Lecture 14: Coreference Resolution

Kevin Clark
Announcements

• Assignment 3 due today at 11:59pm

• Final project milestone due tomorrow
  • ALL teams submit to Gradescope

• Project mentoring:
  • If you haven’t received an email, Richard is your mentor
  • Go to his OH to discuss your project!
Lecture Plan:

• What is Coreference Resolution?

• Mention Detection

• Some Linguistics: Types of Reference

• 3 Kinds of Coreference Resolution Models
  • Including the current state-of-the-art coreference system!
What is Coreference Resolution?

• Identify all mentions that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.
What is Coreference Resolution?

• Identify all **mentions** that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.
What is Coreference Resolution?

- Identify all mentions that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.
What is Coreference Resolution?

- Identify all mentions that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience.
What is Coreference Resolution?

- Identify all mentions that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.
Barack Obama nominated **Hillary Rodham Clinton** as his **secretary of state** on Monday. He chose her because she had foreign affairs experience as a former **First Lady**.
Applications

• Full text understanding
  • information extraction, question answering, summarization, ...
  • “He was born in 1961”
Applications

- Full text understanding
- Machine translation
  - languages have different features for gender, number, dropped pronouns, etc.
Applications

- Full text understanding
- Machine translation
  - languages have different features for gender, number, dropped pronouns, etc.
Applications

• Full text understanding
• Machine translation
• Dialogue Systems

“Book tickets to see James Bond”
“Spectre is playing near you at 2:00 and 3:00 today. How many tickets would you like?”
“Two tickets for the showing at three”
Coreference Resolution is Really Difficult!

• “She poured water from the pitcher into the cup until it was full”

• Requires reasoning /world knowledge to solve
Coreference Resolution is Really Difficult!

- “She poured water from the pitcher into the cup until it was full”
- “She poured water from the pitcher into the cup until it was empty”
- Requires reasoning /world knowledge to solve
Coreference Resolution is Really Difficult!

- “She poured water from the pitcher into the cup until it was full”
- “She poured water from the pitcher into the cup until it was empty”

- The trophy would not fit in the suitcase because it was too big.
- The trophy would not fit in the suitcase because it was too small.

- These are called **Winograd Schema**
Coreference Resolution is Really Difficult!

- “She poured water from the pitcher into the cup until it was full”
- “She poured water from the pitcher into the cup until it was empty”
- The trophy would not fit in the suitcase because it was too big.
- The trophy would not fit in the suitcase because it was too small.

- These are called **Winograd Schema**
  - Recently proposed as an alternative to the Turing test
    - Turing test: how can we tell if we’ve built an AI system? A human can’t distinguish it from a human when chatting with it.
    - But requires a person, people are easily fooled
  - If you’ve fully solved coreference, arguably you’ve solved AI
Coreference Resolution in Two Steps

1. Detect the mentions (easy)
   “[I] voted for [Nader] because [he] was most aligned with [[my] values],” [she] said
   - mentions can be nested!

2. Cluster the mentions (hard)
   “[I] voted for [Nader] because [he] was most aligned with [[my] values],” [she] said
Mention Detection

- Mention: span of text referring to some entity
- Three kinds of mentions:

1. Pronouns
   - I, your, it, she, him, etc.

2. Named entities
   - People, places, etc.

3. Noun phrases
   - “a dog,” “the big fluffy cat stuck in the tree”
Mention Detection

• Span of text referring to some entity
• For detection: use other NLP systems

1. Pronouns
   • Use a part-of-speech tagger

2. Named entities
   • Use a NER system (like hw3)

3. Noun phrases
   • Use a constituency parser (next lecture!)
Mention Detection: Not so Simple

- Marking all pronouns, named entities, and NPs as mentions over-generates mentions
- Are these mentions?
  - It is sunny
Mention Detection: Not so Simple

- Marking all pronouns, named entities, and NPs as mentions over-generates mentions
- Are these mentions?
  - It is sunny
  - Every student
Mention Detection: Not so Simple

• Marking all pronouns, named entities, and NPs as mentions over-generates mentions

• Are these mentions?
  • It is sunny
  • Every student
  • No student
Mention Detection: Not so Simple

- Marking all pronouns, named entities, and NPs as mentions over-generates mentions
- Are these mentions?
  - It is sunny
  - Every student
  - No student
  - The best donut in the world
Mention Detection: Not so Simple

- Marking all pronouns, named entities, and NPs as mentions over-generates mentions
- Are these mentions?
  - *It* is sunny
  - *Every* student
  - *No* student
  - *The best donut in the world*
  - *100* miles
Mention Detection: Not so Simple

- Marking all pronouns, named entities, and NPs as mentions over-generates mentions

- Are these mentions?
  - It is sunny
  - Every student
  - No student
  - The best donut in the world
  - 100 miles

- Some gray area in defining “mention”: have to pick a convention and go with it
How to deal with these bad mentions?

• Could train a classifier to filter out spurious mentions

• Much more common: keep all mentions as “candidate mentions”
  • After your coreference system is done running discard all singleton mentions (i.e., ones that have not been marked as coreference with anything else)
Can we avoid a pipelined system?

- We could instead train a classifier specifically for mention detection instead of using a POS tagger, NER system, and parser.

- Or even jointly do mention-detection and coreference resolution end-to-end instead of in two steps
  - Will cover later in this lecture!
On to Coreference! First, some linguistics

- **Coreference** is when two mentions refer to the same entity in the world
  - *Barack Obama traveled to ... Obama*

- Another kind of reference is **anaphora**: when a term (anaphor) refers to another term (antecedent) and the interpretation of the anaphor is in some way determined by the interpretation of the antecedent
  - *Barack Obama said he would sign the bill.*
    antecedent       anaphor
Anaphora vs Coreference

• Coreference with named entities
  text
  world

  Barack Obama

  Obama

• Anaphora
  text
  world

  Barack Obama
  he
Anaphora vs. Coreference

• Not all anaphoric relations are coreferential

We went to see a concert last night. The tickets were really expensive.

• This is referred to as bridging anaphora.
Cataphora

• Usually the antecedent comes before the anaphor (e.g., a pronoun), but not always
“From the corner of the divan of Persian saddle-bags on which he was lying, smoking, as was his custom, innumerable cigarettes, Lord Henry Wotton could just catch the gleam of the honey-sweet and honey-coloured blossoms of a laburnum...”

(Oscar Wilde – The Picture of Dorian Gray)
Cataphora

“From the corner of the divan of Persian saddle-bags on which he was lying, smoking, as was his custom, innumerable cigarettes, Lord Henry Wotton could just catch the gleam of the honey-sweet and honey-coloured blossoms of a laburnum...”

(Oscar Wilde)
Next Up: Three Kinds of Coreference Models

- Mention Pair
- Mention Ranking
- Clustering
“I voted for Nader because he was most aligned with my values,” she said.
Coreference Models: Mention Pair

- Train a binary classifier that assigns every pair of mentions a probability of being coreferent: $p(m_i, m_j)$

- e.g., for “she” look at all candidate antecedents (previously occurring mentions) and decide which are coreferent with it

"I voted for Nader because he was most aligned with my values," she said.

coreferent with she?
Coreference Models: Mention Pair

- Train a binary classifier that assigns every pair of mentions a probability of being coreferent: \( p(m_i, m_j) \)
  
  - e.g., for “she” look at all candidate antecedents (previously occurring mentions) and decide which are coreferent with it

“I voted for Nader because he was most aligned with my values,” she said.

Positive examples: want \( p(m_i, m_j) \) to be near 1
Coreference Models: Mention Pair

- Train a binary classifier that assigns every pair of mentions a probability of being coreferent: \( p(m_i, m_j) \)
  - e.g., for “she” look at all candidate antecedents (previously occurring mentions) and decide which are coreferent with it

“I voted for Nader because he was most aligned with my values,” she said.

Negative examples: want \( p(m_i, m_j) \) to be near 0
Mention Pair Training

- $N$ mentions in a document
- $y_{ij} = 1$ if mentions $m_i$ and $m_j$ are coreferent, -1 if otherwise
- Just train with regular cross-entropy loss (looks a bit different because it is binary classification)

$$J = - \sum_{i=2}^{N} \sum_{j=1}^{i} y_{ij} \log p(m_j, m_i)$$

Iterate through mentions
Iterate through candidate antecedents (previously occurring mentions)
Coreferent mentions pairs should get high probability, others should get low probability
Mention Pair Test Time

- Coreference resolution is a clustering task, but we are only scoring pairs of mentions... what to do?
Coreference resolution is a clustering task, but we are only scoring pairs of mentions... what to do?

Pick some threshold (e.g., 0.5) and add coreference links between mention pairs where $p(m_i, m_j)$ is above the threshold.

“I voted for Nader because he was most aligned with my values,” she said.
Mention Pair Test Time

- Coreference resolution is a clustering task, but we are only scoring pairs of mentions... what to do?
- Pick some threshold (e.g., 0.5) and add coreference links between mention pairs where $p(m_i, m_j)$ is above the threshold
- Take the transitive closure to get the clustering

“I voted for Nader because he was most aligned with my values,” she said.

Even though the model did not predict this coreference link, I and my are coreferent due to transitivity
Mention Pair Test Time

- Coreference resolution is a clustering task, but we are only scoring pairs of mentions... what to do?
- Pick some threshold (e.g., 0.5) and add coreference links between mention pairs where $p(m_i, m_j)$ is above the threshold
- Take the transitive closure to get the clustering

“I voted for Nader because he was most aligned with my values,” she said.

Adding this extra link would merge everything into one big coreference cluster!
Mention Pair Models: Disadvantage

• Suppose we have a long document with the following mentions
  • Ralph Nader ... he ... his ... him ... <several paragraphs>
  ... voted for Nader because he ...

Ralph Nader → he → his → him → Nader → he

almost impossible

Relatively easy
Mention Pair Models: Disadvantage

- Suppose we have a long document with the following mentions
  - **Ralph Nader** ... *he* ... *his* ... *him* ... <several paragraphs>
    ... *voted for Nader because he* ...

- Many mentions only have one clear antecedent
  - But we are asking the model to predict all of them
- Solution: instead train the model to predict only one antecedent for each mention
  - More linguistically plausible
Coreference Models: Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything

best antecedent for *she*?
Coreference Models: Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything

Positive examples: model has to assign a high probability to either one (but not necessarily both)
Coreference Models: Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything

```
p(NA, she) = 0.1
p(I, she) = 0.5
p(Nader, she) = 0.1
p(he, she) = 0.1
p(my, she) = 0.2
```

Apply a softmax over the scores for candidate antecedents so probabilities sum to 1

best antecedent for she?
Coreference Models: Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model.
- Dummy NA mention allows model to decline linking the current mention to anything.

Apply a softmax over the scores for candidate antecedents so probabilities sum to 1.

\[
\begin{align*}
  p(NA, she) &= 0.1 \\
  p(I, she) &= 0.5 \\
  p(Nader, she) &= 0.1 \\
  p(he, she) &= 0.1 \\
  p(my, she) &= 0.2
\end{align*}
\]
Coreference Models: Training

- We want the current mention $m_j$ to be linked to any one of the candidate antecedents it’s coreferent with.
- Mathematically, we want to maximize this probability:

$$
\sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)p(m_j, m_i)
$$

• Iterate through candidate antecedents (previously occurring mentions)
• For ones that are coreferent to $m_j$...
• ...we want the model to assign a high probability
Coreference Models: Training

- We want the current mention $m_j$ to be linked to *any one* of the candidate antecedents it’s coreferent with.
- Mathematically, we want to maximize this probability:

$$\sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)p(m_j, m_i)$$

Iterate through candidate antecedents (previously occurring mentions)

For ones that are coreferent to $m_j$...

...we want the model to assign a high probability

- The model could produce 0.9 probability for one of the correct antecedents and low probability for everything else, and the sum will still be large
Coreference Models: Training

- We want the current mention $m_j$ to be linked to any one of the candidate antecedents it’s coreferent with.
- Mathematically, we want to maximize this probability:

$$
J = \sum_{i=2}^{N} - \log \left( \sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)p(m_j, m_i) \right)
$$

- Turning this into a loss function:

$$
J = \sum_{i=2}^{N} - \log \left( \sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)p(m_j, m_i) \right)
$$

Iterate over all the mentions in the document

Usual trick of taking negative log to go from likelihood to loss
Mention Ranking Models: Test Time

- Pretty much the same as mention-pair model except each mention is assigned only one antecedent
Mention Ranking Models: Test Time

- Pretty much the same as mention-pair model except each mention is assigned only one antecedent
How do we compute the probabilities?

1. Non-neural statistical classifier

2. Simple neural network

3. More advanced model using LSTMs, attention
1. Non-Neural Coref Model: Features

- Person/Number/Gender agreement
  - Jack gave Mary a gift. She was excited.
- Semantic compatibility
  - ... the mining conglomerate ... the company ...
- Certain syntactic constraints
  - John bought him a new car. [him can not be John]
- More recently mentioned entities preferred for referenced
  - John went to a movie. Jack went as well. He was not busy.
- Grammatical Role: Prefer entities in the subject 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.
2. Neural Coref Model

- Standard feed-forward neural network
  - Input layer: word embeddings and a few categorical features
2. Neural Coref Model: Inputs

- Embeddings
  - Previous two words, first word, last word, head word, ... of each mention
    - The **head** word is the “most important” word in the mention – you can find it using a parser. e.g., *The fluffy cat stuck in the tree*

- Still need some other features:
  - Distance
  - Document genre
  - Speaker information
3. End-to-end Model

- Current state-of-the-art model for coreference resolution (Lee et al., EMNLP 2017)
- Mention ranking model
- Improvements over simple feed—forward NN
  - Use an LSTM
  - Use attention
  - Do mention detection and coreference end-to-end
    - No mention detection step!
    - Instead consider every span of text (up to a certain length) as a candidate mention
      - a span is just a contiguous sequence of words
3. End-to-end Model

- First embed the words in the document using a word embedding matrix and a character-level CNN
3. End-to-end Model

- Then run a bidirectional LSTM over the document.

---

**Bidirectional LSTM** $(x^*)$

**Word & character embedding** $(x)$

- General
- Electric
- said
- the
- Postal
- Service
- contacted
- the
- company
3. End-to-end Model

- Next, represent each span of text $i$ going from START($i$) to END($i$) as a vector.
3. End-to-end Model

- Next, represent each span of text \(i\) going from \(\text{START}(i)\) to \(\text{END}(i)\) as a vector

- **General, General Electric, General Electric said, ... Electric, Electric said, ...** will all get its own vector representation
3. End-to-end Model

- Next, represent each span of text \( i \) going from START\( (i) \) to END\( (i) \) as a vector.

Span representation \( g \)

Span head \( \hat{x} \)

Bidirectional LSTM \( x^* \)

Word & character embedding \( x \)

General Electric said the Postal Service contacted the company

Span representation: \( g_i = [x^*_{\text{START}(i)}, x^*_{\text{END}(i)}, \hat{x}_i, \phi(i)] \)
3. End-to-end Model

- Next, represent each span of text \( i \) going from \( \text{START}(i) \) to \( \text{END}(i) \) as a vector. For example, for “the postal service”

Span representation \( (g) \)

Span head \( (x) \)

Bidirectional LSTM \( (x^*) \)

Word & character embedding \( (x) \)

General Electric said the Postal Service contacted the company

Span representation: \( g_i = [x^*_{\text{START}(i)}, x^*_{\text{END}(i)}, \hat{x}_i, \phi(i)] \)
3. End-to-end Model

- Next, represent each span of text $i$ going from $\text{START}(i)$ to $\text{END}(i)$ as a vector. For example, for "the postal service"

Span representation ($\mathbf{g}$)

Span head ($\hat{x}$)

Bidirectional LSTM ($\hat{x}^*$)

Word & character embedding ($x$)

General  Electric  said  the  Postal  Service  contacted  the  company

Span representation: $\mathbf{g}_i = [x^*_{\text{START}(i)}, x^*_{\text{END}(i)}, \hat{x}_i, \phi(i)]$

BILSTM hidden states for span’s start and end
3. End-to-end Model

- Next, represent each span of text $i$ going from $\text{START}(i)$ to $\text{END}(i)$ as a vector. For example, for “the postal service”

Span representation ($g$)

Span head ($\hat{x}$)

Bidirectional LSTM ($x^*$)

Word & character embedding ($x$)

General Electric said the Postal Service contacted the company

Span representation: $g_i = [x^*_\text{START}(i), x^*_\text{END}(i), \hat{x}_i, \phi(i)]$

BILSTM hidden states for span’s start and end

Attention-based representation (details next slide) of the words in the span
3. End-to-end Model

- Next, represent each span of text \( i \) going from \( \text{START}(i) \) to \( \text{END}(i) \) as a vector. For example, for “the postal service”

Span representation \((g)\)

Bidirectional LSTM \((x^*)\)

Word & character embedding \((x)\)

Span head \((\hat{x})\)

Span representation: \( g_i = [x_{\text{START}(i)}^*, x_{\text{END}(i)}^*, \hat{x}_i, \phi(i)] \)

- BILSTM hidden states for span’s start and end
- Attention-based representation (details next slide) of the words in the span
- Additional features
3. End-to-end Model

- \( \hat{x}_i \) is an attention-weighted average of the word embeddings in the span.

Span representation (\( g \))

Span head (\( \bar{x} \))

Bidirectional LSTM (\( x^* \))

Word & character embedding (\( x \))

General | Electric | said | the | Postal | Service | contacted | the | company

Attention scores

\[ \alpha_t = w_\alpha \cdot \text{FFNN}_\alpha(x^*_t) \]

dot product of weight vector and transformed hidden state
3. End-to-end Model

- \( \hat{x}_i \) is an attention-weighted average of the word embeddings in the span

\[
\alpha_t = \boldsymbol{w}_\alpha \cdot \text{FFNN}_\alpha(x^*_t)
\]

dot product of weight vector and transformed hidden state

\[
a_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)}
\]

just a softmax over attention scores for the span
3. End-to-end Model

- $\hat{\mathbf{x}}_i$ is an attention-weighted average of the word embeddings in the span

Span representation ($g$)

Span head ($\hat{x}$)

Bidirectional LSTM ($x^*$)

Word & character embedding ($x$)

General Electric said the Postal Service contacted the company

Attention scores

$$\alpha_t = \mathbf{w}_\alpha \cdot \text{FFNN}_\alpha(\mathbf{x}^*_t)$$

dot product of weight vector and transformed hidden state

Attention distribution

$$a_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)}$$

just a softmax over attention scores for the span

Final representation

$$\hat{\mathbf{x}}_i = \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot \mathbf{x}_t$$

Attention-weighted sum of word embeddings
3. End-to-end Model

- Why include all these different terms in the span?

\[ g_i = [\mathbf{x}_{\text{START}(i)}, \mathbf{x}_{\text{END}(i)}, \hat{\mathbf{x}}_i, \phi(i)] \]

- hidden states for span’s start and end
- Attention-based representation
- Additional features
- Represents the context to the left and right of the span
- Represents the span itself
- Represents other information not in the text
3. End-to-end Model

- Lastly, score every pair of spans to decide if they are coreferent mentions

\[ s(i, j) = s_m(i) + s_m(j) + s_a(i, j) \]

- Are spans \( i \) and \( j \) coreferent mentions?
- Is \( i \) a mention?
- Is \( j \) a mention?
- Do they look coreferent?
3. End-to-end Model

- Lastly, score every pair of spans to decide if they are coreferent mentions
  \[ s(i, j) = s_m(i) + s_m(j) + s_a(i, j) \]
  
  Are spans \( i \) and \( j \) coreferent mentions?  
  Is \( i \) a mention?  
  Is \( j \) a mention?  
  Do they look coreferent?

- Scoring functions take the span representations as input
  \[ s_m(i) = w_m \cdot \text{FFNN}_m(g_i) \]
  \[ s_a(i, j) = w_a \cdot \text{FFNN}_a([g_i, g_j, g_i \circ g_j, \phi(i, j)]) \]
3. End-to-end Model

- Lastly, score every pair of spans to decide if they are coreferent mentions

\[ s(i, j) = s_m(i) + s_m(j) + s_a(i, j) \]

Are spans \( i \) and \( j \) coreferent mentions? Is \( i \) a mention? Is \( j \) a mention? Do they look coreferent?

- Scoring functions take the span representations as input

\[ s_m(i) = \mathbf{w}_m \cdot \text{FFNN}_m(g_i) \]
\[ s_a(i, j) = \mathbf{w}_a \cdot \text{FFNN}_a([g_i, g_j, g_i \circ g_j, \phi(i, j)]) \]

include multiplicative interactions between the representations again, we have some extra features
3. End-to-end Model

- Intractable to score every pair of spans
  - \(O(T^2)\) spans of text in a document (\(T\) is the number of words)
  - \(O(T^4)\) runtime!
  - So have to do lots of pruning to make work (only consider a few of the spans that are likely to be mentions)

- Attention learns which words are important in a mention (a bit like head words)

  (A fire in a Bangladeshi garment factory) has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee (the blaze) in the four-story building.
Coreference is a clustering task, let’s use a clustering algorithm!
  - In particular we will use agglomerative clustering

Start with each mention in it’s own singleton cluster

Merge a pair of clusters at each step
  - Use a model to score which cluster merges are good
Coreference Models: Clustering-Based

Google recently ... the company announced Google Plus ... the product features ...
Google recently ... the company announced Google Plus ... the product features ...

Coreference with Agglomerative Clustering

- $s(c_1, c_2) = 5$  ✔ merge
- $s(c_2, c_3) = 4$  ✔ merge
- $s(c_1, c_2) = -3$  ✗ do not merge
Coreference Models: Clustering-Based

Mention-pair decision is difficult

Cluster-pair decision is easier
Clustering Model Architecture

From Clark & Manning, 2016

Merge clusters \( c_1 = \{\text{Google, the company}\} \) and \( c_2 = \{\text{Google Plus, the product}\} \)?

**Mention Pairs**

- (Google, Google Plus)
- (Google, the product)
- (the company, Google Plus)
- (the company, the product)

**Mention-Pair Representations**

**Cluster-Pair Representation**

\( s(\text{MERGE}[c_1, c_2]) \)
Clustering Model Architecture

• First produce a vector for each pair of mentions
  • e.g., the output of the hidden layer in the feedforward neural network model
Clustering Model Architecture

- Then apply a pooling operation over the matrix of mention-pair representations to get a cluster-pair representation.
Score the candidate cluster merge by taking the dot product of the representation with a weight vector

\[ s(\text{MERGE}[c_1, c_2]) = u^T r_c(c_1, c_2) \]
Clustering Model: Training

• Current candidate cluster merges depend on previous ones it already made
  • So can’t use regular supervised learning
  • Instead use something like Reinforcement Learning to train the model
    • Reward for each merge: the change in a coreference evaluation metric
Coreference Evaluation

• Many different metrics: MUC, CEAF, LEA, B-CUBED, BLANC
  • Often report the average over a few different metrics
Coreference Evaluation

- An example: B-cubed
  - For each mention, compute a precision and a recall

\[ P = \frac{4}{5} \]
\[ R = \frac{4}{6} \]
Coreference Evaluation

- An example: B-cubed
  - For each mention, compute a precision and a recall

\[
P = \frac{4}{5} \\
R = \frac{4}{6}
\]  

\[
P = \frac{1}{5} \\
R = \frac{1}{3}
\]  

System Cluster 1

System Cluster 2

Gold Cluster 1

Gold Cluster 2
Coreference Evaluation

- An example: B-cubed
  - For each mention, compute a precision and a recall
  - Then average the individual Ps and Rs

\[
P = \frac{\left[4\left(\frac{4}{5}\right) + 1\left(\frac{1}{5}\right) + 2\left(\frac{2}{4}\right) + 2\left(\frac{2}{4}\right)\right]}{9} = 0.6
\]
Coreference Evaluation

100% Precision, 33% Recall

50% Precision, 100% Recall,
System Performance

- OntoNotes dataset: ~3000 documents labeled by humans
  - English and Chinese data

- Report an F1 score averaged over 3 coreference metrics
## System Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>English</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee et al. (2010)</td>
<td>~55</td>
<td>~50</td>
</tr>
<tr>
<td>Chen &amp; Ng (2012) [CoNLL 2012 Chinese winner]</td>
<td>54.5</td>
<td>57.6</td>
</tr>
<tr>
<td>Fernandes (2012) [CoNLL 2012 English winner]</td>
<td>60.7</td>
<td>51.6</td>
</tr>
<tr>
<td>Wiseman et al. (2015)</td>
<td>63.3</td>
<td>—</td>
</tr>
<tr>
<td>Clark &amp; Manning (2016)</td>
<td>65.4</td>
<td>63.7</td>
</tr>
<tr>
<td>Lee et al. (2017)</td>
<td>67.2</td>
<td>--</td>
</tr>
</tbody>
</table>

- **Rule-based system, used to be state-of-the-art!**
- **Non-neural machine learning models**
- **Neural mention ranker**
- **Neural clustering model**
- **End-to-end neural mention ranker**
Where do neural scoring models help?

- Especially with NPs and named entities with no string matching.

Neural vs non-neural scores:

18.9 $F_1$ vs 10.7 $F_1$ on this type compared to 68.7 vs 66.1 $F_1$

These kinds of coreference are hard and the scores are still low!

**Example Wins**

<table>
<thead>
<tr>
<th>Anaphor</th>
<th>Antecedent</th>
</tr>
</thead>
<tbody>
<tr>
<td>the country’s leftist rebels</td>
<td>the guerillas</td>
</tr>
<tr>
<td>the company</td>
<td>the New York firm</td>
</tr>
<tr>
<td>216 sailors from the “USS cole”</td>
<td>the crew</td>
</tr>
<tr>
<td>the gun</td>
<td>the rifle</td>
</tr>
</tbody>
</table>
Conclusion

• Coreference is a useful, challenging, and linguistically interesting task
  • Many different kinds of coreference resolution systems
• Systems are getting better rapidly, largely due to better neural models
  • But overall, results are still not amazing
• Try out a coreference system yourself!
  https://huggingface.co/coref/