

# **The Mention-Pair Architecture**

Natalie Parde

UIC CS 421

# The Mention-Pair Architecture

## Simple premise:

- Given:
  - Pair of mentions (candidate anaphor and candidate antecedent)
- Decide:
  - Whether or not they corefer

## How does this work?

- Compute coreference probabilities for every plausible pair of mentions
- Goal: High probability for actual coreferring pairs, and low probability for other pairs

# The Mention-Pair Architecture

The **University of Illinois at Chicago** is an excellent place to study **natural language processing**. **UIC** has many **faculty** currently working in **NLP**, including but not limited to **Natalie Parde**, **Barbara Di Eugenio**, **Cornelia Caragea**, **Bing Liu**, and **Philip Yu**. The school is located in bustling downtown **Chicago** and as a **bonus** it will be opening a snazzy new (non-brutalist) **CS building** in 2022.

# The Mention-Pair Architecture

The diagram illustrates the Mention-Pair Architecture. A blue box contains a paragraph of text. Red circles highlight specific mentions within the text: 'University of Illinois at Chicago', 'natural language processing', 'UIC', 'faculty', 'NLP', 'Natalie Parde', 'Barbara Di Eugenio', 'Cornelia Caragea', 'Bing Liu', 'Philip Yu', 'Chicago', 'bonus', 'CS building', and '2022'. Blue lines connect these mentions to a central point above the text, representing the architecture's structure.

The University of Illinois at Chicago is an excellent place to study natural language processing. UIC has many faculty currently working in NLP including but not limited to Natalie Parde, Barbara Di Eugenio, Cornelia Caragea, Bing Liu, and Philip Yu. The school is located in bustling downtown Chicago and as a bonus it will be opening a snazzy new (non-brutalist) CS building in 2022.

# The Mention-Pair Architecture

The diagram illustrates the Mention-Pair Architecture by showing a text snippet with various mentions and pairs highlighted. The text is: "The University of Illinois at Chicago is an excellent place to study natural language processing. UIC has many faculty currently working in NLP, including but not limited to Natalie Parde, Barbara Di Eugenio, Cornelia Caragea, Bing Liu, and Philip Yu. The school is located in bustling downtown Chicago and as a bonus it will be opening a snazzy new (non-brutalist) CS building in 2022." The mentions are circled in red: "University of Illinois at Chicago", "UIC", "NLP", "Natalie Parde", "Barbara Di Eugenio", "Cornelia Caragea", "Bing Liu", "Philip Yu", "Chicago", "bonus", "snazzy new (non-brutalist) CS building", and "2022". The pairs are indicated by grey arcs connecting the mentions: "University of Illinois at Chicago" to "UIC", "NLP" to "Natalie Parde", "Barbara Di Eugenio", "Cornelia Caragea", "Bing Liu", and "Philip Yu", "Chicago" to "snazzy new (non-brutalist) CS building", and "bonus" to "2022".

The University of Illinois at Chicago is an excellent place to study natural language processing. UIC has many faculty currently working in NLP, including but not limited to Natalie Parde, Barbara Di Eugenio, Cornelia Caragea, Bing Liu, and Philip Yu. The school is located in bustling downtown Chicago and as a bonus it will be opening a snazzy new (non-brutalist) CS building in 2022.

# The Mention-Pair Architecture

The diagram illustrates the Mention-Pair Architecture. It features a text box with a blue border containing the following text: "The University of Illinois at Chicago is an excellent place to study natural language processing. UIC has many faculty currently working in NLP, including but not limited to Natalie Parde, Barbara Di Eugenio, Cornelia Caragea, Bing Liu, and Philip Yu. The school is located in bustling downtown Chicago, and as a bonus it will be opening a snazzy new (non-brutalist) CS building in 2022." Red annotations highlight specific mentions and their relationships: "University of Illinois at Chicago" is circled, "UIC" is circled, "The school" is circled, and "it" is circled. Red arrows show the relationships: one arrow points from "University of Illinois at Chicago" to "UIC", another from "The school" to "it", and a third from "UIC" to "it".

The University of Illinois at Chicago is an excellent place to study natural language processing. UIC has many faculty currently working in NLP, including but not limited to Natalie Parde, Barbara Di Eugenio, Cornelia Caragea, Bing Liu, and Philip Yu. The school is located in bustling downtown Chicago, and as a bonus it will be opening a snazzy new (non-brutalist) CS building in 2022.

# How do we learn these probabilities?

- Select training samples
  - One positive instance  $(m_i, m_j)$  where  $m_j$  is the closest antecedent to  $m_i$
  - A negative instance  $(m_i, m_k)$  for each  $m_k$  between  $m_j$  and  $m_i$
- Extract features
  - Hand-built features, and/or
  - Implicitly learned representations
- Train classification model

# How do we make predictions?

- Apply the trained classifier to each test instance in a clustering step
  - **Closest-first clustering**
    - For mention  $i$ , classifier is run backwards through prior  $i-1$  mentions
    - First antecedent with probability  $> 0.5$  is selected and linked to  $i$
  - **Best-first clustering**
    - Classifier is run on all possible  $i-1$  antecedents
    - Mention with highest probability is selected as the antecedent for  $i$





# Mention-Pair Architecture

- Advantage:
  - **Simplest** coreference resolution architecture
- Disadvantage:
  - **Doesn't directly compare candidate antecedents** with one another
  - **Considers only mentions**, not overall entities



# How can we address these limitations?

- One option: The **Mention-Rank Architecture**
  - Directly compares antecedents with one another
  - Selects the highest-scoring antecedent for each anaphor
- How does this work?
  - For a mention  $i$ , we have:
    - Random variable  $y_i$  ranging over the values  $Y(i) = \{1, \dots, i - 1, \varepsilon\}$ 
      - $\varepsilon$  = dummy mention meaning  $i$  does not have an antecedent
  - At test time, for  $i$  the model computes a softmax over all possible antecedents
  - When training:
    - Use heuristics to determine the best antecedent for an anaphor (e.g., closest = best)
    - Or, learn more optimal ways to model latent antecedents using machine learning