• Syntactic NLP news today --
  new release of “universal dependencies” for multiple languages
  http://universaldependencies.github.io/docs/
Logistics

- Two more homeworks
  - Tomorrow: HW4 out, on coref. Due in 2 weeks
  - Later: a short HW5
Do within-document coreference in the following document by assigning the mentions entity numbers:

[The government]___ said [today]___ [it]___ ’s going to cut back on [[[the enormous number]___ of [people]___]___ who descended on [Yemen]___ to investigate [[the attack]___ on [the “ USS Cole]___]___]. ” [[[So many people]___ from [several agencies]___]___]___ wanting to participate that [the Yemenis]___ are feeling somewhat overwhelmed in [[their]___ own country]___.

[Investigators]___ have come up with [[another theory]___ on how [the terrorists]___ operated]___.

[[ABC ’s]___ John Miller]___ on [[the house]___ with [a view]___]___

High on [[a hillside]___, in [[a run - down section]___ of [Aden]___]___, [[the house]___ with [the blue door]___]___ has [[a perfect view]___ of [the harbor]___].

[American and Yemeni investigators]___ believe [that view]___ is what convinced [[a man]___ who used [[the name]___ [Abdullah]___]___]___ to rent [the house]___ [[several weeks]___ before [[the bombing]___ of [the “ USS Cole]___]]. ”

Early
• 1. Within-document coreference
• 2. Cross-document coreference
Kinds of Reference

- Referring expressions
  - *John Smith*
  - *President Smith*
  - *the president*
  - *the company’s new executive*

- Free variables
  - Smith saw *his pay* increase

- Bound variables
  - The dancer hurt *herself.*
• Types of coref subtasks
  • 1. Pronoun resolution (anaphora resolution)
  • 2. Common nouns and names

• Typical pipeline
  • 1. Identify candidate mentions
    (ideally, referential mentions: exclude times, etc)
  • 2. Cluster the candidate mentions
Syntactic vs Semantic cues

• State-of-the-art coref uses first two
Syntactic vs Semantic cues

• Syntactic cues
  • [John], a [lawyer], bought [himself] a book.
  • [John], a [lawyer], bought [him] a book.

• State-of-the-art coref uses first two
Syntactic vs Semantic cues

• Syntactic cues
  • [John], a [lawyer], bought [himself] a book.
  • [John], a [lawyer], bought [him] a book.

• Shallow semantic cues
  • John saw Mary. She was eating salad.
  • John saw Mary. He was eating salad.

• State-of-the-art coref uses first two
Syntactic vs Semantic cues

• Syntactic cues
  • [John], a [lawyer], bought [himsel] a book.
  • [John], a [lawyer], bought [him] a book.

• Shallow semantic cues
  • John saw Mary. She was eating salad.
  • John saw Mary. He was eating salad.

• Deeper semantics (world knowledge)
  • The city council denied the demonstrators a permit because they feared violence.
  • The city council denied the demonstrators a permit because they advocated violence.

• State-of-the-art coref uses first two
Hary Potter was a wizard. Lord Voldemort attempted to murder him.

- View gold standard as defining links between mention pairs
- Think of as binary classification problem: take random pairs as negative examples
- Issues: many mention pairs. Also: have to resolve local decisions into entities
Antecedent selection model

- View as antecedent selection problem: which previous mention do I corefer with?
- Makes most sense for pronouns, though can use model for all expressions
- Process mentions left to right. For the \( n \)'th mention, \( n \)-way multi-class classification problem: antecedent is one of the \( n-1 \) mentions to the left, or NULL.
- Features are asymmetric!
- Use a limited window for antecedent candidates e.g. last 5 sentences (for news...)
- Score each candidate by a linear function of features. Predict antecedent to be the highest-ranking candidate.

Hary Potter was a wizard. Lord Voldemort attempted to murder him.
Antecedent selection model

- Prediction: select the highest-scoring candidate as the antecedent. (Though multiple may be ok.)
- Using for applications: take these links and form entity clusters from connected components [whiteboard]
- Training: simple way is to process the gold standard coref chains (entity clusters) into positive and negative links. Train binary classifier.

Hary Potter was a wizard. Lord Voldemort attempted to murder him.
Features for pronoun resolution
Features for pronoun resolution

- English pronouns grammar/semantic matching. Use as features against antecedent candidate properties.
Features for pronoun resolution

• English pronouns grammar/semantic matching. Use as features against antecedent candidate properties.

• Number agreement
  • he/she/it vs. they/them
  • MATCH TO: singular/plural nouns ("person", "people")
Features for pronoun resolution

- English pronouns grammar/semantic matching. Use as features against antecedent candidate properties.
- Number agreement
  - he/she/it vs. they/them
  - MATCH TO: singular/plural nouns ("person", "people")
- Animacy/human-ness agreement
  - it vs. he/she/him/her/his
  - MATCH TO: name-or-not vs. "person" vs. "car"
    (need lexical semantic DB: e.g. wordnet?)
Features for pronoun resolution

- English pronouns grammar/semantic matching. Use as features against antecedent candidate properties.
- Number agreement
  - he/she/it vs. they/them
  - MATCH TO: singular/plural nouns (“person”, “people”)
- Animacy/human-ness agreement
  - it vs. he/she/him/her/his
  - MATCH TO: name-or-not vs. “person” vs. “car”
    (need lexical semantic DB: e.g. wordnet?)
- Gender agreement
  - he/him/his vs. she/her vs. it ---- MATCH TO: name gender?
  - MATCH TO: gender of names, common nouns
Features for pronoun resolution
Features for pronoun resolution

- Grammatical person - interacts with dialogue/discourse structure
  - first person: I/me
  - second person: you/y’all
  - third person: he/she/it/they
Features for pronoun resolution

- Grammatical person - interacts with dialogue/discourse structure
  - first person: I/me
  - second person: you/y’all
  - third person: he/she/it/they
- Reflexives: bind to close subject (usually forbidden)
  - John knew that Bob bought him a book.
  - Bob knew that John bought himself a book.
Other syntactic constraints

• High-precision patterns
  • Predicate-Nominatives: “X was a Y …”
  • Appositives: “X, a Y, …”
  • Role Appositives: “[president] [Lincoln]”

• Maybe you’re happy with a high-precision, low-recall system?
Structural features for pronoun resolution

• Preferences:
  – Recency: More recently mentioned entities are more likely to be referred to
    • John went to a movie. Jack went as well. He was not busy.
  – Grammatical Role: Entities in the subject position is more likely to be referred to than entities in the object position
    • John went to a movie with Jack. He was not busy.
  – Parallelism:
    • John went with Jack to a movie. Joe went with him to a bar.
Structural features for pronoun resolution

• Preferences:
  – Verb Semantics: Certain verbs seem to bias whether the subsequent pronouns should be referring to their subjects or objects
    • John telephoned Bill. He lost the laptop.
    • John criticized Bill. He lost the laptop.
  – Selectional Restrictions: Restrictions because of semantics
    • John parked his car in the garage after driving it around for hours.

• Encode all these and maybe more as features
• How to combine information

• Features in supervised ML -- easiest to do, if you have training data  
  [Berkeley Coref -- Durrett and Klein]

• Rule-based approach.  [Stanford DCoref, Lee et al.]  
  Typically, use a priority ordering:
  • Go through each high-precision rule. If it fires: take it. Done.
  • Else: filter out mentions based on semantic agreement and forbidden syntactic configurations. Choose syntactically closest mention.

• Other multistage approaches  e.g. Bamman et al’s book-nlp:
  • 1. Cluster names based on string match / similarity
  • 2. Resolve pronouns with antecedent model
Features for non-pronoun resolution

• String match ... substring match ... edit distance
  • “Abraham Lincoln” ... “President Lincoln”
  • “Bill Clinton” ... “Hillary Clinton” ... “Clinton” ... “Mr. Clinton”
  • special-case name parsing (firstname vs surname)?

• Head string match
  • I saw a green house. The house was old.

• Many harder cases
  • “Bill” ... “the boy”
  • “Novartis” ... “the company”
Within-doc coref performance

- Have to evaluate: how well do system’s predicted clusters match gold-standard clusters?
- Current systems get 70-80ish % accuracy depending on genre and how you view this
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Features</th>
</tr>
</thead>
</table>

Friday, November 20, 15
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record linkage</td>
<td></td>
</tr>
<tr>
<td>DB1 of entities &lt;=&gt; DB2 of entities</td>
<td></td>
</tr>
<tr>
<td>e.g. Match voter records against Facebook profiles (Bond et al.)</td>
<td></td>
</tr>
</tbody>
</table>
DB/Cross-doc coref

Tasks

• Record linkage
  • DB1 of entities <=> DB2 of entities
  • e.g. Match voter records against Facebook profiles (Bond et al.)

• Entity Linking
  • DB Entities <=> mentions in corpus

Features
DB/Cross-doc coref

Tasks

• Record linkage
  • DB1 of entities <=> DB2 of entities
  • e.g. Match voter records against Facebook profiles (Bond et al.)

• Entity Linking
  • DB Entities <=> mentions in corpus

• Cross-doc coref
  • Discover the entities: like within-doc coref.
    (Building your own entity DB)
  • Clustering problem across all mentions in all docs!
DB/Cross-doc coref

Tasks

• Record linkage
  • DB1 of entities <=> DB2 of entities
  • e.g. Match voter records against Facebook profiles (Bond et al.)

• Entity Linking
  • DB Entities <=> mentions in corpus

• Cross-doc coref
  • Discover the entities: like within-doc coref.
    (Building your own entity DB)
  • Clustering problem across all mentions in all docs!

Features

• Name matching is really important
DB/Cross-doc coref

**Tasks**

- Record linkage
  - DB1 of entities $\leftrightarrow$ DB2 of entities
  - e.g. Match voter records against Facebook profiles (Bond et al.)
- Entity Linking
  - DB Entities $\leftrightarrow$ mentions in corpus
- Cross-doc coref
  - Discover the entities: like within-doc coref.
    (Building your own entity DB)
  - Clustering problem across all mentions in all docs!

**Features**

- Name matching is really important
- Fuzzy matching for e.g. middle initials, multiple surnames (token level?)
  e.g. transliterations: Qaddafî, Gaddafî, el-Qaddafî (character level)
DB/Cross-doc coref

Tasks

• Record linkage
  • DB1 of entities <=> DB2 of entities
  • e.g. Match voter records against Facebook profiles (Bond et al.)

• Entity Linking
  • DB Entities <=> mentions in corpus

• Cross-doc coref
  • Discover the entities: like within-doc coref.
    (Building your own entity DB)
  • Clustering problem across all mentions in all docs!

Features

• Name matching is really important

• Fuzzy matching for e.g. middle initials, multiple surnames (token level?)
  e.g. transliterations: Qaddafi, Gaddafi, el-Qaddafi (character level)

• Jaro-Winkler edit distance: especially customized for names (at least, names typical for the U.S. Census)
DB/Cross-doc coref

Tasks

• Record linkage
  • DB1 of entities <=> DB2 of entities
  • e.g. Match voter records against Facebook profiles (Bond et al.)

• Entity Linking
  • DB Entities <=> mentions in corpus

• Cross-doc coref
  • Discover the entities: like within-doc coref. (Building your own entity DB)
  • Clustering problem across all mentions in all docs!

Features

• Name matching is really important
  • Fuzzy matching for e.g. middle initials, multiple surnames (token level?)
    e.g. transliterations: Qaddafí, Gaddafí, el-Qaddafí (character level)

• Jaro-Winkler edit distance: especially customized for names (at least, names typical for the U.S. Census)
  • TF-IDF weighting
DB/Cross-doc coref

**Tasks**

- Record linkage
  - DB1 of entities $\leftrightarrow$ DB2 of entities
  - e.g. Match voter records against Facebook profiles (Bond et al.)
- Entity Linking
  - DB Entities $\leftrightarrow$ mentions in corpus
- Cross-doc coref
  - Discover the entities: like within-doc coref. (Building your own entity DB)
  - Clustering problem across all mentions in all docs!

**Features**

- Name matching is really important
- Fuzzy matching for e.g. middle initials, multiple surnames (token level?)
  e.g. transliterations: Qaddafi, Gaddafi, el-Qaddafi (character level)
- Jaro-Winkler edit distance: especially customized for names (at least, names typical for the U.S. Census)
- TF-IDF weighting
- Context e.g. bag-of-words near the mention