Coreference Resolution

Natural Language Processing

(based on revision of Chris Manning Lectures)

Announcement

- TA announcements (if any)...

Suggested Readings

1. https://web.stanford.edu/~jurafsky/slp3/21.pdf (coreference resolution)

Intro

Coreference Resolution

Identify all **mentions** that refer to the same entity in the world

> Victoria Chen, CFO of Megabucks Banking, saw her pay jump to $2.3 million, as the 38-year-old became the company's president. It is widely known that she came to Megabucks from rival Lotsabucks.

Each of the underlined phrases is referring to Victoria Chen. We called linguistic expressions like *her* or *Victoria Chen* **mentions** or **referring expressions**, and the discourse entity that is referred to Victoria Chen the **referent**. Two or more referring expressions that are used to refer to the same discourse entity are said to **corefer**; thus, *Victoria Chen* and *she* **corefer**.

In the coreference resolution algorithm, the output would need to find at least four coreference **chains/clusters**: {Victoria Chen, her, the 38 year-old, She}, {Megabucks Banking, the company, Megabucks}, {her pay}, {Lotsabucks}. Anything with single mentions is called **Singleton**.

Anaphora vs. Coreference vs. Entity Linking

- **Coreference** is when **two mentions refer to the same entity in the world**
  - ***Chaklam*** *loves deep learning.* ***He*** *plays soccer.* ***Chaky*** *also loves coding*.
    - *Chaky*, *He*, *Chaklam* are said to **corefer**, where *Chaklam* is the **referent**
    - Coreference resolution find **whether two mentions corefer**
- **Anaphora** is the **reference** in a text to an entity that has been previously introduced
  - ***Chaklam*** *said* ***he*** *would give a quiz*.
    - Chaklam is the antecedent, and he is the anaphor
  - Anaphora **can or cannot be** coreference
    - ***Every dancer*** *twisted* ***her*** *knee.*
      - Every dancer is the antecedent, and her is the anaphor. But they DO NOT corefer
- **Entity linking** is the process of mapping a discourse entity to some real-world individual
  - *Washington is at United States*. Does Washington mapped to George Washington or State of Washington?
  - Coreference resolution and entity linking can work together in a NLP pipeline

Coreference Resolution

Clearly, coreference resolution is an important criteria for successful NLP tasks such as question-answering, translation, dialogue and much more…

Dialogue

- *"There is a 2pm flight on United and a 4pm one on Cathay Pacific"*
- User said in chatbot: *"I want the second one"*

Question answering

- Context: *"Chaky loves deep learning. He is born in Hong Kong"*
- Question: *Where is Chaky born?*

Coreference Tasks and Datasets

Coreference Resolution Task

Given a text ***T***, find all entities and the coreference links between them

[Victoria Chen]^1_a, CFO of [Megabucks Banking]^2_a, saw [[her]^1_b pay]^3_a jump to \$2.3 million, as [the 38-year-old]^1_c also became [[the company]^2_b's president. It is widely known that [she]^1_d came to [Megabucks]^2_c from rival [Lotsabucks]^4_a.

The output could be (many variants):

1. {*Victoria Chen, her, the 38-year-old, She*}
2. {*Megabucks Banking, the company, Megabucks*}
3. { *her pay*}
4. { *Lotsabucks*}

Coreference Resolution Task

- What is counted as **mention** and what **links** are annotated differ from task to task and dataset to dataset

- Some datasets **do not label singletons**, making the task easier
  - **OntoNotes** contains hand-annotated Chinese and English coreference datasets of roughly one million words each, consisting of newswire, magazine articles, broadcast news, etc., as well as 300,000 words of annotated Arabic newswire
    - Does not label singletons

- Some tasks use **gold mention-detection**, i.e., the system is given human-labeled mention boundaries and the task is just to cluster these gold mentions
  - Eliminates the need to detect and segment mentions from running text

Mention Detection

Mention Detection

- Obviously, the first stage of coreference is **mention detection**
  - Find the spans of text that constitute each mention
- Many traditional NLP works well:
  - Pronouns:
    - Use a POS tagger
  - Named entities
    - Use a NER system
  - Noun phrases
    - Use a parser (constituency parser - next week!)
- A neural model like **BERT masking mention span** would also work well
- Not as easy as you think
  - **Every** student, **No** student, **The best donut in the world**, **100 miles**
  - Solution: After coreference resolution, discard all singletons

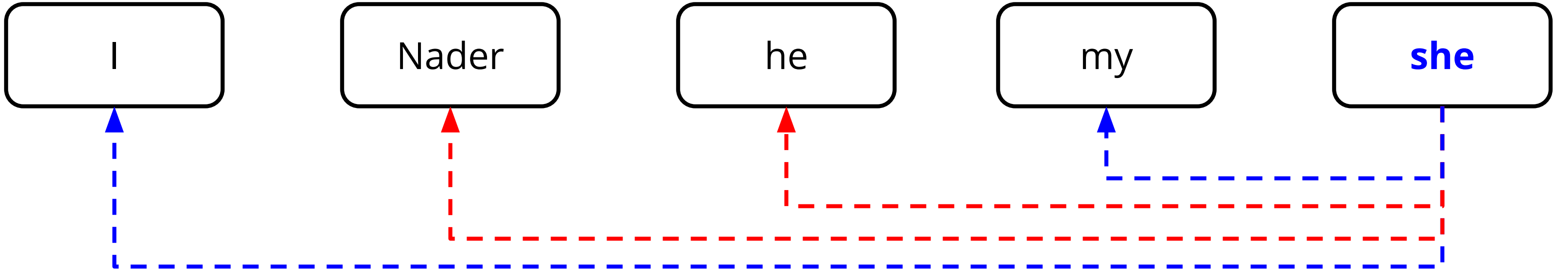
Coreference Algorithms

Method 1: Mention-Pair

- Simplest and most influential
- Based around a classifier, in which given a pair of mentions, i.e., a **candidate anaphor**, and **all potential antecedents**, and makes a **binary classification** whether they are coreferring or not: $p(\mathbf{m_i}, \mathbf{m_j})$
  - For positive examples, p is near 1, for negative samples, p is near 0

Given this example, and let say we are currently working on "she" as the anaphor candidate.

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



Method 1: Mention-Pair Training

- N mentions in a document
- y_{ij} = 1 if mentions $\mathbf{m_i}$ and $\mathbf{m_j}$ are coreference, -1 if otherwise
- Just train with regular cross-entropy (note that it is simply binary logit loss...)

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

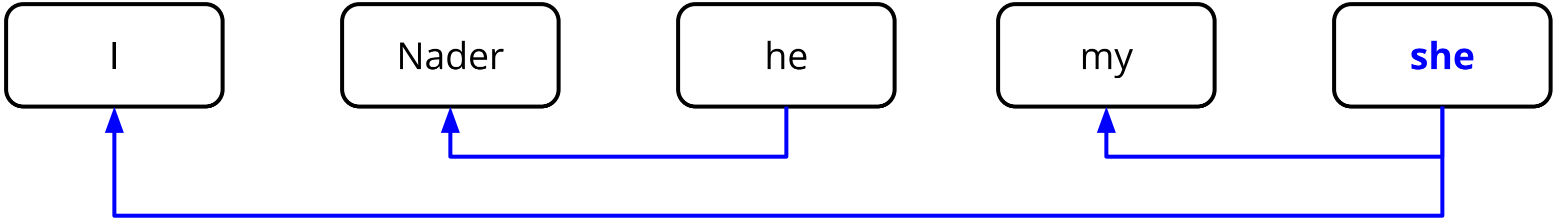
Iterate through mentions (previously occurring mentions)

Iterate through candidate antecedents (previously occurring mentions)

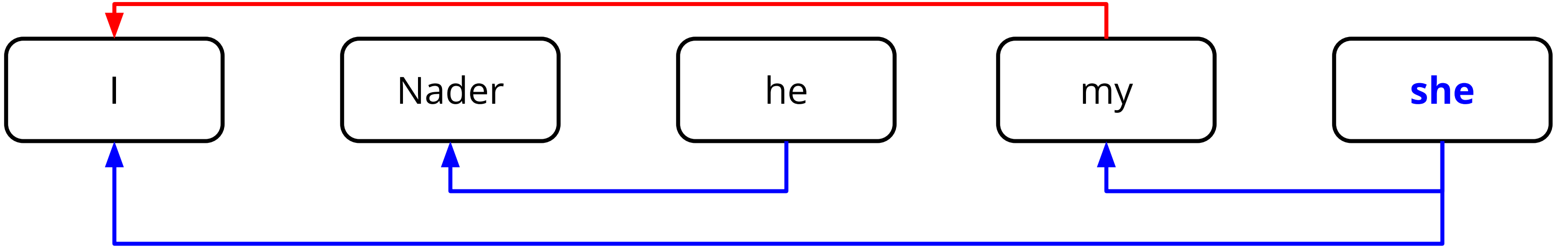
Coreferent mentions pairs should get high probability, others should get low prob

Method 1: Mention-Pair Test Time

- On inference, pick some threshold (e.g., 0.5) and add coreference links between mention pairs where $p(\mathbf{m_i}, \mathbf{m_j})$ is above the threshold



- Take the **transitive** closure to get the clustering



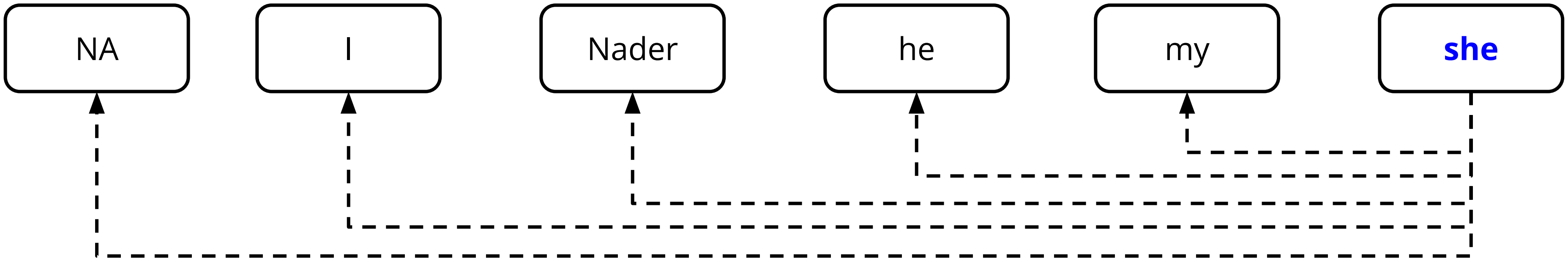
Method 1: Mention-Pair Disadvantages

- Clear advantage is simplicity, but has one main problem
    - Does not directly **compare** candidate antecedents to each other, so it’s not trained to decide, between two likely antecedents, which one is in fact better

Method 2: Mention Ranking

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

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

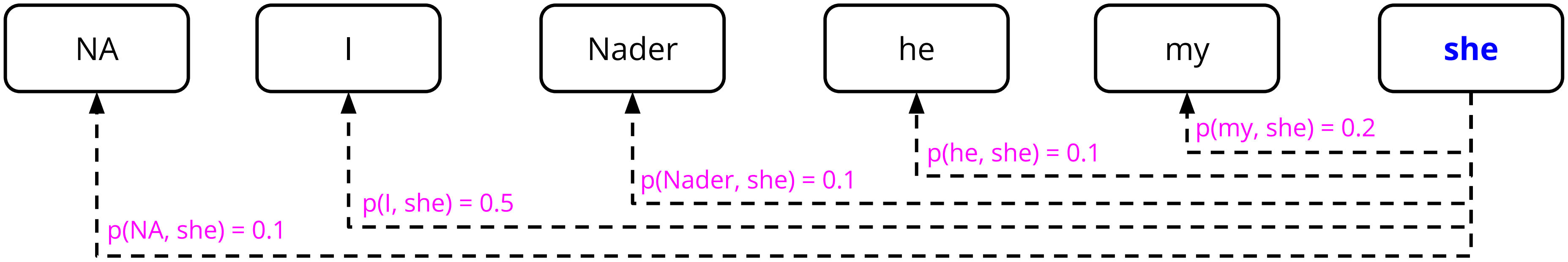


best antecedent for *she*?

Method 2: Mention Ranking

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

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

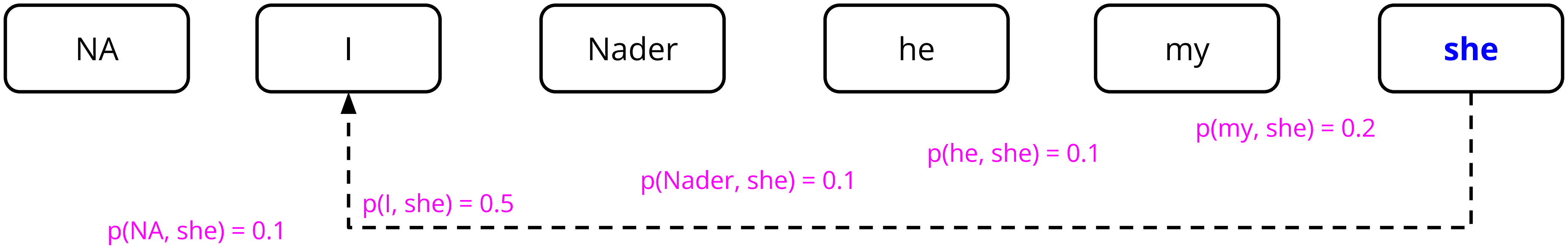


calculate the softmax prob

Method 2: Mention Ranking

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

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



Only add the highest scoring coreference link

Method 2: Mention Ranking

Mathematically, we want to maximize this probability:

$$\sum_{j=1}^{i-1} 1(y_{ij} = 1)p(\mathbf{m}_j, \mathbf{m}_i)$$

Iterate through candidate antecedents (previously occurring mentions)

For ones that are coreferent to $\mathbf{m}_j$

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

How do we compute the probabilities?

For both mention-pair and mention ranking, there is a probability term that we have to compute. We can compute using three main ways:

1. A non-neural statistical classifier (use features)
2. Simple neural network
3. More advanced model using LSTMs, attention, transformers

Coreference Algorithms

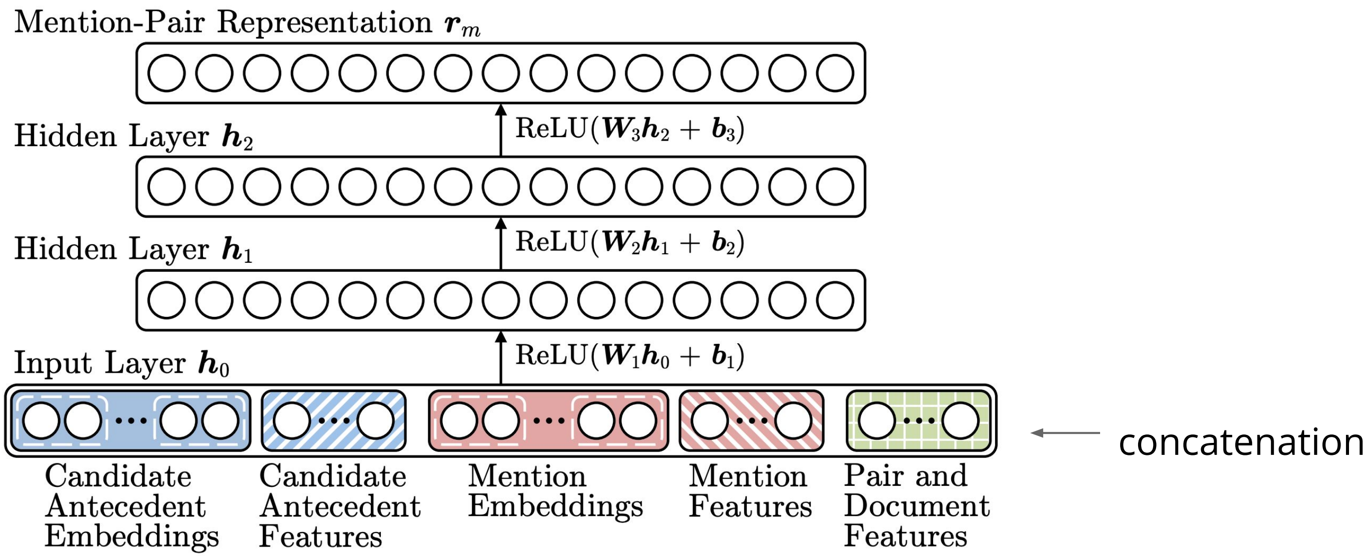
1. A non-neural statistical classifier

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

2. Neural Coref Model [Clark and Manning, ACL 2016]

Standard feed-forward neural network (uses mention-pair)

- Input layer: word embeddings and a few categorical features



Improving Coreference Resolution by Learning Entity-Level Distributed Representations, Clark and Manning 2016, https://arxiv.org/pdf/1606.01323.pdf

3. End-to-end Neural Coref Model [Lee et al., EMNLP 2017]

biLSTM to learning the representations; consider every span of text (uses mention-ranking)

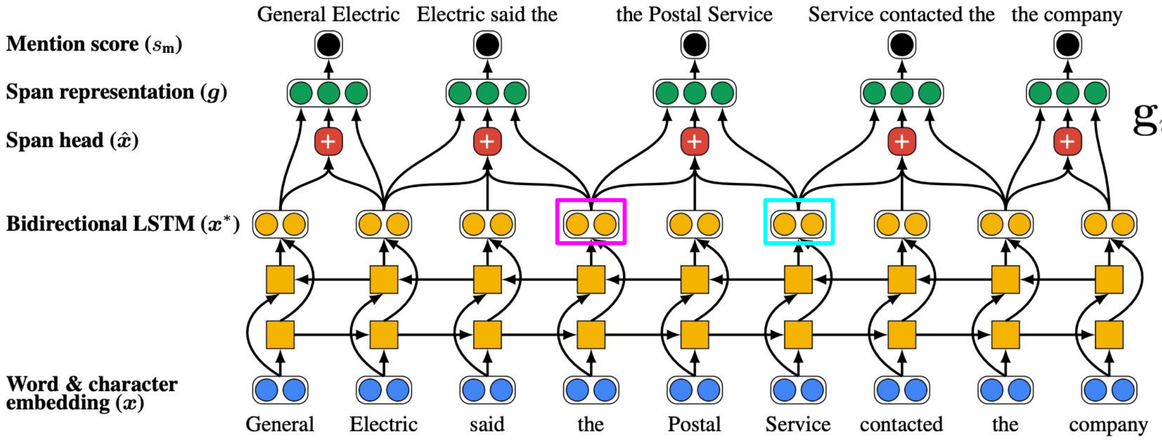


Figure 1: First step of the end-to-end coreference resolution model, which computes embedding representations of spans for scoring potential entity mentions. Low-scoring spans are pruned, so that only a manageable number of spans is considered for coreference decisions. In general, the model considers all possible spans up to a maximum width, but we depict here only a small subset.

additional features

Span representation for "the Postal Service":

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

Attention-based representation of the span words (i.e., the Postal Service)

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

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

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

End-to-end Neural Coreference Resolution, Lee et al. 2017,, https://arxiv.org/pdf/1707.07045.pdf

3. End-to-end Neural Coref Model

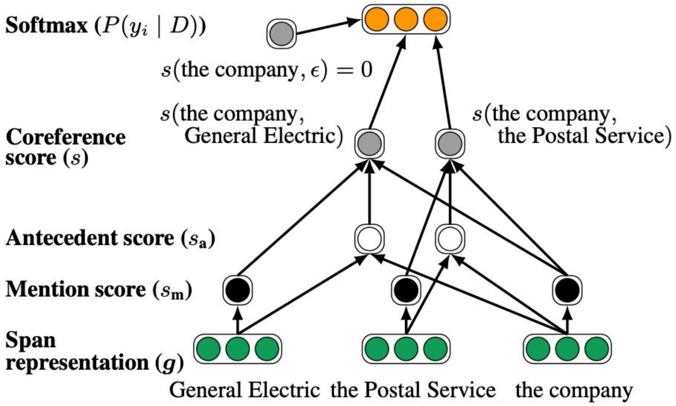


Figure 2: Second step of our model. Antecedent scores are computed from pairs of span representations. The final coreference score of a pair of spans is computed by summing the mention scores of both spans and their pairwise antecedent score.

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

$$s(i,j) = s_m(i) + s_m(j) + s_a(i,j)$$

Are spans i and j coreferent mentions?

Is *i* a mention?

Is *j* a mention?

Do they look coreferent?

$$s_m(i) = \mathbf{w}_m \cdot \mathrm{FFNN}_m(\mathbf{g}_i)$$

$$s_a(i,j) = \mathbf{w}_a \cdot \mathrm{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i,j)])$$

Multiplicative interactions between the representations

Extra features

3. End-to-end Neural Coref Model

- Main problem
  - Very computationally expensive to consider every spans
- Solution
  - Prune some spans that are not likely a mention

4. BERT-based coref [Joshi et al., ACL 2019]

- **SpanBERT**: pretrains BERT using span masking techniques so it can perform well on span-based tasks such as coref or QA

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$

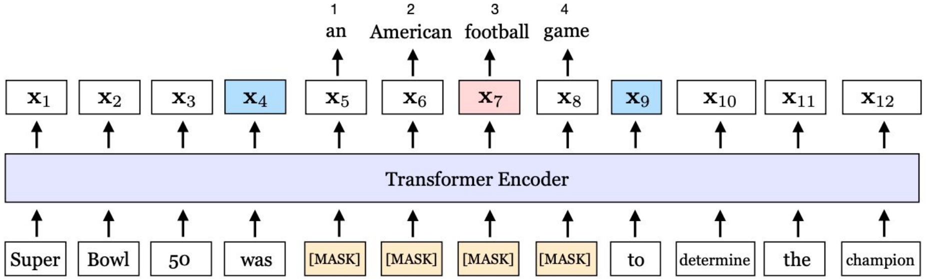


Figure 1: An illustration of SpanBERT training. The span *an American football game* is masked. The span boundary objective (SBO) uses the output representations of the boundary tokens, $\mathbf{x}_4$ and $\mathbf{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and SBO loss terms for predicting the token, *football* (in pink), which as marked by the position embedding $\mathbf{p}_3$, is the *third* token from x_4.

SpanBERT: Improving Pretraining by Representing and Predicting Spans, Joshi et al. 2019, https://arxiv.org/pdf/1907.10529.pdf

4. BERT-based coref [Wu et al., ACL 2020]

CorefQA: treats coreference as QA

Original Passage

In addition , *many people* were poisoned when **toxic gas** was released. *They* were poisoned and did not know how to protect *themselves* against **the poison**.

Our formulation

Q1: Who were poisoned when toxic gas was released?

A1: [*They*, *themselves*]

Q2: What was released when many people were poisoned?

A2: [**the poison**]

Q3: Who were poisoned and did not know how to protect themselves against the poison?

A3: [*many people*, *themselves*]

Q4: Whom did they not know how to protect against the poison?

A4: [*many people*, *They*]

Q5: They were poisoned and did not know how to protect themselves against what?

A5: [**toxic gas**]

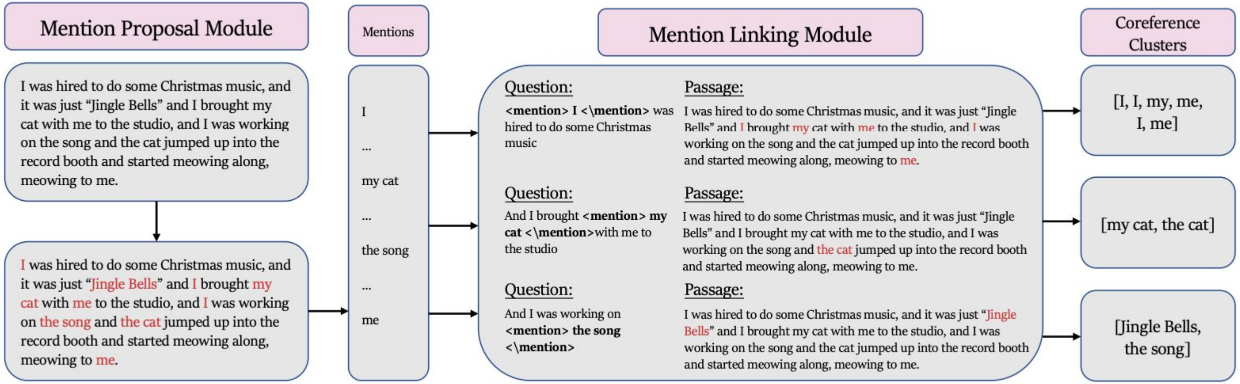


Figure 2: The overall architecture of our CorefQA model. The input passage is first fed into the Mention Proposal Module 3.3 to obtain candidate mentions. Then the Mention Linking Module 3.4 is used to extract coreferent mentions from the passage for each proposed mention. The coreference clusters are obtained using the scores produced in the above two stages.

CorefQA: Coreference Resolution as Query-based Span Pediction, Wu et al. 2020, https://aclanthology.org/2020.acl-main.622.pdf

Evaluation

Evaluation

- Essentially a clustering problem
  - For each mention, compute a precision and a recall

$$\text{Precision} = \sum_{i=1}^{N} w_i \frac{\#\ of\ correct\ mentions\ in\ hypothesis\ chain\ containing\ entity_i}{\#\ of\ mentions\ in\ hypothesis\ chain\ containing\ entity_i}$$

$$\text{Recall} = \sum_{i=1}^{N} w_i \frac{\#\ of\ correct\ mentions\ in\ hypothesis\ chain\ containing\ entity_i}{\#\ of\ mentions\ in\ reference\ chain\ containing\ entity_i}$$

- We evaluate by comparing a set of **hypothesis chains** or clusters H, against a set of **gold or reference chains** R or clusters from human labeling

Coref Performance

Evaluated on **CoNLL-2021** shared task (based on OntoNotes we mentioned earlier). The average F1 of MUC, B^3, and CEAF (all basically based on H and R) and is used.

Model	**English**	**Chinese**
Lee et al. (2010) - Rule-based	55	50
Clark and Manning (2016) - neural coref model	65.4	63.7
Lee et al. (2017) - end-to-end neural coref model	67.2	-
Joshi et al. (2019 - SpanBERT	79.6	-
Wu et al. (2020) - corefQA	83.1	-

Summary

- **Coreference** is a useful, challenging, and linguistically interesting task
  - Main dataset is OntoNotes and CoNLL shared task
  - Two clustering algorithms: **mention pair** and **mention clustering**
  - Many ways to compute probabilities
    - Feature-based - require a lot of manual labor
    - Neural-based - still seems requiring some additional features
    - BERT-based - masking seems to greatly improve; does not require features

- Systems are getting better but **most models still make many mistakes**
  - OntoNotes coref is pretty easy, because it's based on news
  - Imagine a novel or actual complicated how-to instructions