CMPE 297, INSTRUCTOR: JORJETA JETCHEVA

COREFERENCE RESOLUTION

COREFERENCE RESOLUTION INTRODUCTION

- Determining who is being referred to in a piece of text is an important language understanding task, e.g.,

  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.

- All the references to Victoria Chen (in magenta above) are called mentions or referring expressions
- The entity referred to (Victoria Chen) is called a referent
- Two or more referring expressions that refer to the same referent are said to corefer

MOTIVATING EXAMPLES

- Dialogue System:

  Virtual Assistant: There is a 2pm flight on United and a 4pm one on Cathay Pacific.
  User: I'll take the second one.

- QA system asking about Marie Curie's birthplace, needs to be able to figure out that *She* refers to Marie Curie in the sentence *She was born in Warsaw*

RELEVANT CONCEPTS

- A discourse model is the mental model that a person builds in their mind as they read text
  - E.g., of entities and their properties and relationships
- NLU is based on interpreting text in the context of the discourse model that has been built so far
- When a referent is first mentioned in text, it is said that a representation of it is evoked into the model
- Subsequent references to the referent are described as an access of its representation from the model

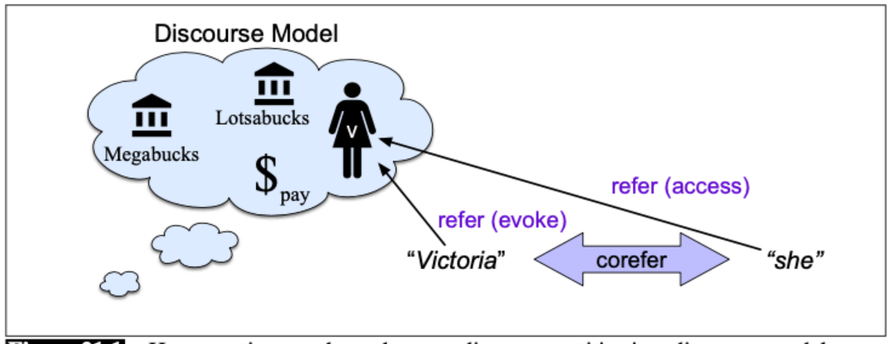

Figure 21.1 How mentions evoke and access discourse entities in a discourse model.

SOME TERMINOLOGY

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.

- Singleton - an entity mentioned in text only once (no references to it later on)
- Anaphora - reference to a previously introduced entity
- Anaphor, Anaphoric - the referring expression (e.g., she) to a previously introduced entity
- Antecedent - a prior mention that an anaphor corefers with (e.g., Victoria Chen)
- Coreference chain/cluster - the set of coreferring expressions
- Entity linking/resolution: mapping a discourse entity to a real-world individual (typically implemented as a reference in an ontology, e.g., Wikipedia)

OTHER TYPES OF COREFERENCE BESIDES ONES FOCUSED ON ENTITIES

- Event coreference: deciding whether two event mentions refer to the same event
  - E.g. the buy and sell events in the example below:

AMD agreed to [**buy**] Markham, Ontario-based ATI for around $5.4 billion in cash and stock, the companies announced Monday.
The [**acquisition**] would turn AMD into one of the world's largest providers of graphics chips.

- Discourse Deixis: an anaphor refers back to a discourse segment which is hard to delimit and categorize (not as simple as a basic entity like a person being referred to)
  - E.g., the referent of *that* is a speech act in a. below, a proposition in b., and a manner of description in c.

According to Soleil, Beau just opened a restaurant

a. But *that* turned out to be a lie.
b. But *that* was false.
c. *That* struck me as a funny way to describe the situation.

NON-REFERRING EXPRESSIONS

- Appositives
  - An appositional structure is a noun phrase that describes another (head) noun phrase, e.g.:

    (21.23) Victoria Chen, CFO of Megabucks Banking, saw ...
    (21.24) United, a unit of UAL, matched the fares.

- Predicative and Prenominal NPs
  - Describe properties of the head noun, e.g., *"$2.3 million"* is a property of *"her pay"*, *"the company's president"* is a property of *"the 38-year-old"*:

    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.

NON-REFERRING EXPRESSIONS (CONT.)

- Expletives: uses of pronouns like “it”, e.g.:
  - *It is raining*
  - *It was Emma Goldman who founded Mother Earth*
  - *It surprised me that there was a herring hanging on her wall*
- Generics: expressions that are generic references (but don’t refer to an actual evoked entity in the text), e.g.:
  - *I love mangoes. They are very tasty.*
  - *In July in San Francisco you have to wear a jacket.*

COREFERENCE TASKS

- More formally, we can formulate the coreference resolution task as follows: given a text T, find all entities and the coreference links between them.
  - Model output is compared to human annotations of links between entities (gold coreference annotations on T)
- In the example below, the superscripts indicate the cluster IDs for the coreference clusters/chain, and the letters differentiate the different mentions within a cluster (first mention is a, second mention is b, etc.)

[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.

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

- Coreference resolution approaches typically detect all mentions (referring expressions for entities) in the input and then link them into clusters.

MENTION DETECTION

- The first step in coreference is to find the spans of text that are mentions (mention detection)
- Mention detection algorithms emphasize recall
  - And thus return many candidate mentions that are not actual mentions and require subsequent filtering
  - For example, PoS or NER taggers are used to extract every span that is a NP, possessive pronoun, or named entity, resulting in the candidate mentions in the table below:

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

Victoria Chen	$2.3 million	she
CFO of Megabucks Banking	the 38-year-old	Megabucks
Megabucks Banking	the company	Lotsabucks
her	the company's president	
her pay	It	

  - Some mention detection systems extract all n-grams with n up to 10, and then need to do extensive filtering, e.g., using rules

    - Trying to removing some of the pleonastic (useless) "it" for example can be done using the following rule:

```
It seems/appears/means/follows (that) S
```

MENTION DETECTION (CONT.)

- Rule-based systems do not perform well
- Mention detection can start with all nouns and named entities and then use anaphoricity classifiers or referentiality classifiers to filter out non-mentions
  - E.g., a classifier can be trained using labeled anaphoric referring expressions + use all other NPs or named entities as negative training examples
  - Features used by the classifiers include head word (most important word grammatically), surrounding words, length, etc.
  - Using this kind of classification to perform hard filtering does not lead to good performance since different classifier thresholds may work better for different datasets
- Current systems perform mention detection, anaphoricity, and coreference jointly in a single end-to-end model (to be discussed in a few slides)

Note that mention detection is still an open research problem which limits the performance of coreference resolution systems.

MENTION-PAIR ARCHITECTURE

- The mention-pair architecture is a classic model that helps to introduce the features used by more complex models (though is not the best performing)
- The mention-pair classifier
  - Takes as input two mentions: a candidate anaphor and a candidate antecedent,
  - Outputs a binary class: coreferring or not
  - May rely on hand-built or neural-model based features
- The training set is commonly auto constructed by
  - Taking the closest antecedent (to a mention) as a positive example
  - And all mentions in between them as a negative examples

  I.e., for each anaphor mention m_i, create:
  - 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

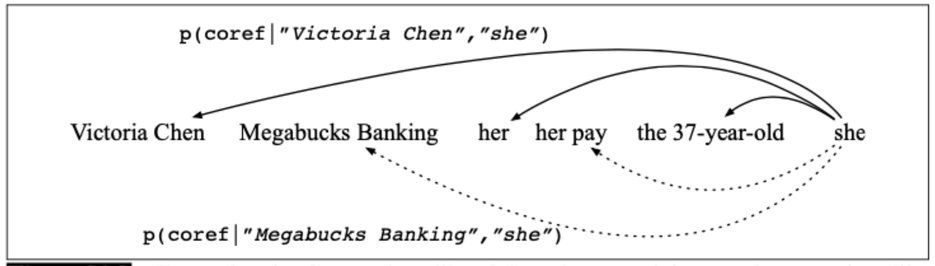

Figure 21.2 For each pair of a mention (like *she*), and a potential antecedent mention (like *Victoria Chen* or *her*), the mention-pair classifier assigns a probability of a coreference link.

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.

- Once the classification is done, all the coreferring expressions (per the binary classification) will be clustered together (different approaches can be used here)

MENTION-PAIR ARCHITECTURE SHORTCOMINGS

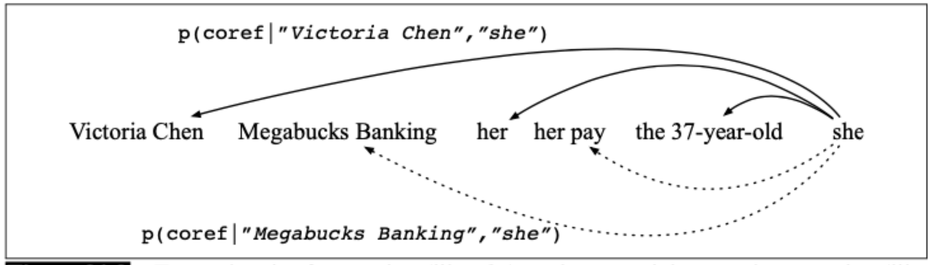

Figure 21.2 For each pair of a mention (like *she*), and a potential antecedent mention (like *Victoria Chen* or *her*), the mention-pair classifier assigns a probability of a coreference link.

- Ignores the discourse model by only looking at mentions and not entities
  - This is addressed by entity-based models which link each mention to a previous discourse entity (an entity is represented by a cluster of mentions) rather than to a previous mention
- The classifier does not directly compare candidate antecedents to each other and thus cannot decide the most likely antecedent
  - This is addressed by the mention-rank architecture (discussed next)

MENTION-RANK ARCHITECTURE

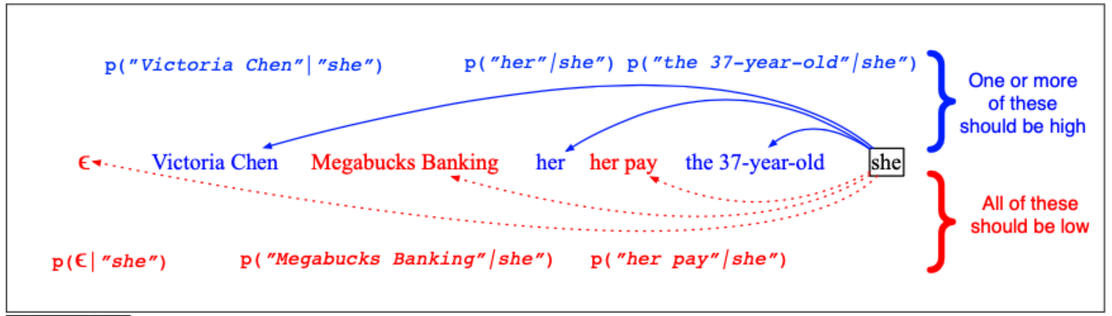

Figure 21.3 For each candidate anaphoric mention (like *she*), the mention-ranking system assigns a probability distribution over all previous mentions plus the special dummy mention ϵ.

- The mention ranking model compares candidate antecedent to each other, and chooses the highest scoring antecedent for each anaphor
- For the i-th mention (anaphor), we have a random variable y_i ranging over the values $Y(i) = \{1,....,i-1,\epsilon\}$
  - ϵ is a dummy variable that allows us to capture the fact that the i-th mention is not an anaphor or does not have an antecedent (e.g., it's a new referent/entity)
- The output of the classifier is a softmax over all the antecendents (and ϵ), which indicates the highest probability antecedent, or that there is none (if ϵ has the highest probability)
- Once the best antecedent for each anaphor is output, a clustering is done by performing transitive closure over the pairwise decisions
- Hand-built features and neural-based feature representations can be used with mention-rank models

HAND-BUILT FEATURES USED FOR CLASSIFICATION

- Most feature-based classifiers make use of three types of features:
  - Features of the anaphor
  - Features of the candidate antecedent
  - Features of the relationship between the pair
- Hand-built features can be used for both logistic regression and other non-neural classifiers, as well as for neural network-based classifiers
- Neural classifiers use contextual embeddings, and can benefit from the addition of hand-built features like mention length, distance between mentions, and genre

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

Features of the Anaphor or Antecedent Mention		
First (last) word	Victoria/she	First or last word (or embedding) of antecedent/anaphor
Head word	Victoria/she	Head word (or head embedding) of antecedent/anaphor
Attributes	Sg-F-A-3-PER/ Sg-F-A-3-PER	The number, gender, animacy, person, named entity type attributes of (antecedent/anaphor)
Length	2/1	length in words of (antecedent/anaphor)
Grammatical role	Sub/Sub	The grammatical role—subject, direct object, indirect object/PP—of (antecedent/anaphor)
Mention type	P/Pr	Type: (P)roper, (D)efinite, (I)ndefinite, (Pr)onoun) of antecedent/anaphor
Features of the Antecedent Entity		
Entity shape	P-Pr-D	The 'shape' or list of types of the mentions in the antecedent entity (cluster), i.e., sequences of (P)roper, (D)efinite, (I)ndefinite, (Pr)onoun.
Entity attributes	Sg-F-A-3-PER	The number, gender, animacy, person, named entity type attributes of the antecedent entity
Ant. cluster size	3	Number of mentions in the antecedent cluster
Features of the Pair of Mentions		
Longer anaphor	F	True if anaphor is longer than antecedent
Pairs of any features	Victoria/she, 2/1, P/Pr, etc.	For each individual feature, pair of type of antecedent+ type of anaphor
Sentence distance	1	The number of sentences between antecedent and anaphor
Mention distance	4	The number of mentions between antecedent and anaphor
i-within-i	F	Anaphor has i-within-i relation with antecedent
Cosine		Cosine between antecedent and anaphor embeddings
Appositive	F	True if the anaphor is in the syntactic apposition relation to the antecedent. Useful even if appositives aren't mentions (to know to attach the appositive to a preceding head)
Features of the Pair of Entities		
Exact String Match	F	True if the strings of any two mentions from the antecedent and anaphor clusters are identical.
Head Word Match	F	True if any mentions from antecedent cluster has same headword as any mention in anaphor cluster
Word Inclusion	F	All words in anaphor cluster included in antecedent cluster
Features of the Document		
Genre/source	N	The document genre— (D)ialog, (N)ews, etc,

Figure 21.4 Feature-based coreference: sample feature values for anaphor "she" and potential antecedent "Victoria Chen".

NEURAL MENTION-RANKING: ALGORITHM OVERVIEW

Below is a modified version of the e2e-coref algorithm, where given a document D with T words:

- The model considers all text spans in D (in practice n-grams with n up to 10)
  and assigns an antecedent y_i to each span i from among the possible antecedents and the dummy ϵ
  - If the dummy has the highest probability output by the classifier, then i does not have an antecedent (e.g., it is new or non-anaphoric)
- To compute $P(y_i)$ for a span i, the algorithms assigns a score $s(i,j)$ for each pair of spans i and j that captures the "strength" of their coreference:

$$P(y_i) = \frac{\exp(s(i,y_i))}{\sum_{y' \in Y(i)} \exp(s(i,y_i))}$$

$s(i,j) = m(i) + m(j) + c(i,j)$, where factor $m(i)$ indicates whether span i is a mention (and same for $m(j)$), and $c(i,j)$ indicates whether j is the antecedent of i

Note that $s(i,\epsilon) = 0$, and if any non-dummy scores are positive, the model predicts the highest scoring antecedent, and when all are negative, it doesn't make a prediction.

COMPUTING SPAN REPRESENTATIONS FOR NEURAL COREFERENCE RESOLUTION

- Span can be represented as a concatenation of the representation of 3 tokens: the first and last word of the span, and an attention vector
  - The attention vector is used to identify likely head word in the span
- The token representations are embeddings computed by BERT, where span i is represented by the following vector g_i:

$$\mathbf{g}_i = [\mathbf{h}_{START_{(i)}}, \mathbf{h}_{END_{(i)}}, \mathbf{h}_{ATT_{(i)}}]$$

The likely syntacticly most important word (the head word) is represented as an attention vector, computed as follows:

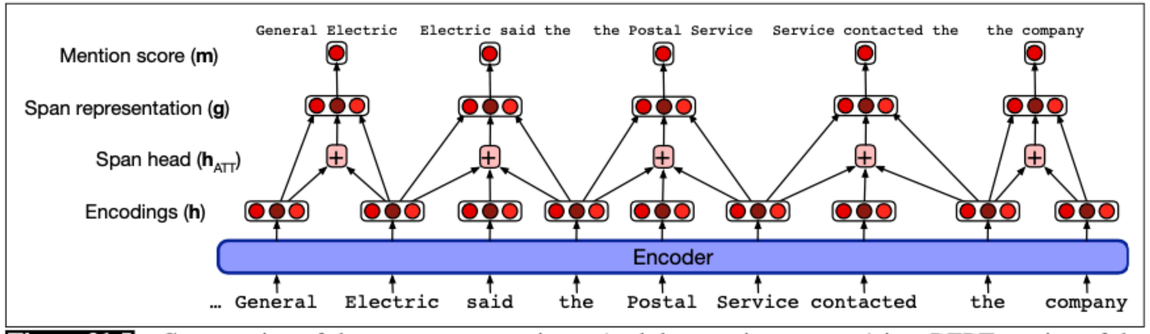

Figure 21.5 Computation of the span representation **g** (and the mention score **m**) in a BERT version of the e2e-coref model (Lee et al. 2017b, Joshi et al. 2019). The model considers all spans up to a maximum width of say 10; the figure shows a small subset of the bigram and trigram spans.

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

Attention is computed by learning a weight vector w_α and computing the dot product between w_α and the hidden state h_t transformed through a feed forward neural network

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

The attention score is normalized into a probability distribution through softmax

$$\mathbf{h}_{\text{ATT}(i)} = \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot \mathbf{w}_t$$

The attention distribution is used to create the attention vector (w_t are the words in the span)

COMPUTING THE MENTION AND ANTECEDENT SCORES FOR NEURAL MODELS

Recall that our goal is to compute a score in order to find the highest probability antecedent: $s(i,j) = m(i) + m(j) + c(i,j)$, where factor $m(i)$ indicates whether span i is a mention (and same for $m(j)$), and $c(i,j)$ indicates whether j is the antecedent of i

- Given $\mathbf{g}_i = [\mathbf{h}_{START_{(i)}}, \mathbf{h}_{END_{(i)}}, \mathbf{h}_{ATT_{(i)}}]$, we now compute $m(i)$ and $c(i,j)$ using FFNNs as follows:

$$m(i) = w_m \cdot \text{FFNN}_m(\mathbf{g}_i)$$
$$c(i,j) = w_c \cdot \text{FFNN}_c([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j,])$$

$\mathbf{g}_i$ and $\mathbf{g}_j$ are the representations of the two spans under consideration

$\mathbf{g}_i \circ \mathbf{g}_j$ is an element-wise multiplication of the span vectors which indicates their element-wise similarity

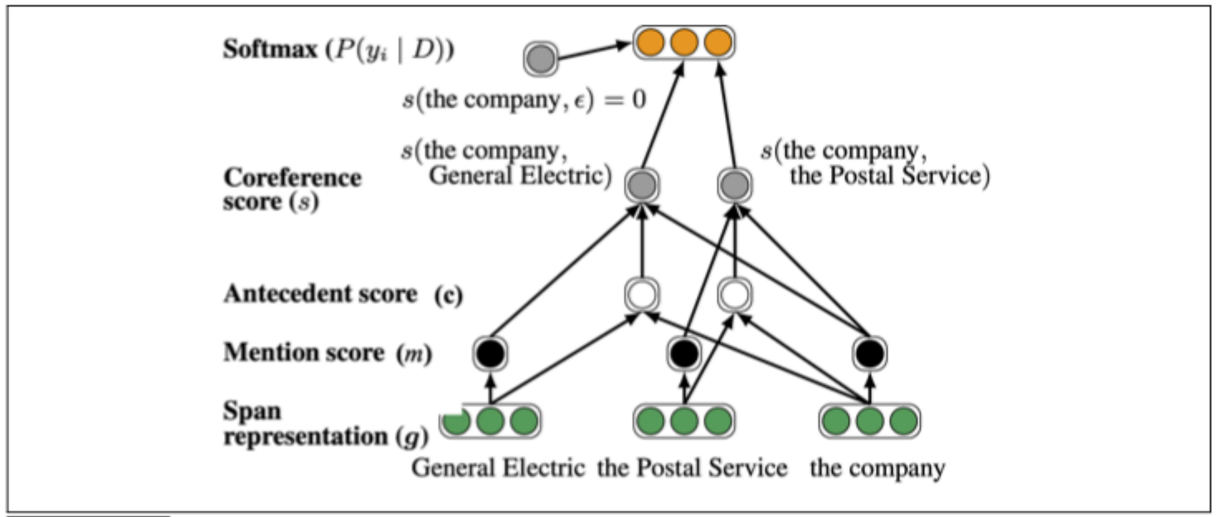

Figure 21.6 The computation of the score s for the three possible antecedents of *the company* in the example sentence from Fig. 21.5. Figure after Lee et al. (2017b).

- We compute transitive closure on the antecedents to create a final clustering of the referring expressions

NEURAL MODEL CHALLENGES: EXAMPLE

- Neural models make mistakes when there is a similarity of embeddings between words (assuming they corefer)

We are looking for **(a region of central Italy bordering the Adriatic Sea)**. **(The area)** is mostly mountainous and includes Mt. Corno, the highest peak of the Apennines. **(It)** also includes a lot of sheep, good clean-living, healthy sheep, and an Italian entrepreneur has an idea about how to make a little money of them.

(The flight attendants) have until 6:00 today to ratify labor concessions. **(The pilots')** union and ground crew did so yesterday.

Figure 21.7 Sample predictions from the Lee et al. (2017b) model, with one cluster per example, showing one correct example and one mistake. Bold, parenthesized spans are mentions in the predicted cluster. The amount of red color on a word indicates the head-finding attention weight $a_{i,t}$ in (21.52). Figure adapted from Lee et al. (2017b).

EVALUATING THE PERFORMANCE OF COREFERENCE RESOLUTION ALGORITHMS

- We compare a set of clusters H (also referred to as hypothesis chains) against a set of gold standard/reference clusters R
- Some example metrics are shown below but often multiple metrics are used and combined (e.g., averaged)
- MUC F-measure: counts the number of coreference links (pairs of mentions) common to H and R
  - Precision: number of common links divided by the number of links in H
  - Recall: the number of common links divided by the number of links in R
  - Shortcomings: biased toward systems that produce long chains, and ignores singletons (as there is no chain)
- B^3 focuses on mentions rather than links and uses a weight w_i for each entity; the weight that can be modified, resulting in different versions of the algorithm:

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

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

WINOGRAD SCHEMA PROBLEMS

- Introduced by Winograd (1972)
- Difficult examples that require knowledge of the world - grammar does not provide clues as to the preferred antecedent

The city council denied the demonstrators a permit because

a. they feared violence.
b. they advocated violence.

- Levesque (2012) proposed a challenges task: the Winograd Schema Challenge to encourage focus on common-sense reasoning
- The Winograd Schema Challenge contains coreference problems that are easy for humans, but not solvable through brute force/naive approaches:

1. The problems each have two parties
2. A pronoun preferentially refers to one of the parties, but could grammatically also refer to the other
3. A question asks which party the pronoun refers to
4. If one word in the question is changed, the human-preferred answer changes to the other party

However, pre-trained language models have been able to handle the challenge even though they are not capable of common sense reasoning, so researchers are designing new challenge datasets.

GENDER BIAS IN COREFERENCE ALGORITHMS

- Gender bias has been shown to be present across all types of both classical and neural-based approaches, e.g.,
  - Stereotypes about professions
  - Stereotypes about violence

The secretary called the physician$_i$ and told him$_i$ about a new patient [pro-stereotypical]

The secretary called the physician$_i$ and told her$_i$ about a new patient [anti-stereotypical]

Simple approach that has been shown to work well:

- Create a second version of each dataset which has the male and female pronouns and entities swapped, and train algorithms on both the original and modified version of the dataset