



北京航空航天大学  
BEIHANG UNIVERSITY

# 自然语言处理

人工智能研究院

主讲教师 沙磊



# 共指消解

# 1. What is Coreference Resolution?

- Identify all mentions that refer to the same entity in the word

A couple of years later, Vanaja met Akhila at the local park. Akhila's son Prajwal was just two months younger than her son Akash, and they went to the same school. For the pre-school play, Prajwal was chosen for the lead role of the naughty child Lord Krishna. Akash was to be a tree. She resigned herself to make Akash the best tree that anybody had ever seen. She bought him a brown T-shirt and brown trousers to represent the tree trunk. Then she made a large cardboard cutout of a tree's foliage, with a circular opening in the middle for Akash's face. She attached red balls to it to represent fruits. It truly was the nicest tree.

From The Star by Shruthi Rao, with some shortening.

# Applications

- Full text understanding
  - information extraction, question answering, summarization, ...
  - “He was born in 1961”      (Who?)

# Applications

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

The image displays two examples of Google Translate's machine translation interface, illustrating how it handles gender and number in Spanish sentences.

**Example 1:**

- Spanish input:** A Alicia le gusta Juan porque es inteligente
- English output:** Alicia likes Juan because he's smart

**Example 2:**

- Spanish input:** A Juan le gusta Alicia porque es inteligente
- English output:** Juan likes Alicia because he's smart

The interface includes language selection buttons (Spanish, English, French, Detect language), a 'Translate' button, and a 'Suggest an edit' link.

# Applications

- Full text understanding
- Machine translation
- Dialogue Systems
  - “Book tickets to see JamesBond”
  - “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 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



### 3. Mention Detection

- Mention: A 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.: Paris, Joe Biden, Nike
  3. Noun phrases
    - “a dog,” “the big fluffy cat stuck in the tree”

# Mention Detection

- Mention: A span of text referring to some entity
- For detection: traditionally, use a pipeline of other NLP systems
  1. Pronouns
    - Use a part-of-speech tagger
  2. Named entities
    - Use a NamedEntity Recognition system
  3. Noun phrases
    - Use a parser (especially a constituency parser!)

# Mention Detection: Not Quite So Simple

- Marking all pronouns, named entities, and NPs as mentions over-generates mentions
- Are these mentions?
  - It is sunny
  - The best donut in the world
  - 100 miles

## 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)
    - But you might well want to know about referential singletons!

## 4. On to Coreference! First, some linguistics

- **Coreference** is when two mentions refer to the same entity in the world
  - *Barack Obama* traveled to ... *Obama* ...
- A different-but-related linguistic concept is **anaphora**: when a term (anaphor) refers to another term (antecedent)
  - 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

Barack Obama

Obama

world



## Anaphora

text

Barack Obama

he

world



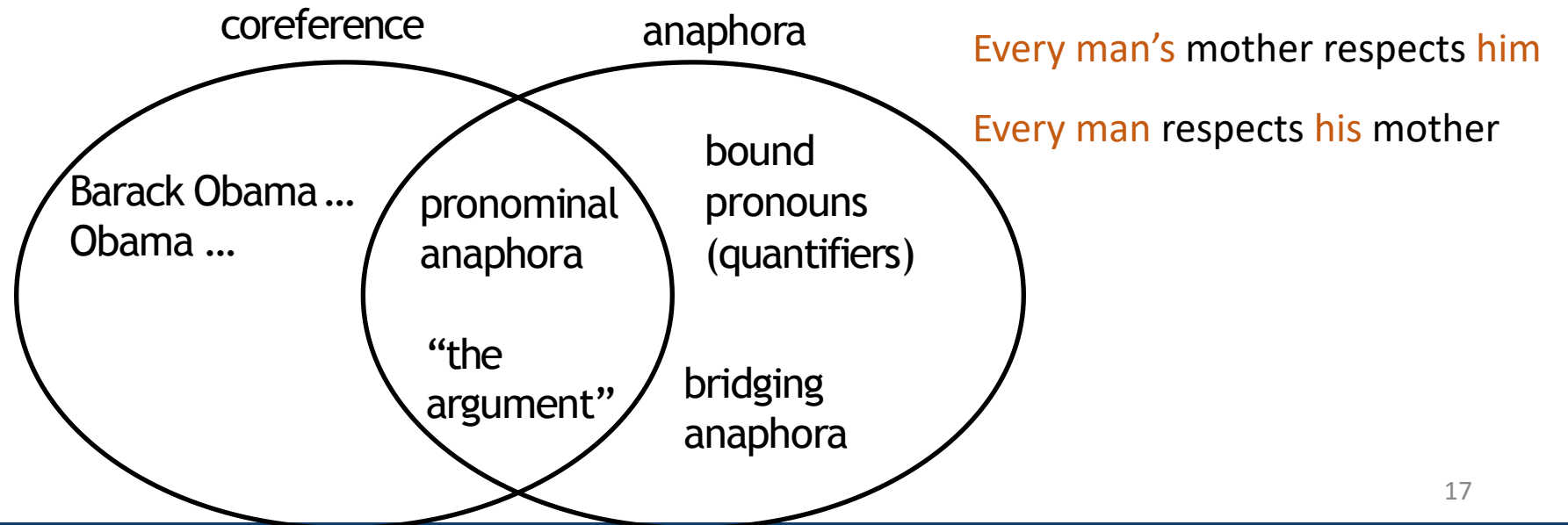
## Not all anaphoric relations are coreferential

- Not all nounphrases have reference
- *Every dancer* twisted *her knee*.
- *No dancer* twisted *her knee*.
- There are three NPs in each of these sentences; because the first one is non-referential (or a group), the other two aren't either.



# 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**.



## Anaphora vs. Cataphora(后指代)

- Usually, the antecedent comes before the anaphor (e.g., a pronoun), but not always

## Cataphora (后指代)

*“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)



# Taking stock ...

- It's often said that language is interpreted "in context"
- We've seen some examples, like word-sense disambiguation:
  - I took money out of **the bank** vs. The boat disembarked from **the bank**
- Coreference is another key example of this:
  - **Obama** was the president of the U.S. from 2008 to 2016. **He** was born in Hawaii.
- As we progress through an article, or dialogue, or webpage, we build up a (potentially very complex) **discourse model**, and we interpret new sentences/utterances with respect to our model of what's come before.
- Coreference and anaphora are all we see in this class of whole-discourse meaning
  - But it's a big part of human language understanding!

# Three Coreference Models

- Rule-based (pronominal anaphora resolution)
- Mention Pair/Mention Ranking
- End-to-end neural coreference

# 5. Traditional pronominal anaphora resolution: Hobbs' naive algorithm



1. Begin at the NP immediately dominating the pronoun
2. Go up tree to first NP or S. Call this X, and the path p.
3. Traverse all branches below X to the left of p, left-to-right, breadth-first. Propose as antecedent any NP that has a NP or S between it and X
4. If X is the highest S in the sentence, traverse the parse trees of the previous sentences in the order of recency. Traverse each tree left-to-right, breadth first. When an NP is encountered, propose as antecedent. If X not the highest node, go to step 5.

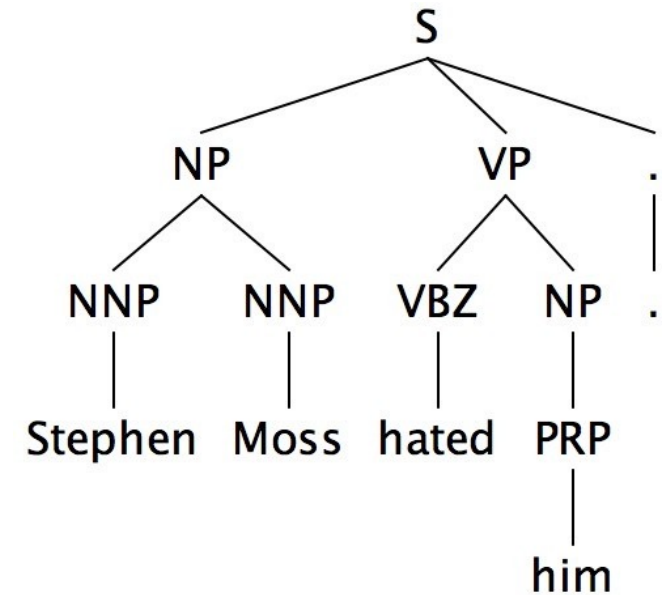
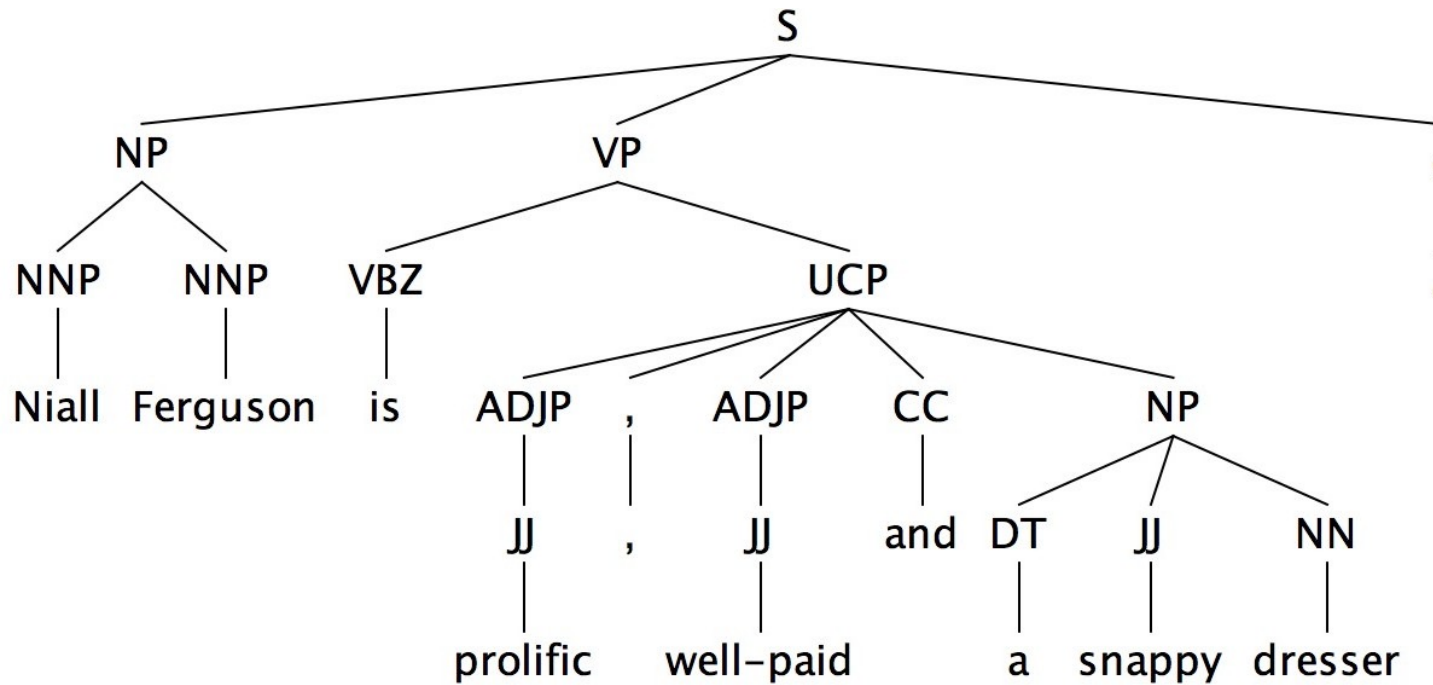
## Hobbs' naive algorithm (1976)

5. From node X, go up the tree to the first NP or S. Call it X, and the path p.
6. If X is an NP and the path p to X came from a non-head phrase of X (a specifier or adjunct, such as a possessive, PP, apposition, or relative clause), propose X as antecedent
  - (The original said “did not pass through the N' that X immediately dominates”, but the Penn Treebank grammar lacks N' nodes....)
7. Traverse all branches below X to the left of the path, in a left-to-right, breadth first manner. Propose any NP encountered as the antecedent
8. If X is an S node, traverse all branches of X to the right of the path but do not go below any NP or S encountered. Propose any NP as the antecedent.
9. Go to step4

Until deep learning still often used as a feature in ML systems!



# Hobbs Algorithm Example

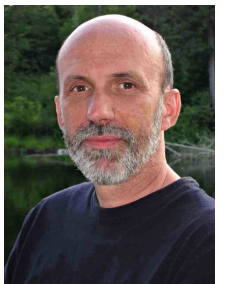
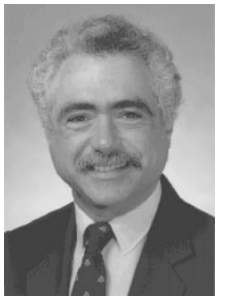


1. Begin at the NP immediately dominating the pronoun
2. Go up tree to first NP or S. Call this X, and the path p.
3. Traverse all branches below X to the left of p, left-to-right, breadth-first. Propose as antecedent any NP that has a NP or S between it and X
4. If X is the highest S in the sentence, traverse the parse trees of the previous sentences in the order of recency. Traverse each tree left-to-right, breadth first. When an NP is encountered, propose as antecedent. If X not the highest node, go to step5.

5. From node X, go up the tree to the first NP or S. Call it X, and the path p.
6. If X is an NP and the path p to X came from a non-head phrase of X (a specifier or adjunct, such as a possessive, PP, apposition, or relative clause), propose X as antecedent
7. Traverse all branches below X to the left of the path, in a left-to-right, breadth first manner. Propose any NP encountered as the antecedent
8. If X is an S node, traverse all branches of X to the right of the path but do not go below any NP or S encountered. Propose any NP as the antecedent.
9. Go to step4

# Knowledge-based Pronominal Coreference

- 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 city council refused the women a permit because they feared violence.
- The city council refused the women a permit because they advocated violence.
  - Winograd (1972)
- These are called **WinogradSchema**
  - Recently proposed as an alternative to the Turing test
    - See: Hector J. Levesque "On our best behaviour" IJCAI 2013  
<http://www.cs.toronto.edu/~hector/Papers/ijcai-13-paper.pdf>
    - <http://commonsensereasoning.org/winograd.html>
  - If you've fully solved coreference, arguably you've solved AI !!!



# Hobbs' algorithm: commentary

- *“... the naïve approach is quite good. Computationally speaking, it will be a long time before a semantically based algorithm is sophisticated enough to perform as well, and these results set a very high standard for any other approach to aim for.”*
- *“Yet there is every reason to pursue a semantically based approach. The naïve algorithm does not work. Any one can think of examples where it fails. In these cases, it not only fails; it gives no indication that it has failed and offers no help in finding the real antecedent.”*
  - — Hobbs (1978), *Lingua*, p. 345

## 6. Coreference Models: Mention Pair

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

I

Nader

he

my

she

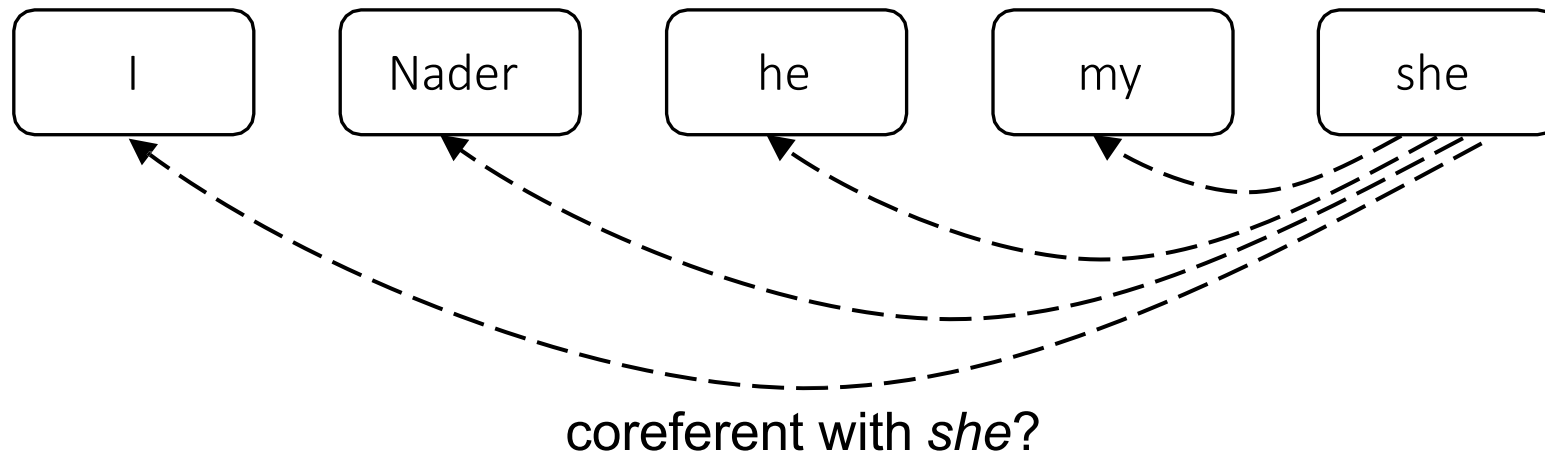
Coreference Cluster 1

Coreference Cluster 2

# 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

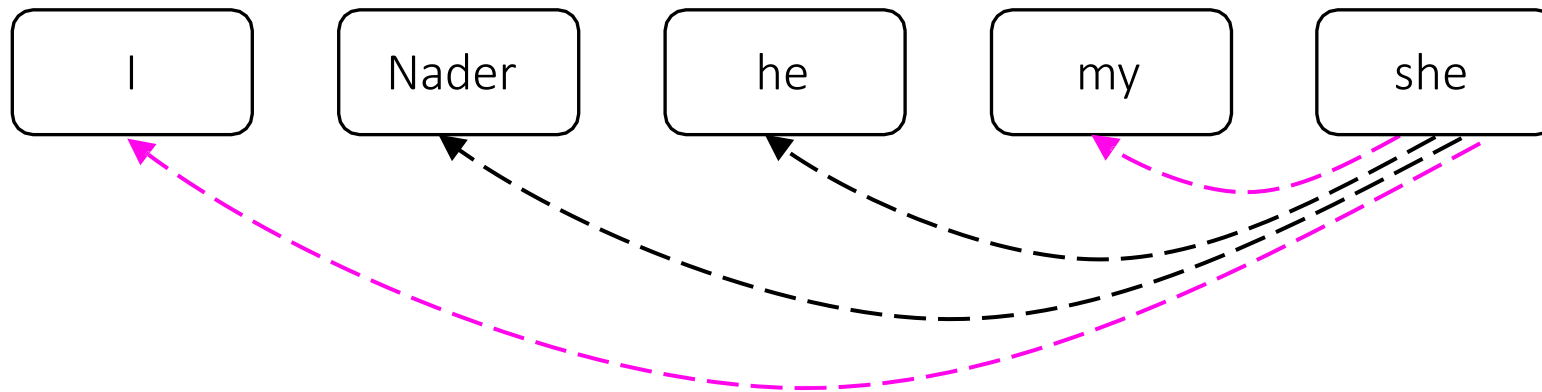
*“I voted for **Nader** because **he** was most aligned with **my** values,” **she** said.*



# 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.*

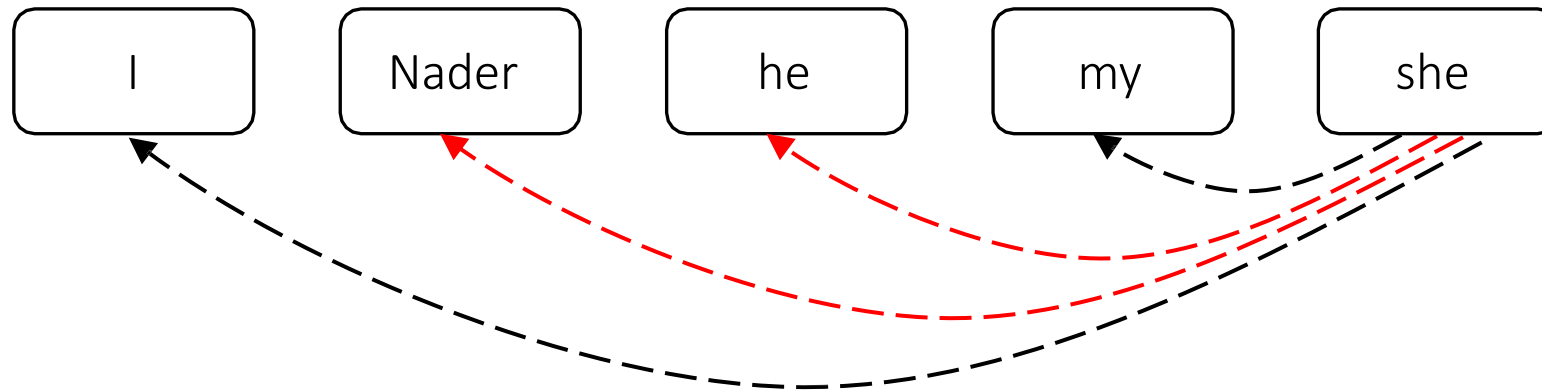


**Positive** examples: want  $p(m_i, m_j)$  to be near 1

# 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.*

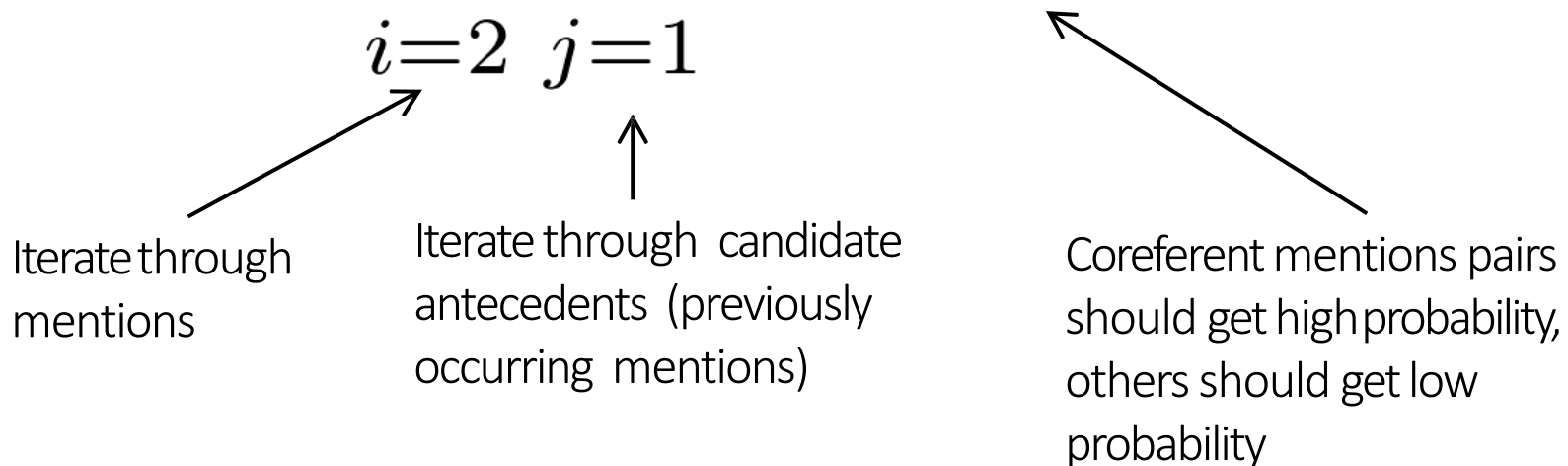


**Negative** examples: want  $p(m_i, m_j)$  to be near 0



# 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 TestTime

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

I

Nader

he

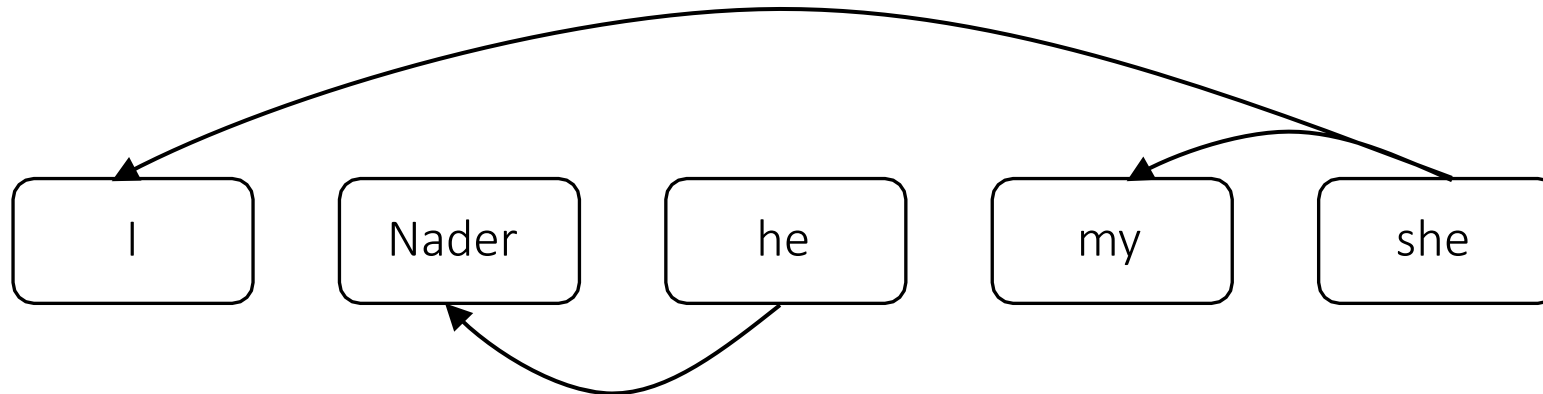
my

she

# 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

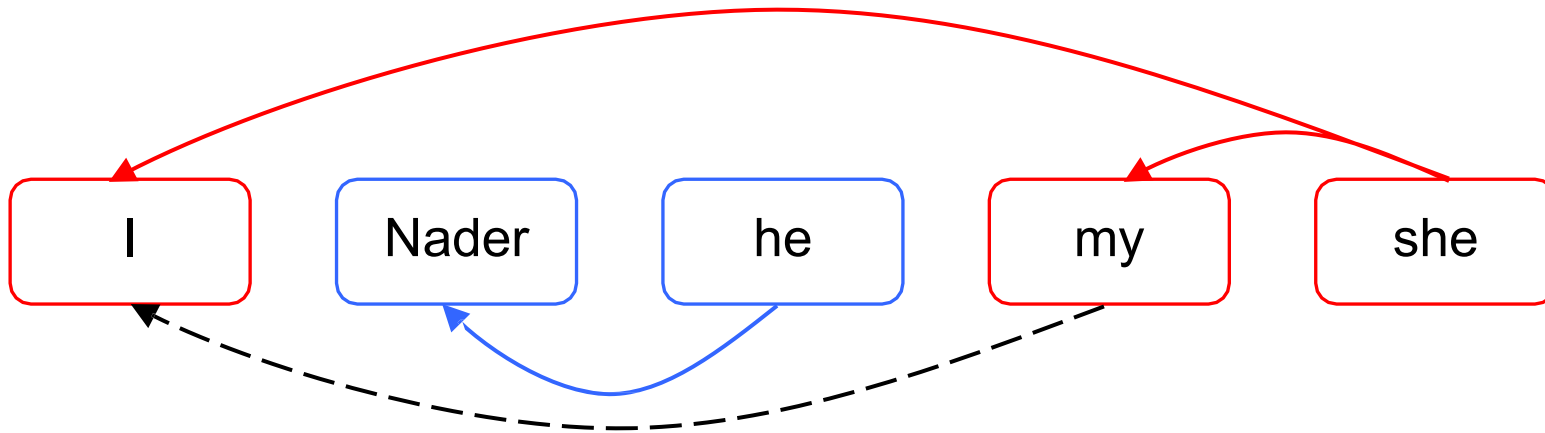
*"I voted for **Nader** because **he** was most aligned with **my** values," **she** said.*



# Mention Pair TestTime

- 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.*

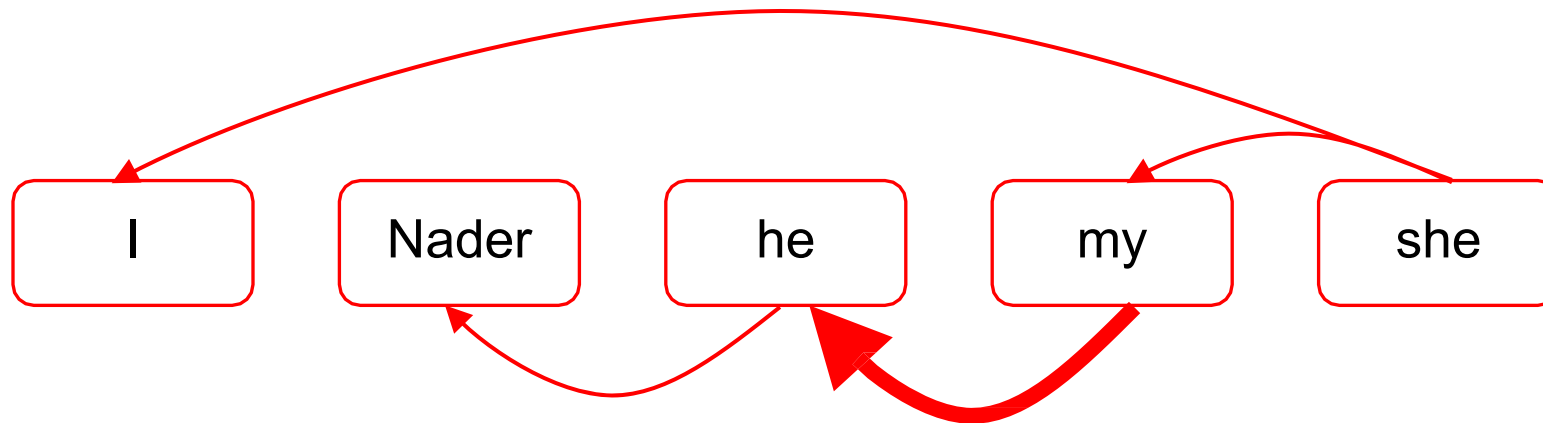


Even though the model did not predict this coreference link,  
*I* and *my* are coreferent due to transitivity

## Mention Pair TestTime

- 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

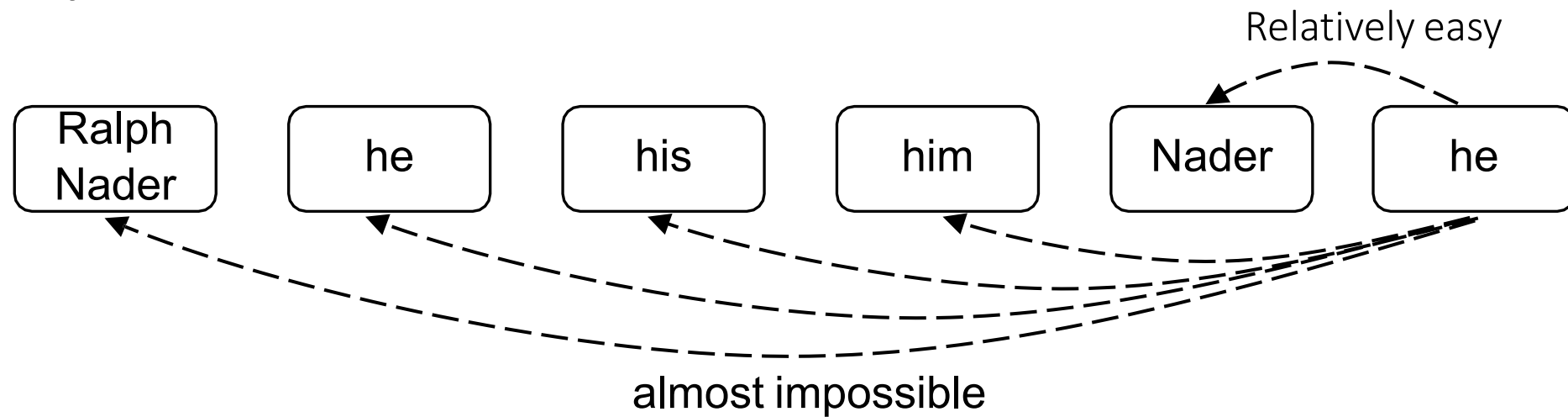
*"I voted for **Nader** because **he** was most aligned with **my** values," **she** said.*



Adding this extra link would merge everything into one big coreference cluster!

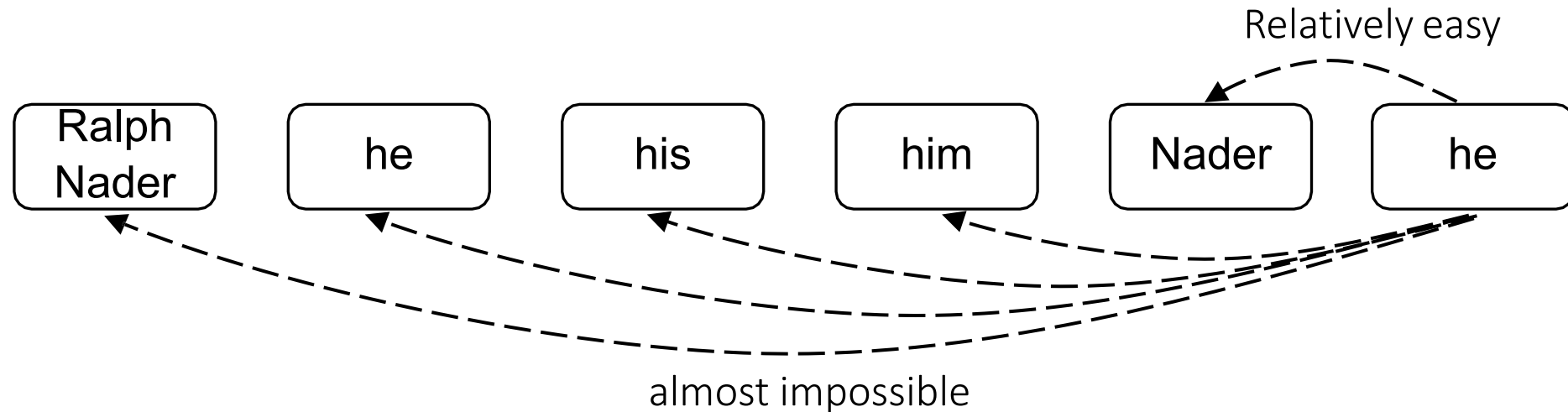
# Mention Pair Models: Disadvantage

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



# 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*...

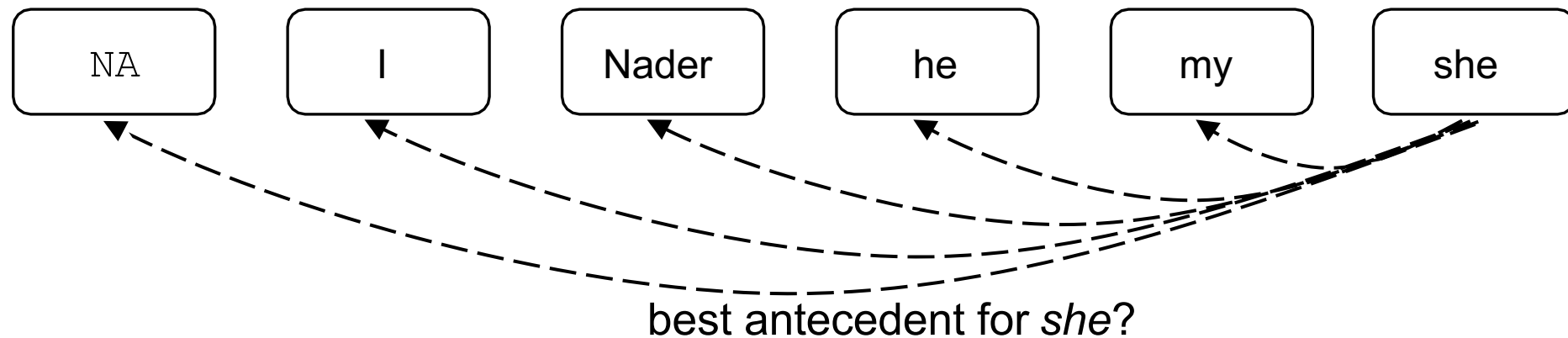


- Many mentions only have one clear antecedent
  - But we are asking the model to predict all of them
- Solution: instead train the model to predict only one antecedent for each mention
  - More linguistically plausible



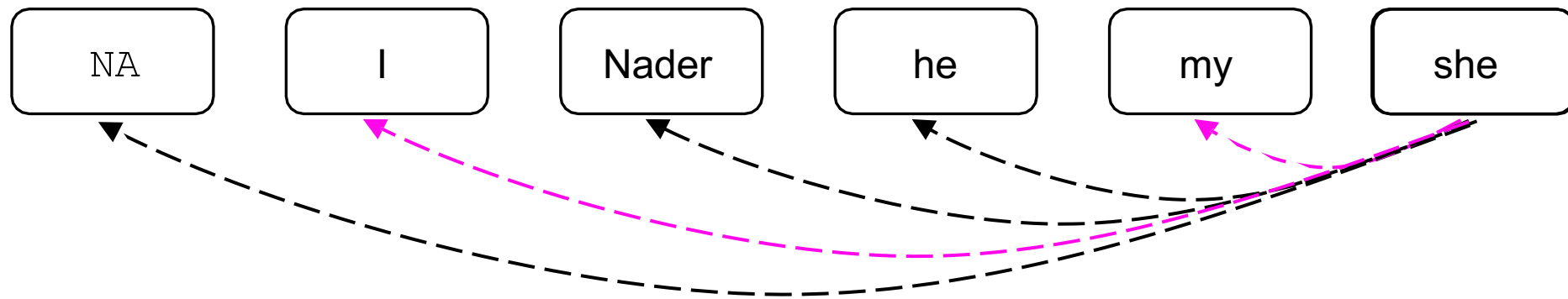
## 7. Coreference Models: Mention Ranking

- 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 (“singleton” or “first” mention)



## 7. Coreference Models: Mention Ranking

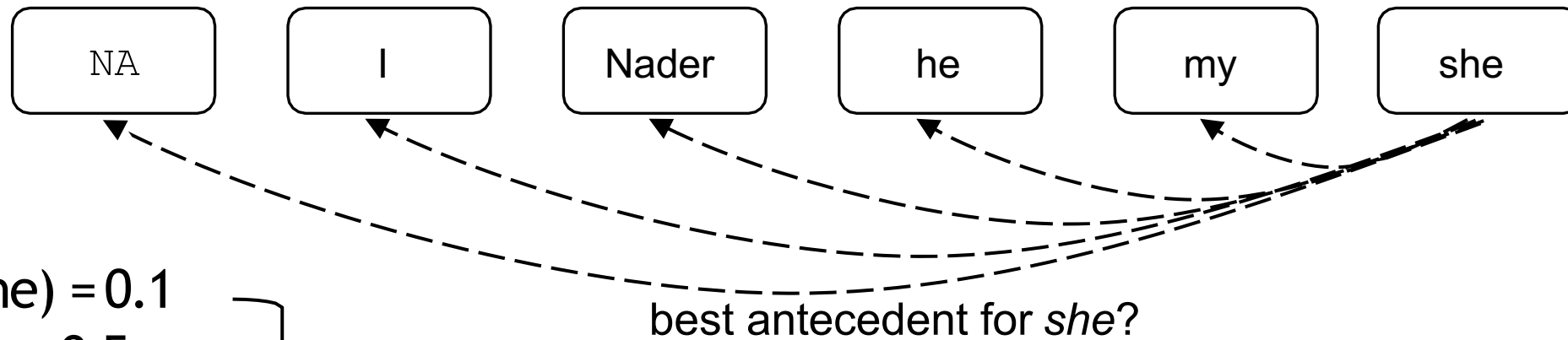
- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything (“singleton” or “first” mention)



**Positive examples:** model has to assign a high probability to either one (but not necessarily both)

## 7. 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 (“singleton” or “first” mention)



$$p(\text{NA}, \text{she}) = 0.1$$

$$p(\text{I}, \text{she}) = 0.5$$

$$p(\text{Nader}, \text{she}) = 0.1$$

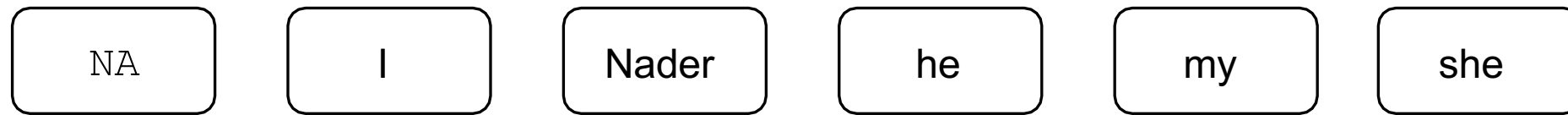
$$p(\text{he}, \text{she}) = 0.1$$

$$p(\text{my}, \text{she}) = 0.2$$

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

## 7. 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 (“singleton” or “first” mention)



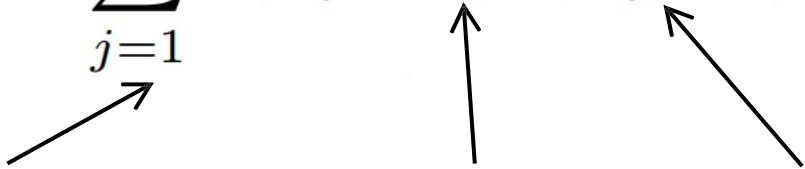
$$\begin{aligned} p(\text{NA}, \text{she}) &= 0.1 \\ p(\text{I}, \text{she}) &= 0.5 \\ p(\text{Nader}, \text{she}) &= 0.1 \\ p(\text{he}, \text{she}) &= 0.1 \\ p(\text{my}, \text{she}) &= 0.2 \end{aligned}$$

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

only add highest scoring coreference link

# 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

# How do we compute the probabilities?

- A. Non-neural statistical classifier
- B. Simple neural network
- C. More advanced model using LSTMs, attention, transformers

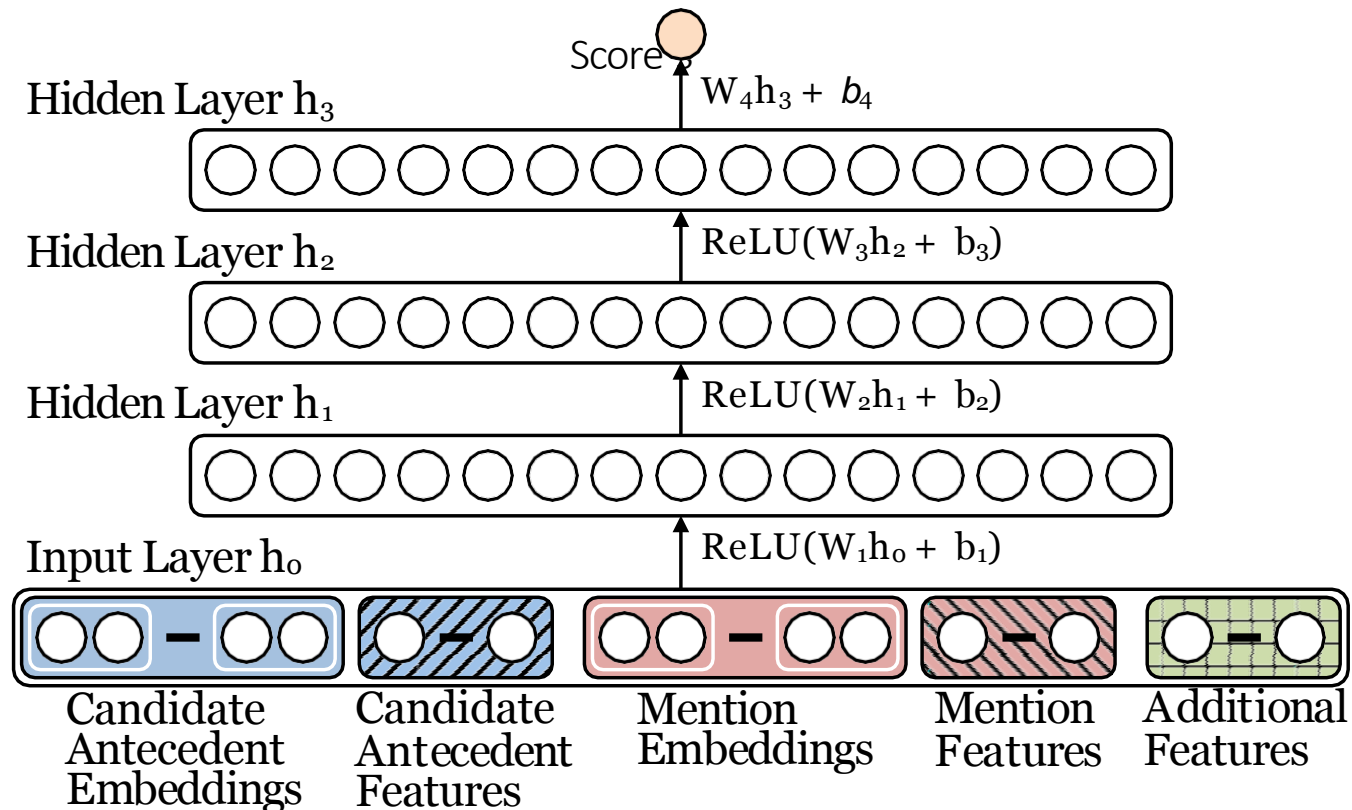
## A. Non-Neural Coref Model: Features

- 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.
- ...

Aslight change needed  
here for singular *they*!

## B. Neural Coref Model [Clark and Manning2016]

- Standard feed-forward neural network
  - Input layer: word embeddings and a few categorical features





# Neural Coref Model: Inputs

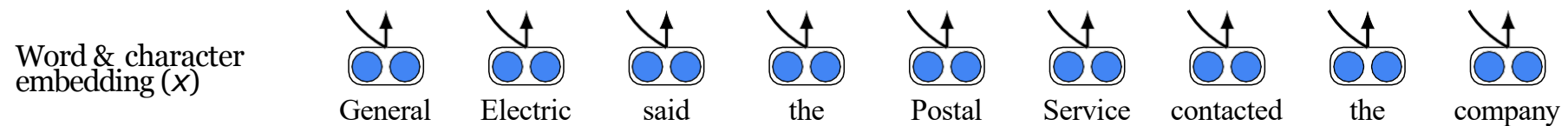
- Embeddings
  - Previous two words, first word, last word, head word, ... of each mention
    - The **head** word is the “most important” word in the mention – you can find it using a parser. e.g., *The fluffy **cat** stuck in the tree*
- Still need some other features to get a strongly performing model:
  - Distance
  - Document genre
  - Speaker information

## 7. End-to-end Neural Coref Model

- Current state-of-the-art models for coreference resolution
  - Kenton Lee et al. from UW (EMNLP 2017) et seq.
- Mention ranking model
- Improvements over simple feed-forward NN
  - Use an LSTM (or more)
  - 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

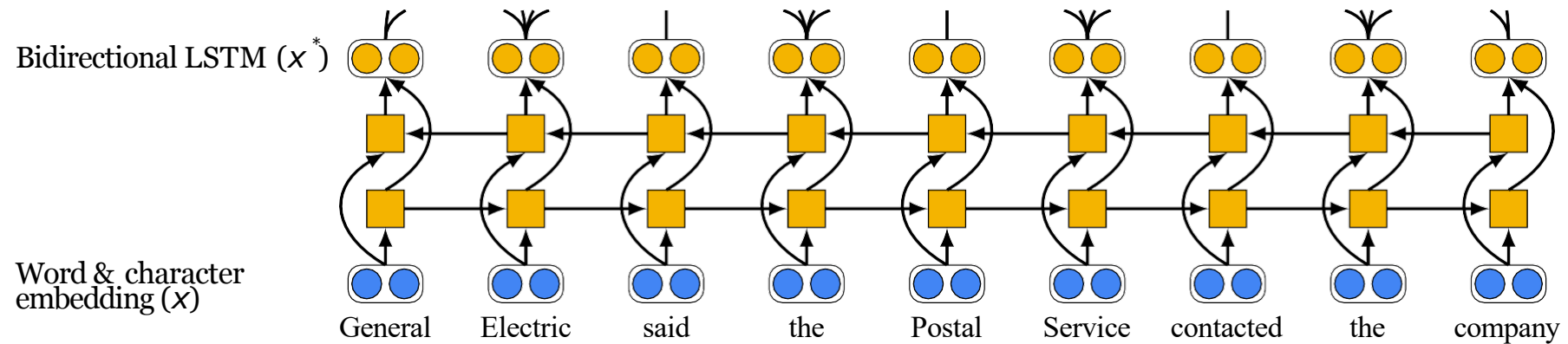
# End-to-end Model

- First embed the words in the document using a word embedding matrix and a character-level CNN



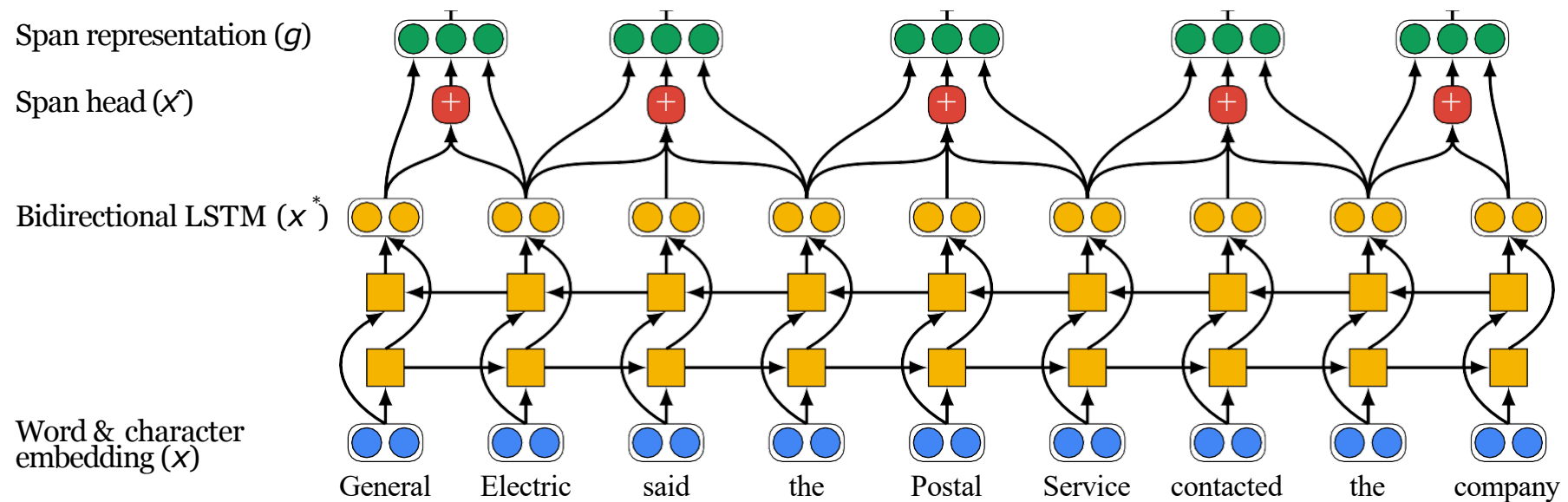
# End-to-end Model

- Then run a bidirectional LSTM over the document



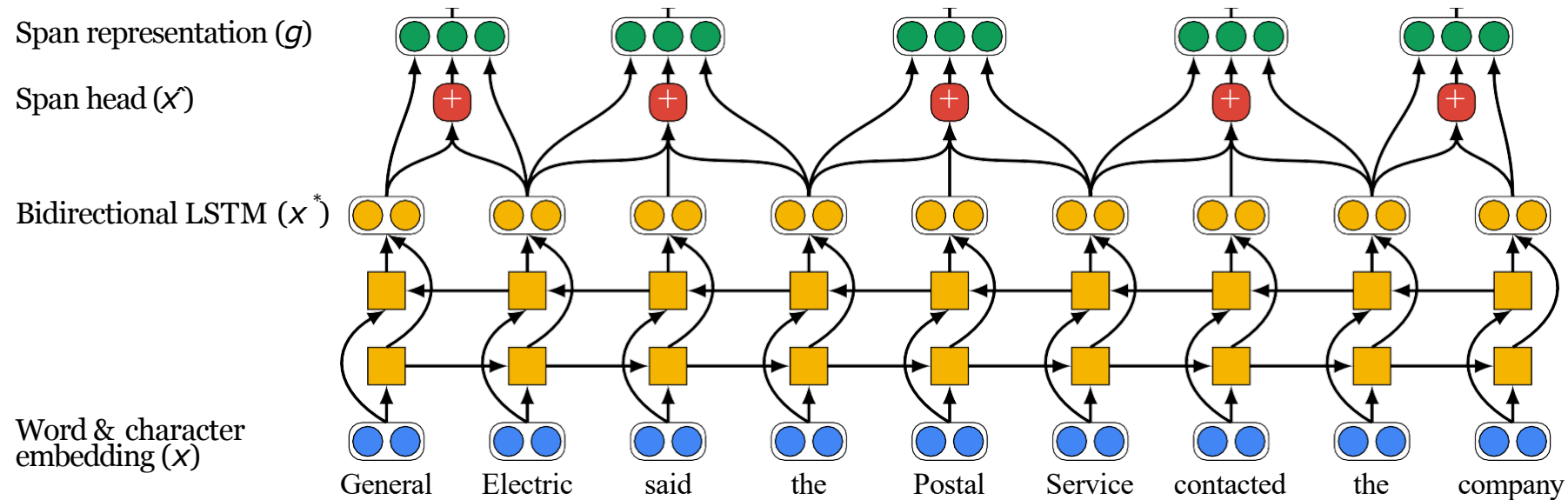
# End-to-end Model

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



# End-to-end Model

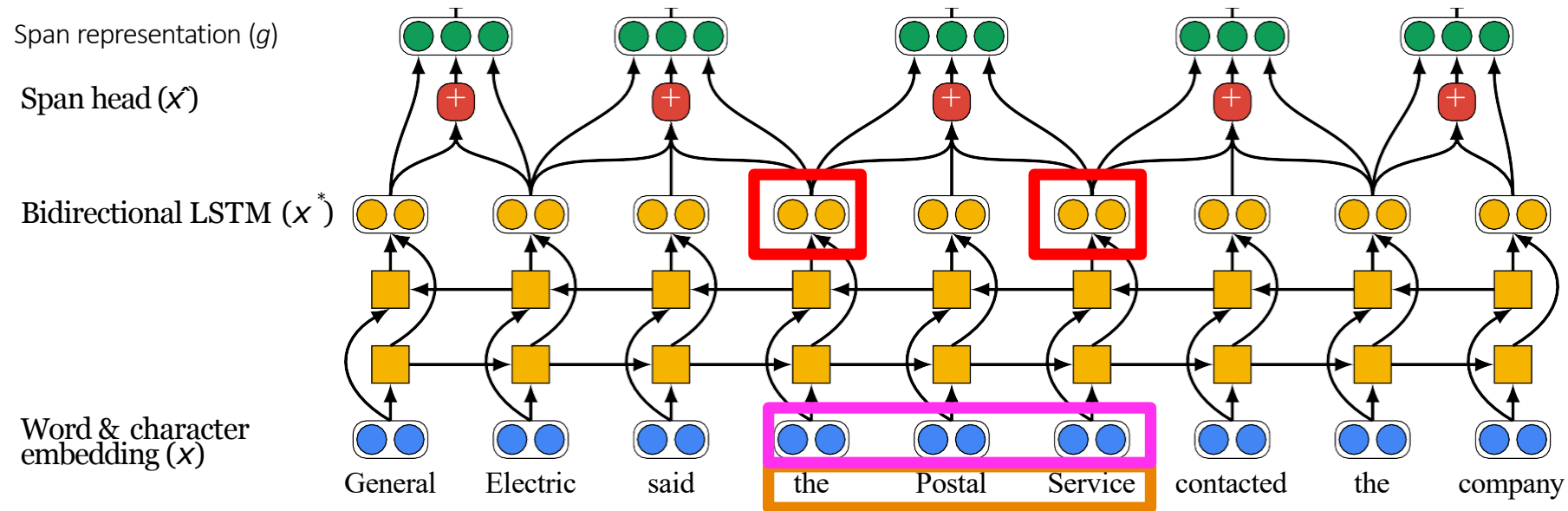
- Next, represent each span of text  $i$  going from  $\text{START}(i)$  to  $\text{END}(i)$  as a vector



- General, General Electric, General Electric said, ..., Electric, Electric said, ...* will all get its own vector representation

# 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 postalservice”



Span representation:

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

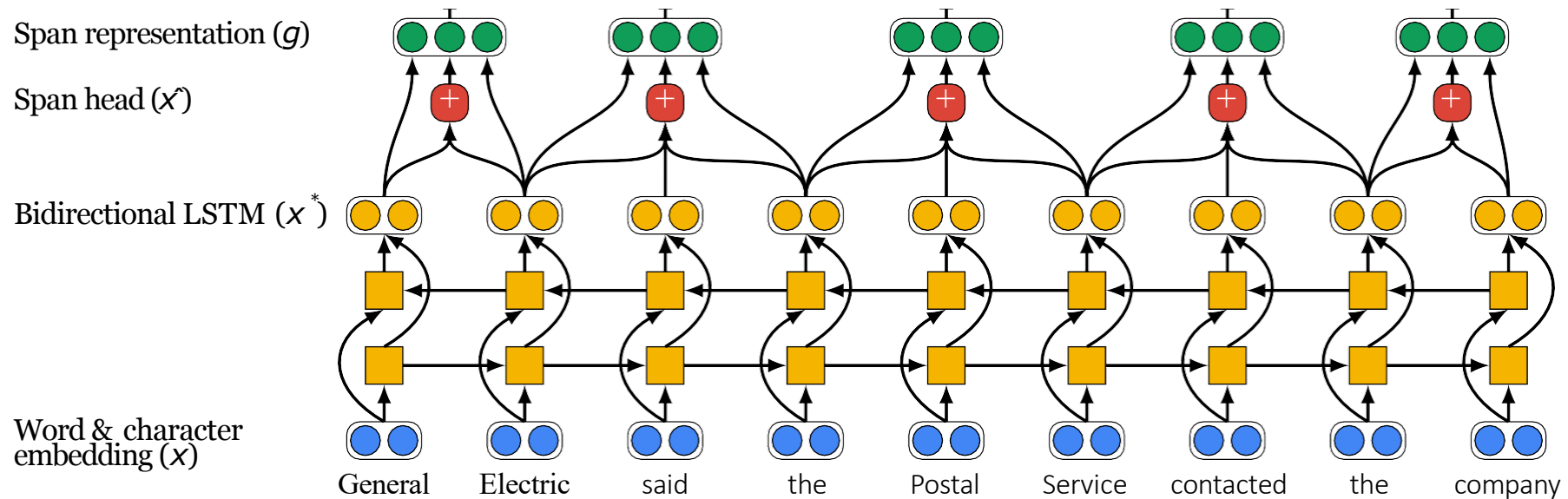
BILSTM hidden states for span's start and end

Attention-based representation (details next slide) of the words in the span

Additional features

# End-to-end Model

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



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



# End-to-end Model

- Why include all these different terms in the span?

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

hidden states for  
span's start and  
end

Attention-based  
representation

Additional  
features

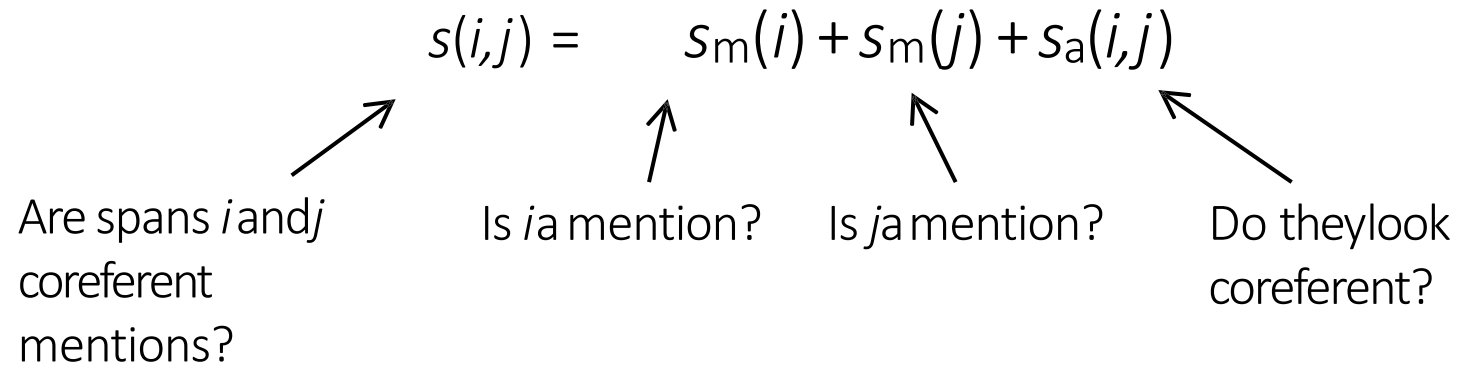
Represents the  
context to the left  
and right of the  
span

Represents the span itself

Represents other  
information not in  
the text

# End-to-end Model

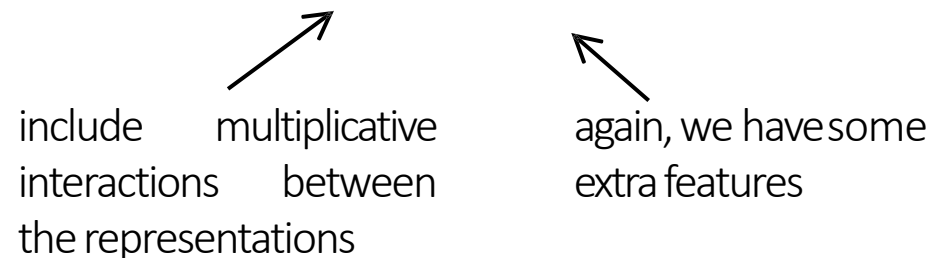
- Lastly, score every pair of spans to decide if they are coreferent mentions



- Scoring functions take the span representations as input

$$s_m(i) = w_m \cdot \text{FFNN}_m(g_i)$$

$$s_a(i,j) = w_a \cdot \text{FFNN}_a([g_i, g_j, g_i \circ g_j, \phi(i,j)])$$



# 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)

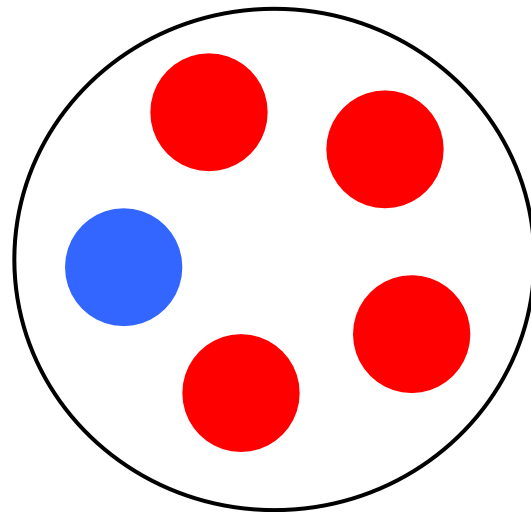
(A **fire** in a Bangladeshi garment factory) has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee (the **blaze**) in the four-story building.

# BERT-based coref: Now has the best results!

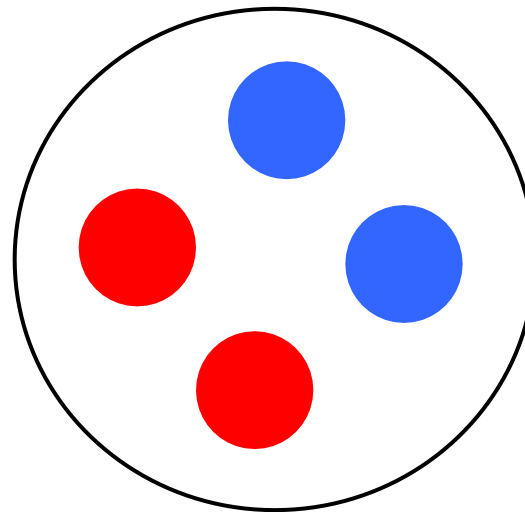
- Pretrained transformers can learn long-distance semantic dependencies in text.
- Idea 1, SpanBERT: Pretrains BERT models to be better at span-based prediction tasks like coref and QA
- Idea 2, BERT-QA for coref: Treat Coreference like a deep QA task
  - “Point to” a mention, and ask “what is its antecedent”
  - Answer span is a coreference link
- Idea 3: Maybe you don’t have to do it with spans after all, and you can go back to a representation of a word (maybe the head) and make things  $O(T^2)$ 
  - Current best model: Dobrovolskii (2021) <https://arxiv.org/abs/2109.04127>
  - Sort of makes sense given richness of transformers

## 8. Coreference Evaluation

- Many different metrics: MUC, CEAF, LEA, B-CUBED, BLANC
  - People often report the average over a few different metrics
- Essentially the metrics think of coreference as a clustering task and evaluate the quality of the clustering



System Cluster 1



System Cluster 2

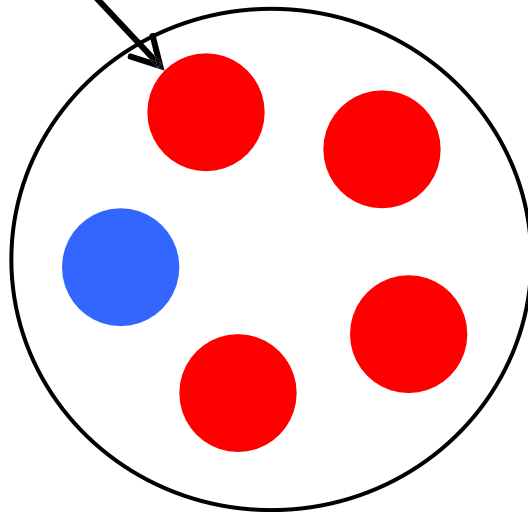
Gold Cluster 1

Gold Cluster 2

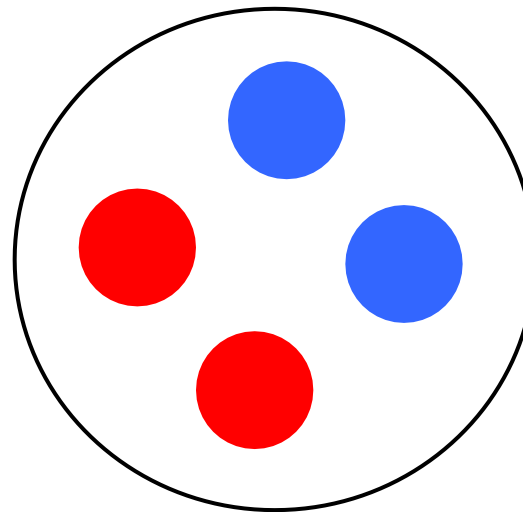
# Coreference Evaluation

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

$$P = 4/5$$
$$R = 4/6$$



System Cluster 1



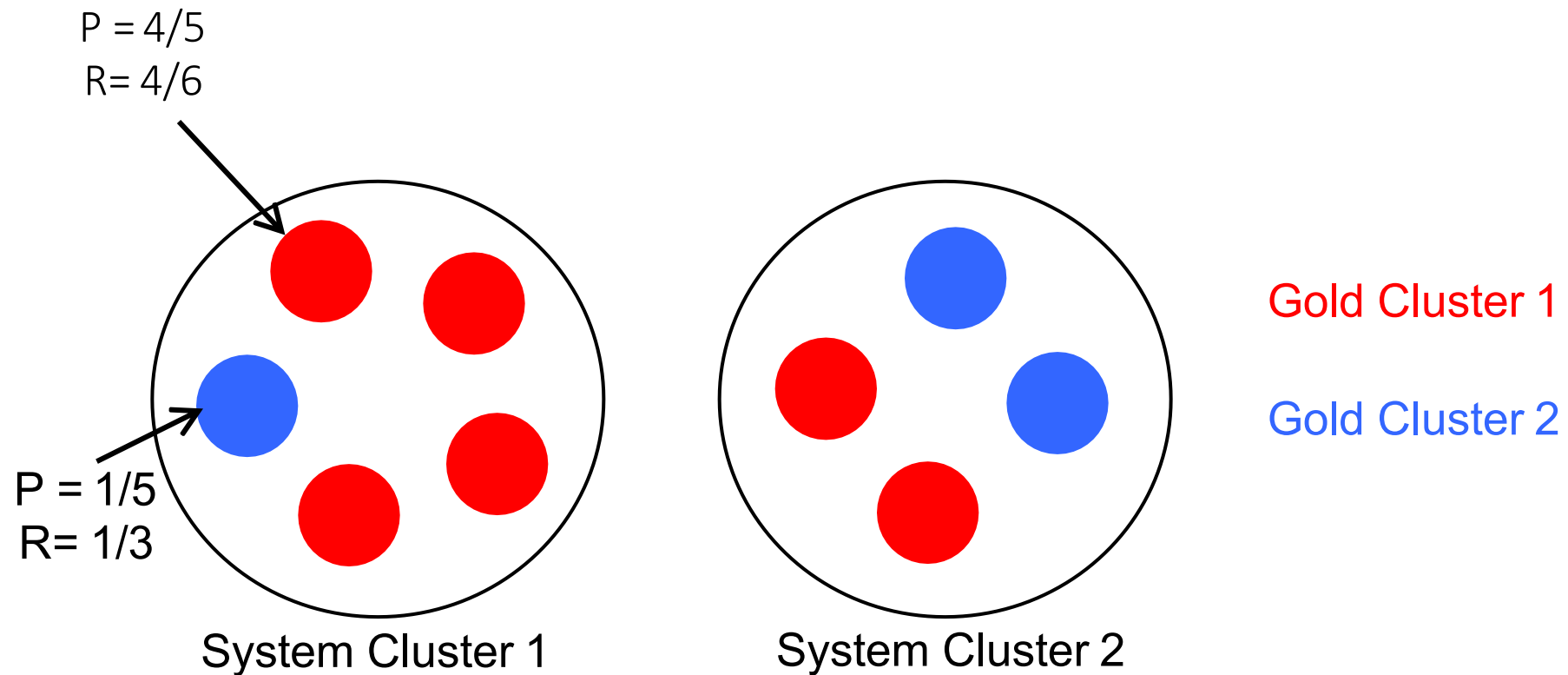
System Cluster 2

Gold Cluster 1

Gold Cluster 2

# Coreference Evaluation

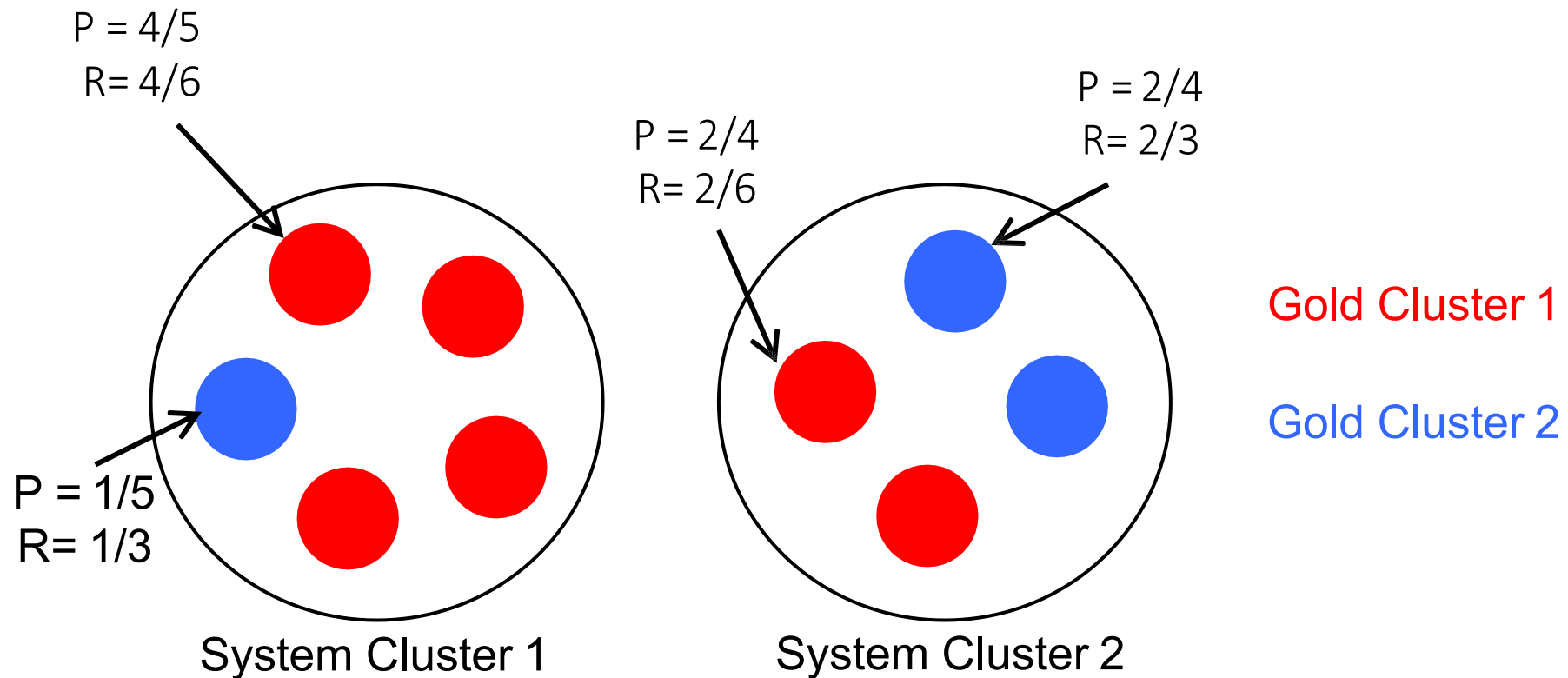
- An example: B-cubed
  - For each mention, compute a precision and a recall



# Coreference Evaluation

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

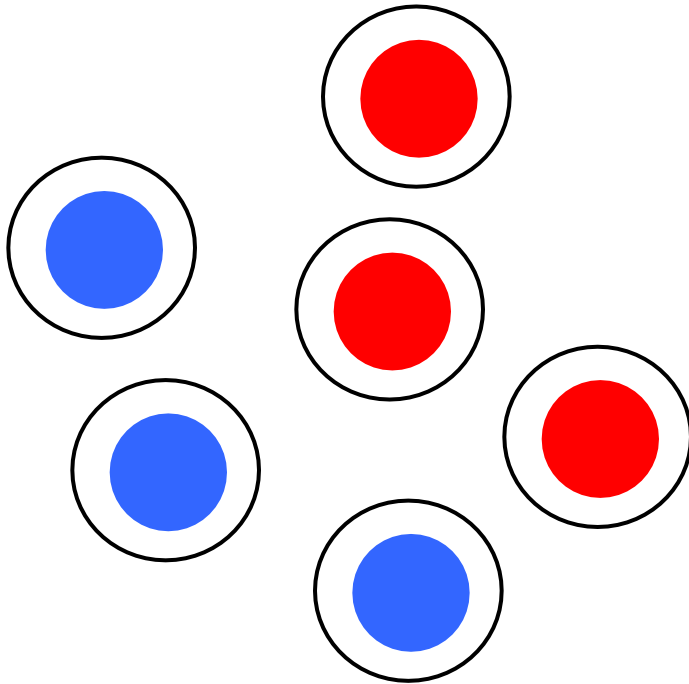
$$P = [4(4/5) + 1(1/5) + 2(2/4) + 2(2/4)] / 9 = 0.6$$



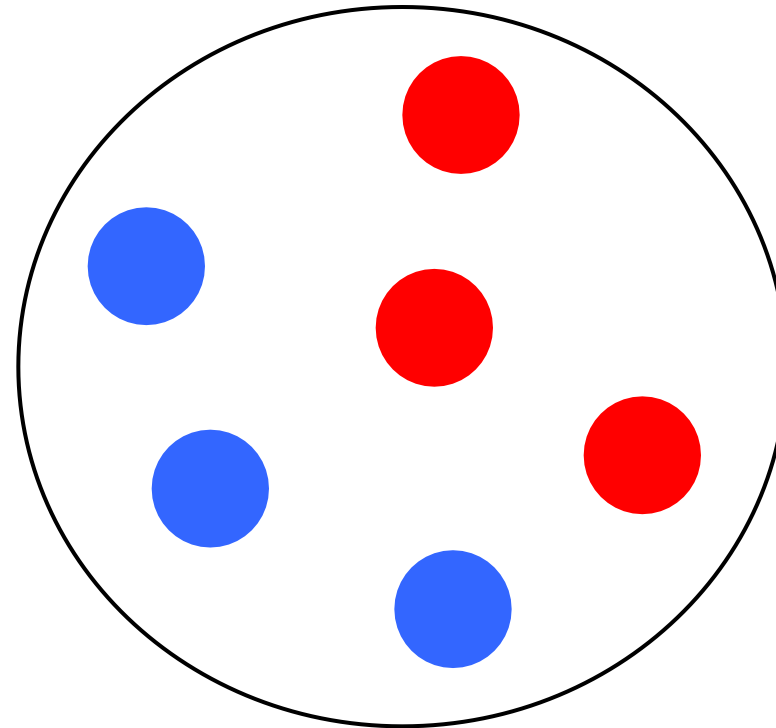


# Coreference Evaluation

100% Precision, 33% Recall



50% Precision, 100% Recall,



# System Performance

- OntoNotes dataset: ~3000 documents labeled by humans
  - English and Chinese data
- Standard evaluation: an F1 score averaged over 3 coreference metrics

# System Performance

Model	English	Chinese	
Lee et al. (2010)	~55	~50	Rule-based system, used to be state-of-the-art! Non-neural machine learning models
Chen & Ng (2012) [CoNLL 2012 Chinese winner]	54.5	57.6	
Fernandes (2012) [CoNLL 2012 English winner]	60.7	51.6	
Wiseman et al. (2015)	63.3	—	Neural mention ranker
Clark & Manning (2016)	65.4	63.7	Neural clustering model
Lee et al. (2017)	67.2	—	End-to-end neural ranker
Joshi et al. (2019)	79.6	—	End-to-end neural mention ranker + <b>SpanBERT</b> CorefQA
Wu et al. (2019) [CorefQA]	79.9	—	
Xu et al. (2020)	80.2		CorefQA + SpanBERT rulez
Dobrovolskii (2021)	81.0		

# Where do neural scoring models help?

- Especially with NPs and named entities with no string matching. Neural vs non-neural scores:

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

## Example Wins

Anaphor	Antecedent
the country's leftist rebels	the guerillas
the company	the New York firm
216 sailors from the "USS Cole"	the crew
the gun	the rifle

# 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 most models still make many mistakes – OntoNotes coref is easy newswire case
- Try out a coreference system yourself!
  - <http://corenlp.run/> (ask for coref in Annotations)
  - <https://huggingface.co/coref/>

Thank you!