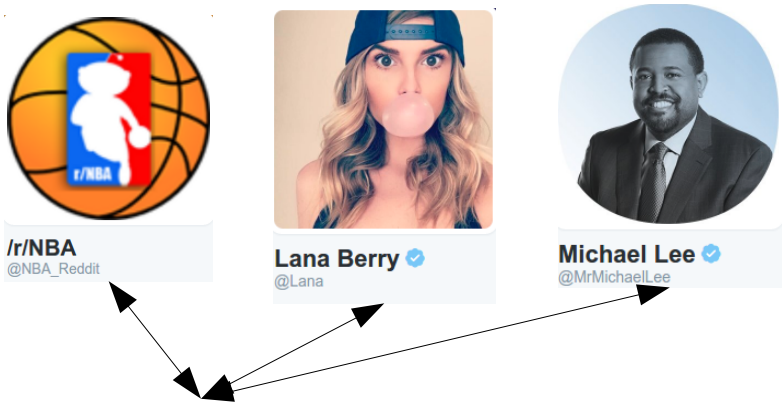
Shea Serrano
@SheaSerrano

Follow

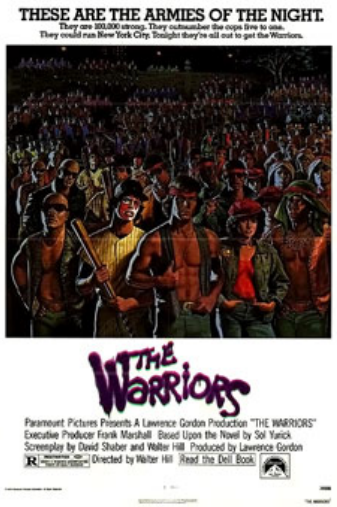
an absolutely perfect response by the warriors



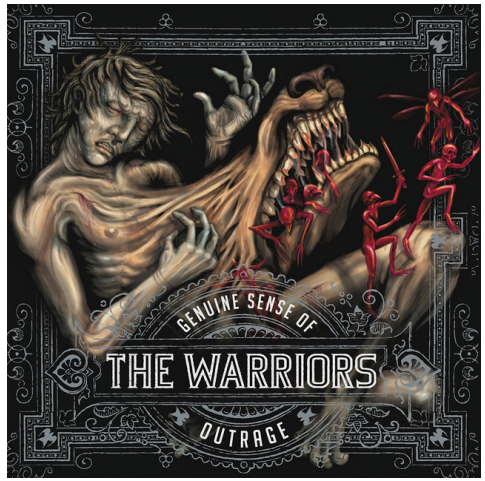


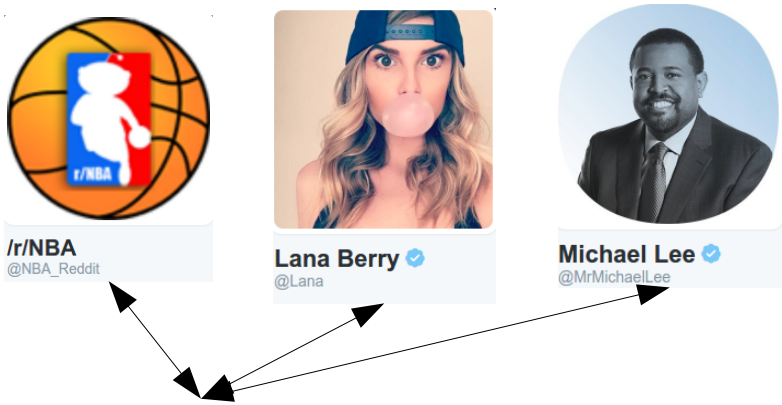

/r/NBA
@NBA_Reddit
Lana Berry
@Lana
Michael Lee
@MrMichaelLee


Shea Serrano
@SheaSerrano
Follow
an absolutely perfect response by the warriors


THESE ARE THE ARMIES OF THE NIGHT.
They are 100,000 strong. They outnumber the cops five to one.
They could run New York City. Tonight they're all out to get the Warriors.
THE WARRIORS
Paramount Pictures Presents A Lawrence Gordon Production "THE WARRIORS"
Executive Producer Frank Marshall Based Upon the Novel by Sol Yurick
Screenplay by David Shaber and Walter Hill Produced by Lawrence Gordon
R Directed by Walter Hill Read the Dell Book

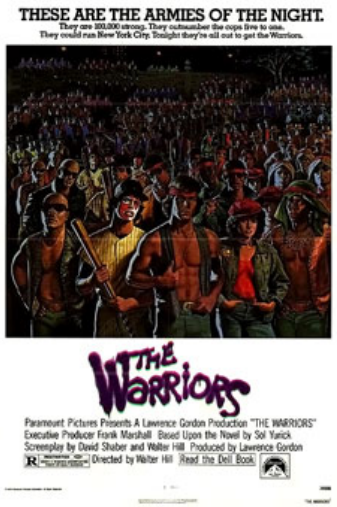

GOLDEN STATE
WARRIORS


GENUINE SENSE OF
THE WARRIORS
OUTRAGE



The return of Clutch Dirk Nowitzki is one of the more exciting, unexpected developments in an already bonkers NBA season

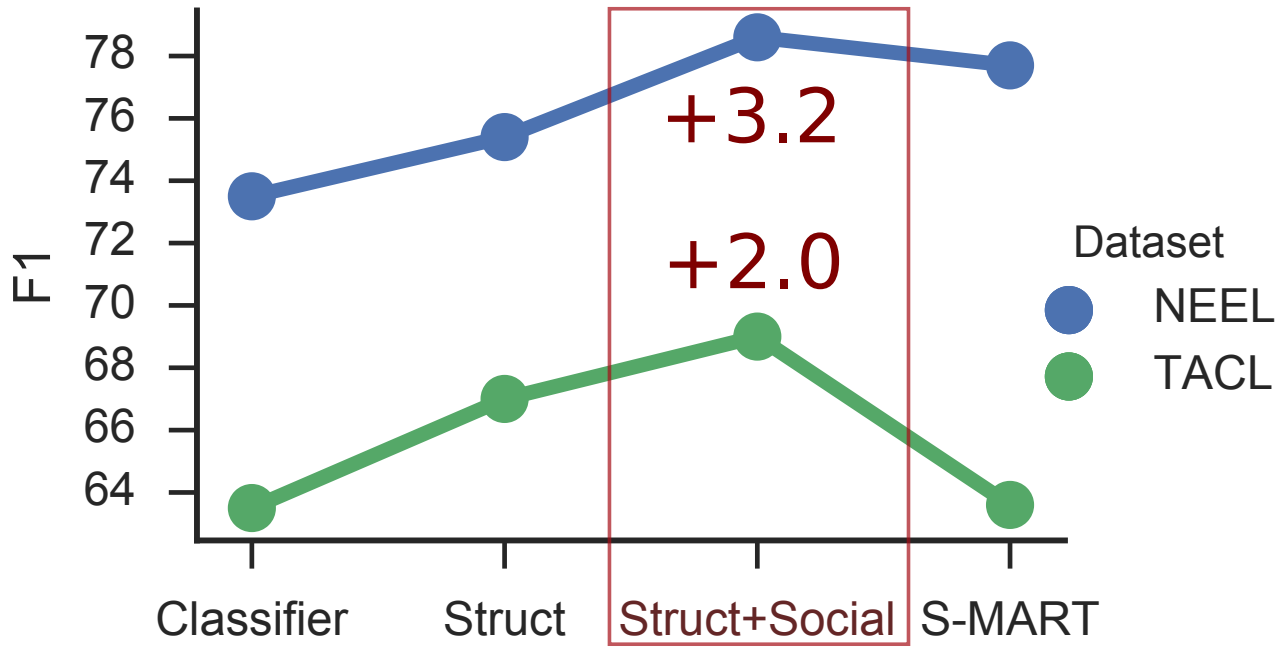
an absolutely perfect response by the warriors



We project embeddings for entities, words, and authors into a shared semantic space.

Inner products in this space indicate compatibility.

Results



- Structure prediction improves accuracy.
- Social context yields further improvements.
- S-MART is the prior state-of-the-art (Yang & Chang, 2015).

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:

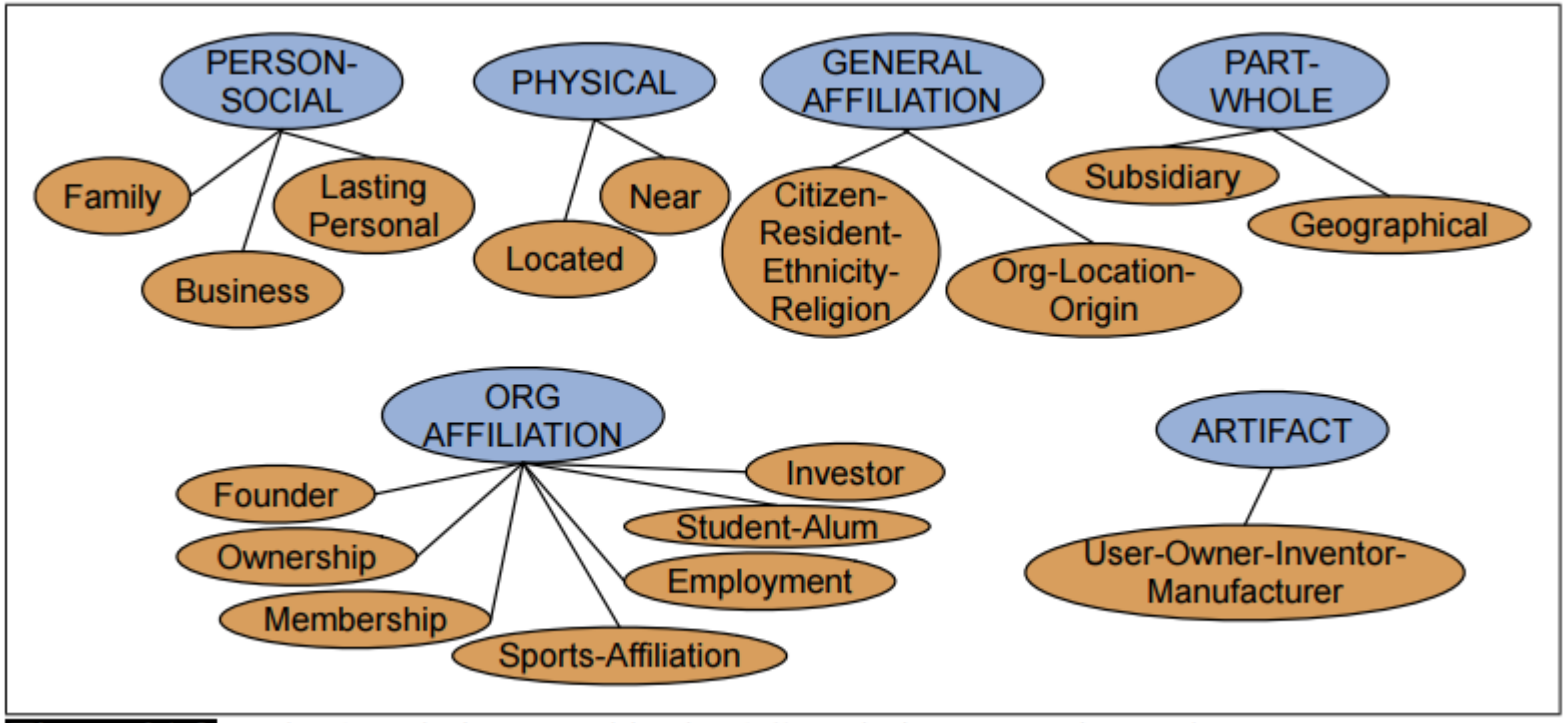


Figure 21.8 The 17 relations used in the ACE relation extraction task.

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
  - 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

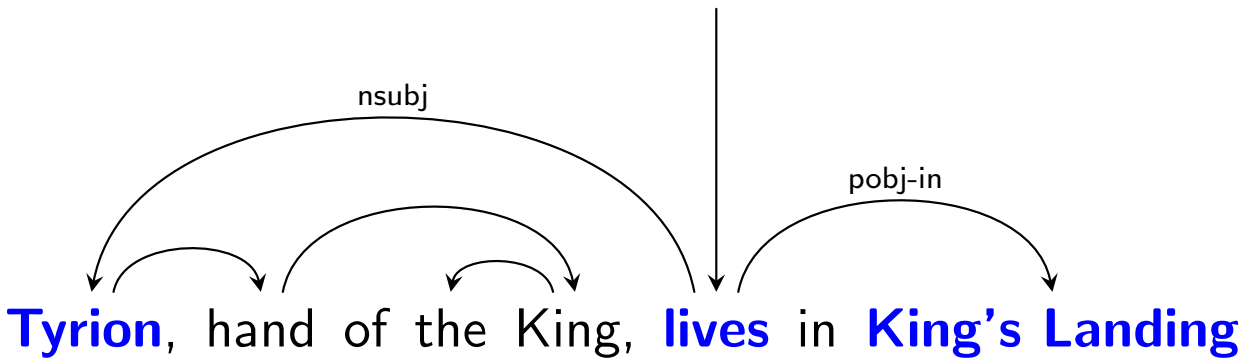
- Early idea: lexical patterns, like regular expressions
- Possible patterns for PERSON LIVES-IN LOCATION:
  - PERSON <VGROUP BASE=LIVE> in LOCATION
  - PERSON resides in LOCATION
- Can we generalize beyond lexical patterns?
  - morphological analysis
  - phrase chunking

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:

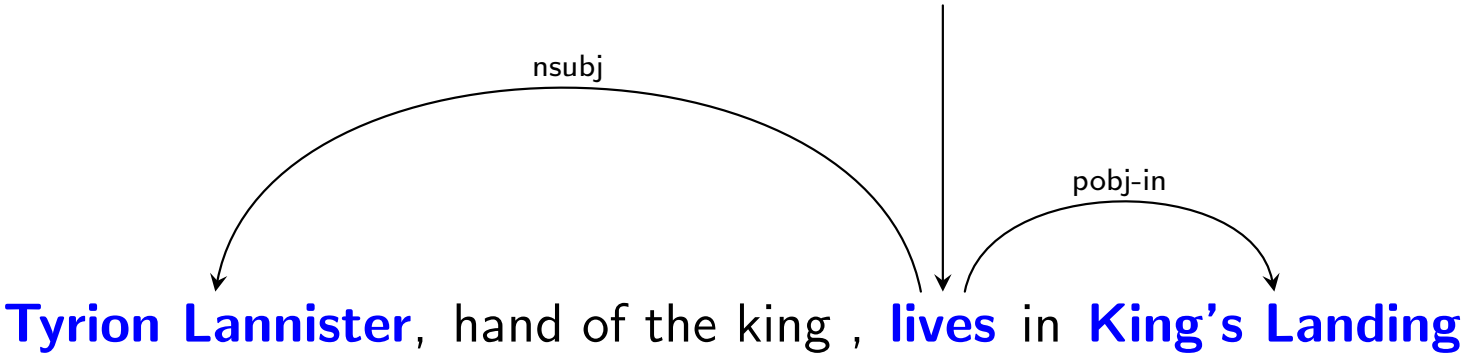


Supervised relation extraction

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

- Feature-based classification
  - Compute features of each proposed relation
  - Learn weights from labeled data
- Kernel-based classification
  - Kind of like K-nearest-neighbors classification
  - The label for a test instance should be based on similar training instances

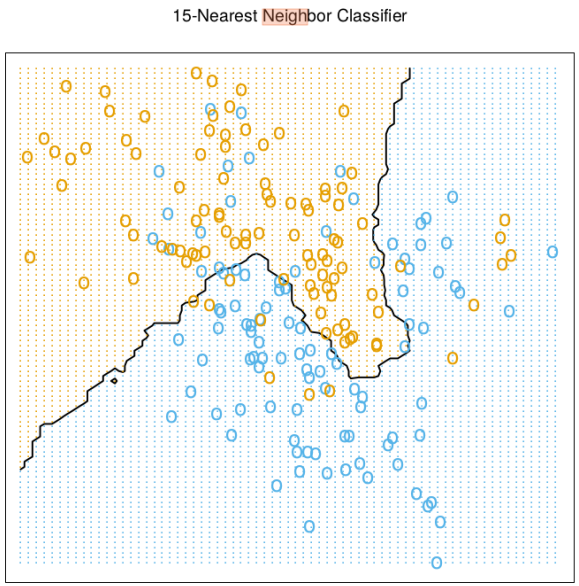
Feature-based relation extraction



- **Heads**: Lannister, Landing, lives
- **POS**: NNP, VBZ, NNP
- **Types**: PER, LOC
- **Distance**: six words, zero entities
- **Words between entities**: hand; of; the; King; lives; in
- **Path**: NSUB↑-POBJ-IN↓
- **Path-words**: lives-in

K-nearest neighbor classification

- Most of the learning methods considered in the course are linear classifiers.
- What if we need a non-linear classification rule?
- **K-nearest-neighbors**: let "most similar" instances in training set vote.
- **Kernel** methods generalize this idea.



Kernel-based classification

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

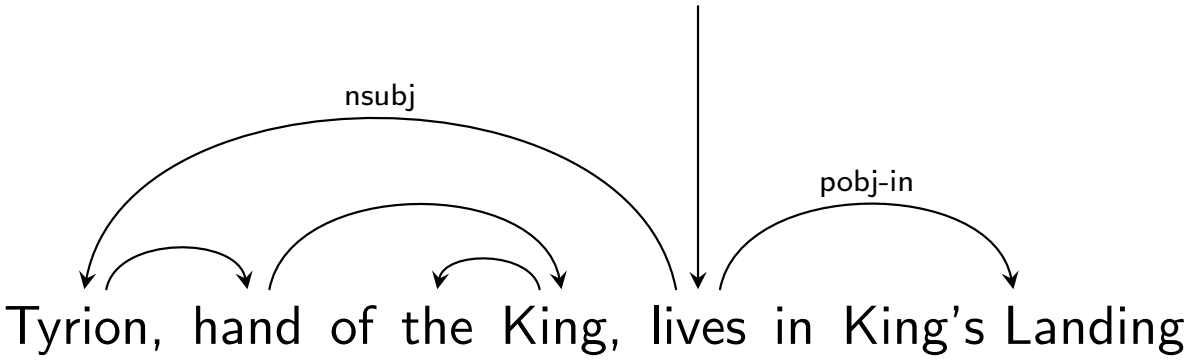
$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$$

- $K(x_1, x_2)$ arises from inner product of (implicit) feature vectors in margin constraint,

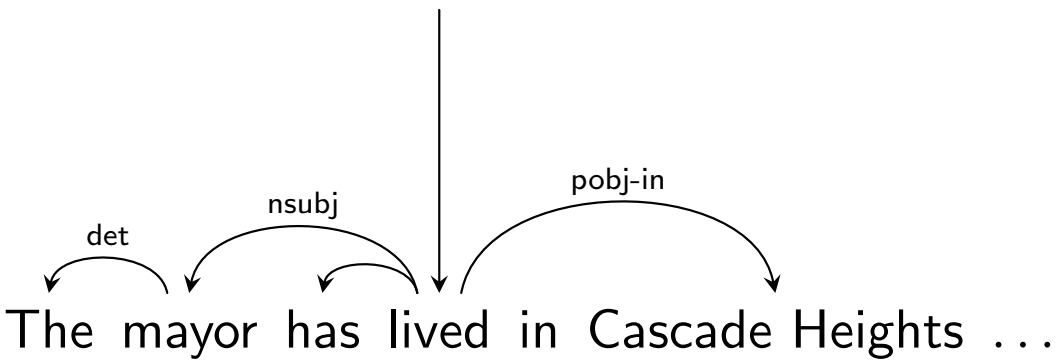
$$K(x_1, x_2) = \mathbf{f}(x_1) \cdot \mathbf{f}(x_2). \tag{1}$$

  - Large if x_1 and x_2 are similar.
  - Matrix K must be positive semi-definite.
- However, we define K explicitly, rather than f.

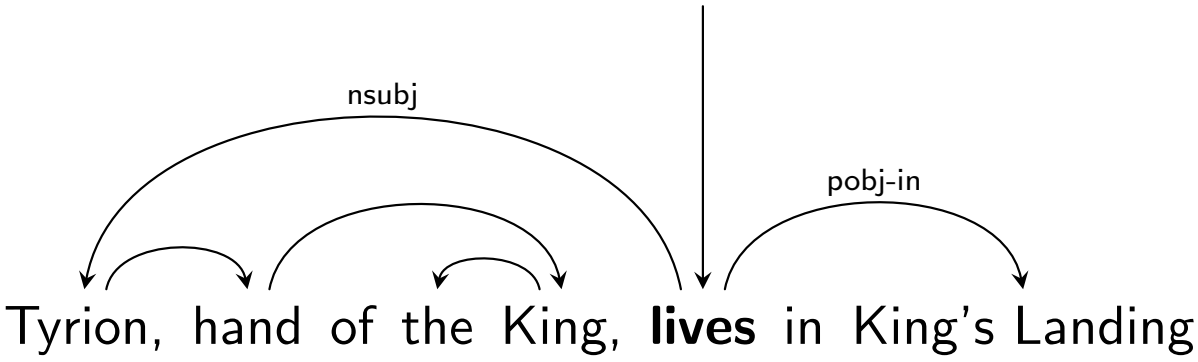
Dependency kernel example



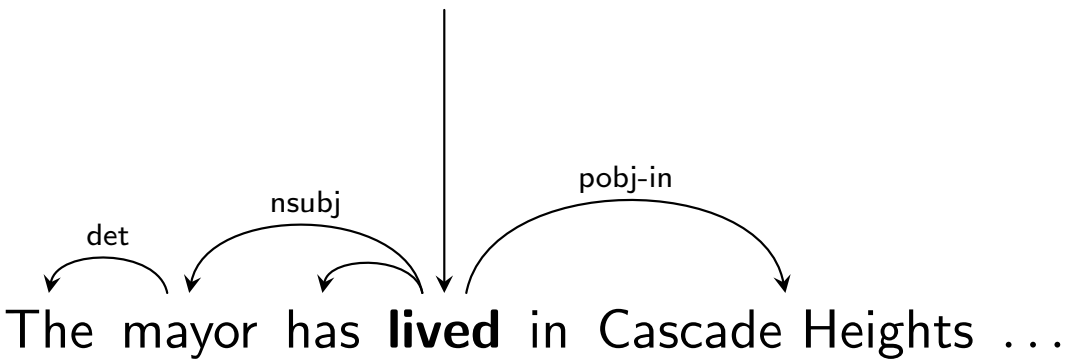
$$K(x_1, x_2) =$$



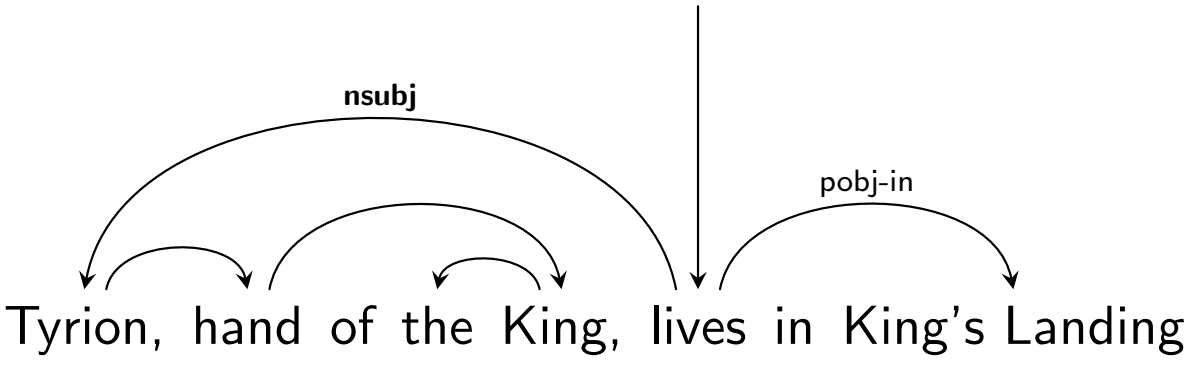
Dependency kernel example



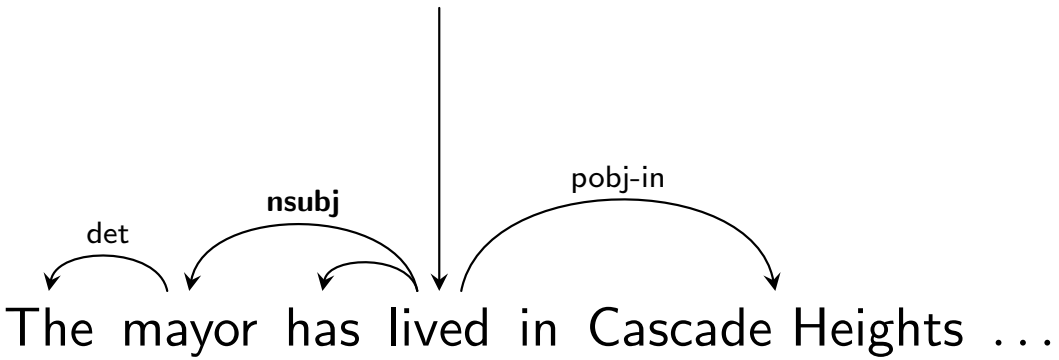
$$K(x_1, x_2) = \quad 1$$



Dependency kernel example

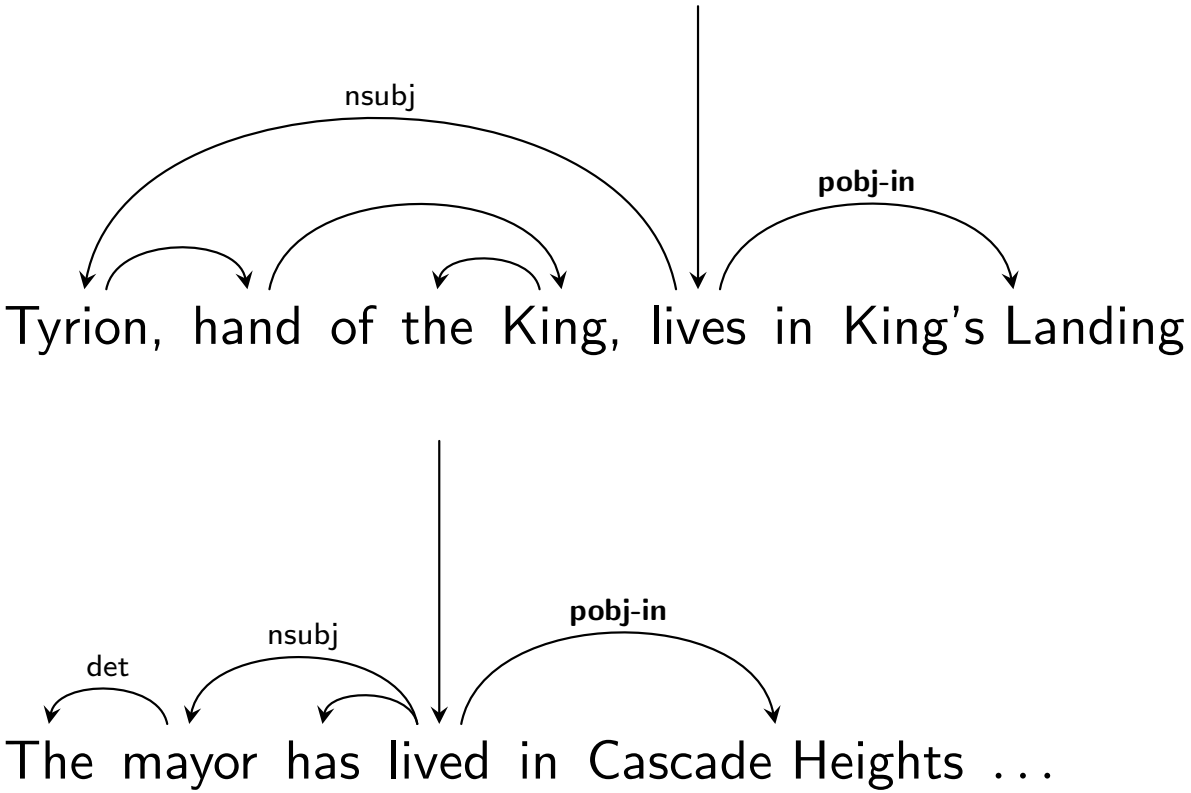


$$K(x_1, x_2) = 1$$



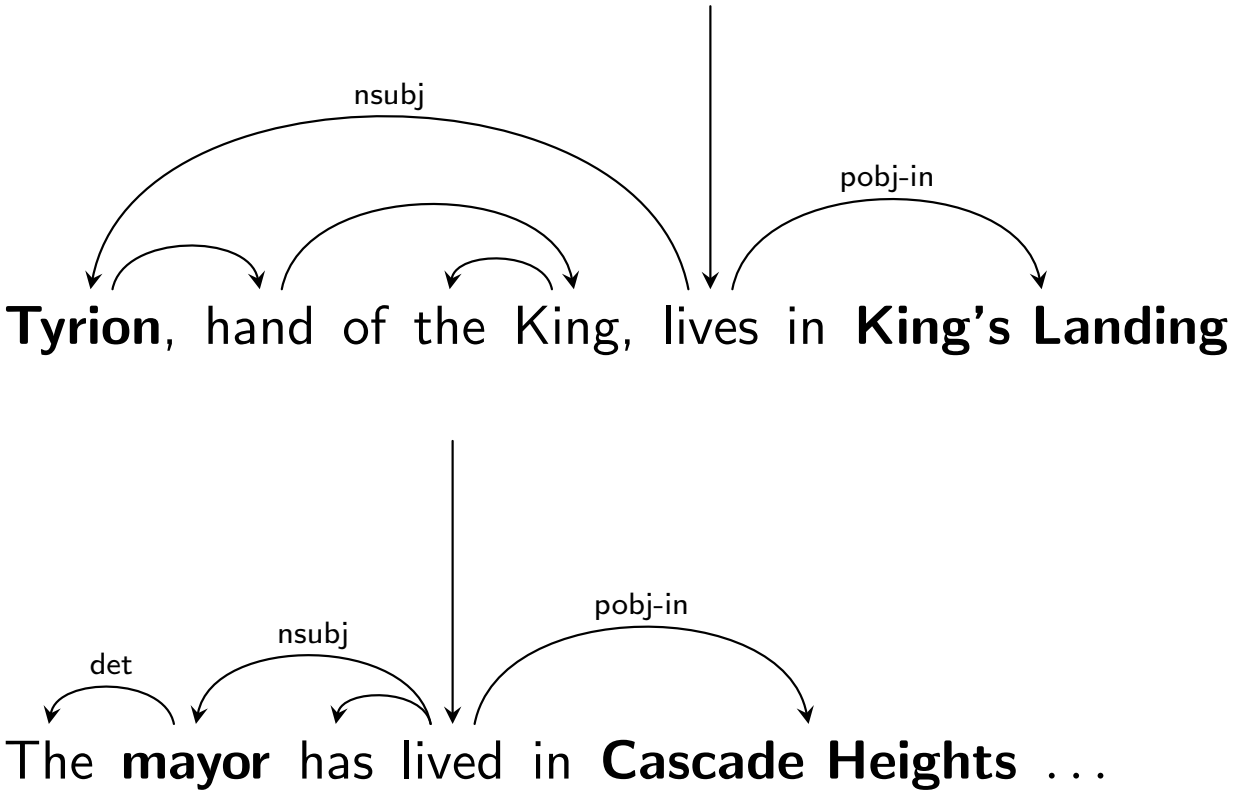
$$+ 1$$

Dependency kernel example



$$K(x_1, x_2) = 1 + 1 + 1$$

Dependency kernel example



$$
\begin{aligned}
K(x_1, x_2) = \; & 1 \\
& + 1 \\
& + 1 \\
& + 0 \\
= & 3
\end{aligned}
$$

Kernel-based classification

Binary classification rule, for $y_i \in \{-1, 1\}$

$$\hat{y}(\boldsymbol{x}) = \text{sign}(\sum_{i}^{N} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b)$$

Each $\alpha_i \geq 0$ is a parameter, which must be learned.

Kernel-based classification

Binary classification rule, for $y_i \in \{-1, 1\}$

$$\hat{y}(\boldsymbol{x}) = \text{sign}(\sum_{i}^{N} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b)$$

Each $\alpha_i \geq 0$ is a parameter, which must be learned.

$$\max_{\alpha} L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_j y_i y_j \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$s.t. \quad \alpha_i \geq 0, \forall i$$

$$\sum_i \alpha_i y_i = 0$$

Learning typically involves inverting the kernel matrix K.

Other training paradigms: bootstrapping

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

DIPRE (Brin, 1998)

- Relation of interest : (author, book)
- DIPRE's algorithm:
  - Given a small seed set of (author, book) pairs

1. Use the seed examples to label some data.
2. Induces patterns from the labeled data.
3. Apply the patterns to unlabeled data to get new set of (author,book) pairs, and add to the seed set.
4. Return to step 1, and iterate until convergence criteria is reached

Seed: (Arthur Conan Doyle, The Adventures of Sherlock Holmes)

A Web crawler finds all documents contain the pair.

...

Read The Adventures of Sherlock Holmes by Arthur Conan Doyle online or in you email

...

Extract **tuple**:

[0, Arthur Conan Doyle, The Adventures of Sherlock Holmes, Read, online or, by]

A tuple of 6 elements: [order, author, book, prefix, suffix, middle]

order = 1 if the author string occurs before the book string, = 0 otherwise

prefix and *suffix* are strings contain the 10 characters occurring to the left/right of the match

middle is the string occurring between the author and book

…

know that Sir Arthur Conan Doyle wrote The Adventures of Sherlock Holmes, in 1892

…

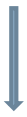
Extract tuple:

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, now that Sir, in 1892, wrote]

…

When Sir Arthur Conan Doyle wrote the adventures of Sherlock Holmes in 1892 he was high

…



Extract tuple:

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, When Sir, in 1892 he, wrote]

Extracted list of tuples:

[0, Arthur Conan Doyle, The Adventures of Sherlock Holmes, Read, online or, by]

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, now that Sir, in 1892, wrote]

[1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, When Sir, in 1892 he, wrote]

…

Group tuples by matching *order* and *middle* and induce patterns

Induce patterns from group of tuples:

[longest-common-suffix of prefix strings, author, middle, book, longest-common-prefix of suffix strings]

Pattern:

[Sir, Arthur Conan Doyle, wrote, The Adventures of Sherlock Holmes, in 1892]

Pattern with wild card expression:

[Sir, .*?, wrote, .*?, in 1892]

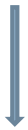
Use the wild card patterns **[Sir, .*?, wrote, .*?, in 1892]**

search the Web to find more documents

...

Sir Arthur Conan Doyle **wrote** Speckled Band **in 1892**, that is around 62 years apart which would make the stories

...



Extract new relations:

(Arthur Conan Doyle, Speckled Band)

Repeat the algorithm with the new relation.

Other training paradigms: distant supervision

Problems with bootstrapping (Mintz et al, 2009)

- [Steven Spielberg]'s film [Saving Private Ryan] is loosely based on the brother's story
  (Spielberg could be producer, actor)
- Allison co-produced the award-winning [Saving Private Ryan], directed by [Steven Spielberg]
  (Saving Private Ryan might not be a film)

Other training paradigms: distant supervision

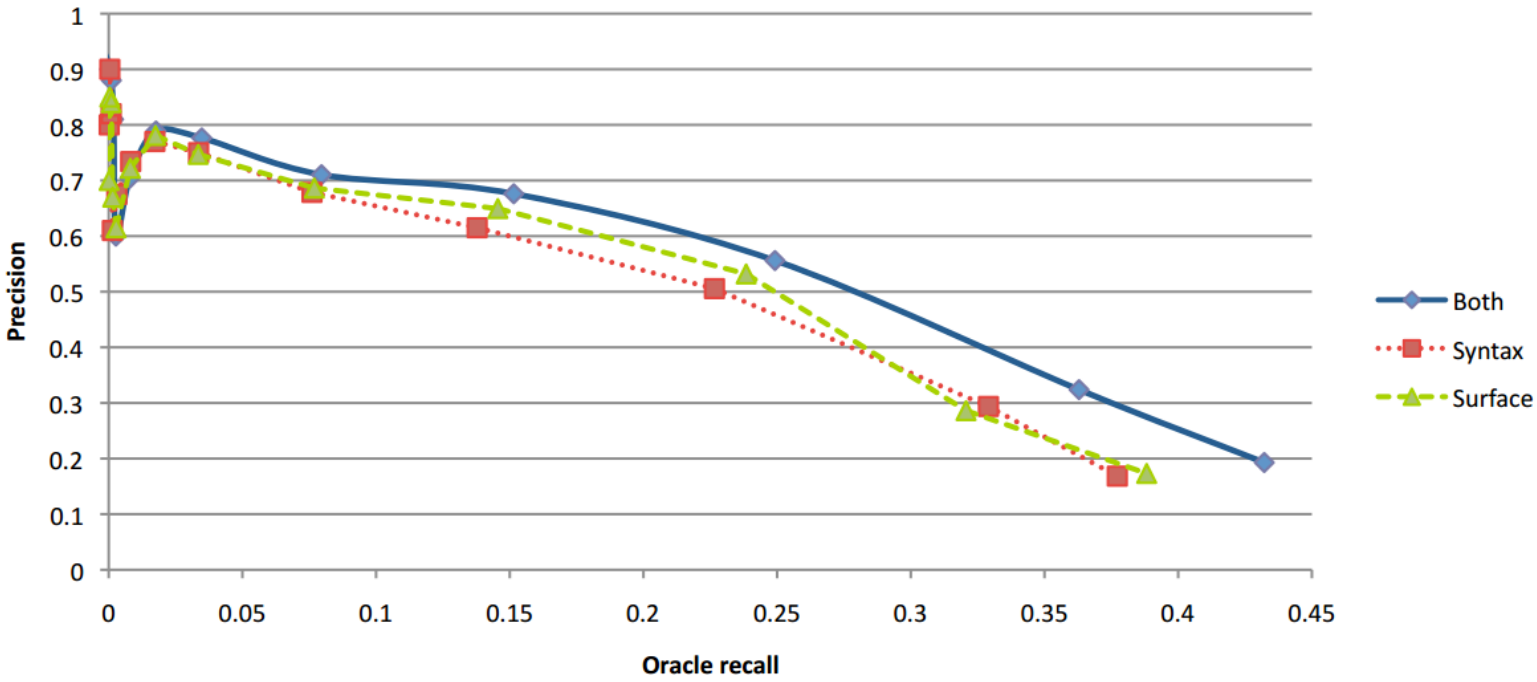
Problems with bootstrapping (Mintz et al, 2009)

- [Steven Spielberg]'s film [Saving Private Ryan] is loosely based on the brother's story
  (Spielberg could be producer, actor)
- Allison co-produced the award-winning [Saving Private Ryan], directed by [Steven Spielberg]
  (Saving Private Ryan might not be a film)

Distant supervision

- 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).

Distant supervision performance



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

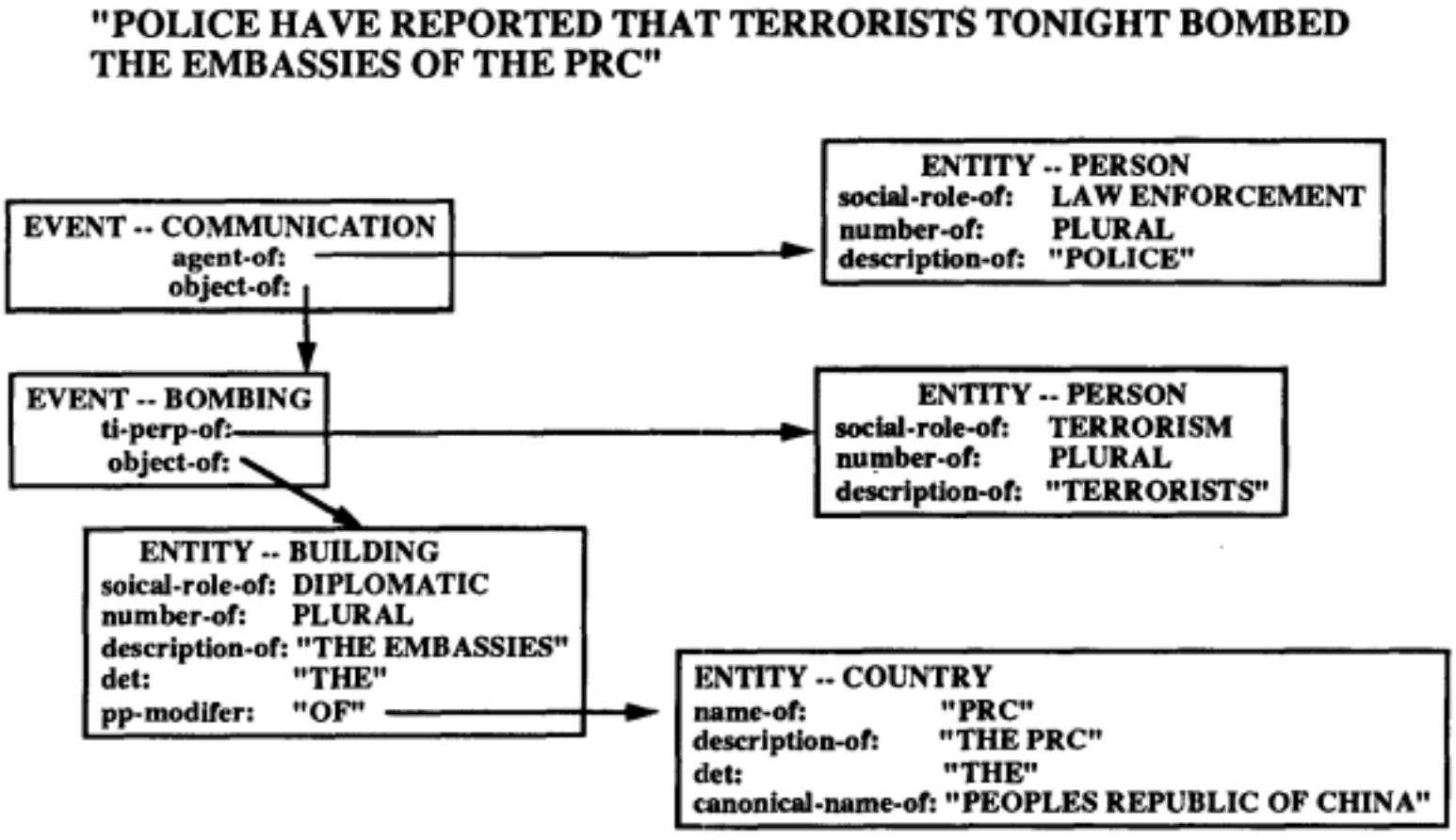
- Unstructured source: At a meeting of the Thirteen, Pyat Pree tells Daenerys he has her dragons in the House of the Undying.
- Annotated entities: At a meeting of <ORG>the Thirteen</ORG>, <PER>Pyat Pree</PER> tells <PER>Daenerys</PER> that he has <OBJ>her dragons</OBJ> in the <PER>House of the Undying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

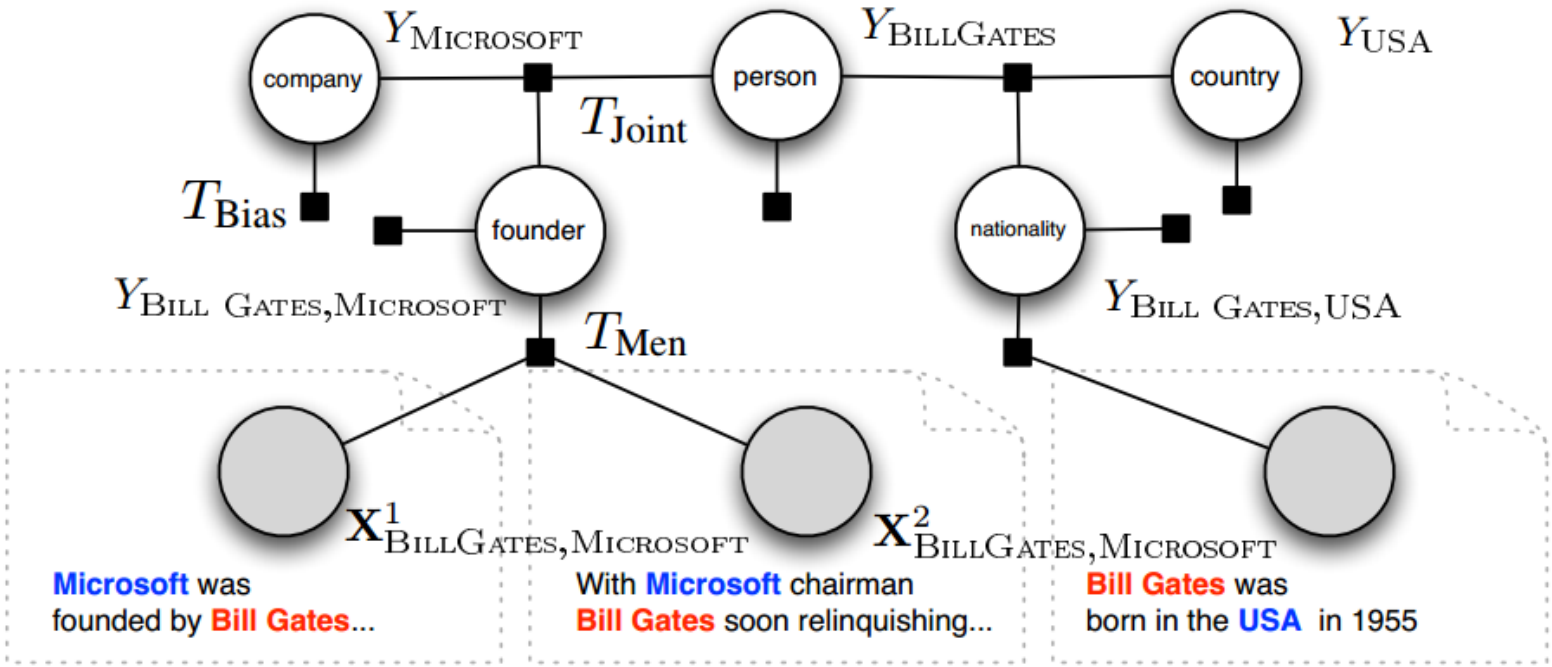
In supervised event extraction, each event type has a template of relevant attributes.

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

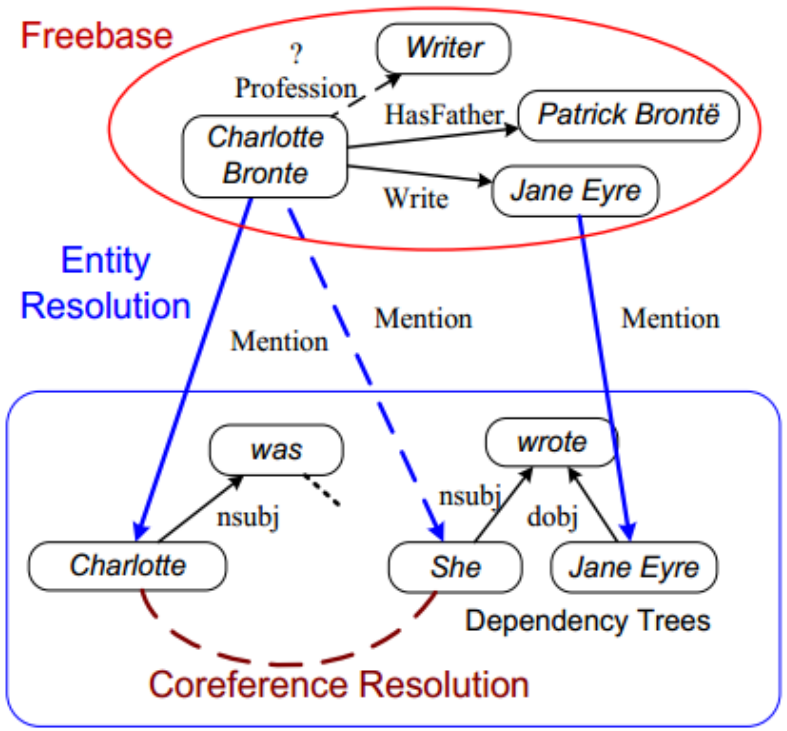
Joint reasoning about both language understanding and the underlying semantics.



(Yao, Riedel, and McCallum, 2010)

Collective relation extraction

Joint reasoning about both language understanding and the underlying semantics.



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

Next steps: Processes

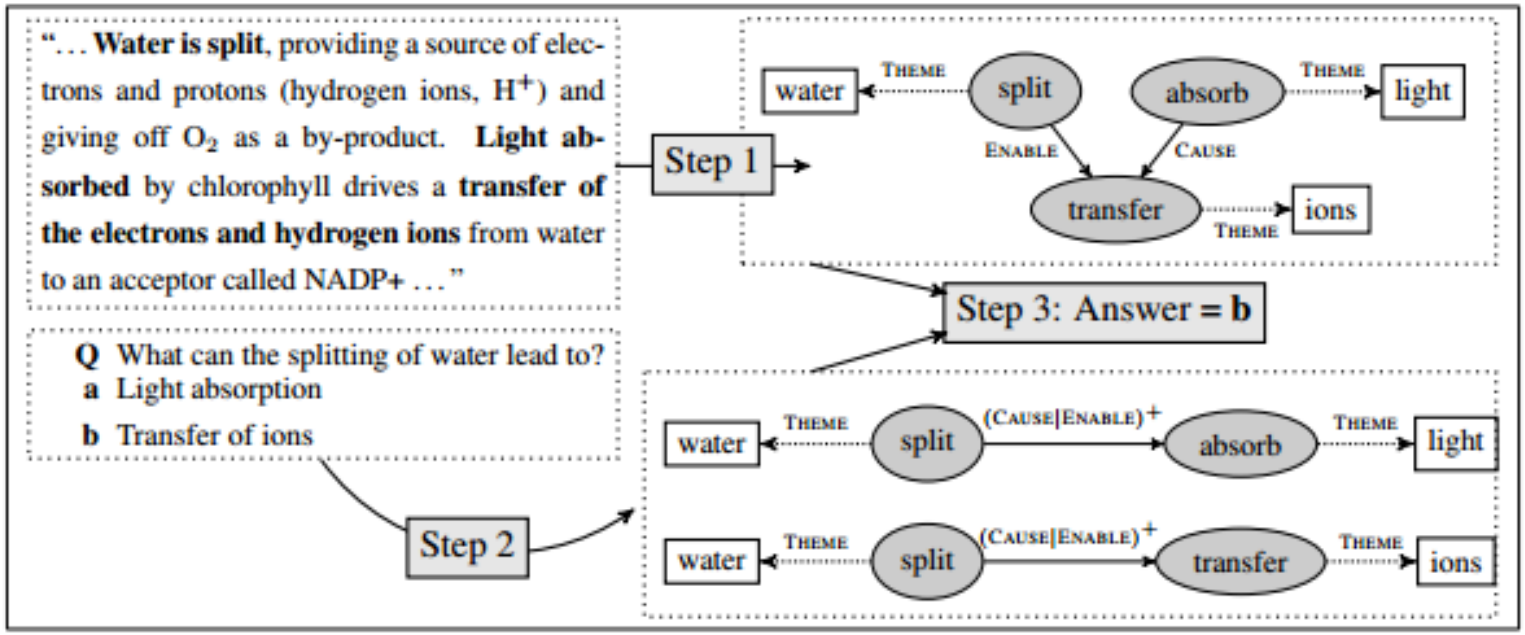


Figure 1: An overview of our reading comprehension system. First, we predict a structure from the input paragraph (the top right portion shows a partial structure skipping some arguments for brevity). Circles denote events, squares denote arguments, solid arrows represent event-event relations, and dashed arrows represent event-argument relations. Second, we map the question paired with each answer into a query that will be answered using the structure. The bottom right shows the query representation. Last, the two queries are executed against the structure, and a final answer is returned.

(Berant et al, 2014)

Processes

Berant et al (2014): a process is a directed graph, involving

Event triggers introduce events, e.g. split. They are nodes in the graph.

Arguments are entities that participate in events. They are nodes, connected to event triggers by edges labeled by semantic role.

Event-event relations are edges between event trigger nodes, including

- **cause**, **enable**, **prevent**, and their disjunctions and conjunctions
- **super**: one event is part of another

Process graph induction is formulated as an integer linear program.

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 factuality	Izvestiya **said** that the G-7 leaders **pretended** everything was OK in Russia's economy.
Epistemic marking	He saw the gunman, The editorialist speculated ...

FactBank

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

Table 1
FactBank annotation scheme. CT = certain; PR = probable; PS = possible; U = underspecified; + = positive; − = negative; u = unknown.

Value	Definition	Count
CT+	According to the source, it is certainly the case that X	7,749 (57.6%)
PR+	According to the source, it is probably the case that X	363 (2.7%)
PS+	According to the source, it is possibly the case that X	226 (1.7%)
CT−	According to the source, it is certainly not the case that X	433 (3.2%)
PR−	According to the source it is probably not the case that X	56 (0.4%)
PS−	According to the source it is possibly not the case that X	14 (0.1%)
CTu	The source knows whether it is the case that X or that not X	12 (0.1%)
Uu	The source does not know what the factual status of the event is, or does not commit to it	4,607 (34.2%)
		13,460

FactBank Annotations and Modeling

Magna International Inc.'s chief financial officer, James McAlpine, **resigned** and its chairman, Frank Stronach, is stepping in to help turn the automotive-parts manufacturer around, the company said.

Normalization: James McAlpine resigned

Annotations: CT+: 10

In the air, U.S. Air Force fliers say they have **engaged** in "a little cat and mouse" with Iraqi warplanes.

Normalization: U.S. Air Force fliers have engaged in "a little cat and mouse" with Iraqi warplanes

Annotations: CT+: 9, PS+: 1

(de Marneffe, Manning, and Potts 2012)

FactBank Annotations and Modeling

(a) If the heavy outflows **continue**, fund managers will face increasing pressure to sell off some of their junk to pay departing investors in the weeks ahead.

Normalization: the heavy outflows will continue

Annotations: Uu: 7, PS+: 2, CT+: 1

(b) A unit of DPC Acquisition Partners said it would seek to liquidate the computer-printer maker "as soon as possible," even if a merger isn't **consummated**.

Normalization: a merger will be consummated

Annotations: Uu: 8, PS+: 2

(de Marneffe, Manning, and Potts 2012)