IJCAI 2016 Tutorial

Coreference Resolution: Successes and Challenges

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas

http://www.hlt.utdallas.edu/~vince/ijcai-2016/coreference

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

- Inherently a clustering task
  - the coreference relation is transitive
    - Coref(A,B) $\wedge$ Coref(B,C) → Coref(A,C)

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention, m_j

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention, m_j
  - Does Queen Elizabeth have a preceding mention coreferent with it? If so, what is it?

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention, m_j
  - Does her have a preceding mention coreferent with it? If so, what is it?

Entity Coreference

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention, m_j
  - Does husband have a preceding mention coreferent with it? If so, what is it?

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Why it's hard

- Many sources of information play a role
  - **lexical** / **word**: head noun matches
    - President Clinton = Clinton =? Hillary Clinton
  - **grammatical**: number/gender agreement, …
  - **syntactic**: syntactic parallelism, binding constraints
    - John helped himself to... vs. John helped him to…
  - **discourse**: discourse focus, salience, recency, …
  - **semantic**: semantic class agreement, …
  - **world knowledge**

- Not all knowledge sources can be computed easily

Why it's hard

- Many sources of information play a role
  - **lexical** / **word**: head noun matches
    - President Clinton = Clinton =? Hillary Clinton
  - **grammatical**: number/gender agreement, …
  - **syntactic**: syntactic parallelism, binding constraints
    - John helped himself to... vs. John helped him to…
  - **discourse**: discourse focus, salience, recency, …
  - **semantic**: semantic class agreement, …
  - **world knowledge**
- Not all knowledge sources can be computed easily

Why It's Hard

No single source is a completely reliable indicator

- number and gender
  - assassination (of Jesuit priests) =? these murders
  - the woman = she = Mary =? the chairman

Why It's Hard

Coreference strategies differ depending on the mention type

- definiteness of mentions
  - … Then Mark saw the man walking down the street.
  - … Then Mark saw a man walking down the street.

- pronoun resolution alone is notoriously difficult
  - There are pronouns whose resolution requires world knowledge
    - The Winograd Schema Challenge (Levesque, 2011)
  - pleonastic pronouns refer to nothing in the text

I went outside and it was snowing.

Why it's hard

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Why it's hard

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

Queen Elizabeth set about transforming her husband,

King George VI, into a viable monarch. Logue,

a renowned speech therapist, was summoned to help

the King overcome his speech impediment...

Why it's hard

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

~~Queen Elizabeth~~ set about transforming her husband,

King George VI, into a viable monarch. Logue,

a renowned speech therapist, was summoned to help

the King overcome his speech impediment...

Why it's hard

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

Queen Elizabeth set about transforming her husband,
King George VI, into a viable monarch. Logue,
a renowned speech therapist, was summoned to help
the King overcome his speech impediment...

Why it's hard

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

Queen Elizabeth set about transforming her ~~husband~~,

King George VI, into a viable monarch. Logue,

a renowned speech therapist, was summoned to help

the King overcome his speech impediment...

Why it's hard

Resolving a non-anaphoric mention causes a coreference system's **precision** to drop.

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

Queen Elizabeth set about transforming her ~~husband~~,
King George VI, into a viable monarch. Logue,
a renowned speech therapist, was summoned to help
the King overcome his speech impediment...

Why it's hard

If we do a **perfect** job in anaphoricity determination, coreference systems can improve by an F-score of 5% absolute (Stoyanov et al., 2009)

- Anaphoricity determination is a difficult task
  - determine whether a mention has an antecedent
    - check whether it is part of a coreference chain but is not the head of the chain

Queen Elizabeth set about transforming her ~~husband~~,
King George VI, into a viable monarch. Logue,
a renowned speech therapist, was summoned to help
the King overcome his speech impediment...

Not all coreference relations are equally difficult to resolve

Not all coreference relations are equally difficult to resolve

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Not all coreference relations are equally difficult to resolve

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Not all coreference relations are equally difficult to resolve

Queen Elizabeth set about transforming $\boxed{\text{her}}$ $\boxed{\text{husband}}$, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Not all coreference relations are equally difficult to resolve

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Not all coreference relations are equally difficult to resolve

Queen Elizabeth set about transforming her husband, $\boxed{\text{King George VI}}$, into $\boxed{\text{a viable monarch}}$. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Not all coreference relations are equally difficult to resolve

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist was summoned to help the King overcome his speech impediment...

Let’s modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let’s modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let’s modify the paragraph

<u>Queen Mother</u> asked Queen Elizabeth to transform $\boxed{\text{her}}$ sister, Margaret, into <u>an elegant lady</u>. <u>Logue</u> was summoned to help the princess improve her <u>manner</u>...

Let's modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let’s modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let's modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let's modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let's modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let's modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let's modify the paragraph

Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

Let's modify the paragraph

> Queen Mother asked Queen Elizabeth to transform her sister, Margaret, into an elegant lady. Logue was summoned to help the princess improve her manner...

- Not all coreference relations are equally difficult to identify
  - A system will be more confident in predicting some and less confident in predicting others
    - use the more confident ones to help predict the remaining ones?

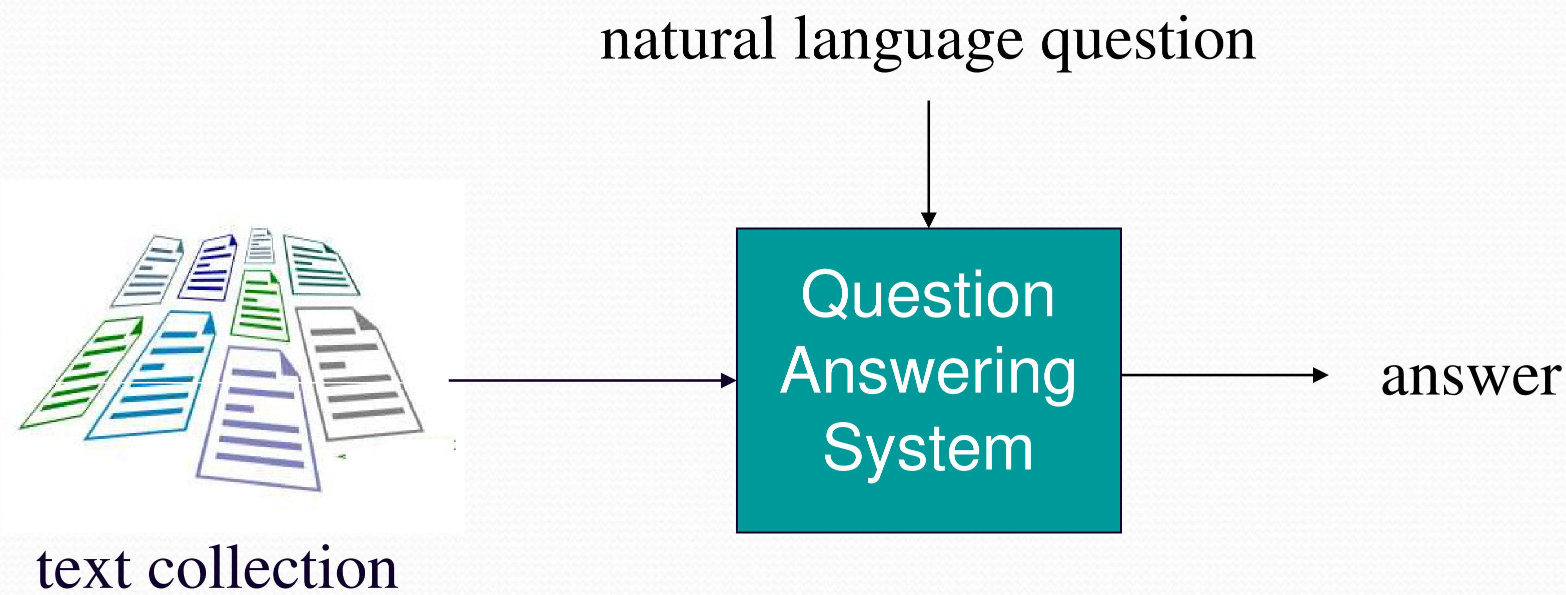
Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Applications of Coreference

- Question answering
- Information extraction
- Machine translation
- Text summarization
- Information retrieval
- ...

Application: Question Answering



Coreference for Question Answering

Where was Mozart born?

Mozart was one of the first classical composers. He was born in Salzburg, Austria, in 27 January 1756. He wrote music of many different genres...

Haydn was a contemporary and friend of Mozart. He was born in Rohrau, Austria, in 31 March 1732. He wrote 104 symphonies...

Coreference for Question Answering

Where was Mozart born?

Mozart was one of the first classical composers. He was born in Salzburg, Austria, in 27 January 1756. He wrote music of many different genres...

Haydn was a contemporary and friend of Mozart. He was born in Rohrau, Austria, in 31 March 1732. He wrote 104 symphonies...

Coreference for Question Answering

Where was Mozart born?

Mozart was one of the first classical composers. He was born in Salzburg, Austria, in 27 January 1756. He wrote music of many different genres...

Haydn was a contemporary and friend of Mozart. He was born in Rohrau, Austria, in 31 March 1732. He wrote 104 symphonies...

Coreference for Question Answering

Where was Mozart born?

Mozart was one of the first classical composers. He was born in Salzburg, Austria, in 27 January 1756. He wrote music of many different genres...

Haydn was a contemporary and friend of Mozart. He was born in Rohrau, Austria, in 31 March 1732. He wrote 104 symphonies...

Application: Information Extraction

AFGANISTAN MAY BE PREPARING FOR ANOTHER TEST

Thousands of people are feared dead following... (voice-over) ...a powerful earthquake that hit Afghanistan today. The quake registered 6.9 on the Richter scale. (on camera) Details now hard to come by, but reports say entire villages were buried by the earthquake.

Disaster Type:
- location:
- date:
- magnitude:
- magnitude-confidence:
- damage:
  - human-effect:
    - victim:
    - number:
    - outcome:
  - physical-effect:
    - object:
    - outcome:

Application: Information Extraction

AFGANISTAN MAY BE PREPARING FOR ANOTHER TEST

Thousands of people are feared dead following... (voice-over) ...a powerful earthquake that hit Afghanistan today. The quake registered 6.9 on the Richter scale. (on camera) Details now hard to come by, but reports say entire villages were buried by the earthquake.

Disaster Type: earthquake
- location: *Afghanistan*
- date: *today*
- magnitude: *6.9*
- magnitude-confidence: high
- damage:
  - human-effect:
    - victim: *Thousands of people*
    - number: *Thousands*
    - outcome: dead
  - physical-effect:
    - object: *entire villages*
    - outcome: damaged

Coreference for Information Extraction

AFGANISTAN MAY BE PREPARING FOR ANOTHER TEST

Thousands of people are feared dead following... (voice-over) ...a powerful earthquake that hit Afghanistan today. The quake registered 6.9 on the Richter scale. (on camera) Details now hard to come by, but reports say entire villages were buried by the earthquake.

Disaster Type: earthquake
- location: *Afghanistan*
- date: *today*
- magnitude: *6.9*
- magnitude-confidence: high
- damage:
  - human-effect:
    - victim: *Thousands of people*
    - number: *Thousands*
    - outcome: dead
  - physical-effect:
    - object: *entire villages*
    - outcome: damaged

Coreference for Information Extraction

AFGANISTAN MAY BE PREPARING FOR ANOTHER TEST
Thousands of people are feared dead following... (voice-over) ...a powerful earthquake that hit Afghanistan today. The quake registered 6.9 on the Richter scale. (on camera) Details now hard to come by, but reports say entire villages were buried by the earthquake.
The last major earthquake in Afghanistan took place in August 2001.

Disaster Type: earthquake

- location: *Afghanistan*
- date: *today*
- magnitude: *6.9*
- magnitude-confidence: high
- damage:
  - human-effect:
    - victim: *Thousands of people*
    - number: *Thousands*
    - outcome: dead
  - physical-effect:
    - object: *entire villages*
    - outcome: damaged

Application: Machine Translation

- Chinese to English machine translation

俄罗斯作为米洛舍夫维奇一贯的支持者，曾经提出调停这场政治危机。

Russia is a consistent supporter of Milosevic, has proposed to mediate the political crisis.

Application: Machine Translation

- Chinese to English machine translation

俄罗斯作为米洛舍夫维奇一贯的支持者，曾经提出调停这场政治危机。

Russia is a consistent supporter of Milosevic, ??? has proposed to mediate the political crisis.

Application: Machine Translation

- Chinese to English machine translation

俄罗斯作为米洛舍夫维奇一贯的支持者，曾经提出调停这场政治危机。

Russia is a consistent supporter of Milosevic, ??? has proposed to mediate the political crisis.

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Rule-Based Approaches

- Popular in the 1970s and 1980s
  - A popular PhD thesis topic
  - Charniak (1972): Children's story comprehension
    - "In order to do pronoun resolution, one had to be able to do everything else."
  - Focus on **sophisticated knowledge** & **inference** mechanisms

- **Syntax-based** approaches (Hobbs, 1976)

- **Discourse-based** approaches / **Centering** algorithms
  - Kantor (1977), Grosz (1977), Webber (1978), Sidner (1979)
  - **Centering** algorithms and alternatives
    - Brennan et al. (1987), …

Rule-Based Approaches

- **Knowledge-poor** approaches (Mitkov, 1990s)

Evaluation of Rule-Based Approaches

- **Small-scale** evaluation
  - a few hundred sentences
  - sometimes by hand
    - algorithm not always implemented/implementable

MUC Coreference

- The MUC conferences
  - Goal: evaluate information extraction systems

- Coreference as a supporting task for information extraction
  - First recognized in MUC-6 (1995)
  - First large-scale evaluation of coreference systems. Need
    - Scoring program
      - MUC scoring program (Vilain et al., 1995)
    - Guidelines for coreference-annotating a corpus
      - Original task definition very ambitious
      - Final task definition focuses solely on identity coreference

Other Types of Coreference

- Non-identity coreference: **bridging**
  - **Part-whole relations**
    - He passed by Jan's house and saw that the door was painted red.
  - **Set-subset relations**

- Difficult cases
  - **Verb phrase ellipsis**
    - John enjoys watching movies, but Mary doesn't.
  - **Reference to abstract entities**
    - Each fall, penguins migrate to Fuji.
    - It happens just before the eggs hatch.

Other Types of Coreference

- Non-identity coreference: **bridging**
  - **Part-whole relations**
    - He passed by Jan's house and saw that the door was painted red.
  - **Set-subset relations**

- Difficult cases
  - **Verb phrase ellipsis**
    - John enjoys watching movies, but Mary doesn't.
  - **Reference to abstract entities**
    - Each fall, penguins migrate to Fuji.
    - That's why I'm going there next month.

MUC Coreference

- MUC-6 coreference task
  - **MUC-6 corpus**: 30 training documents, 30 test documents
  - Despite the fact that 30 training documents are available, all but one resolver are rule-based
    - UMASS (Lenhert & McCarthy, 1995) resolver is learning-based
  - Best-performing resolver, FASTUS, is rule-based

MUC Coreference

- MUC-6 coreference task
  - **MUC-6 corpus**: 30 training documents, 30 test documents
  - Despite the fact that 30 training documents are available, all but one resolver are rule-based
    - UMASS (Lenhert & McCarthy, 1995) resolver is learning-based
  - Best-performing resolver, FASTUS, is rule-based

- MUC-7 coreference task
  - **MUC-7 corpus**: 30 training documents, 20 test documents
  - none of the 7 participating resolvers used machine learning

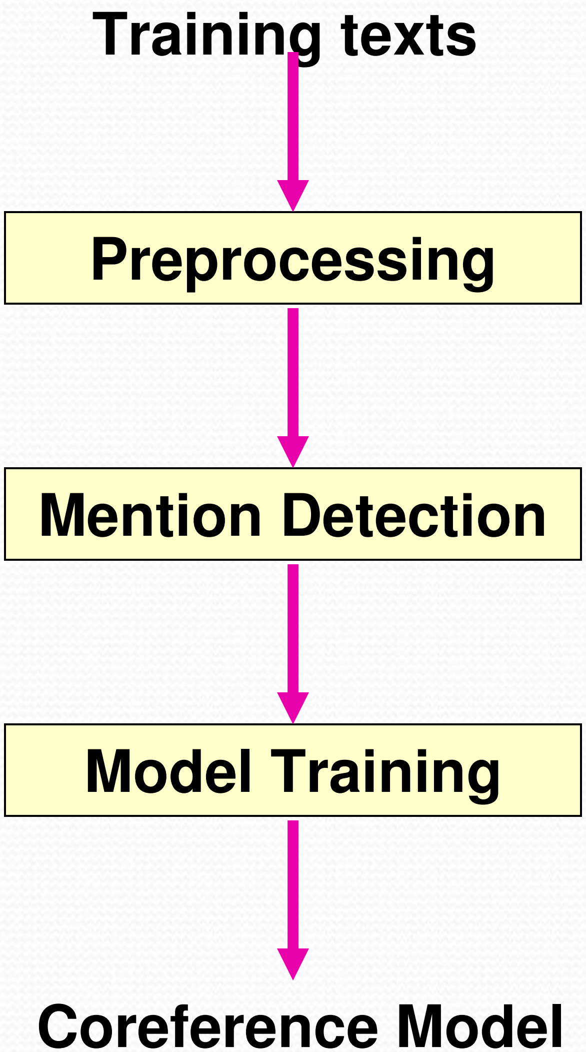
MUC Coreference

- MUC-6 coreference task
  - **MUC-6 corpus**: 30 training documents, 30 test documents
  - Despite the fact that 30 training documents are available, all but one resolver are rule-based
    - UMASS (Lenhert & McCarthy, 1995) resolver is learning-based
  - Best-performing resolver, FASTUS, is rule-based
- MUC-7 coreference task
  - **MUC-7 corpus**: 30 training documents, 20 test documents
  - none of the 7 participating resolvers used machine learning
- Learning-based approaches were not the mainstream
  - But … interest grew when Soon et al. (2001) showed that a learning-based resolver can offer competitive performance

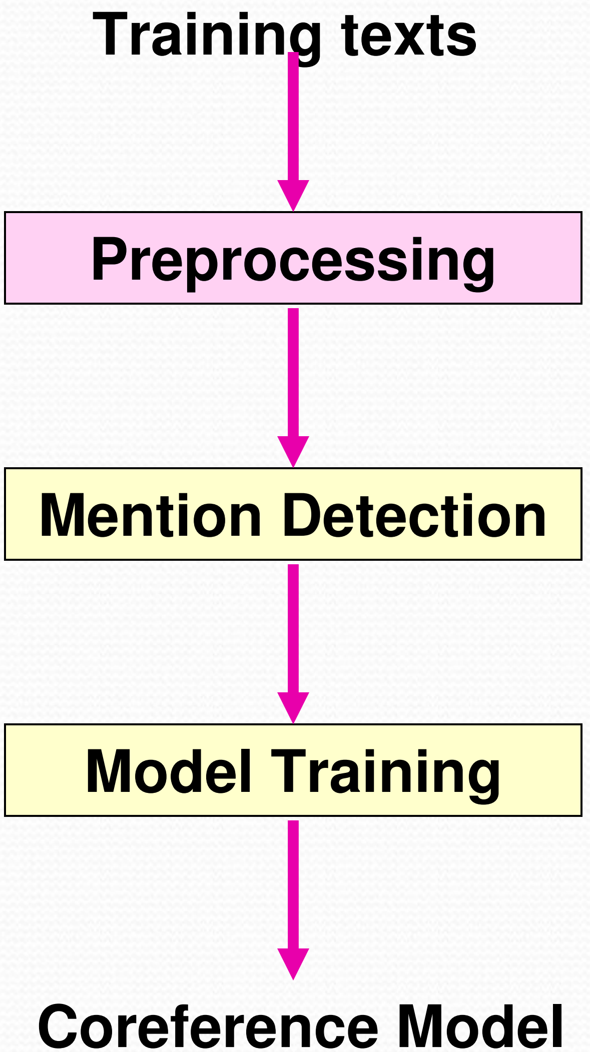
Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
  - Employing semantics and world knowledge
  - Resources and evaluation (corpora, evaluation metrics, …)
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

System Architecture: Training

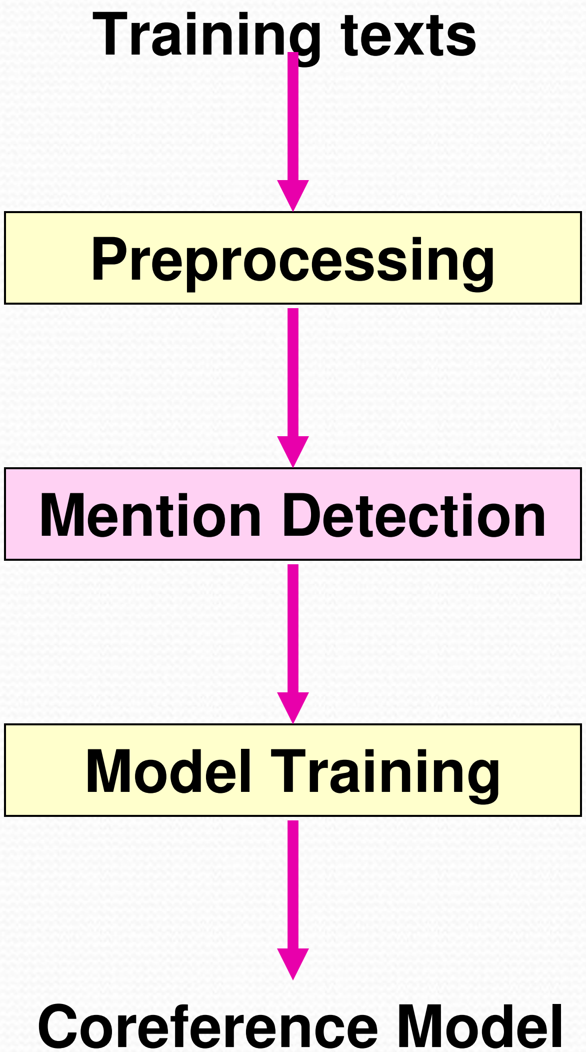


System Architecture: Training



- Tokenization
- Sentence splitting
- part-of-speech tagging

System Architecture: Training



- Tokenization
- Sentence splitting
- part-of-speech tagging

- **Trivial**: extract the hand-annotated (i.e., gold) mentions from training texts

System Architecture: Training

Training texts

Preprocessing

- Tokenization
- Sentence splitting
- part-of-speech tagging

Mention Detection

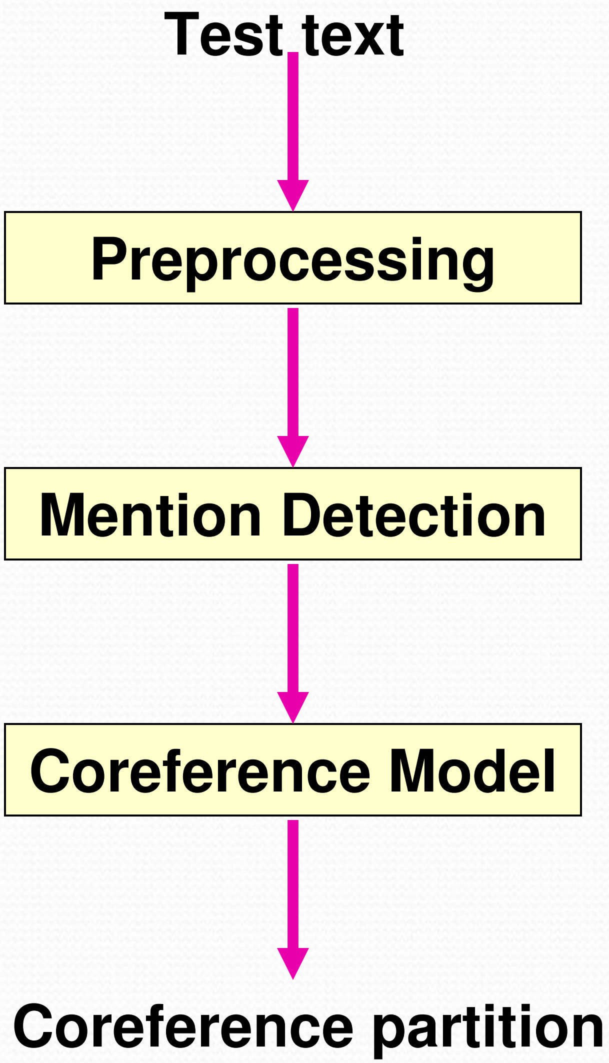
- **Trivial**: extract the hand-annotated (i.e., gold) mentions from training texts

Model Training

- Feature extraction
- Model design and learning

Coreference Model

System Architecture: Testing



System Architecture: Testing

Test text

Preprocessing

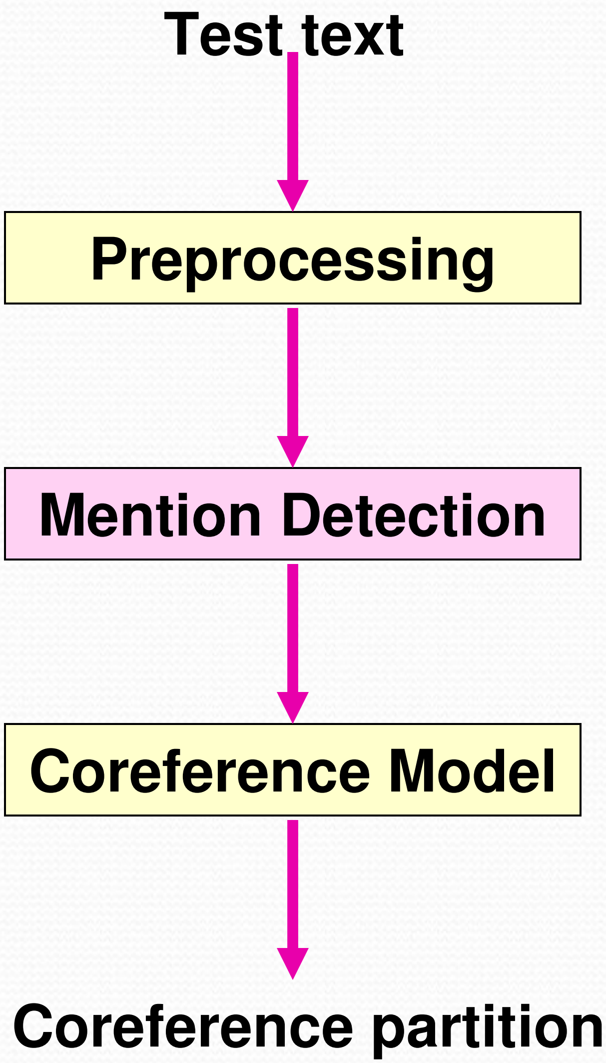
Mention Detection

Coreference Model

Coreference partition

- Tokenization
- Sentence splitting
- part-of-speech tagging

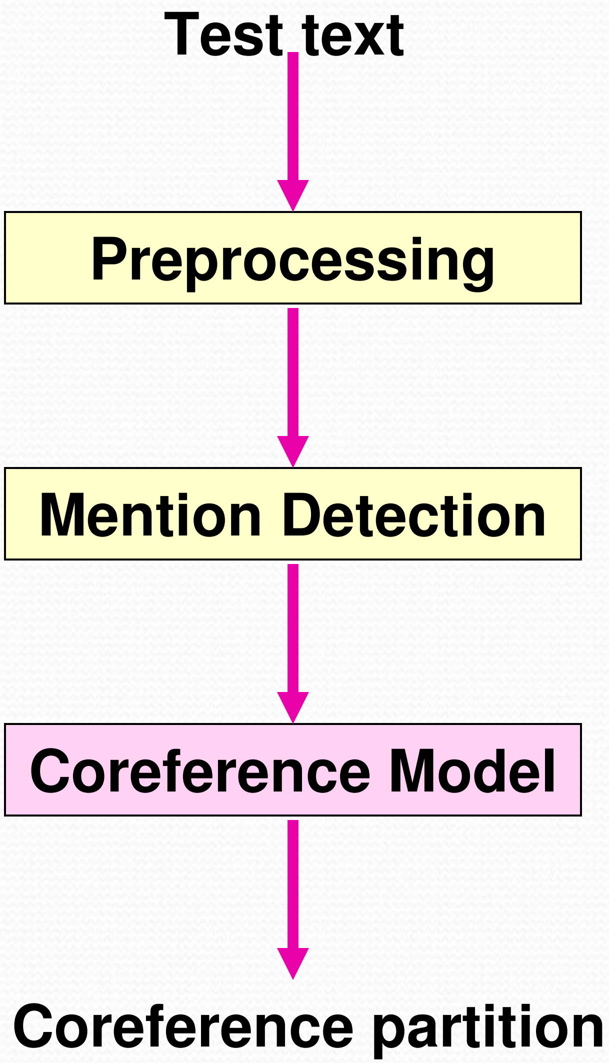
System Architecture: Testing



- Tokenization
- Sentence splitting
- part-of-speech tagging

- **Not-so-trivial**: extract the mentions (pronouns, names, nominals, nested NPs)
- Some researchers reported results on gold mentions, not system mentions
  - Substantially simplified the coref task
    - F-scores of 80s rather than 60s

System Architecture: Testing



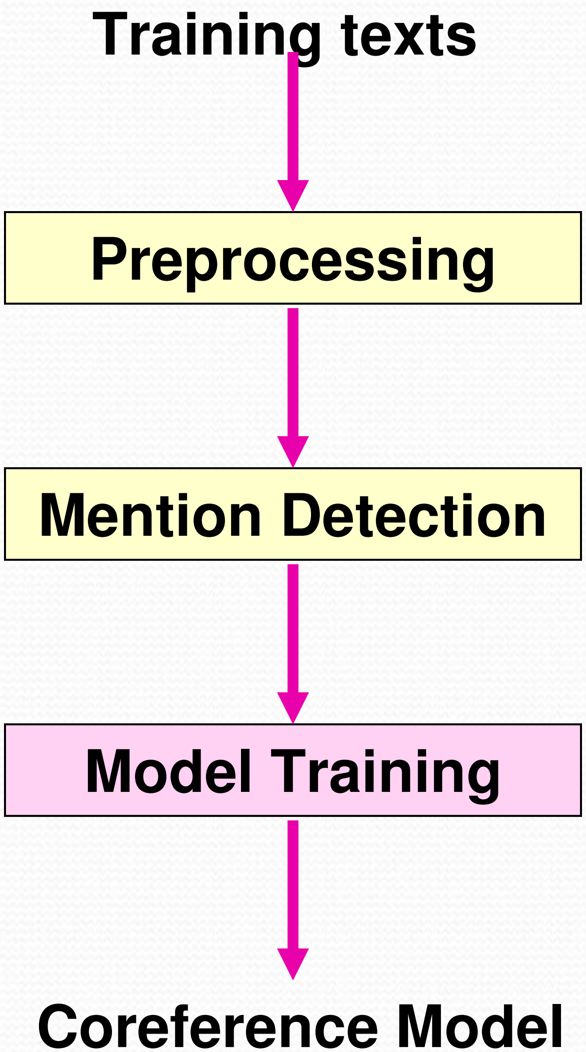
- Tokenization
- Sentence splitting
- part-of-speech tagging

- **Not-so-trivial**: extract the mentions (pronouns, names, nominals, nested NPs)
- Some researchers reported results on gold mentions, not system mentions
  - Substantially simplified the coref task
    - F-scores of 80s rather than 60s

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
  - Employing semantics and world knowledge
  - Resources and evaluation (corpora, evaluation metrics, …)
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

System Architecture: Training



The Mention-Pair Model

- a classifier that, given a description of two mentions, m_i and m_j, determines whether they are coreferent or not
  - coreference as a pairwise classification task

How to train a mention-pair model?

- Training instance creation
  - create one training instance for each pair of mentions from texts annotated with coreference information

[Mary] said [John] hated [her] because [she] …

How to train a mention-pair model?

- Training instance creation
  - create one training instance for each pair of mentions from texts annotated with coreference information

negative

[Mary] said [John] hated [her] because [she] …

How to train a mention-pair model?

- Training instance creation
  - create one training instance for each pair of mentions from texts annotated with coreference information

$$\underbrace{\overbrace{\text{[Mary] said [John]}}^{\textit{negative}}\ \overbrace{\text{hated [her]}}^{\textit{negative}}}\ \text{because [she] ...}$$

How to train a mention-pair model?

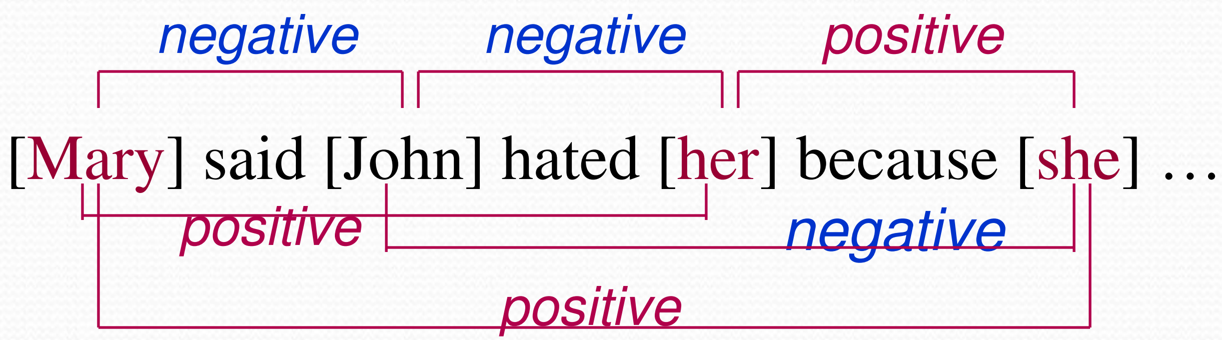
- Training instance creation
  - create one training instance for each pair of mentions from texts annotated with coreference information

negative *negative* *positive*

[Mary] said [John] hated [her] because [she] …

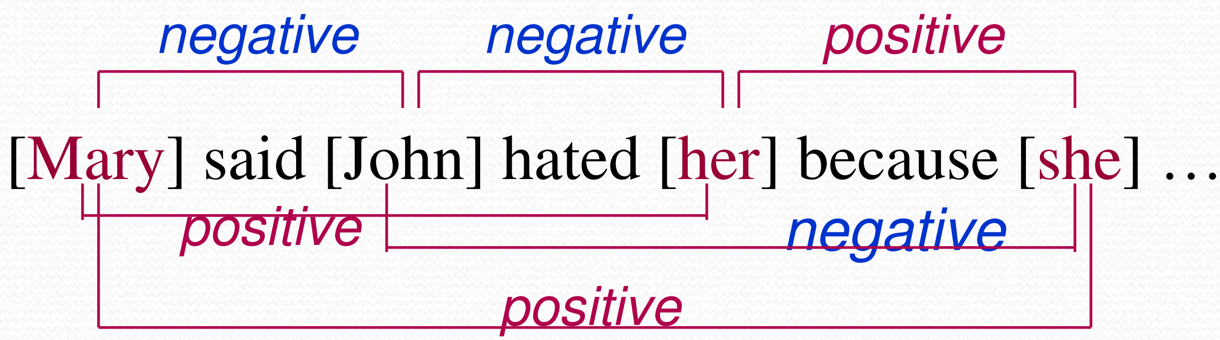
How to train a mention-pair model?

- Training instance creation
  - create one training instance for each pair of mentions from texts annotated with coreference information



How to train a mention-pair model?

- Training instance creation
  - create one training instance for each pair of mentions from texts annotated with coreference information

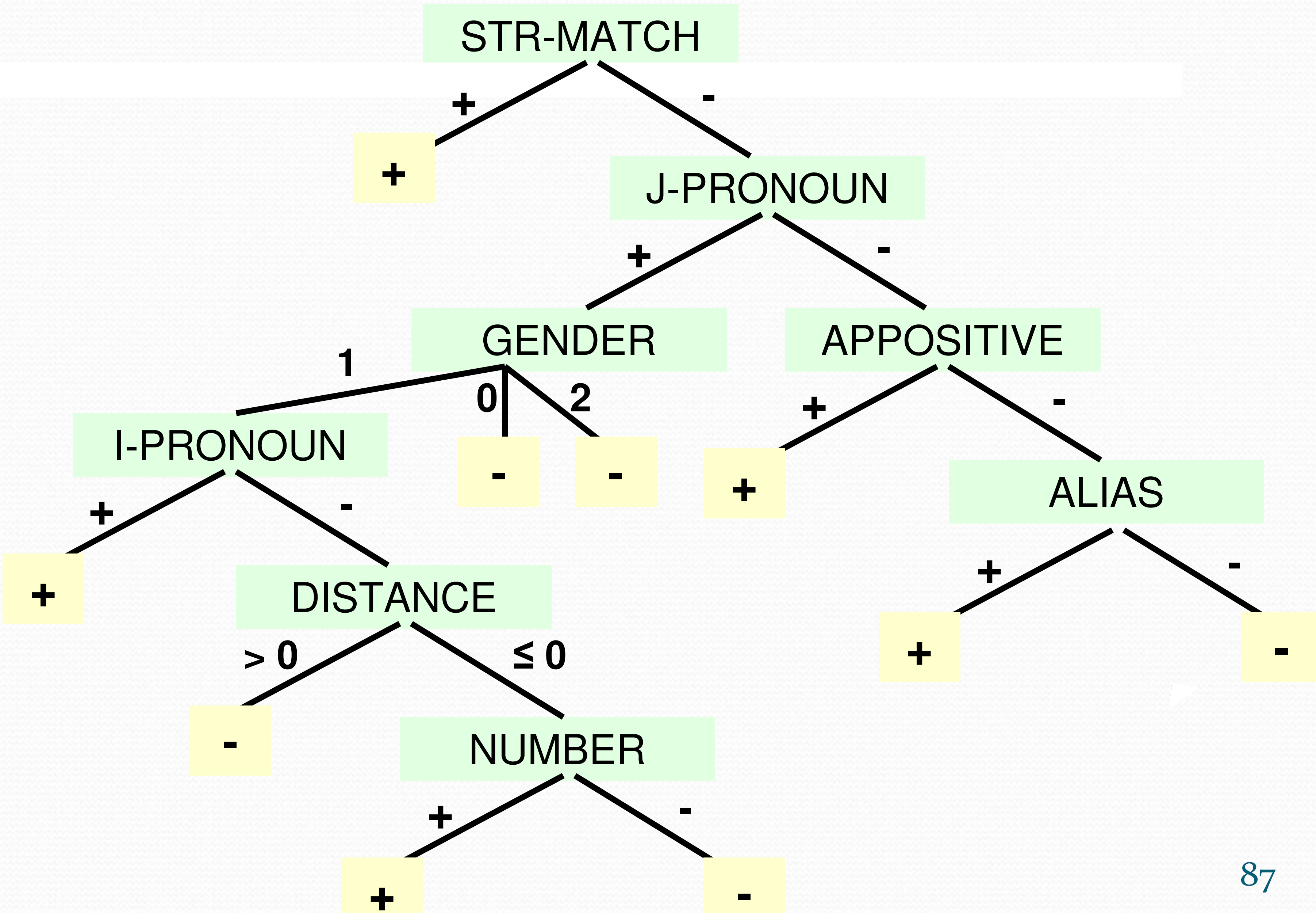


  - Problem: all mention pairs produce a large and skewed data set
  - Soon et al.'s (2001) heuristic instance creation method

How to train a mention-pair model?

- Soon et al.'s feature vector: describes two mentions
  - Exact string match
    - are m_i and m_j the same string after determiners are removed?
  - Grammatical
    - gender and number agreement, $Pronoun_i$?, $Pronoun_j$?, …
  - Semantic
    - semantic class agreement
  - Positional
    - distance between the two mentions
- Learning algorithm
  - C5 decision tree learner

Decision Tree Learned for MUC-6 Data



Applying the mention-pair model

- After training, we can apply the model to a test text
  - Classify each pair of mentions as coreferent or not coreferent
  - **Problem**: the resulting classifications may violate transitivity!

positive

negative

[Jing] likes [him] but [she] …

positive

How to resolve the conflicts?

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive *negative*

[Jing] likes [him] but [she] …

positive

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive

negative

[Jing] likes [him] but [she] …

positive

No antecedent

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive

negative

[Jing] likes [him] but [she] …

positive

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive

negative

[Jing] likes [him] but [she] …

positive

Antecedent is Jing

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive *negative*

[Jing] likes [him] but [she]…

positive

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive

negative

[Jing] likes [him] but [she]…

positive

Antecedent is Jing

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive *negative*

[Jing] likes [him] but [she]…

positive

In the end, all three mentions will be in the same cluster

How to resolve the conflicts?

- Given a test text,
  - process the mentions in a left-to-right manner
  - for each m_j,
    - select as its antecedent the closest preceding mention that is classified as coreferent with m_j
    - otherwise, no antecedent is found for m_j

positive *negative*

[Jing] likes [him] but [she]…

positive

In the end, all three mentions will be in the same cluster

Single-link clustering

MUC Scoring Metric

- **Link-based** metric

- **Key**: {A, B, C}, {D}
- **Response**: {A, B}, {C, D}

- Two links are needed to create the key clusters
  - Response recovered one of them, so recall is ½

- Out of the two links in the response clusters, one is correct
  - So precision is ½

- F-measure

MUC Scoring Metric

- **Link-based** metric

- **Key**: {A, B, C}, {D}, {E}
- **Response**: {A, B}, {C, D}, {E}

- Two links are needed to create the key clusters
  - Response recovered one of them, so recall is ½

- Out of the two links in the response clusters, one is correct
  - So precision is ½

- F-measure

Soon et al. Results

MUC-6			MUC-7		
R	P	F	R	P	F
58.6	67.3	**62.6**	56.1	65.5	**60.4**

Anaphoricity Determination

Determines whether a mention has an antecedent

Anaphoricity Determination

- In single-link clustering, when selecting an antecedent for m_j,
  - select the closest preceding mention that is coreferent with m_j
  - if no such mention exists, m_j is classified as non-anaphoric

Anaphoricity Determination

- In single-link clustering, when selecting an antecedent for m_j,
  - select the closest preceding mention that is coreferent with m_j
  - if no such mention exists, m_j is classified as non-anaphoric

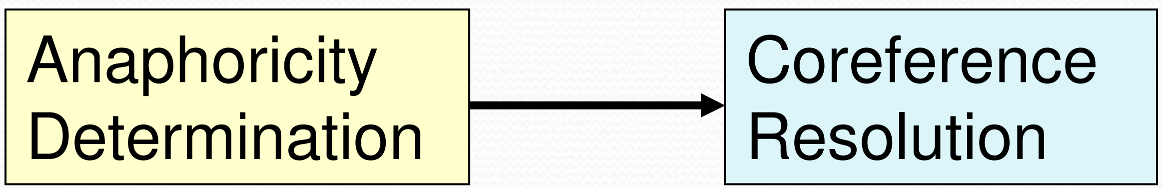
- Why not explicitly determine whether m_j is anaphoric prior to coreference resolution (anaphoricity determination)?

Anaphoricity Determination → Coreference Resolution

Anaphoricity Determination

- In single-link clustering, when selecting an antecedent for m_j,
  - select the closest preceding mention that is coreferent with m_j
  - if no such mention exists, m_j is classified as non-anaphoric

- Why not explicitly determine whether m_j is anaphoric prior to coreference resolution (anaphoricity determination)?



  - If m_j is not anaphoric, shouldn't bother to resolve it
  - pipeline architecture

    → filter non-anaphoric NPs prior to coreference resolution

Anaphoricity Determination

- train a classifier to determine whether a mention is anaphoric (i.e., whether an mention has an antecedent)
  - one training instance per mention
  - 37 features
  - class value is *anaphoric* or *not anaphoric*

Anaphoricity Determination

- train a classifier to determine whether a mention is anaphoric (i.e., whether an mention has an antecedent)
  - one training instance per mention
  - 37 features
  - class value is *anaphoric* or *not anaphoric*

- **Result** on MUC-6/7 (Ng & Cardie, 2002)
  - coreference F-measure drops
    - precision increases, recall drops abruptly
      - many anaphoric mentions are misclassified

        → Error propagation

Some Questions (Circa 2003)

- Is there a model better than the mention-pair model?
- Can anaphoricity determination benefit coreference resolution?

Some Questions (Circa 2003)

- Is there a model better than the mention-pair model?
- Can anaphoricity determination benefit coreference resolution?

Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

Mr. Clinton   *Clinton*   *she*

Weaknesses of the Mention-Pair Model

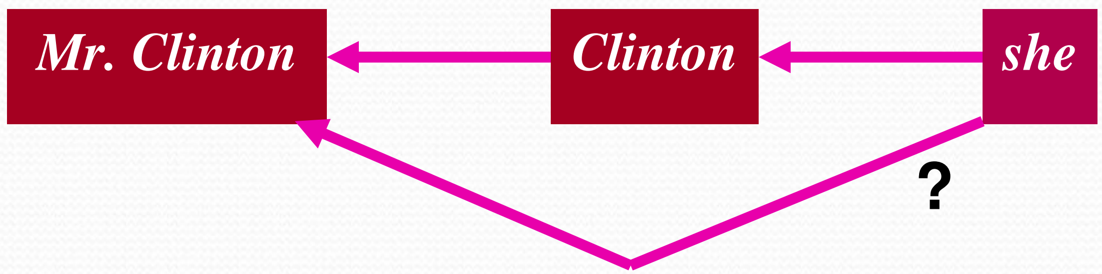
- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision



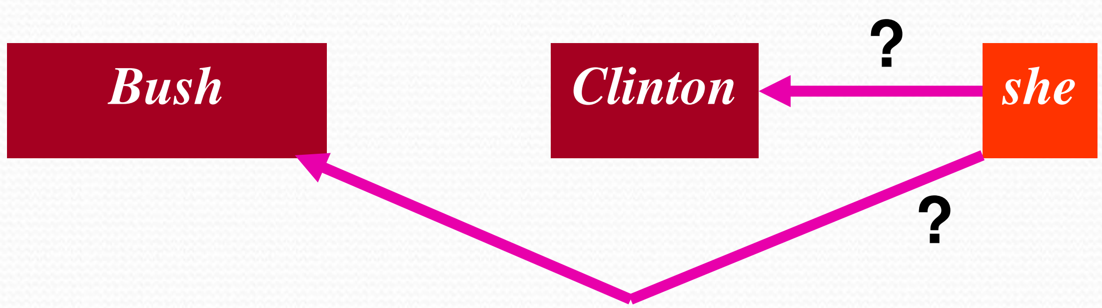
Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision
- **Can't determine which candidate antecedent is the best**
  - only determines how good a candidate is relative to the mention to be resolved, not how good it is relative to the others

Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

- **Can't determine which candidate antecedent is the best**
  - only determines how good a candidate is relative to the mention to be resolved, not how good it is relative to the others

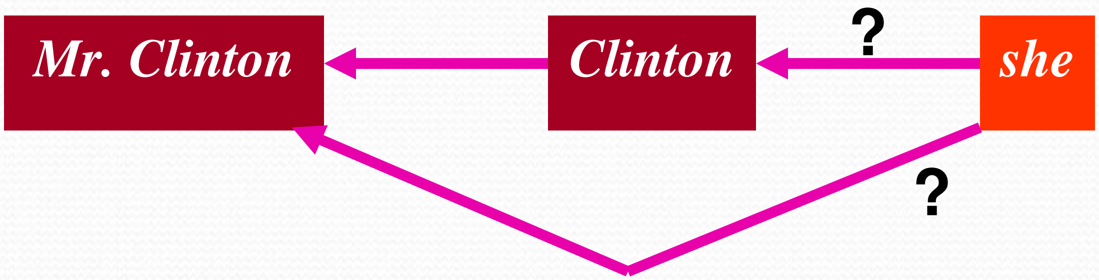


Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

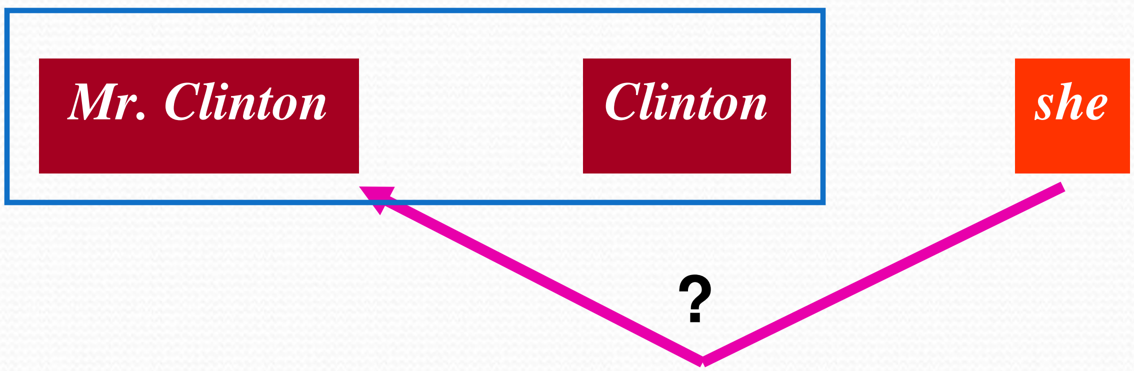
- **Can't determine which candidate antecedent is the best**
  - only determines how good a candidate is relative to the mention to be resolved, not how good it is relative to the others

Improving Model Expressiveness



- Want a coreference model that can tell us whether "she" and a preceding cluster of "she" are coreferent

Improving Model Expressiveness



- Want a coreference model that can tell us whether “she” and a preceding cluster of “she” are coreferent

The Entity-Mention Model

- a classifier that determines whether (or how likely) a mention belongs to a preceding coreference cluster

- more **expressive** than the mention-pair model
  - an instance is composed of a mention and a preceding cluster
  - can employ cluster-level features defined over any subset of mentions in a preceding cluster

Pasula et al. (2003), Luo et al. (2004), Yang et al. (2004, 2008), Daume & Marcu (2005), Culotta et al. (2007), …

The Entity-Mention Model

- a classifier that determines whether (or how likely) a mention belongs to a preceding coreference cluster

- more **expressive** than the mention-pair model
  - an instance is composed of a mention and a preceding cluster
  - can employ cluster-level features defined over any subset of mentions in a preceding cluster
    - is a mention gender-compatible with all mentions in a preceding cluster?
    - is a mention gender-compatible with most of the mentions in it?
    - is a mention gender-compatible with none of them?

Pasula et al. (2003), Luo et al. (2004), Yang et al. (2004, 2008), Daume & Marcu (2005), Culotta et al. (2007), …

Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two mentions may not be sufficient for making an informed coreference decision

- **Can’t determine which candidate antecedent is the best**
  - only determine how good a candidate is relative to the mention to be resolved, not how good it is relative to the others

How to address this problem?

- Idea: train a model that imposes a **ranking** on the candidate antecedents for a mention to be resolved
  - so that it assigns the highest rank to the correct antecedent

How to address this problem?

- Idea: train a model that imposes a **ranking** on the candidate antecedents for a mention to be resolved
  - so that it assigns the highest rank to the correct antecedent
- A ranker allows all candidate antecedents to be compared
  - allows us find the best candidate antecedent for a mention

How to address this problem?

- Idea: train a model that imposes a **ranking** on the candidate antecedents for a mention to be resolved
  - so that it assigns the highest rank to the correct antecedent
- A ranker allows all candidate antecedents to be compared
  - allows us find the best candidate antecedent for a mention
- There is a natural resolution strategy for a mention-ranking model
  - A mention is resolved to the highest-ranked candidate antecedent

Yang et al. (2003), Iida et al., (2003), Denis & Baldridge (2007, 2008), …

Caveat

- Since a mention ranker only imposes a ranking on the candidates, it cannot determine whether a mention is anaphoric
  - Need to train a classifier to perform anaphoricity determination

Recap

Problem	Entity Mention	Mention Ranking
Limited expressiveness	✓	✗
Cannot determine best candidate	✗	✓

Recap

Problem	Entity Mention	Mention Ranking
Limited expressiveness	✓	✗
Cannot determine best candidate	✗	✓

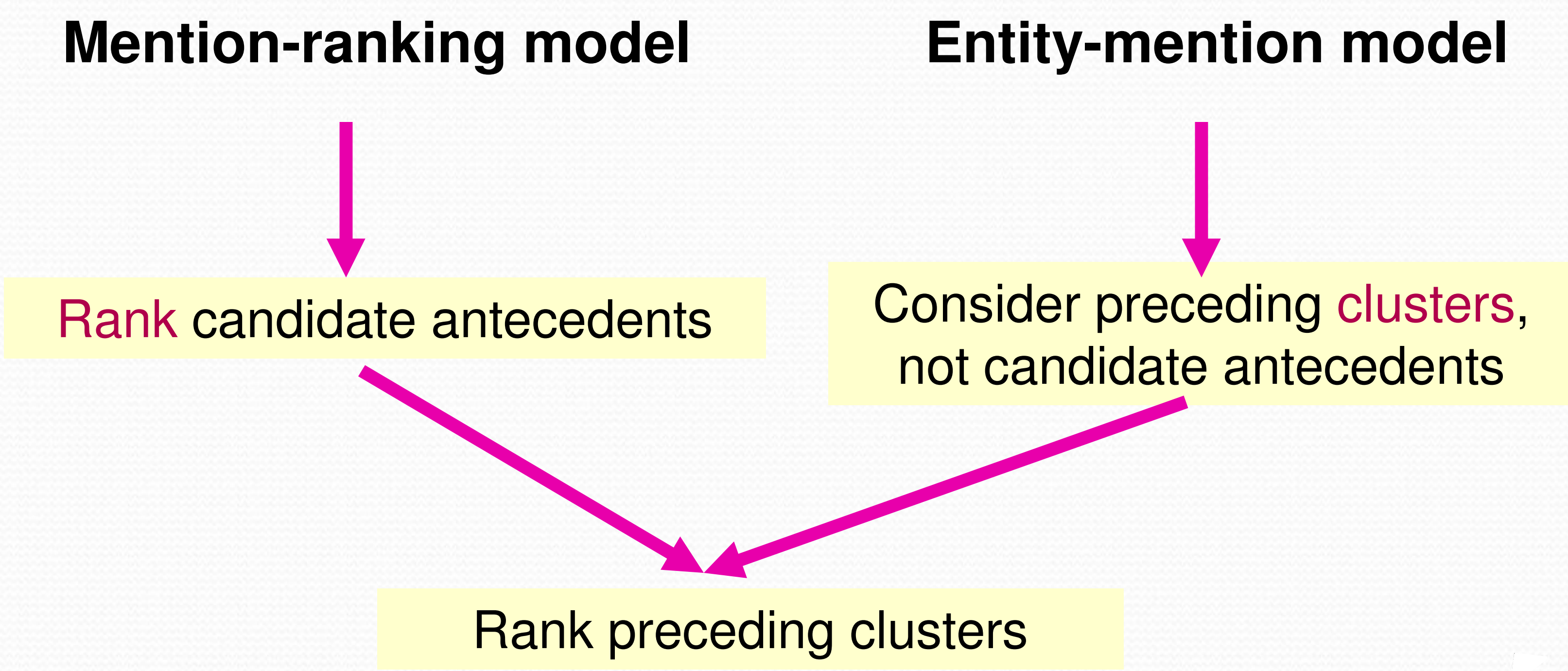
Can we combine the strengths of these two model?

Mention-ranking model

Rank candidate antecedents

Entity-mention model

Consider preceding clusters, not candidate antecedents


Mention-ranking model
Entity-mention model
Rank candidate antecedents
Consider preceding clusters,
not candidate antecedents
Rank preceding clusters

The Cluster-Ranking Model

Mention-ranking model

Rank candidate antecedents

Entity-mention model

Consider preceding clusters, not candidate antecedents

Rank preceding clusters

Rahman & Ng EMNLP'09

The Cluster-Ranking Model

- **Training**
  - train a ranker to rank preceding clusters
- **Testing**
  - resolve each mention to the highest-ranked preceding cluster

The Cluster-Ranking Model

- **Training**
  - train a ranker to rank preceding clusters
- **Testing**
  - resolve each mention to the highest-ranked preceding cluster

After many years of hard work … finally came up with cluster rankers, which are conceptually similar to Lappin & Leass' (1994) pronoun resolver --- Bonnie Webber (2010)

The Cluster-Ranking Model

- As a ranker, the cluster-ranking model cannot determine whether a mention is anaphoric
  - Before resolving a mention, we still need to use an anaphoricity classifier to determine if it is anaphoric
    - yields a pipeline architecture
- Potential problem
  - errors made by the anaphoricity classifier will be propagated to the coreference resolver

Potential Solution

- Jointly learn anaphoricity and coreference

How to jointly learn anaphoricity and coreference resolution?

How to jointly learn anaphoricity and coreference resolution?

- Currently, the cluster-ranking model is trained to rank preceding clusters for a given mention, m_j

- In joint modeling, the cluster-ranking model is trained to rank preceding clusters + null cluster for a given mention, m_j
  - want to train the model such that the null cluster has the highest rank if m_j is non-anaphoric

- Joint training allows the model to simultaneously learn whether to resolve an mention, and if so, which preceding cluster is the best

How to apply the joint model?

- During testing, resolve m_j to the highest-ranked cluster
  - if highest ranked cluster is null cluster, m_j is non-anaphoric

- Same idea can be applied to mention-ranking models

Experimental Setup

- The English portion of the ACE 2005 training corpus
  - 599 documents coref-annotated on the ACE entity types
  - 80% for training, 20% for testing
- Mentions extracted automatically using a mention detector
- Scoring programs: recall, precision, F-measure
  - B^3 (Bagga & Baldwin, 1998)
  - CEAF (Luo, 2005)

B^3 Scoring Metric

- Mention-based metric
  - Computes per-mention recall and precision
  - Aggregates per-mention scores into overall scores

- Key: {A, B, C}, {D}
- Response: {A, B}, {C, D}

- To compute the recall and precision for A:
  - A's key cluster and response cluster have 2 overlapping mentions
  - 2 of the 3 mentions in key cluster is recovered, so recall = 2/3
  - 2 mentions in response cluster, so precision = 2/2

B^3 Scoring Metric

- Mention-based metric
  - Computes per-mention recall and precision
  - Aggregates per-mention scores into overall scores

- Key: {A, B, C}, {D}, {E}
- Response: {A, B}, {C, D}, {E}

- To compute the recall and precision for A:
  - A's key cluster and response cluster have 2 overlapping mentions
  - 2 of the 3 mentions in key cluster is recovered, so recall = 2/3
  - 2 mentions in response cluster, so precision = 2/2

CEAF Scoring Metric

- Entity/Cluster-based metric

- Computes the best bipartite matching between the set of key clusters and the set of response clusters

- Key: {A, B, C}, {D, E}
- Response: {A, B}, {C, D}, {E}

CEAF Scoring Metric

- Entity/Cluster-based metric

- Computes the best bipartite matching between the set of key clusters and the set of response clusters

- Key: {A, B, C}, {D, E}
- Response: {A, B}, {C, D}, {E}

- Recall: (2+1)/(3+2)
- Precision: (2+1)/3

CEAF Scoring Metric

- Entity/Cluster-based metric

- Computes the best bipartite matching between the set of key clusters and the set of response clusters

- Key: {A, B, C}, {D, E}
- Response: {A, B}, {C, D}, {E}

- Recall: (2+1)/(3+2)
- Precision: (2+1)/3

Experimental Setup

- Three baseline coreference models
  - mention-pair, entity-mention, mention-ranking models

Results (Mention-Pair Baseline)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**

Results (Entity-Mention Baseline)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**

Results (Pipeline Mention-Ranking)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**
Mention-Ranking Baseline (Pipeline)	52.3	61.8	**56.6**	51.6	56.7	**54.1**

- Apply an anaphoricity classifier to filter non-anaphoric NPs

Results (Joint Mention-Ranking)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**
Mention-Ranking Baseline (Pipeline)	52.3	61.8	**56.6**	51.6	56.7	**54.1**
Mention-Ranking Baseline (Joint)	50.4	65.5	**56.9**	53.0	58.5	**55.6**

Results (Pipeline Cluster Ranking)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**
Mention-Ranking Baseline (Pipeline)	52.3	61.8	**56.6**	51.6	56.7	**54.1**
Mention-Ranking Baseline (Joint)	50.4	65.5	**56.9**	53.0	58.5	**55.6**
Cluster-Ranking Model (Pipeline)	55.3	63.7	**59.2**	54.1	59.3	**56.6**

- Apply an anaphoricity classifier to filter non-anaphoric mentions

Results (Joint Cluster Ranking)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**
Mention-Ranking Baseline (Pipeline)	52.3	61.8	**56.6**	51.6	56.7	**54.1**
Mention-Ranking Baseline (Joint)	50.4	65.5	**56.9**	53.0	58.5	**55.6**
Cluster-Ranking Model (Pipeline)	55.3	63.7	**59.2**	54.1	59.3	**56.6**
Cluster-Ranking Model (Joint)	54.4	70.5	**61.4**	56.7	62.6	**59.5**

Results (Joint Cluster Ranking)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**
Mention-Ranking Baseline (Pipeline)	52.3	61.8	**56.6**	51.6	56.7	**54.1**
Mention-Ranking Baseline (Joint)	50.4	65.5	**56.9**	53.0	58.5	**55.6**
Cluster-Ranking Model (Pipeline)	55.3	63.7	**59.2**	54.1	59.3	**56.6**
Cluster-Ranking Model (Joint)	54.4	70.5	**61.4**	56.7	62.6	**59.5**

- In comparison to the best baseline (joint mention-ranking),
  - significant improvements in F-score for both B^3 and CEAF
  - due to simultaneous rise in recall and precision

Results (Joint Cluster Ranking)

	B^3			CEAF		
	R	P	F	R	P	F
Mention-Pair Baseline	50.8	57.9	**54.1**	56.1	51.0	**53.4**
Entity-Mention Baseline	51.2	57.8	**54.3**	56.3	50.2	**53.1**
Mention-Ranking Baseline (Pipeline)	52.3	61.8	**56.6**	51.6	56.7	**54.1**
Mention-Ranking Baseline (Joint)	50.4	65.5	**56.9**	53.0	58.5	**55.6**
Cluster-Ranking Model (Pipeline)	55.3	63.7	**59.2**	54.1	59.3	**56.6**
Cluster-Ranking Model (Joint)	54.4	70.5	**61.4**	56.7	62.6	**59.5**

- Joint modeling is better than pipeline modeling

The CoNLL Shared Tasks

- Much recent work on entity coreference resolution was stimulated in part by the availability of the OntoNotes corpus and its use in two coreference shared tasks
  - CoNLL-2011 and CoNLL-2012

- OntoNotes coreference: unrestricted coreference

Two Top Shared Task Systems

- Multi-pass sieve approach (Lee et al., 2011)
  - Winner of the CoNLL-2011 shared task
    - English coreference resolution
- Latent tree-based approach (Fernandes et al.,2012)
  - Winner of the CoNLL-2012 shared task
    - Multilingual coreference resolution (English, Chinese, Arabic)

Two Recent Approaches

- Multi-pass sieve approach (Lee et al., 2011)
  - Winner of the CoNLL-2011 shared task
    - English coreference resolution
- Latent tree-based approach (Fernandes et al.,2012)
  - Winner of the CoNLL-2012 shared task
    - Multilingual coreference resolution (English, Chinese, Arabic)

Stanford's Sieve-Based Approach

- Rule-based resolver
  - Each rule enables coreference links to be established
  - Rules are partitioned into 12 components (or sieves) arranged as a pipeline

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

The 12 Sieves

Each sieve is composed of a set of rules for establishing coreference links

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Two mentions are coreferent if they have the same string

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Two mentions are coreferent if the strings obtained by dropping the text after their head words are identical

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Two mentions are coreferent if they are in an appositive construction
- Two mentions are coreferent if they are in a copular construction
- Two mentions are coreferent if one is a relative pronoun that modifies the head of the antecedent NP
- …

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

Sieves implementing different kinds of string matching

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

Posits two mentions as coreferent if they are linked by a WordNet lexical chain

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

Resolves a pronoun to a mention that agree in number, gender, person, number & semantic class

A Few Notes on Sieves

- Each sieve is composed of a set of rules for establishing coreference links

- Sieves are ordered in decreasing order of precision

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Rules in the DP sieve has the highest precision

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Rules in the DP sieve has the highest precision
- … followed by those in Exact String Match (Sieve 2)

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Rules in the DP sieve has the highest precision
- … followed by those in Exact String Match (Sieve 2)
- … followed by those in Relaxed String Match (Sieve 3)

The 12 Sieves

- Discourse Processing
- Exact String Match
- Relaxed String Match
- Precise Constructs
- Strict Head Matching A,B,C
- Proper Head Word Match
- Alias Sieve
- Relaxed Head Matching
- Lexical Chain
- Pronouns

- Rules in the DP sieve has the highest precision
- … followed by those in Exact String Match (Sieve 2)
- … followed by those in Relaxed String Match (Sieve 3)
- Those in the Pronouns sieve have the lowest precision

A Few Notes on Sieves

- Each sieve is composed of a set of rules for establishing coreference links
- Sieves are ordered in decreasing order of precision
- Coreference clusters are constructed incrementally
  - Each sieve builds on the partial coreference clusters constructed by the preceding sieves
    - Enables the use of rules that link two clusters
      - Rules can employ features computed over one or both clusters
        - E.g., are there mentions in the 2 clusters that have same head?
      - Resemble entity-mention models

Evaluation

- Corpus
  - Training: CoNLL-2011 shared task training corpus
  - Test: CoNLL-2011 shared task test corpus

- Scoring programs: recall, precision, F-measure
  - MUC (Vilain et al., 1995)
  - B^3 (Bagga & Baldwin, 1998)
  - $CEAF_e$ (Luo, 2005)
  - CoNLL (unweighted average of MUC, B^3 and $CEAF_e$ F-scores)

Results (Closed Track)

System	MUC	B^3	$CEAF_e$	CoNLL
Rank 1: Multi-Pass Sieves	59.6	68.3	45.5	57.8
Rank 2: Label Propagation	59.6	67.1	41.3	56.0

- Caveat
  - Mention detection performance could have played a role
    - Best system's mention detection results: R/75, P/67, F/71
    - 2^{nd} best system's mention detection results: R/92, P/28, F/43

Lessons

- Easy-first coreference resolution
  - Exploit easy relations to discover hard relations

- Results seem to suggest that humans are better at combining features than machine learners
  - Better feature induction methods for combining primitive features into more powerful features?

Another Sieve-Based Approach

- Ratinov & Roth (EMNLP 2012)

- **Learning-based**
  - Each sieve is a machine-learned classifier

- Later sieves can **override** earlier sieves' decisions
  - Can recover from errors as additional evidence is available

Ratinov & Roth's 9 Sieves (Easy First)

- Each sieve is a mention-pair model applicable to a subset of mention pairs

1. Nested (e.g., {city of {Jurusalem}})
2. Same Sentence both Named Entities (NEs)
3. Adjacent (Mentions closest to each other in dependency tree)
4. Same Sentence NE&Nominal (e.g., Barack Obama, president)
5. Different Sentence two NEs
6. Same Sentence No Pronouns
7. Different Sentence Closest Mentions (no intervening mentions)
8. Same Sentence All Pairs
9. All Pairs

Information Propagation

- Encoded as features

- Decision-encoding features at sieve i
  - whether m_j and m_k are posited as coreferent by sieve 1, sieve 2, …, sieve i-1
  - whether m_j and m_k are in the same coreference cluster after sieve 1, sieve 2, …, sieve i-1
  - the results of various set operations applied to the cluster containing m_j and the cluster containing m_k
    - set identity, set containment, set overlap, …

Two Recent Approaches

- Multi-pass sieve approach (Lee et al., 2011)
  - Winner of the CoNLL-2011 shared task
    - English coreference resolution
- Latent tree-based approach (Fernandes et al.,2012)
  - Winner of the CoNLL-2012 shared task
    - Multilingual coreference resolution (English, Chinese, Arabic)

Training the Mention-Pair Model

- The mention-pair model determines whether two mentions are coreferent or not
- Each training example corresponds to two mentions
  - Class value indicates whether they are coreferent or not
- Soon et al. train the model using a decision tree learner
- But we can train it using other learners, such as the perceptron learning algorithm

The Perceptron Learning Algorithm

- Parameterized by a weight vector w
- Learns a linear function
  - Output of perceptron y = w • x

weight vector

feature vector (features computed based on two mentions)

The Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + x_i$ // update the weights

  until convergence

- An iterative algorithm
- An error-driven algorithm:
  weight vector is updated whenever a mistake is made

- Observation (McCallum & Wellner, 2004):
  - Since the goal is to output a coreference partition, why not learn to predict a partition directly?

- They modified the perceptron algorithm in order to learn to predict a coreference partition
  - each training example corresponds to a document
    - Class value is the correct coreference partition of the mentions

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_{i,j,l} \lambda_l f_l(x_i, x_j, y_{ij}) + \sum_{i,j,k,l'} \lambda_{l'} f_{l'}(y_{ij}, y_{jk}, y_{ik}) \right)$$

The Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + x_i$ // update the weights

  until convergence

The Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

The Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- Each example x_i corresponds to a partition. What features should be used to represent a partition?

The Perceptron Learning Algorithm

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

- Each example x_i corresponds to a partition. What features should be used to represent a partition?
  - still pairwise features, but they are computed differently
    - **Before**: do the two mentions have compatible gender?
    - **Now**: how many coreferent pairs have compatible gender?

The Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

each value in $F(y_i')$ is the sum of the feature values over all mention pairs in y_i'

- Each example x_i corresponds to a
  should be used to represent a par
  - still pairwise features, but they are computed differently
    - **Before**: do the two mentions have compatible gender?
    - **Now**: how many coreferent pairs have compatible gender?

The Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- How can we predict most prob. partition using the current w?

Method 1:

- For each possible partition p, compute $w \bullet F(p)$
- Select the p with the largest $w \bullet F(p)$

The Perceptron Learning Algorithm

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

- How can we predict most prob. partition using the current w?

Method 1:

- For each possible partition p, compute w • F(p)
- Select the p with the largest w • F(p)

Computationally intractable

The Perceptron Learning Algorithm

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

- How can we predict most prob. partition using the current w?

  Method 2:

  - Approximate the optimal partition given the current w using correlation clustering

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- How can we predict most prob. partition using the current w?

Method 2:

- Approximate the optimal partition given the current w using correlation clustering

Observation

- Now we have an algorithm that learns to partition
  - Can we further improve?

Observation

- Now we have an algorithm that learns to partition
  - Can we further improve?
- Recall that to compute each pairwise feature, we sum the values of the pairwise feature over the coreferent pairs
  - E.g., number of coreferent pairs that have compatible gender

Observation

- Now we have an algorithm that learns to partition
  - Can we further improve?

- Recall that to compute each pairwise feature, we sum the values of the pairwise feature over the coreferent pairs
  - E.g., number of coreferent pairs that have compatible gender

- We are learning a partition from all coreferent pairs
  - But … learning from all coreferent pairs is hard
    - Some coreferent pairs are hard to learn from
  - And … we don't need to learn from all coreferent pairs
    - We do **not** need all coreferent pairs to construct a partition

Observation

- To construct a coreference partition, we need to construct each coreference cluster
  - To construct a coreference cluster with n mentions, we need only n-1 links

Queen Elizabeth
her

husband
King George VI
the King
his

a viable monarch

speech impediment

a renowned speech therapist

Observation

- To construct a coreference partition, we need to construct each coreference cluster
  - To construct a coreference cluster with n mentions, we need only n-1 links

Queen Elizabeth

her

husband

King George VI

the King

his

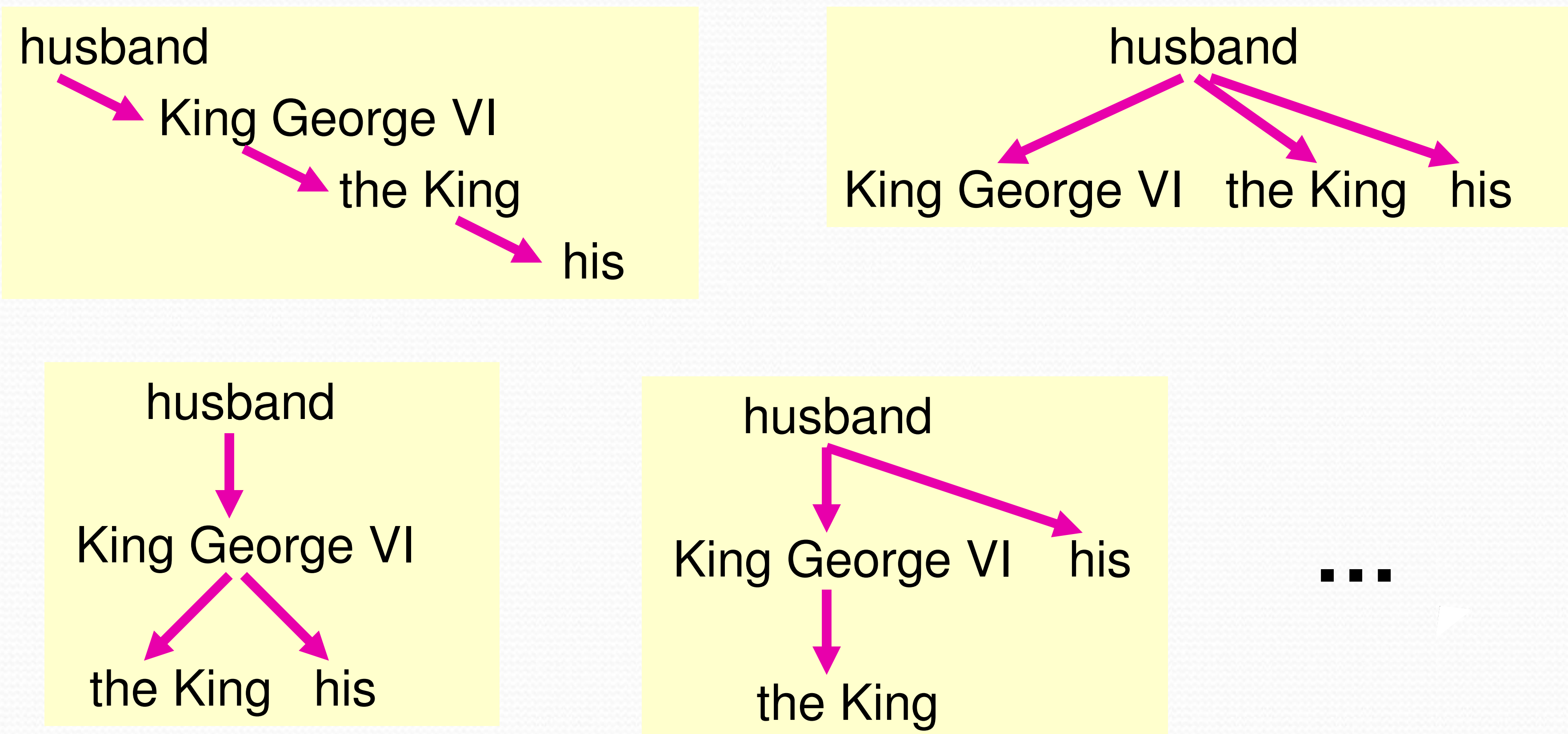
a viable monarch

speech impediment

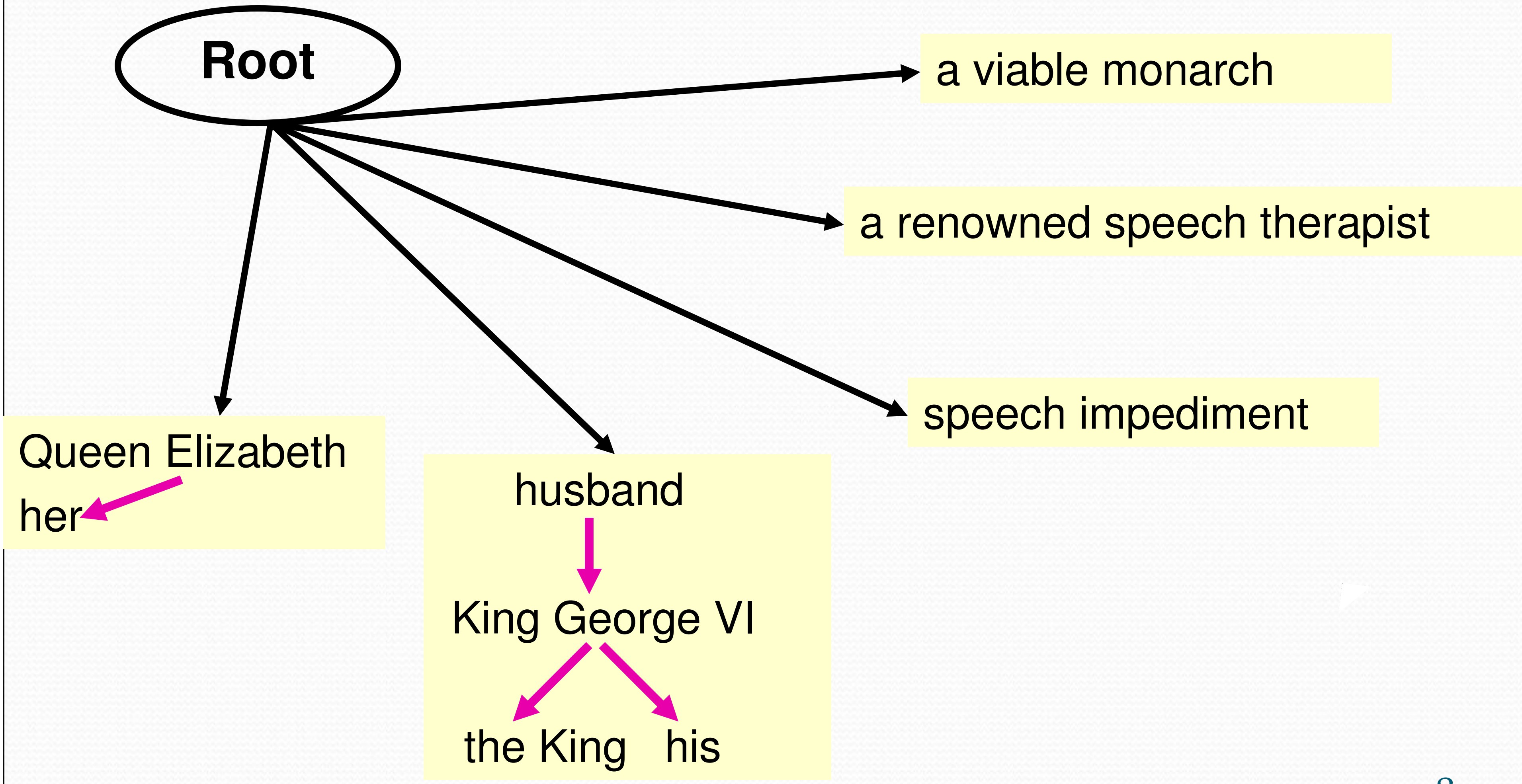
a renowned speech therapist

Observations

- There are many ways links can be chosen



Coreference Tree



Coreference Tree

- A coreference tree is an equivalent representation of a coreference partition
  - Latent tree-based approach (Fernandes et al., 2012)
    - Learn coreference trees rather than coreference partitions
    - But … many coreference trees can be created from one coreference partition … which one should we learn?

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct partition y_i

predict the most probable partition y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

for each training example x_i

predict the most probable tree y_i'

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- How can we predict most prob. tree y_i' using the current w?

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

- How can we predict most prob. tree y_i' using the current w?

  Method 1:
  - For each possible tree t, compute $w \bullet F(t)$
  - Select the t with the largest $w \bullet F(t)$

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

each value in $F(y_i')$ is the sum of the feature values over all the edges in y_i'

- How can we predict most prob. tr

  Method 1:

  - For each possible tree t, compute w • F(t)
  - Select the t with the largest w • F(t)

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

  for each training example x_i

    (1) predict the class of x_i using the current w

    (2) if the predicted class is not equal to the correct class

      $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

 until convergence

- How can we predict most prob. tree y_i' using the current w?

  Method 2:
  - Run the Chu-Liu/Edmonds' algorithm to find max spanning tree

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- Each x_i is labeled with correct tree y_i. Since many correct trees can be created from a partition, which one should be used?

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- Each x_i is labeled with correct tree y_i. Since many correct trees can be created from a partition, which one should be used?
  - Heuristically?

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

for each training example x_i

(1) predict the class of x_i using the current w

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- Each x_i is labeled with correct tree y_i. Since many correct trees can be created from a partition, which one should be used?
  - Select the correct y_i with the largest $w \bullet F(y_i)$

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

x_i = document w/ correct tree y_i

for each training example x_i

(1) predict the class of x_i using the current w

predict the most probable tree y_i'

(2) if the predicted class is not equal to the correct class

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

until convergence

- Eac… …ct tree y_i. Since many correct trees can… …on, which one should be used?
  - …e largest $w \bullet F(y_i)$

Select the correct tree that is best given the current model

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (1) predict the class of x_i using the current w

  (2) if the predicted class is not equal to the correct class

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

x_i = document w/ correct tree y_i

predict the most probable tree y_i'

- Eac[...]ct tr[...]t trees can[...]on,[...]ed?
  - S[...]e largest y_i

Select the correct tree that is best given the current model

A different correct tree will be selected in each iteration

The Structured Perceptron Learning Algorithm

- Initialize w
- Loop

  for each training example x_i

  (0) select the correct tree y_i using the current w

  (1) predict the most prob. tree y_i' of x_i using the current w

  (2) if the predicted tree is not equal to the correct tree

  $w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

  until convergence

The Latent Structured Perceptron Learning Algorithm (Joachims & Yu, 2009)

- Initialize w
- Loop

for each training example x_i

(0) select the correct tree y_i using the current w

(1) predict the most prob. tree y_i' of x_i using the current w

(2) if the predicted tree is not equal to the correct tree

$w \leftarrow w + F(y_i) - F(y_i')$ // update the weights

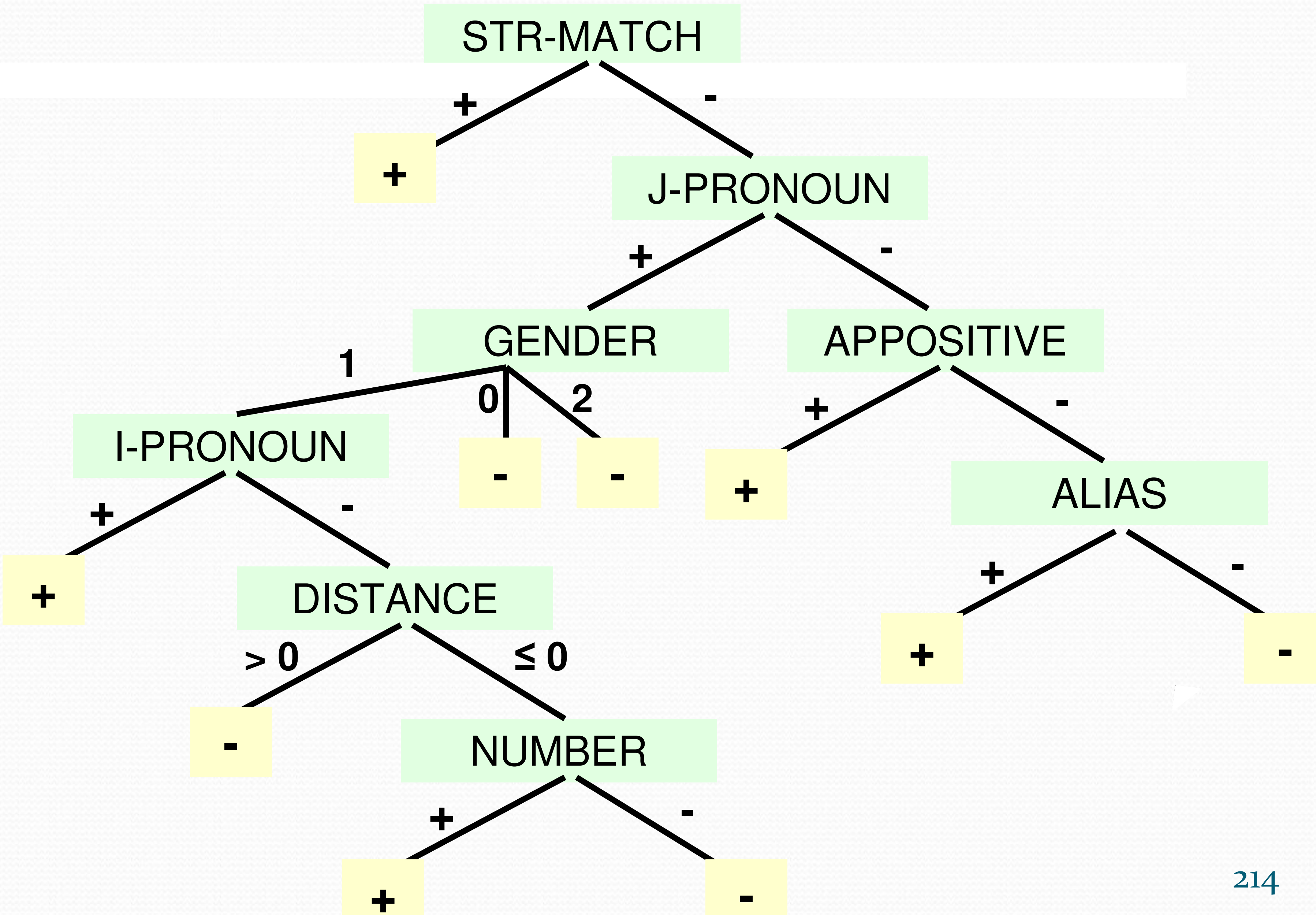
until convergence

What's left?

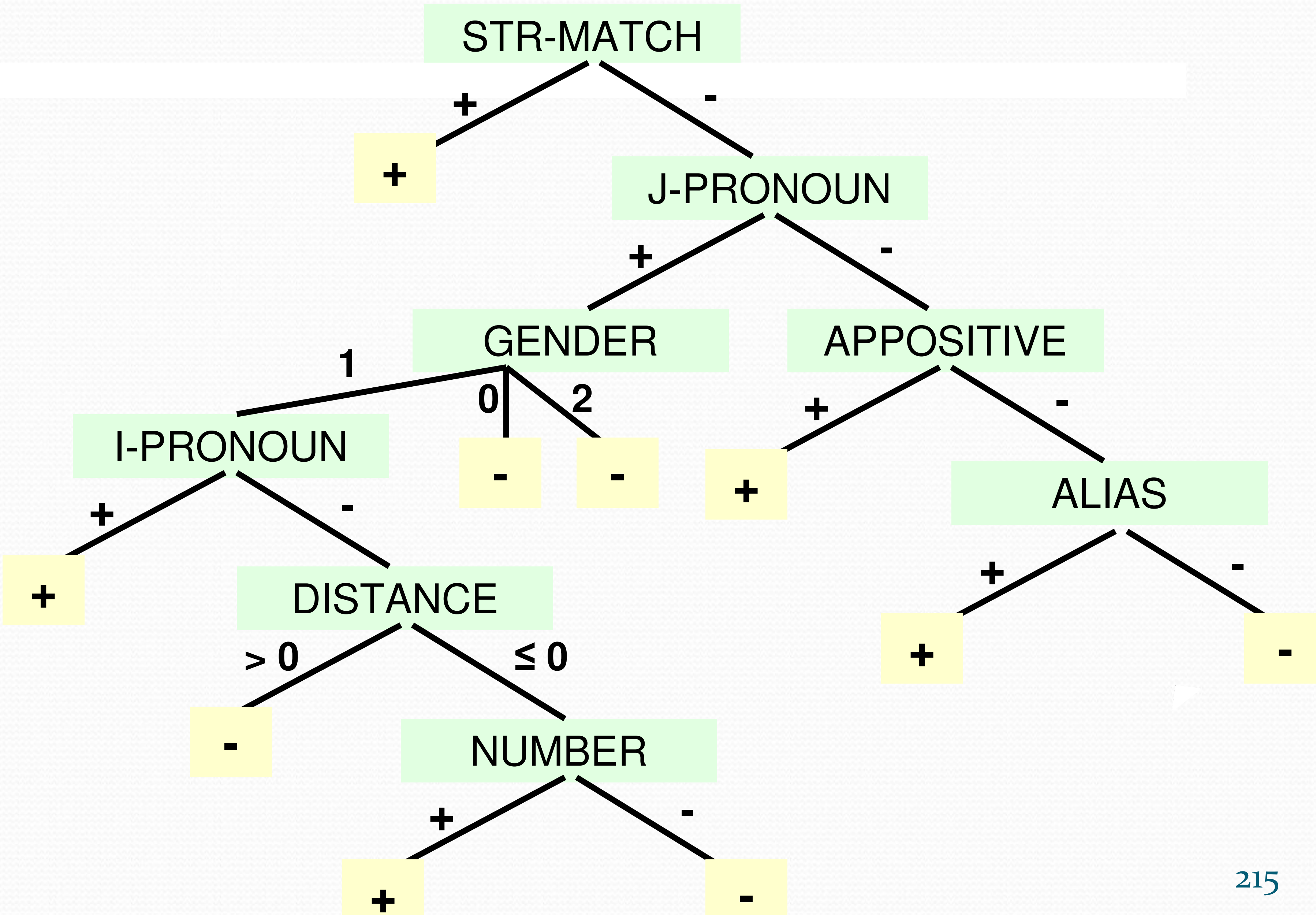
- Recall that …

- Humans are better at combining features than machine learners
  - Better feature induction methods for combining primitive features into more powerful features?

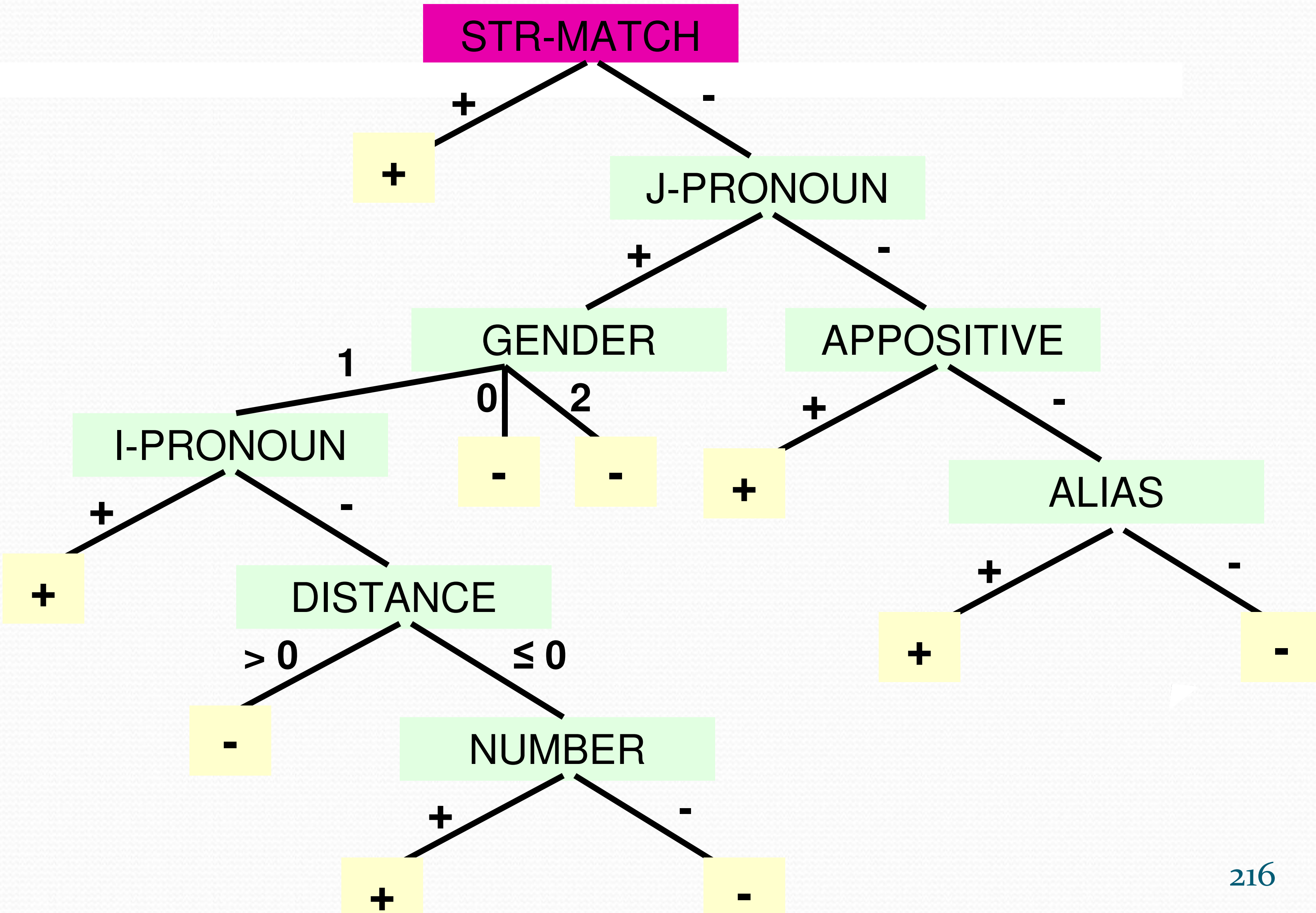
- The latent tree-based model employs feature induction
  - Entropy-based feature induction
    - given the same training set used to train a mention-pair model, train a decision tree classifier


STR-MATCH
+
-
+
J-PRONOUN
+
-
GENDER
APPOSITIVE
1
0
2
I-PRONOUN
-
-
+
-
+
+
ALIAS
+
-
+
DISTANCE
> 0
≤ 0
+
-
-
NUMBER
+
-
+
-

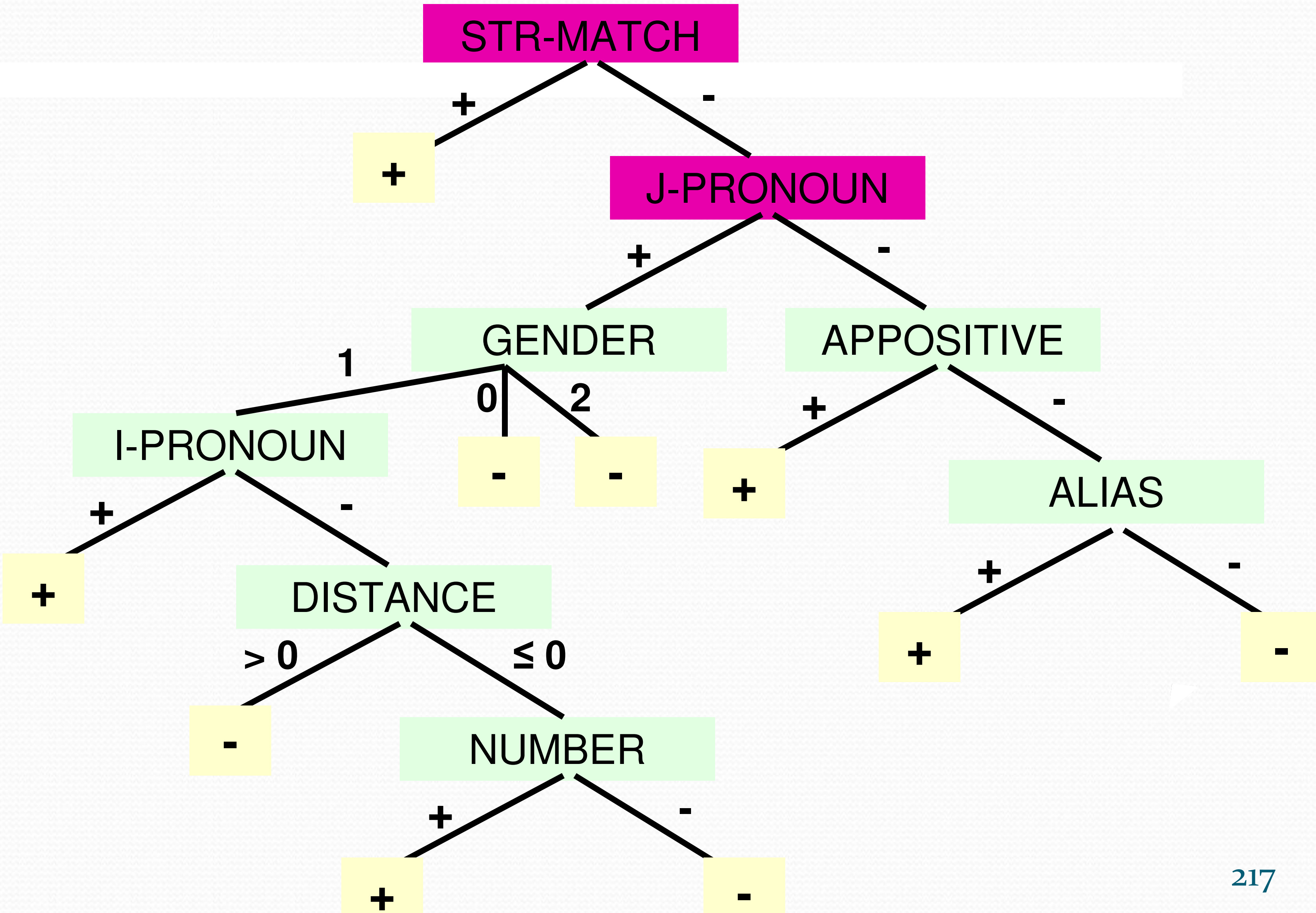
- Generate feature combinations using all paths of all possible lengths starting the root node



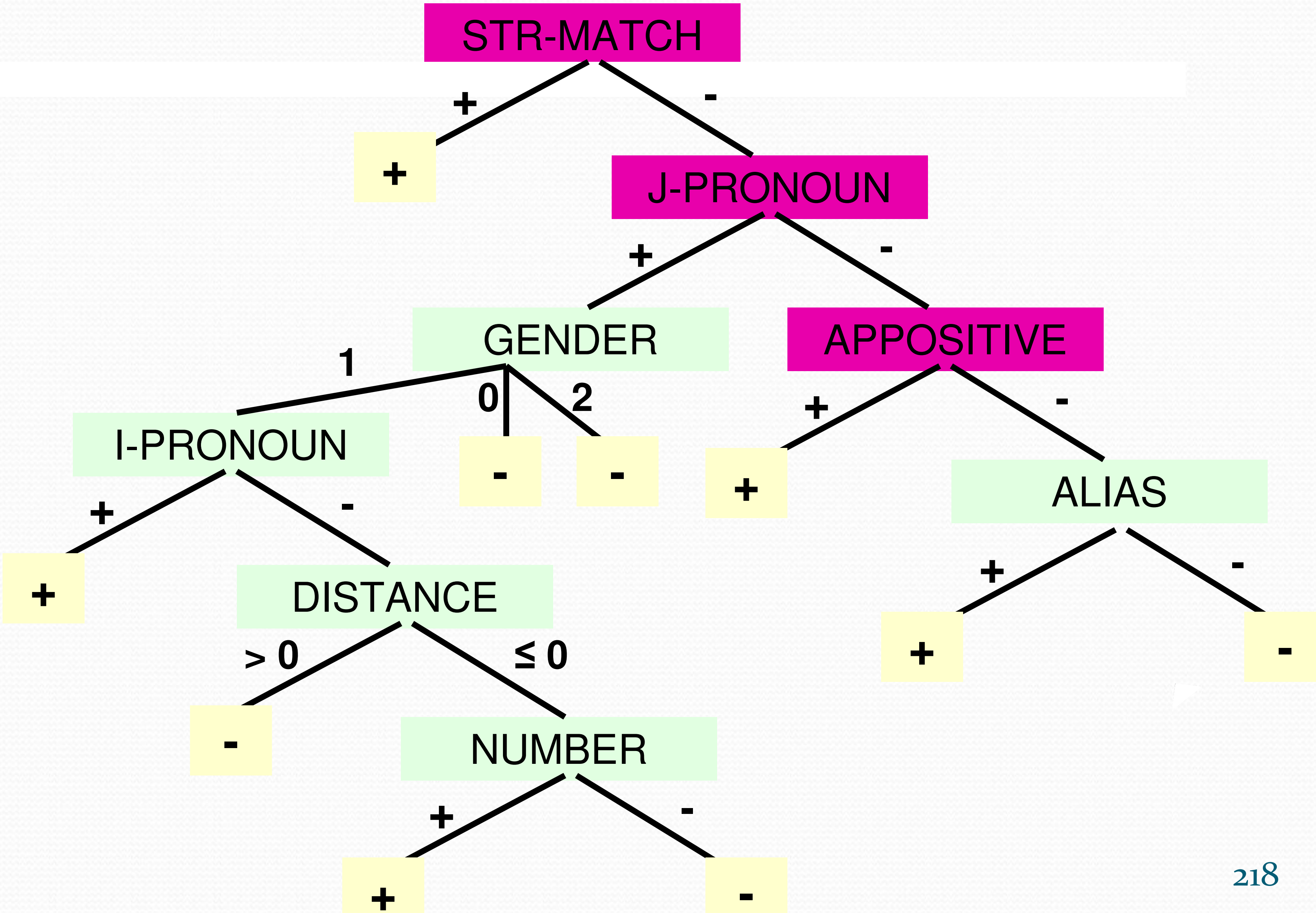
- Generate feature combinations using all paths of all possible lengths starting the root node



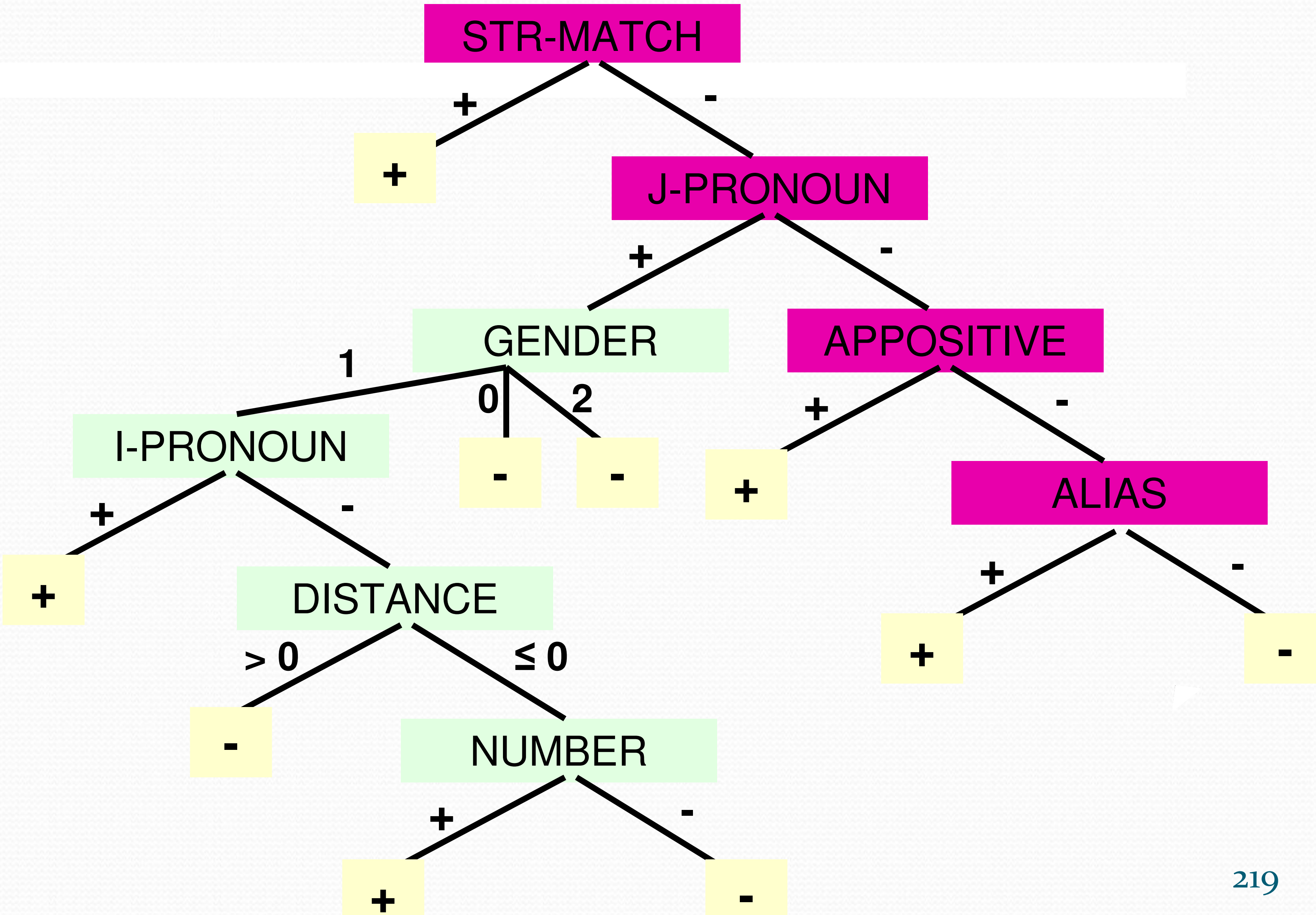
- Generate feature combinations using all paths of all possible lengths starting the root node



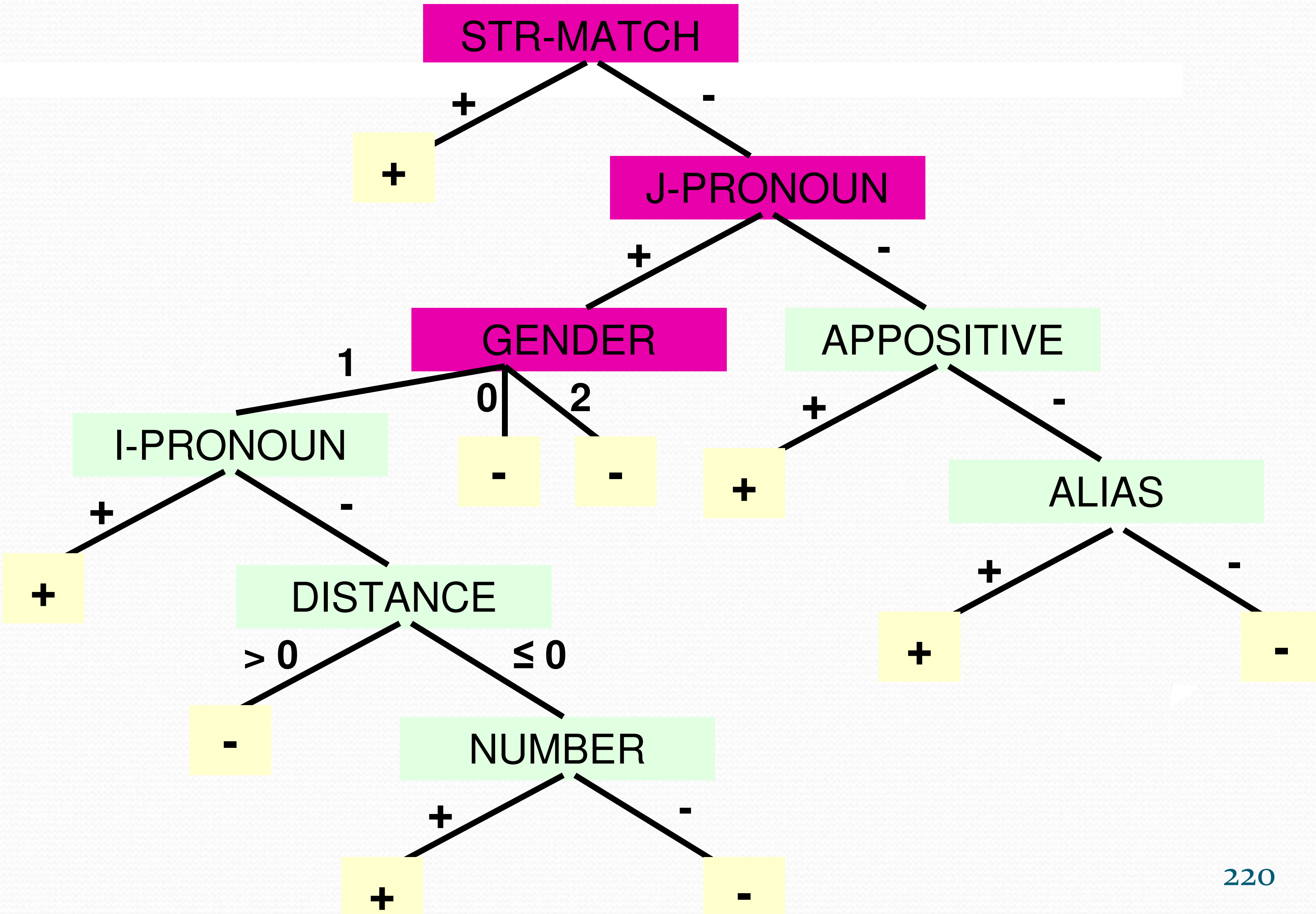
- Generate feature combinations using all paths of all possible lengths starting the root node



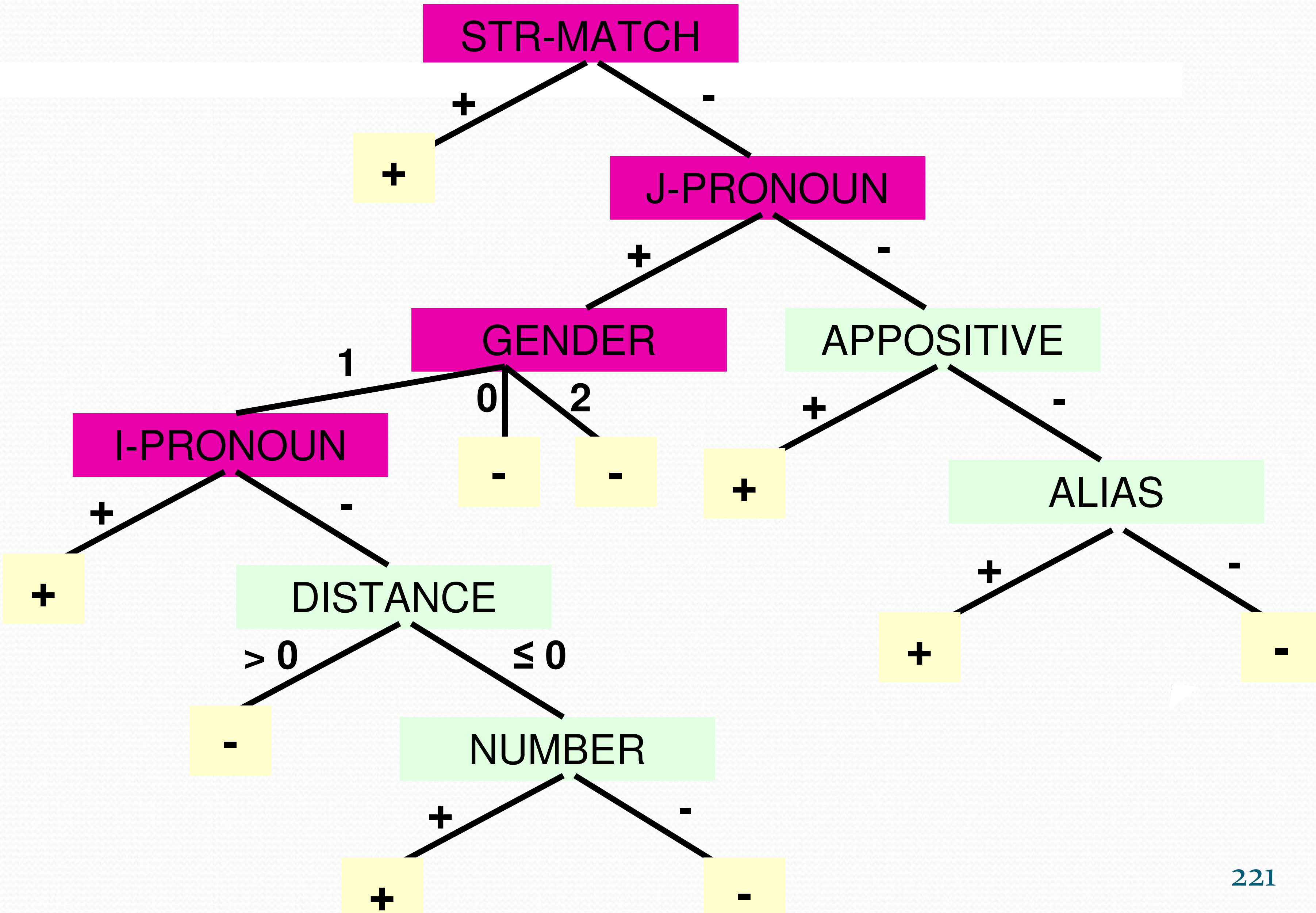
- Generate feature combinations using all paths of all possible lengths starting the root node



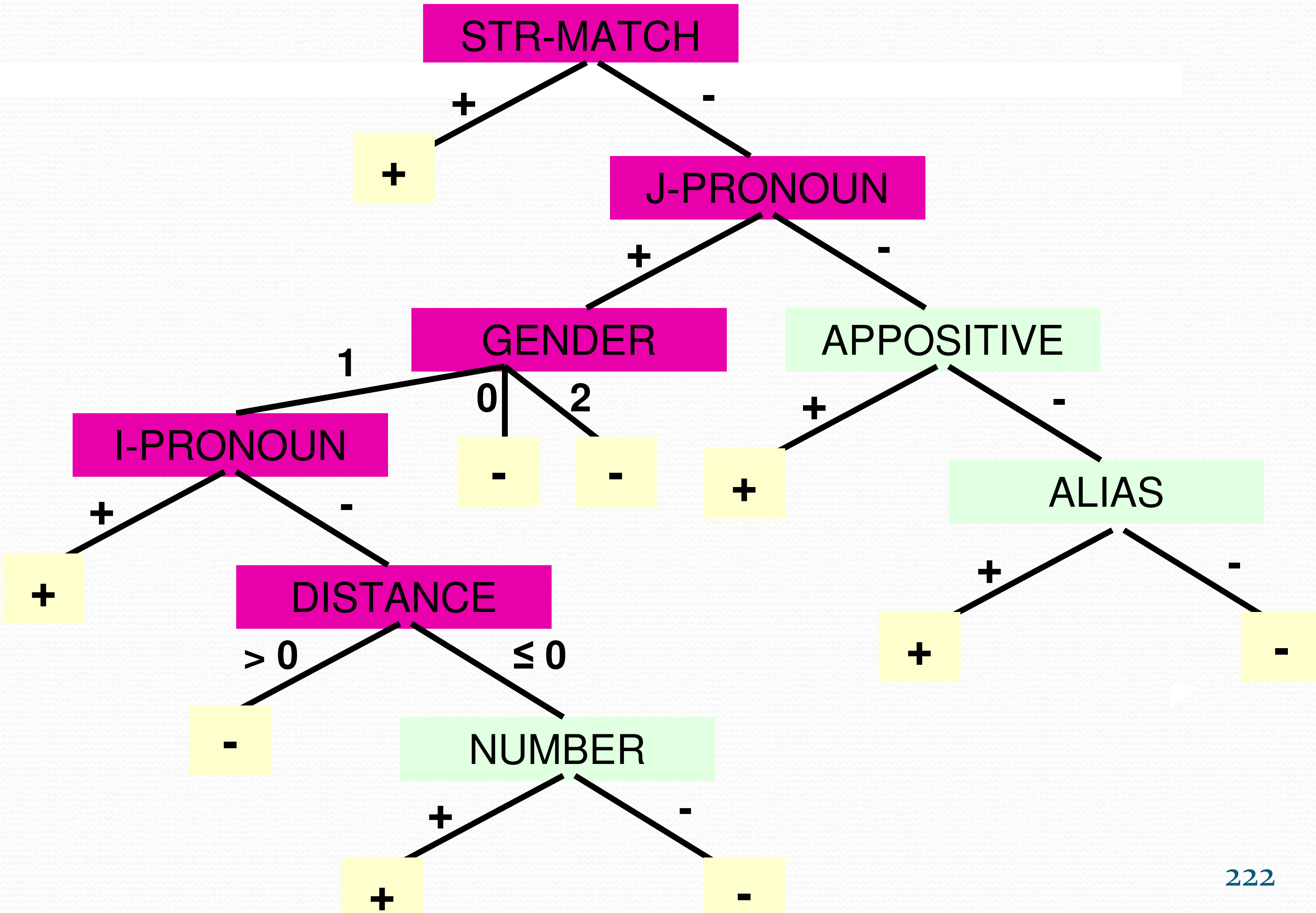
- Generate feature combinations using all paths of all possible lengths starting the root node



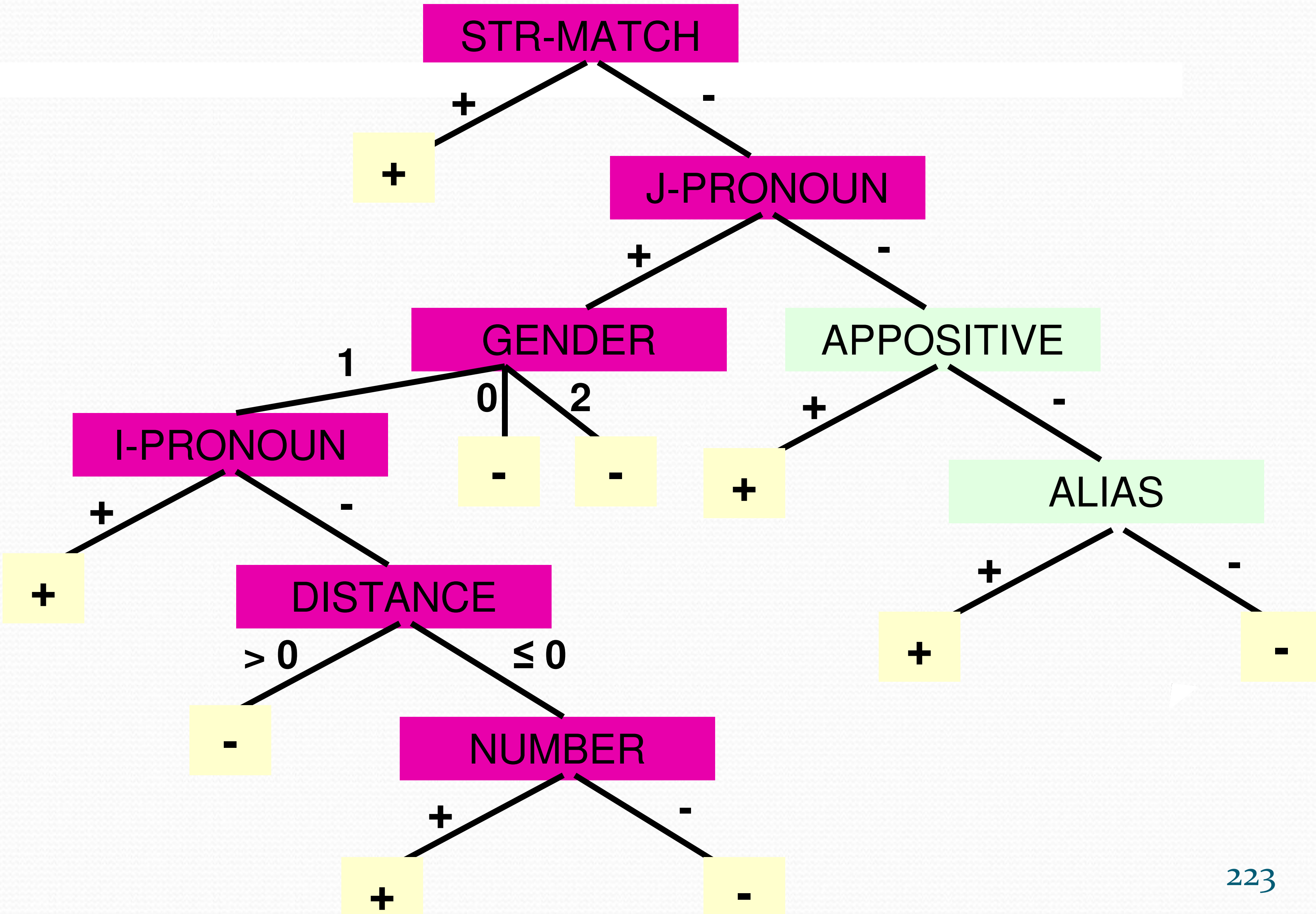
- Generate feature combinations using all paths of all possible lengths starting the root node



- Generate feature combinations using all paths of all possible lengths starting the root node



- Generate feature combinations using all paths of all possible lengths starting the root node



Feature Induction

- Use the resulting feature combinations, rather than the original features, to represent a training/test instance

Evaluation

- Corpus
  - Training: CoNLL-2012 shared task training corpus
  - Test: CoNLL-2012 shared task test corpus
- Scoring programs: recall, precision, F-measure
  - MUC (Vilain et al., 1995)
  - B^3 (Bagga & Baldwin, 1998)
  - $CEAF_e$ (Luo, 2005)
  - CoNLL (unweighted average of MUC, B^3 and $CEAF_e$ F-scores)

Results (Closed Track)

System	English	Chinese	Arabic	Avg
Rank 1: Latent Trees	**63.4**	58.5	**54.2**	58.7
Rank 2: Mention-Pair	61.2	60.0	53.6	58.3
Rank 3: Multi-Pass Sieves	59.7	**62.2**	47.1	56.4

- Usual caveat
  - Mention detection performance could have played a role

Latent Tree-Based Approach: Main Ideas

- Represent a coreference partition using a tree
  - Avoid learning from the hard coreference pairs
- Allow gold tree to change in each perceptron learning iteration
- Reduce number of candidate trees using Stanford sieves
- Use feature induction to better combine available features

- Which of them are effective?

Recent Models

- **Revival of mention-ranking models**
  - Durrett & Klein (2013): “There is no polynomial-time dynamic program for inference in a model with arbitrary entity-level features, so systems that use such features make decisions in a pipelined manner and sticking with them, operating greedily in a left-to-right fashion or in a multi-pass, sieve-like manner”

- Two recent mention-ranking models
  - Durrett & Klein (EMNLP 2013)
  - Wiseman et al. (ACL 2015)

Recent Models

- **Revival of mention-ranking models**
  - Durrett & Klein, 2013: "There is no polynomial-time dynamic program for inference in a model with arbitrary entity-level features, so systems that use such features make decisions in a pipelined manner and sticking with them, operating greedily in a left-to-right fashion or in a multi-pass, sieve-like manner"

- Two recent mention-ranking models
  - Durrett & Klein (EMNLP 2013)
  - Wiseman et al. (ACL 2015)

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^{\top}\mathbf{f}(i, a_i, x)\right)$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

document
context

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

ith mention

document context

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^{\top}\mathbf{f}(i, a_i, x)\right)$$

ith mention

document
context

antecedent
chosen for
mention i

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

sum over all mentions

ith mention

antecedent chosen for mention i

document context

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^{\top}\mathbf{f}(i, a_i, x)\right)$$

Vector of antecedents chosen for the document, one per mention

sum over all mentions

ith mention

antecedent chosen for mention i

document context

- Similar to mention-ranking model, except that we train it to jointly maximize the likelihood of selecting all antecedents

Durrett & Klein (EMNLP 2013)

- **Goal**
  - maximize the likelihood of the antecedent vector
- **Problem**
  - a mention may have more than one antecedent, so which one should we use for training?
- **Solution**
  - sum over all antecedent structures licensed by gold clusters

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda \|\mathbf{w}\|_1$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda \|\mathbf{w}\|_1$$

weight parameters

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda \|\mathbf{w}\|_1$$

weight parameters

sum over all training docs

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

weight parameters

sum over all training docs

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

weight parameters

sum over all training docs

sum over all antecedent structures

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

weight parameters

sum over all training docs

sum over all antecedent structures

L1 regularizer

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

weight parameters

sum over all training docs

sum over all antecedent structures

L1 regularizer

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda \|\mathbf{w}\|_1$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda \|\mathbf{w}\|_1$$

$$P'(a|x_k) \quad \propto \quad P(a|x_k)\exp(l(a, C_k^*))$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

$$P'(a|x_k) \quad \propto \quad P(a|x_k)\exp(l(a, C_k^*))$$

Durrett & Klein (EMNLP 2013)

- Log linear model of the conditional distribution P(a|x)

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

- Log likelihood function

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{a \in \mathcal{A}(C_k^*)} P'(a|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

$$P'(a|x_k) \quad \propto \quad P(a|x_k)\exp(l(a, C_k^*))$$

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}} \mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}} \mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}} \mathrm{WL}(a, C^*)$$

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}}\mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}}\mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}}\mathrm{WL}(a, C^*)$$

antecedent
vector

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}} \mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}} \mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}} \mathrm{WL}(a, C^*)$$

antecedent
vector

gold
clusters

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}} \mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}} \mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}} \mathrm{WL}(a, C^*)$$

antecedent
vector

gold
clusters

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}}\mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}}\mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}}\mathrm{WL}(a, C^*)$$

antecedent
vector

False Anaphoric

gold
clusters

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}} \mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}} \mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}} \mathrm{WL}(a, C^*)$$

antecedent vector

gold clusters

False Anaphoric

False New

The Loss Function

$$l(a, C^*) = \alpha_{\mathrm{FA}} \mathrm{FA}(a, C^*) + \alpha_{\mathrm{FN}} \mathrm{FN}(a, C^*) + \alpha_{\mathrm{WL}} \mathrm{WL}(a, C^*)$$

antecedent vector

gold clusters

False Anaphoric

False New

Wrong Link

The Loss Function

$$l(a, C^*) = \alpha_{FA} \mathrm{FA}(a, C^*) + \alpha_{FN} \mathrm{FN}(a, C^*) + \alpha_{WL} \mathrm{WL}(a, C^*)$$

antecedent vector

gold clusters

False Anaphoric

False New

Wrong Link

Surface Features

- Features computed on each of the two mentions
  - mention type (pronoun, name, nominal)
  - complete string, semantic head
  - first word, last word, preceding word, following word
  - length (in words)
- Features computed based on both mentions
  - Exact string match, head match
  - Distance in number of sentences and number of mentions
- Feature conjunctions
  - Attach to each feature the mention type (or the pronoun itself if it's a pronoun)

Results on the CoNLL-2011 test set

	MUC	B^3	CEAF$_e$	Avg.
STANFORD	60.46	65.48	47.07	57.67
IMS	62.15	65.57	46.66	58.13
SURFACE	**64.39**	**66.78**	**49.00**	**60.06**

- **Easy victories**
  - Using surface features (rather than standard coreference features) allows their system to outperform the state of the art

Why?

- D&K's explanation
  - These standard features do capture the same phenomena as standard coreference features, just implicitly

- Examples
  - Rather than using rules targeting person, number, or gender of mentions, they use conjunctions of pronoun identity
  - Rather than using a feature encoding definiteness, the first word of a mention would capture this
  - Rather than encoding grammatical role (subject/object), such information can be inferred from the surrounding words

Recent Models

- **Revival of mention-ranking models**
  - Durrett & Klein, 2013: "There is no polynomial-time dynamic program for inference in a model with arbitrary entity-level features, so systems that use such features make decisions in a pipelined manner and sticking with them, operating greedily in a left-to-right fashion or in a multi-pass, sieve-like manner"
- Two recent mention-ranking models
  - Durrett & Klein (EMNLP 2013)
  - Wiseman et al. (ACL 2015)

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\top} \phi(x, y)$$

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

anaphor

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?
- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

anaphor

candidate antecedent

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\mathrm{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

  - Another way of expressing the scoring function

$$s_{\mathrm{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\mathrm{a}}(x) \\ \phi_{\mathrm{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\mathrm{a}}(x) & \text{if } y = \epsilon \end{cases}$$

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^\mathsf{T} \phi(x, y)$$

  - Another way of expressing the scoring function

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^\mathsf{T} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\mathsf{T} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

non-null antecedent

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}} \quad \phi(x, y)$$

Features on anaphor

  - Another way of expressing the scoring function

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

non-null antecedent

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

s_{lin} Features on anaphor $\phi($ Features on both anaphor and candidate antecedent

  - Another way of expressing the scoring function

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} u^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x, y) \end{bmatrix} & \text{if } y \neq \epsilon \\ v^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

non-null antecedent

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

  - Another way of expressing the scoring function

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

null antecedent

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x,y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x,y)$$

  - Another way of e[…] scoring function

Features on anaphor

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

null antecedent

Wiseman et al. (ACL 2015)

- **Observation**: recent mention-ranking models are all linear; why not train a non-linear mention-ranking model?

- Recap
  - Scoring function for linear mention-ranking models

$$s_{\text{lin}}(x, y) \triangleq \boldsymbol{w}^{\mathsf{T}} \phi(x, y)$$

  - Another way of expressing the scoring function

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x, y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

Raw/Unconjoined features

Raw/Unconjoined Features

- Mention's head, complete string, first word, last word, …

- Researchers don't seem to like them
  - they almost always use conjoined features
    - created by hand or obtained via feature induction
    - can add some non-linearity to the linear model

- Why?
  - Wiseman et al. empirically showed that raw/unconjoined features are not predictive for the coreference task

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}\right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task

- Scoring function for **linear** mention-ranking model

$$s_{\text{lin}+}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix} \right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task

- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix} \right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

$$\boldsymbol{h}_{\text{a}}(x) \triangleq \tanh(\boldsymbol{W}_{\text{a}}\, \phi_{\text{a}}(x) + \boldsymbol{b}_{\text{a}})$$

$$\boldsymbol{h}_{\text{p}}(x, y) \triangleq \tanh(\boldsymbol{W}_{\text{p}}\, \phi_{\text{p}}(x, y) + \boldsymbol{b}_{\text{p}})$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^\mathsf{T} \begin{bmatrix} \phi_\text{a}(x) \\ \phi_\text{p}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\mathsf{T} \phi_\text{a}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^\mathsf{T} \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_\text{a}(x) \\ \boldsymbol{h}_\text{p}(x,y) \end{bmatrix} \right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\mathsf{T} \boldsymbol{h}_\text{a}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

$$\boldsymbol{h}_\text{a}(x) \triangleq \tanh(\boldsymbol{W}_\text{a}\, \phi_\text{a}(x) + \boldsymbol{b}_\text{a})$$

$$\boldsymbol{h}_\text{p}(x,y) \triangleq \tanh(\boldsymbol{W}_\text{p}\, \phi_\text{p}(x,y) + \boldsymbol{b}_\text{p})$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin}+}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix} \right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

$$\boldsymbol{h}_{\text{a}}(x) \triangleq \tanh(\boldsymbol{W}_{\text{a}}\, \phi_{\text{a}}(x) + \boldsymbol{b}_{\text{a}})$$

$$\boldsymbol{h}_{\text{p}}(x,y) \triangleq \tanh(\boldsymbol{W}_{\text{p}}\, \phi_{\text{p}}(x,y) + \boldsymbol{b}_{\text{p}})$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix} \right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

$$\boldsymbol{h}_{\text{a}}(x) \triangleq \tanh(\boldsymbol{W}_{\text{a}} \phi_{\text{a}}(x) + \boldsymbol{b}_{\text{a}})$$
$$\boldsymbol{h}_{\text{p}}(x,y) \triangleq \tanh(\boldsymbol{W}_{\text{p}} \phi_{\text{p}}(x,y) + \boldsymbol{b}_{\text{p}})$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

$$\boldsymbol{h}_{\text{a}}(x) \triangleq \tanh(\boldsymbol{W}_{\text{a}} \phi_{\text{a}}(x) + \boldsymbol{b}_{\text{a}})$$
$$\boldsymbol{h}_{\text{p}}(x,y) \triangleq \tanh(\boldsymbol{W}_{\text{p}} \phi_{\text{p}}(x,y) + \boldsymbol{b}_{\text{p}})$$

Network's parameters

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^\mathsf{T} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\mathsf{T} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^\mathsf{T} \boldsymbol{g}(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\mathsf{T} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

$$\boldsymbol{h}_{\text{a}}(x) \triangleq \tanh(\boldsymbol{W}_{\text{a}} \phi_{\text{a}}(x) + \boldsymbol{b}_{\text{a}})$$

$$\boldsymbol{h}_{\text{p}}(x,y) \triangleq \tanh(\boldsymbol{W}_{\text{p}} \phi_{\text{p}}(x,y) + \boldsymbol{b}_{\text{p}})$$

Can learn feature combinations

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task

- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x, y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

  - **Option 1**: let g be the identity function
    - Neural net model same as linear model except that it's defined over non-linear feature representations

Wiseman et al.'s Proposal

- Learn feature representations that are useful for the task
- Scoring function for **linear** mention-ranking model

$$s_{\text{lin+}}(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \begin{bmatrix} \phi_{\text{a}}(x) \\ \phi_{\text{p}}(x,y) \end{bmatrix} & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \phi_{\text{a}}(x) & \text{if } y = \epsilon \end{cases}$$

- Scoring function for Wiseman's **neural net**-based model

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{g}(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^{\mathsf{T}} \boldsymbol{h}_{\text{a}}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

- **Option 2:** $\boldsymbol{g}(\begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix}) = \tanh(\boldsymbol{W} \begin{bmatrix} \boldsymbol{h}_{\text{a}}(x) \\ \boldsymbol{h}_{\text{p}}(x,y) \end{bmatrix} + \boldsymbol{b})$

functions as an additional hidden layer

Wiseman et a.'s Proposal

- Pros
  - Can learn (non-linear) feature representations from raw features
    - Don't have to conjoin features by hand
- Cons
  - Training a non-linear model is more difficult than training a linear model
    - Model performance sensitive to weight initializations

Training

$$s(x,y) \triangleq \begin{cases} \boldsymbol{u}^\top \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{h}_\mathrm{a}(x) \\ \boldsymbol{h}_\mathrm{p}(x,y) \end{bmatrix}\right) + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\top \boldsymbol{h}_\mathrm{a}(x) + v_0 & \text{if } y = \epsilon \end{cases}$$

1. Train two neural nets separately
   - One for anaphoricity and one for coreference
2. Use the weight parameters learned as initializations for the combined neural net

- Objective function similar to Durrett & Klein's, except that it is based on margin loss rather than log loss

Results on CoNLL 2012 test set

System	MUC			B^3			$CEAF_e$			
	P	R	F_1	P	R	F_1	P	R	F_1	CoNLL
BCS (2013)	74.89	67.17	70.82	64.26	53.09	58.14	58.12	52.67	55.27	61.41
This work (g_2)	76.96	68.10	72.26	66.90	54.12	59.84	59.02	53.34	56.03	62.71
This work (g_1)	76.23	69.31	72.60	66.07	55.83	**60.52**	59.41	54.88	**57.05**	**63.39**

Wiseman et al. (NAACL 2016)

- Improved their neural net model by incorporating entity-level information
  - CoNLL score increased by 0.8
- State-of-the-art results on the English portion of the CoNLL 2012 test data
- Software available from the Harvard NLP group page

Unsupervised Models

- **EM**
  - Cherry & Bergsma (2005), Charniak & Elsner (2009)
- **Clustering**
  - Cardie & Wagstaff (1999), Cai & Strube (2010)
- **Nonparametric models**
  - Haghighi & Klein (2007, 2010)
- **Markov Logic networks**
  - Poon & Domingos (2008)
- **Bootstrapping**
  - Kobdani et al. (2011)

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Semantics and World Knowledge

- Coreference resolution is considered one of the most difficult tasks in NLP in part because of its reliance on sophisticated knowledge sources
- The importance of semantics and world knowledge in coreference resolution has long been recognized
  - Hobbs (1976)
    - Syntactic approach (the naïve algorithm)
    - Semantic approach
- Shift in research trends
  - Knowledge-rich approaches (1970s and 1980s)
  - Knowledge-lean approaches (1990s)
  - Knowledge-rich approaches (2000 onwards)

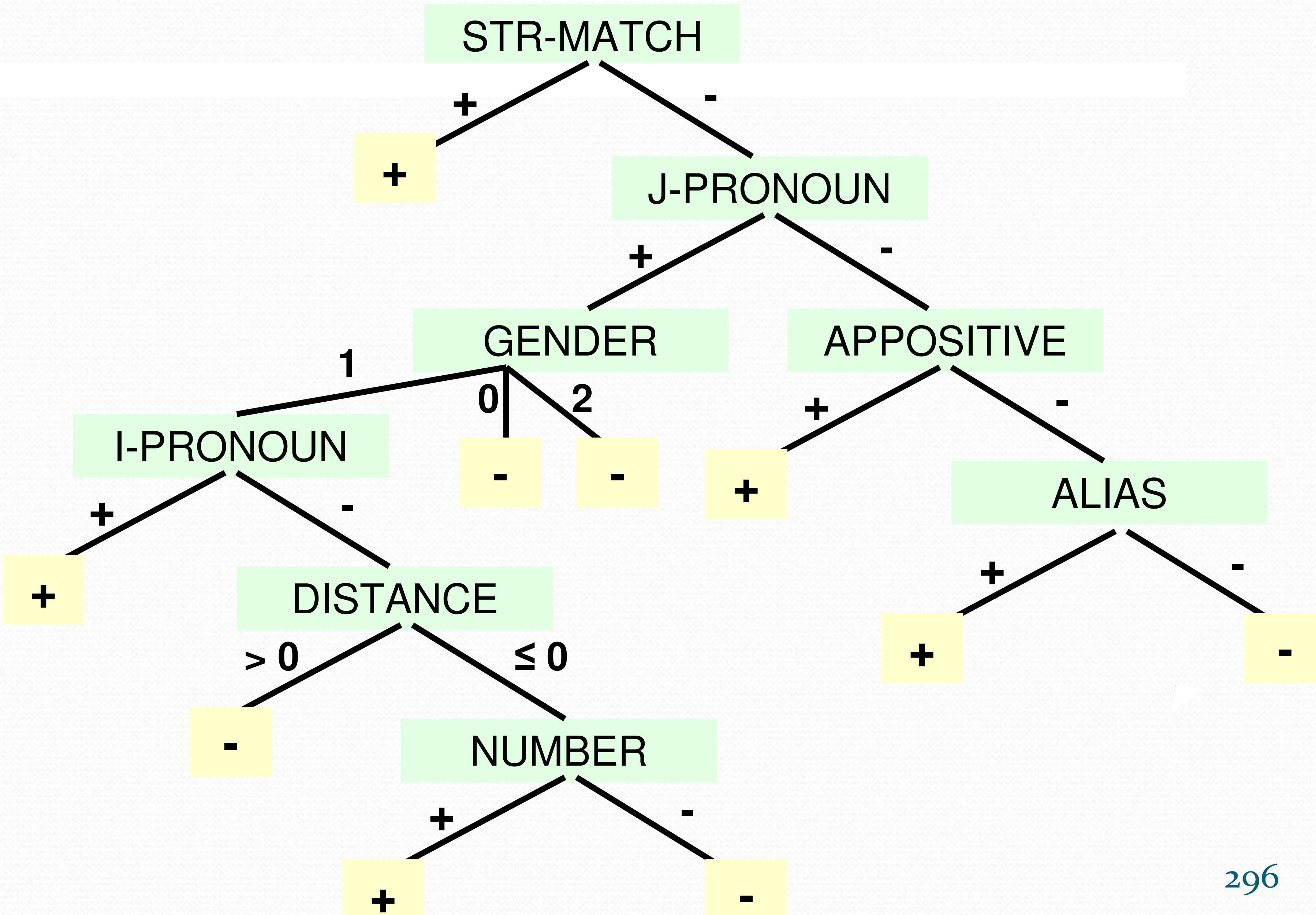
Semantics and World Knowledge

- Have researchers been successful in employing semantics and world knowledge to improve learning-based coreference resolution systems?
- Are these features useful in the presence of morpho-syntactic (knowledge-lean, robustly computed) features?

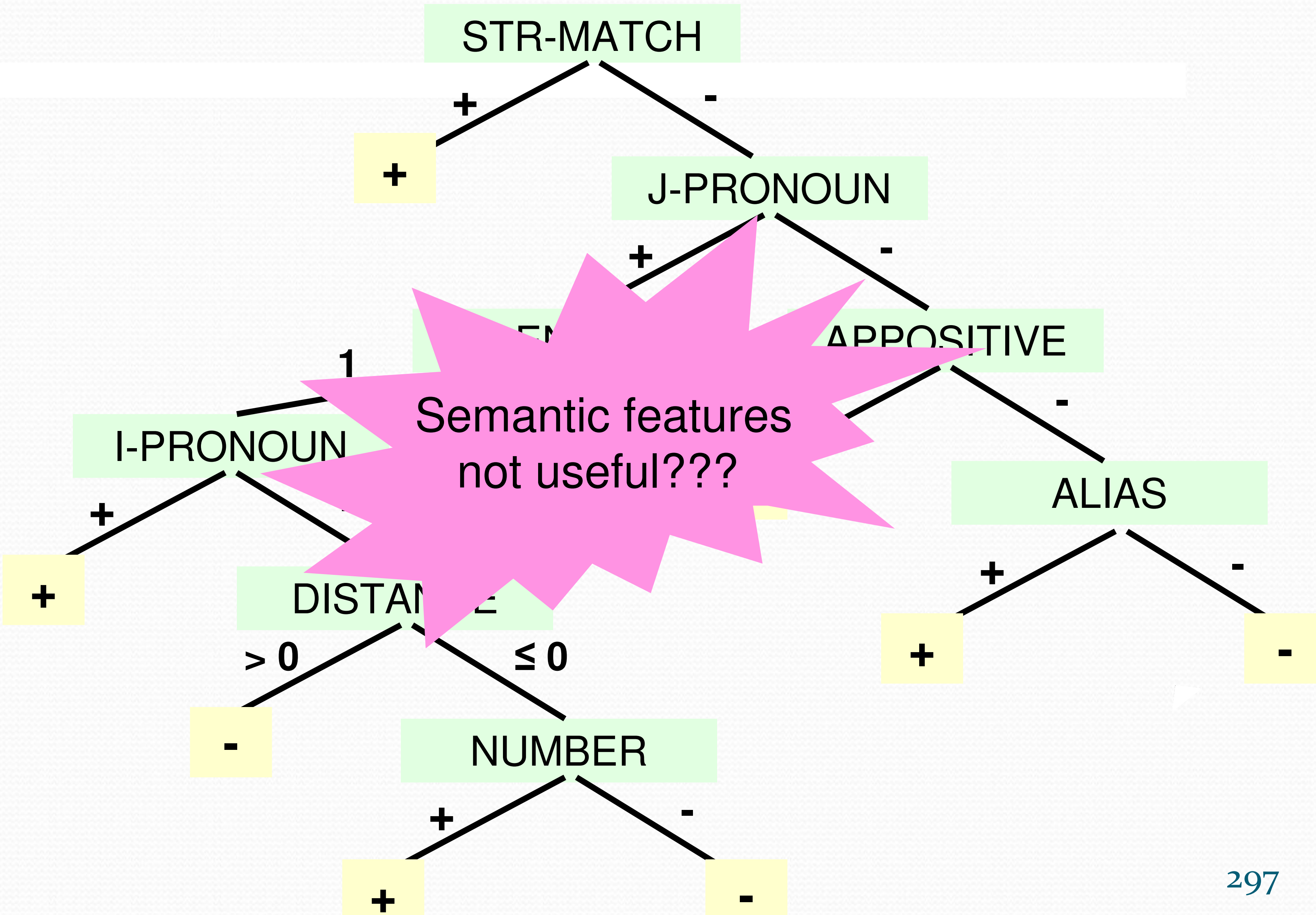
Soon et al. (2001)

- One of the first learning-based coreference systems
- Mention-pair model trained using C4.5
- 12 features
  - Mostly morpho-syntactic features
  - One semantic feature computed based on WordNet (1^{st} sense)
    - U if one/both mentions have undefined WordNet semantic class
    - T if the two have the same WordNet semantic class
    - F otherwise
- No separate evaluation of how useful the WordNet semantic feature is, but…

Decision Tree Learned for MUC-6 Data



Decision Tree Learned for MUC-6 Data



Ng & Cardie (ACL 2002)

- A large-scale expansion of the features used by Soon et al.
  - 53 lexical, grammatical, and semantic features

Four WordNet-based Semantic Features

- Whether m_j is the closest mention preceding mention that has the same WordNet semantic class as m_k
- Whether the two mentions have an ancestor-descendent relationship in WordNet; if yes,
  - Encode the sense numbers in WordNet that give rise to this ancestor-descendent relationship
  - compute the distance between the two WordNet synsets

Evaluation

- No separate evaluation of how useful the WordNet semantic feature is, but…
  - A hand-selected subset of the features was used to train a mention-pair model that yielded better performance
    - This subset did not include any of these 4 semantic features

Evaluation

- No separate evaluation of how useful the WordNet semantic feature is, but…
  - A hand-selected subset of the feat res was used to train a mention-pair model that yi ded ter performance
    - The subset did nc 4 semantic features

Semantic features not useful???

Kehler et al. (NAACL 2004)

- Approximate world knowledge using predicate-argument statistics

He worries that Trump's initiative would push his industry over the edge, forcing it to shift operations elsewhere.

- forcing_industry is a more likely verb-object combination in naturally-occurring data than forcing_initiative or forcing_edge

- Such predicate-argument (e.g., subject-verb, verb-object) statistics can be collected from a large corpus
  - TDT-2 corpus (1.32 million subj-verb; 1.17 million verb-obj)

Kehler et al. (NAACL 2004)

- **Goal**: examine whether predicate-argument statistics, when encoded as features, can improve a pronoun resolver trained on state-of-the-art morpho-syntactic features

- Morpho-syntactic features
  - Gender agreement
  - number agreement
  - Distance between the two mentions
  - Grammatical role of candidate antecedent
  - NP form (def, indef, pronoun) of the candidate antecedent

- Train a mention-pair model using maximum entropy
- Evaluate on the ACE 2003 corpus

Results

Features	MaxEnt	MaxEnt-Features
none	.6877	.6496
num, gend	.6667	.6745
num, gend, dist	.7336	.7415
num, gend, dist, pos	.7441	.7507
num, gend, dist, pos, lform	.7572	.7572

Results

Features	MaxEnt	MaxEnt-Features
none	[illegible]77	.6496
num, gend	[illegible]	.6745
num, ge[illegible]	[illegible]	.7415
num, gend, di[illegible]	[illegible]	.7507
num, gend, disc[illegible]	[illegible]72	.7572

Semantic features not useful???

Error Analysis

After the endowment was publicly excoriated for having the temerity to award some of its money to art that addressed changing views of gender and race, …

- MaxEnt selected the temerity
- Predicate-argument statistics selected the endowment
- Need a better way to exploit the statistics?

Error Analysis

The dancers were joined by about 70 supporters as they marched around a fountain not far from the mayor's office.

- MaxEnt selected the supporters
- So did the predicate-argument statistics

Kehler et al.'s Observations

- In the cases in which statistics reinforced a wrong answer, no manipulation of features can rescue the prediction
- For the cases in which statistics could help, their successful use will depend on the existence of a formula that can capture these cases without changing the predictions for examples that the model currently classifiers correctly
- **Conclusion**: predicate-argument statistics are a poor substitute for world knowledge, and more to the point, they do not offer much predictive power to a state-of-the-art morphosyntactically-driven pronoun resolution system

Yang et al. (ACL 2005)

- **Goal**: examine whether predicate-argument statistics, when encoded as features, can improve a pronoun resolver trained on state-of-the-art morpho-syntactic features

- But… with the following differences in the setup
  - Web as corpus (to mitigate data sparsity)
  - Pairwise ranking model
  - Baseline morpho-syntactic features defined on m_i and m_j
    - DefNP_i, Pro_i, NE_i, SameSent, NearestNP_i, Parallel_i, FirstNPinSent_i, Reflexive_j, NPType_j
  - Learner: C4.5
  - Corpus: MUC-6 and MUC-7

Results

	Neutral Pro.		Personal Pro.		Overall	
	Corp	Web	Corp	Web	Corp	Web
Baseline	73.9		**91.9**		81.9	
Baseline + predicate-arg statistics	76.7	**79.2**	91.4	**91.9**	83.3	**84.8**

Results

	Neutral Pro.		Personal Pro.		Overall	
	Corp	Web	Corp	Web	Corp	Web
Baseline	73.9		**91.9**		81.9	
Baseline + predicate-arg statistics	76.7	**79.2**	91.4	**91.9**	83.3	**84.8**

Semantic features are useful!!!

Ponzetto & Strube (NAACL 2006)

- **Goal**: improve learning-based coreference resolution by exploiting three knowledge sources
  - WordNet
  - Wikipedia
  - Semantic role labeling

Using WordNet

- **Motivation**:
  - Soon et al. employed a feature that checks whether two mentions have the same WordNet semantic class
    - **Noisy**: had problems with coverage and sense proliferation
- **Solution**: measure the similarity between the WordNet synsets of the two mentions using six similarity measures
  - 3 path-length based measures
  - 3 information-content based measures
- Two features
  - Highest similarity score over all senses and all measures
  - Average similarity score over all senses and all measures

Using Wikipedia

Martha Stewart is hoping people don’t run out on her.
The celebrity indicted on charges stemming from …

- can also resolve the celebrity using syntactic parallelism, but
  - heuristics are not always accurate
  - does not mimic the way humans look for antecedents

- Use world knowledge extracted from Wikipedia

Using Wikipedia

- Given mentions m_i and m_j, retrieve the Wiki pages they refer to, P_i and P_j, with titles m_i and m_j (or their heads)
- Create features for coreference resolution
  - Features based on first paragraph of Wiki page
    - Whether P_i's first paragraph contains m_j
    - Create an analogous feature by reversing the roles of m_i & m_j

Using Wikipedia

- Given mentions m_i and m_j, retrieve the Wiki pages they refer to, P_i and P_j, with titles m_i and m_j (or their heads)

- Create features for coreference resolution
  - Features based on first paragraph of Wiki page
  - Features based on the hyperlinks of Wiki page
    - Whether at least one hyperlink in P_i contains m_j
    - Create an analogous feature by reversing the roles of m_i & m_j

Using Wikipedia

- Given mentions m_i and m_j, retrieve the Wiki pages they refer to, P_i and P_j, with titles m_i and m_j (or their heads)
- Create features for coreference resolution
  - Features based on first paragraph of Wiki page
  - Features based on the hyperlinks of Wiki page
  - Features based on the Wiki categories
    - Whether the categories P_i belongs to contains m_j (or its head)
    - Create analogous feature by reversing the roles of m_i & m_j

Using Wikipedia

- Given mentions m_i and m_j, retrieve the Wiki pages they refer to, P_i and P_j, with titles m_i and m_j (or their heads)

- Create features for coreference resolution
  - Features based on first paragraph of Wiki page
  - Features based on the hyperlinks of Wiki page
  - Features based on the Wiki categories
  - Features based on overlap of first paragraphs
    - Overlap score between first paragraphs of the two Wiki pages

Using Wikipedia

- Given mentions m_i and m_j, retrieve the Wiki pages they refer to, P_i and P_j, with titles m_i and m_j (or their heads)

- Create features for coreference resolution
  - Features based on first paragraph of Wiki page
  - Features based on the hyperlinks of Wiki page
  - Features based on the Wiki categories
  - Features based on overlap of first paragraphs
  - Highest & average relatedness score of all category pairs formed from the categories associated with the two Wiki pages

Semantic Role Labeling (SRL)

Peter Anthony decries program trading as "limiting the game to a few," but he is not sure whether he wants to denounce it because …

- Knowing that "program trading" is the PATIENT of the "decry" predicate and "it" being the PATIENT of "denounce" could trigger the (semantic parallelism based) inference

Semantic Role Labeling (SRL)

- Use the ASSERT semantic parser to identify all verb predicates in a sentence and their semantic arguments
  - Each argument is labeled with its PropBank-style semantic role
    - ARG_1, …, ARG_n
- Two SRL features
  - The role-predicate pair of mention m_i
  - The role-predicate pair of mention m_j

Experimental Setup

- Baseline
  - mention-pair model trained with Soon et al.'s 12 features using MaxEnt
- ACE 2003 (Broadcast News + Newswire)
- Evaluation metric: MUC scorer

Results

	R	P	F_1
baseline	54.5	88.0	67.3
+WordNet	**56.7**	87.1	**68.6**
+Wikipedia	**55.8**	87.5	**68.1**
+SRL	**56.3**	**88.4**	**68.8**
all features	**61.0**	84.2	**70.7**

Results

	R	P	F_1
baseline	[illegible]	88.0	67.3
+WordNe[illegible]	[illegible]	[illegible]	**68.6**
+Wiki[illegible]	[illegible]	[illegible]	**68.1**
+S[illegible]	[illegible]	[illegible]	**68.8**
all fea[illegible]	[illegible]	84.2	**70.7**

Semantic features
are useful!!!

Rahman & Ng (ACL 2011)

- **Goal**: improve learning-based coreference resolution by exploiting two knowledge sources
  - YAGO
  - FrameNet

YAGO (Suchanek et al., 2007)

- contains 5 million facts derived from Wikipedia and WordNet
- each fact is a triple describing a relation between two NPs
  - <NP1, rel, NP2>, rel can be one of 90 YAGO relation types
- focuses on two types of YAGO relations: TYPE and MEANS (Bryl et al., 2010, Uryupina et al., 2011)
  - TYPE: the IS-A relation
    - <AlbertEinstein, TYPE, physicist>
      <BarackObama, TYPE, president>
  - MEANS: addresses synonymy and ambiguity
    - <Einstein, MEANS, AlbertEinstein>,
      <Einstein, MEANS, AlfredEinstein>
  - provide evidence that the two NPs involved are coreferent

Why YAGO?

- combines the information in Wikipedia and WordNet

Martha Stewart is hoping people don't run out on her.
The celebrity indicted on charges stemming from …

- can resolve the celebrity to Martha Stewart
  - neither Wikipedia nor WordNet alone can

- How to use YAGO to resolve?
  1. Heuristically maps each Wiki category in the Wiki page for Martha Stewart to its semantically closest WordNet synset
     - AMERICAN TELEVISION PERSONALITIES → synset for sense #2 of personality
  2. Realizes personality is a direct hyponym of celebrity in WordNet
  3. Extracts the fact <MarthaStewart, TYPE, celebrity>

Using YAGO for Coreference Resolution

- create a new feature for mention-pair model whose value is

  1 if the two NPs are in a TYPE or MEANS relation

  0 otherwise

FrameNet (Baker et al., 1998)

- A lexico-semantic resource focused on semantic frames

- A **frame** contains
  - the lexical predicates that can invoke it
  - the frame elements (i.e., the semantic roles)

- E.g., the JUDGMENT_COMMUNICATION frame describes situations in which a COMMUNICATOR communicates a judgment of an EVALUEE to an ADDRESSEE
  - frame elements: COMMUNICATOR, EVALUEE, ADDRESSEE
  - lexical predicates: acclaim, accuse, decry, denounce, slam, …

Motivating Example

Peter Anthony decries program trading as "limiting the game to a few," but he is not sure whether he wants to denounce it because …

- To resolve it to Peter Anthony, it may be helpful to know
  - decry and decounce are "semantically related"
  - the two mentions have the same semantic role

Motivating Example

Peter Anthony decries program trading as "limiting the game to a few," but he is not sure whether he wants to denounce it because …

- To resolve it to Peter Anthony, it may be helpful to know
  - decry and decounce are "semantically related"
  - the two mentions have the same semantic role

- Missing from Ponzetto & Strube's application of semantic roles
- We model this using FrameNet

Observation

- Features encoding
  - the semantic roles of the two NPs under consideration
  - whether the associated predicates are “semantically related”

  could be useful for identifying coreference relations.

Observation

- Features encoding
  - the semantic roles of the two NPs under consideration
  - whether the associated predicates are "semantically related"

could be useful for identifying coreference relations.

Use ASSERT

- Provides PropBank-style roles (Arg0, Arg1, …)

Observation

- Features encoding
  - the semantic roles of the two NPs under consideration
  - whether the associated predicates are “semantically related”

could be useful for identifying coreference relations.

Use ASSERT
- Provides PropBank-style roles (Arg0, Arg1, …)

Use PropBank
- Checks whether the two predicates appear in the same frame

Results on ACE 2005 and OntoNotes

- Baseline: mention-pair model trained with 39 features from Rahman & Ng (2009)

	ACE		OntoNotes	
	B^3	CEAF	B^3	CEAF
Baseline	62.4	60.0	53.3	51.5
Baseline+YAGO types	63.1	62.8	54.6	52.8
Baseline+YAGO types & means	63.6	63.2	55.0	53.3
Baseline+YAGO types & means & FN	63.8	63.1	55.2	53.4

Difficulty in Exploiting World Knowledge

- World knowledge extracted from YAGO is noisy
  - Numerous entries for a particular mention, all but one are irrelevant

Ratinov & Roth (EMNLP 2012)

- **Goal**: Improve their learning-based multi-pass sieve approach using world knowledge extracted from Wikipedia
- Extract for each mention **knowledge attributes** from Wiki
  - Find the Wiki page the mention refers to using their context-sensitive entity linking system, which could reduce **noise**
  - Extract from the retrieved page three attributes
    - (1) gender; (2) nationality; (3) fine-grained semantic categories
- Create features from the extracted attributes
  - Whether the two mentions are mapped to the same Wiki page
  - Agreement w.r.t. gender, nationality, and semantic categories
- Augment each sieve's feature set with these new features

Results on ACE 2004 Newswire texts

- System shows a minimum improvement of 3 (MUC), 2 (B^3), and 1.25 (CEAF) F1 points on gold mentions
  - Not always considered an acceptable evaluation setting
  - Improvements on **gold** mentions do not necessarily imply improvements on **automatically extracted** mentions

Durrett & Klein (EMNLP 2013)

- Using only surface features, their log linear model achieved state of the art results on the CoNLL-2011 test set

- Can performance be further improved with semantics?

- Derive semantic features from four sources
  - WordNet hypernymy and synonymy
  - Number and gender for names and nominals
  - Named entity types
  - Latent clusters computed from English Gigaword
    - Each element in a cluster is a nominal head together with the conjunction of its verbal governor and its semantic role

Results on CoNLL-2011 Test Set

	MUC	B^3	CEAF$_e$	Avg.
SURFACE	64.39	66.78	49.00	60.06
SURFACE+SEM	64.70	**67.27**	**49.28**	**60.42**

Results on CoNLL-2011 Test Set

	MUC	B^3	CEAF_e	Avg.
SURFACE	64.39	66.78	49.00	60.06
SURFACE+SEM	64.70	**67.27**	**49.28**	**60.42**

Semantic features not useful???

Results on CoNLL-2011 Test Set

	MUC	B^3	CEAF$_e$	Avg.
SURFACE	64.39	66.78	49.00	60.06
SURFACE+SEM	64.70	**67.27**	**49.28**	**60.42**

- **D&K's explanation**
  - only a small fraction of the mention pairs are coreferent
    - A system needs very strong evidence to overcome the default hypothesis that a pair of mentions is not coreferent
    - The weak (semantic) indicators of coreference will likely have high false positive rates, doing more harm than good

Results on CoNLL-2011 Test Set

	MUC	B^3	CEAF$_e$	Avg.
SURFACE	64.39	66.78	49.00	60.06
SURFACE+SEM	64.70	**67.27**	**49.28**	**60.42**

- **D&K's explanation**
  - only a small fraction of the mention pairs are coreferent
    - A system needs very strong evidence to overcome the default hypothesis that a pair of mentions is not coreferent
    - The weak (semantic) indicators of coreference will likely have high false positive rates, doing more harm than good
- **Our explanation**
  - It's harder to use semantics to improve a strong baseline
  - D&K's semantic features are **shallow** semantic features

Other (Failed) Attempts

- Stanford's multi-pass sieve system
  - beet system in the CoNLL 2011 shared task
  - extended their system with 2 new sieves that exploit semantics from WordNet, Wikipedia infoboxes, and Freebase records
    - the semantics sieves didn't help

Other (Failed) Attempts

- Stanford's multi-pass sieve system
  - beet system in the CoNLL 2011 shared task
  - extended their system with 2 new sieves that exploit semantics from WordNet, Wikipedia infoboxes, and Freebase records
    - the semantics sieves didn't help

- Sepena et al.'s (2013) system
  - 2nd best system in the CoNLL 2011 shared task
  - Proposed a constraint-based hypergraph partitioning approach
  - Used info extracted from Wikipedia as features/constraints
  - **Conclusion**: the problem seems to lie with the extracted info, which is biased in favor of famous/popular entities… including false positives… imbalance against entities with little or no info

Plan for the Talk

- **Part I**: Background
  - Task definition
  - Why coreference is hard
  - Applications
  - Brief history
- **Part II**: Machine learning for coreference resolution
  - System architecture
  - Computational models
    - Resources and evaluation (corpora, evaluation metrics, …)
  - Employing semantics and world knowledge
- **Part III**: Solving **hard** coreference problems
  - Difficult cases of overt pronoun resolution
    - Relation to the Winograd Schema Challenge

Hard-to-resolve Definite Pronouns

- Resolve definite pronouns for which traditional linguistic constraints on coreference and commonly-used resolution heuristics would not be useful

A Motivating Example (Winograd, 1972)

- The city council refused to give the demonstrators a permit because they feared violence.

- The city council refused to give the demonstrators a permit because they advocated violence.

Another Motivating Example (Hirst, 1981)

- When Sue went to Nadia's home for dinner, she served sukiyaki au gratin.

- When Sue when to Nadia's home for dinner, she ate sukiyaki au gratin.

Another Example

- James asked Robert for a favor, but he refused.
- James asked Robert for a favor, but he was refused.

Yet Another Example

- Sam fired Tom but he did not regret doing so.
- Sam fired Tom although he is diligent.

Focus on certain kinds of sentences

- The target pronoun should
  - appear in a sentence that has two clauses with a discourse connective, where the first clause contains two candidate antecedents and the second contains the pronoun
  - agree in gender, number, semantic class with both candidates

When Sue went to Nadia's home for dinner, she served sukiyaki au gratin.

When Sue when to Nadia's home for dinner, she ate sukiyaki au gratin.

Focus on certain kinds of sentences

- The target pronoun should
  - appear in a sentence that has two clauses with a discourse connective, where the first clause contains two candidate antecedents and the second contains the pronoun
  - agree in gender, number, semantic class with both candidates
- We ensure that each sentence has a twin. Two sentences are twins if
  - their first clauses are the same
  - they have lexically identical pronouns with different antecedents

When Sue went to Nadia's home for dinner, she served sukiyaki au gratin.

When Sue when to Nadia's home for dinner, she ate sukiyaki au gratin.

Dataset

- 941 sentence pairs composed by 30 students who took my undergraduate machine learning class in Fall 2011

Our Approach: Ranking

- Create one ranking problem from each sentence
  - Each ranking problem consists of two instances
    - one formed from the pronoun and the first candidate
    - one formed from the pronoun and the second candidate
- Goal: train a ranker that assigns a higher rank to the instance having the correct antecedent for each ranking problem

Eight Components for Deriving Features

- Narrative chains
- Google
- FrameNet
- Semantic compatibility
- Heuristic polarity
- Machine-learned polarity
- Connective-based relations
- Lexical features

Narrative Chains (Chambers & Jurafsky, 2008)

- Narrative chains are learned versions of **scripts**
  - Scripts represent knowledge of stereotypical event sequences that can aid text understanding
    - Reach restaurant, waiter sits you, gives you a menu, order food,..
- Partially ordered sets of events centered around a protagonist
  - e.g., **borrow-s invest-s spend-s pay-s raise-s lend-s**
    - Someone who borrows something may invest, spend, pay, or lend it
  - can contain a mix of "s" (subject role) and "o" (object role)
    - e.g., the restaurant script

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role

- “he” participates in the “try” and “escape” events as a subject

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
- "he" participates in the "try" and "escape" events as a subject

2) Find the event(s) in which the candidates participate
- Both candidates participate in the "punish" event

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
- "he" participates in the "try" and "escape" events as a subject

2) Find the event(s) in which the candidates participate
- Both candidates participate in the "punish" event

3) Pair each candidate event with each pronoun event
- Two pairs are created: (punish, try-s), (punish, escape-s)

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
- “he” participates in the “try” and “escape” events as a subject

2) Find the event(s) in which the candidates participate
- Both candidates participate in the “punish” event

3) Pair each candidate event with each pronoun event
- Two pairs are created: (punish, try-s), (punish, escape-s)

4) For each pair, extract chains containing both elements in pair
- One chain is extracted, which contains punish-o and escape-s

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
- "he" participates in the "try" and "escape" events as a subject

2) Find the event(s) in which the candidates participate
- Both candidates participate in the "punish" event

3) Pair each candidate event with each pronoun event
- Two pairs are created: (punish, try-s), (punish, escape-s)

4) For each pair, extract chains containing both elements in pair
- One chain is extracted, which contains punish-o and escape-s

5) Obtain role played by pronoun in the candidate's event: object

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
  - "he" participates in the "try" and "escape" events as a subject

2) Find the event(s) in which the candidates participate
  - Both candidates participate in the "punish" event

3) Pair each candidate event with each pronoun event
  - Two pairs are created: (punish, try-s), (punish, escape-s)

4) For each pair, extract chains containing both elements in pair
  - One chain is extracted, which contains punish-o and escape-s

5) Obtain role played by pronoun in the candidate's event: object

6) Find the candidate that plays the extracted role: Tim

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
   - "he" participates in the "try" and "escape" events as a subject
2) Find the event(s) in which the candidates participate
   - Both candidates participate in the "punish" event
3) Pair each candidate event with each pronoun event
   - Two pairs are created: (punish, try-s), (punish, escape-s)
4) For each pair, extract chains containing both elements in pair
   - One chain is extracted, which contains punish-o and escape-s
5) Obtain role played by pronoun in the candidate's event: object
6) Find the candidate that plays the extracted role: Tim

- Creates a binary feature that encodes this heuristic decision

Search Engine (Google)

Lions eat zebras because they are predators.

Search Engine (Google)

Lions eat zebras because they are predators.

1) Replace the target pronoun with a candidate antecedent

Lions eat zebras because lions are predators.

Lions eat zebras because zebras are predators.

2) Generate search queries based on lexico-syntactic patterns

- Four search queries for this example: “lions are”, “zebras are”, “lions are predators”, “zebras are predators

3) Create features where the query counts obtained for the two candidate antecedents are compared

FrameNet

John killed Jim, so he was arrested.

FrameNet

John killed Jim, so he was arrested.

- Both candidates are names, so search queries won't return useful counts.

FrameNet

John killed Jim, so he was arrested.

- Both candidates are names, so search queries won't return useful counts.

- Solution: before generating search queries, replace each name with its FrameNet semantic role
  - "John" with "killer", "Jim" with "victim"
  - Search "killer was arrested", "victim was arrested", …

Semantic Compatibility

- Same as what we did in the Search Engine component, except that we obtain query counts from the Google Gigaword corpus

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

- Use polarity information to resolve target pronouns in sentences that involve comparison

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

- Use polarity information to resolve target pronouns in sentences that involve comparison

1) Assign rank values to the pronoun and the two candidates
   - In first sentence, “Jim” is better, “Ed” is worse, “he” is worse
   - In second sentence, “Jim” is better, “Ed” is worse, “he” is better

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

- Use polarity information to resolve target pronouns in sentences that involve comparison

1) Assign rank values to the pronoun and the two candidates
   - In first sentence, “Jim” is better, “Ed” is worse, “he” is worse
   - In second sentence, “Jim” is better, “Ed” is worse, “he” is better

2) Resolve pronoun to the candidate that has the same rank value as the pronoun

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

- Use polarity information to resolve target pronouns in sentences that involve comparison

1) Assign rank values to the pronoun and the two candidates
   - In first sentence, “Jim” is better, “Ed” is worse, “he” is worse
   - In second sentence, “Jim” is better, “Ed” is worse, “he” is better

2) Resolve pronoun to the candidate that has the same rank value as the pronoun

- Create features that encode this heuristic decision and rank values

Machine-Learned Polarity

- Hypothesis
  - rank values could be computed more accurately by employing a sentiment analyzer that can capture contextual information

Machine-Learned Polarity

- Hypothesis
  - rank values could be computed more accurately by employing a sentiment analyzer that can capture contextual information
- Same as Heuristic Polarity, except that OpinionFinder (Wilson et al., 2005) is used to compute rank values

Connective-Based Relations

Google bought Motorola because they are rich.

Connective-Based Relations

Google bought Motorola because they are rich.

- To resolve “they”, we
  1) Count number of times the triple <“buy”, “because”, “rich”> appears in the Google Gigaword corpus

Connective-Based Relations

Google bought Motorola because they are rich.

- To resolve “they”, we
  1) Count number of times the triple <“buy”, “because”, “rich”> appears in the Google Gigaword corpus
  2) If count is greater than a certain threshold, resolve pronoun to candidate that has the same deep grammatical role as pronoun

Connective-Based Relations

Google bought Motorola because they are rich.

- To resolve “they”, we
  1) Count number of times the triple <“buy”, “because”, “rich”> appears in the Google Gigaword corpus
  2) If count is greater than a certain threshold, resolve pronoun to candidate that has the same deep grammatical role as pronoun
  3) Generate feature based on this heuristic resolution decision

Lexical Features

- Exploit information in the coreference-annotated training texts

Lexical Features

- Exploit information in the coreference-annotated training texts
- Antecedent-independent features
  - Unigrams
  - Bigrams (pairing word before connective and word after connective)
  - Trigrams (augmenting each bigram with connective)

Lexical Features

- Exploit information in the coreference-annotated training texts

- Antecedent-independent features
  - Unigrams
  - Bigrams (pairing word before connective and word after connective)
  - Trigrams (augmenting each bigram with connective)

- Antecedent-dependent features
  - pair a candidate's head word with
    - its governing verb
    - its modifying adjective
    - the pronoun's governing verb
    - the pronoun's modifying adjective

Evaluation

- **Dataset**
  - 941 annotated sentence pairs (70% training; 30% testing)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Evaluation metrics**: percentages of target pronouns that are
  - correctly resolved
  - incorrectly resolved
  - unresolved (no decision)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Evaluation metrics**: percentages of target pronouns that are
  - correctly resolved
  - incorrectly resolved
  - unresolved (no decision)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Evaluation metrics**: percentages of target pronouns that are
  - correctly resolved
  - incorrectly resolved
  - unresolved (no decision)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Evaluation metrics**: percentages of target pronouns that are
  - correctly resolved
  - incorrectly resolved
  - unresolved (no decision)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Unadjusted Scores**
  - Raw scores computed based on a resolver's output

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Unadjusted Scores**
  - Raw scores computed based on a resolver's output
- **Adjusted Scores**
  - "Force" a resolver to resolve every pronoun by probabilistically assuming that it gets half of the unresolved pronouns right

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Three baseline resolvers**
  - Stanford resolver (Lee et al., 2011)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Three baseline resolvers**
  - Stanford resolver (Lee et al., 2011)
  - Baseline Ranker: same as our ranking approach, except that ranker is trained using the 39 features from Rahman & Ng (2009)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Three baseline resolvers**
  - Stanford resolver (Lee et al., 2011)
  - Baseline Ranker: same as our ranking approach, except that ranker is trained using the 39 features from Rahman & Ng (2009)
  - The Combined resolver combines Stanford and Baseline Ranker:
    - Baseline Ranker is used only when Stanford can't make a decision

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- Stanford outperforms Baseline ranker
- Combined resolver does not outperform Stanford
- Our system outperforms Stanford by 18 accuracy points

Ablation Experiments

- Remove each of the 8 components one at a time
- Accuracy drops significantly (paired t-test, $p < 0.05$) after each component is removed
- Most useful: Narrative chains, Google, Lexical Features
- Least useful: FrameNet, Learned Polarity

Peng et al. (2015)

- An alternative approach to address this task
- Define two types of predicate schemas
  - Collect statistics for **instantiated** schemas from different knowledge sources: Gigaword, Wikipedia, and the Web

Motivating Example

The bee landed on the flower because it had pollen.

- To correctly resolve 'it', we need to know:

  S(have(m=[the flower], a=pollen)) >
  S(have(m=[the bee], a=pollen))

- Corresponding predicate schema:

$$S(pred_m(m,a))$$

Motivating Example

The bird perched on the limb and it bent.

- To correctly resolve 'it', we need to know:

  S(bend(m=[the limb], a=*)) >
  S(bend(m=[the bird], a=*))

- Corresponding predicate schema:

$$S(\mathrm{pred}_m(m,*))$$

Predicate Schema Type 1

- We saw

$$S(pred_m(m,a)) \text{ and } S(pred_m(m,*))$$

- More generally, we also need

$$S(pred_m(a,m)) \text{ and } S(pred_m(*,m))$$

Predicate Schema Type 1

- We saw

$$S(pred_m(m,a)) \text{ and } S(pred_m(m,*))$$

- More generally, we also need

$$S(pred_m(a,m)) \text{ and } S(pred_m(*,m))$$

Predicate Schema Type 1

- We saw

$$S(pred_m(m,a)) \text{ and } S(pred_m(m,*))$$

- More generally, we also need

$$S(pred_m(a,m)) \text{ and } S(pred_m(*,m))$$

Predicate Schema Type 1

- We saw

$$S(pred_m(m,a)) \text{ and } S(pred_m(m,*))$$

- More generally, we also need

$$S(pred_m(a,m)) \text{ and } S(pred_m(*,m))$$

Motivating Example

Ed was afraid of Tim because <u>he</u> gets scared around new people.

- To correctly resolve 'he', we need to know:
  S(be afraid of(m,a), because, get scared(m,a')) >
  S(be afraid of(a,m), because, get scared(m,a'))

- Corresponding predicate schemas

$$S(\text{pred1}_m(m,a),\ dc,\ \text{pred2}_m(m,a'))$$
$$S(\text{pred1}_m(a,m),\ dc,\ \text{pred2}_m(m,a'))$$

Motivating Example

Ed was afraid of Tim because he gets scared around new people.

- To correctly resolve 'he', we need to know:
  S(be afraid of(m,a), because, get scared(m,a')) >
  S(be afraid of(a,m), because, get scared(m,a'))

- Corresponding predicate schemas
  $S(pred1_m(m,a), dc, pred2_m(m,a'))$
  $S(pred1_m(a,m), dc, pred2_m(m,a'))$

Motivating Example

Ed was afraid of Tim because he gets scared around new people.

- To correctly resolve 'he', we need to know:
  S(be afraid of(m,a), because, get scared(m,a')) >
  S(be afraid of(a,m), because, get scared(m,a'))

- Corresponding predicate schemas

$$S(pred1_m(m,a), dc, pred2_m(m,a'))$$
$$S(pred1_m(a,m), dc, pred2_m(m,a'))$$

Predicate Schema Type 2

- So far, we have

$S(pred1_m(m,a), dc, pred2_m(m,a'))$

$S(pred1_m(a,m), dc, pred2_m(m,a'))$

- More generally, we also need:

$S(pred1_m(m,a), dc, pred2_m(a',m))$

$S(pred1_m(a,m), dc, pred2_m(a',m))$

$S(pred1_m(m,*), dc, pred2_m(*,m))$

$S(pred1_m(*,m), dc, pred2_m(*,m))$

Predicate Schema Type 2

- So far, we have

$S(pred1_m(m,a), dc, pred2_m(m,a'))$

$S(pred1_m(a,m), dc, pred2_m(m,a'))$

- More generally, we also need:

$S(pred1_m(m,a), dc, pred2_m(a',m))$

$S(pred1_m(a,m), dc, pred2_m(a',m))$

$S(pred1_m(m,*), dc, pred2_m(*,m))$

$S(pred1_m(*,m), dc, pred2_m(*,m))$

Predicate Schema Type 2

- So far, we have

$S(pred1_m(m,a), dc, pred2_m(m,a'))$

$S(pred1_m(a,m), dc, pred2_m(m,a'))$

- More generally, we also need:

$S(pred1_m(m,a), dc, pred2_m(a',m))$

$S(pred1_m(a,m), dc, pred2_m(a',m))$

$S(pred1_m(m,*), dc, pred2_m(*,m))$

$S(pred1_m(*,m), dc, pred2_m(*,m))$

Collecting Statistics for the Schemas

- … from Gigaword, Wikipedia, and the Web

Using the Statistics

- As features and/or as constraints for their coreference resolver

Peng et al.'s results on our dataset

Metric	Illinois	IlliCons	Rahman and Ng (2012)	KnowFeat	KnowCons	KnowComb
Precision	51.48	53.26	73.05	71.81	74.93	**76.41**

Summary

- There is a recent surge of interest in these hard, but incredibly interesting pronoun resolution tasks
- They could serve as an alternative to the Turing Test (Levesque, 2011)
  - This challenge is known as the Winograd Schema Challenge
    - Announced as a shared task in AAAI 2014
    - Sponsored by Nuance
- Additional test cases available from Ernest Davis' website:

  https://www.cs.nyu.edu/**davis**e/papers/WS.html

Challenges

Challenges: New Models

- Can we jointly learn coreference resolution with other tasks?
  - Exploit cross-task constraints to improve model learning
  - Durrett & Klein (2014): jointly learn coreference with two tasks
    - Named entity recognition (coarse semantic typing)
    - Entity linking (matching to Wikipedia entities)

    using a graphical model, encoding soft constraints in factors
    - Use semantic info in Wikipedia for better semantic typing
    - Use semantic types to disambiguate tricky Wikipedia links
    - Ensure consistent type predictions across coreferent mentions
    - …

Challenges: New Models

- Can we jointly learn coreference resolution with other tasks?
  - Exploit cross-task constraints to improve model learning
  - Durrett & Klein (2014): jointly learn coreference with two tasks
    - Named entity recognition (coarse semantic typing)
    - Entity linking (matching to Wikipedia entities)

    using a graphical model, encoding soft constraints in factors
    - Use semantic info in Wikipedia for better semantic typing
    - Use semantic types to disambiguate tricky Wikipedia links
    - Ensure consistent type predictions across coreferent mentions
    - …
  - Hajishirzi et al. (2013): jointly learn coreference w/ entity linking
  - Can we jointly learn entity coreference with event coreference?

Challenges: New Features

- There is a limit on how far one can improve coreference resolution using machine learning methods
  - A good model can profitably exploit the available features, but if the knowledge needed is not present in the data, there isn't much that the model can do

Challenges: New Features

- There is a limit on how far one can improve coreference resolution using machine learning methods
  - A good model can profitably exploit the available features, but if the knowledge needed is not present in the data, there isn't much that the model can do

- We know that semantics and world knowledge are important
  - But it's hard to use them to improve state-of-the-art systems
  - Wiseman: learn non-linear representations from raw features
  - What if we learn such representations from complex features, including those that encode world knowledge?
  - Can we leverage recent advances in distributional lexical semantics (e.g., word embeddings)?

Challenges: New Languages

- Low-resource languages
  - Large lexical knowledge bases may not be available
    - Can we learn world knowledge from raw text?
      - **Idea**: using appositive constructions
        - E.g., Barack Obama, president of the United States, …

Challenges: New Languages

- Low-resource languages
  - Large lexical knowledge bases may not be available
    - Can we learn world knowledge from raw text?
      - **Idea**: using appositive constructions
        - E.g., Barack Obama, president of the United States, …
  - Large coreference-annotated corpora may not be available
    - Can we employ weakly supervised learning or active learning?
    - Can we exploit resources from a resource-rich language?
      - **Idea**: translation-based coreference annotation projection

Translation-Based Projection: Example

玛丽告诉约翰她非常喜欢他。

Translation-Based Projection: Example

玛丽告诉约翰她非常喜欢他。

1. Machine-translate document from target to source

Translation-Based Projection: Example

玛丽告诉约翰她非常喜欢他。

Mary told John that she liked him a lot.

1. Machine-translate document from target to source

Translation-Based Projection: Example

玛丽告诉约翰她非常喜欢他。

Mary told John that she liked him a lot.

2. Run resolver on the translated document
   - to extract mentions and produce coreference chains

Translation-Based Projection: Example

玛丽告诉约翰她非常喜欢他。

Mary told John that she liked him a lot.

2. Run resolver on the translated document
   - to extract mentions and produce coreference chains

Translation-Based Projection: Example

玛丽告诉约翰她非常喜欢他。

Mary told John that she liked him a lot.

3. Project annotations from source back to target
   - project mentions

Translation-Based Projection: Example

[玛丽]告诉[约翰][她]非常喜欢[他]。

Mary told John that she liked him a lot.

3. Project annotations from source back to target
   - project mentions

Translation-Based Projection: Example

[玛丽]告诉**[**约翰**][她]非常喜**欢**[他]**。

Mary told John that she liked him a lot.

3. Project annotations from source back to target
   - project mentions
   - project coreference chains

Translation-Based Projection: Example

[玛丽]告诉[约翰][她]非常喜欢[他]。

Mary told John that she liked him a lot.

3. Project annotations from source back to target
   - project mentions
   - project coreference chains

Challenges: New Coreference Tasks

- **Bridging** [Non-identity coreference]
  - Set-subset relation, part-whole relation
- **Event coreference resolution**
  - Determines which event mentions refer to the same event
  - Difficult because for two events to be coreferent, one needs to check whether their arguments/participants are compatible
- **Partial coreference relation** [Non-identity coreference]
  - **subevent**
    - Subevent relations form a sterotypical sequence of events
      - e.g., bombing → destroyed → wounding
  - **membership**
    - multiple instances of the same kind of event
      - e.g., I attended three parties last month. The 1st one was the best.

Challenges: New Evaluation Metrics

- Designing evaluation metrics is a challenging task
- There are four commonly used coreference evaluation metrics (MUC, B^3, CEAF, BLANC), but it's not clear which of them is the best
  - Can we trust them? (Moosavi & Strube, 2016)
  - **Weaknesses**
    - Linguistically agnostic
      - Are all links equally important?
        - E.g., 3 mentions: Hillary Clinton, she, she
        - System 1: Clinton-she; System 2: she-she
    - Hard to interpret the resulting F-scores
      - Can the scores tell which aspects of a system can be improved?