Lec 13. Coreference(동일지시어) Resolution

1. What is Coreference Resolution?

: Identify all mentions that refer to the same entity in the world
eg. Vanaja -> her, she, herself, she, she, she
Akhila -> Akila's/ Prajwal-> Akila's son/ Akash -> a tree?

: split antecedence

2. Applications of coreference resolution

: full text undertanding
: Machine translation -> languages have different feautures for gender, number, dropped pronoun ..
: Dialogue Systems

: Two steps -> 1. Detect the mentions(can be nested) 2. Cluster the mentions

3. Mention Detection

: Mention: A span of text referring to some entity
: 3 kinds of mentions -> 1. Pronouns 2. Named entities 3. Noun phrases

: 1. Pronoun: Use a part-of-speech-tagger
: 2. Named entities: Use a Named Entity Recognigion system
: 3. Noun phrases: Use a parser(especially a constituency parser)

: Not so simple -> Marking all pronouns, named entities, and NPs as mentions over-generates mentions

: How to deal with these bad mentions? -> training a classifier to filter out spurious mentions, keep all mentions as "candidate mentions".

: Avoding a traditional pipeline system -> we could instead train a classifier specifically for mention detection instead of using a POS tagger, NER system, and parser.

: Or we can not even try to do mention detection explicitly: we can build a model that begins with all pans and jointly does mention-detection and coreference resolution end-to-end in one model

4. Some Linguistics: Types of Reference

: Coreference is when two mentions refer to the same entity in the world.
: A different-but-related linguistic concept is anaphora: when a term refers to another term(antecedent) -> the interpretation of the anaphor is some way determined by the interpretation of the antecedent

: Anaphor vs. Coreference -> Barack Obama = Obama vs. he

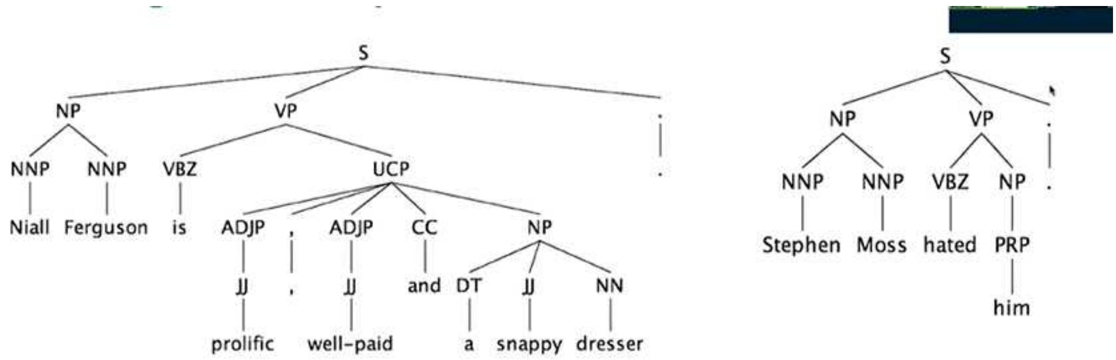
: Not all anaphoric relations are coreferential
: Not all noun phrases have reference
: bridging anaphora
: Usually the antecedent comes before the anaphor, but not always.
: As we progress through an article, or dialogue, or webpage, we build up a discourse model, and we interpret new sentence/utterances with respect to our model of what's come before.

:Four kinds of coreference models -> rule-based/Mention pair/Mention Ranking/Clustering

5. Rule-Based
: Traditional pronominal anaphora resolution -> Hobb's naive algorithm

1. Begin at the NP immediately dominating the pronoun
2. Go up tree to first NP or S. Call this X, and the path p.
3. Traverse all branches below X to the left of p, left-to-right, breadth-first. Propose as antecedent any NP that has a NP or S between it and X
4. If X is the highest S in the sentence, traverse the parse trees of the previous sentences in the order of recency. Traverse each tree left-to-right, breadth first. When an NP is encountered, propose as antecedent. If X not the highest node, go to step 5.
5. From node X, go up the tree to the first NP or S. Call it X, and the path p.
6. If X is an NP and the path p to X came from a non-head phrase of X (a specifier or adjunct, such as a possessive, PP, apposition, or relative clause), propose X as antecedent

   (The original said "did not pass through the N' that X immediately dominates", but the Penn Treebank grammar lacks N' nodes....)
7. Traverse all branches below X to the left of the path, in a left-to-right, breadth first manner. Propose any NP encountered as the antecedent
8. If X is an S node, traverse all branches of X to the right of the path but do not go below any NP or S encountered. Propose any NP as the antecedent.
9. Go to step 4



:Knowledge-based Pronominal Coreference

- She poured water from the pitcher into the cup until it was full.
- She poured water from the pitcher into the cup until it was empty.

- The city council refused the women a permit because they feared violence.
- The city council refused the women a permit because they advocated violence.
  - Winograd (1972)

- These are called **Winograd Schema**
  - Recently proposed as an alternative to the Turing test

6. Mention-pair and mention-ranking models

: Mention-pair -> train a binary classsifier that assigns every pair of mentions a probability of being coreferent

- N mentions in a document
- y_{ij} = 1 if mentions m_i and m_j are coreferent, -1 if otherwise
- Just train with regular cross-entropy loss (looks a bit different because it is binary classification)

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

Iterate through mentions

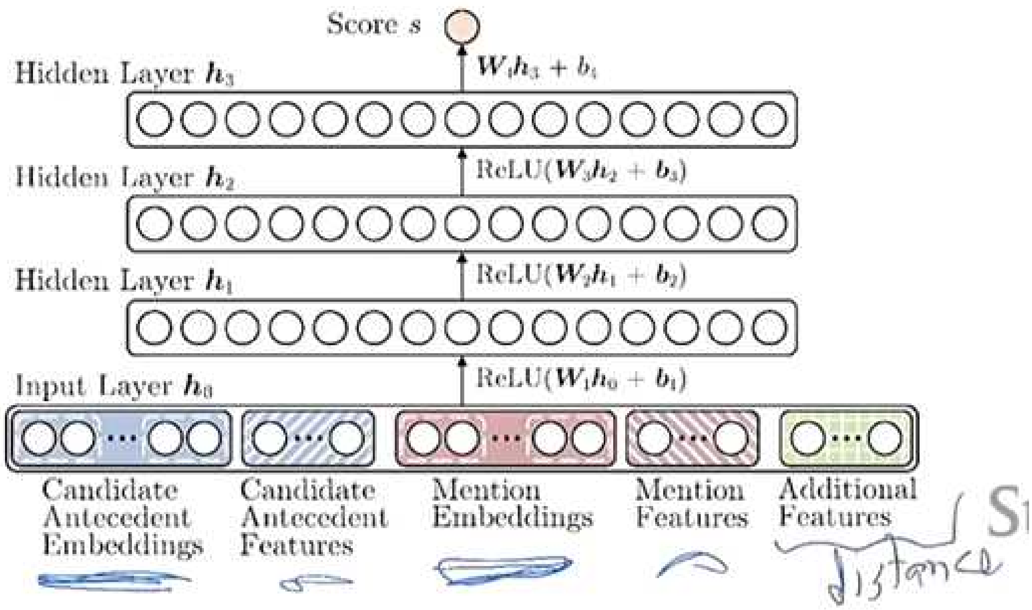
Iterate through candidate antecedents (previously occurring mentions)

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

: Mention Ranking-> assign each mention its highest scoring candidate antecedent according to the model.
: Dummy NA mention allows model to decline linking the current mention to anything.

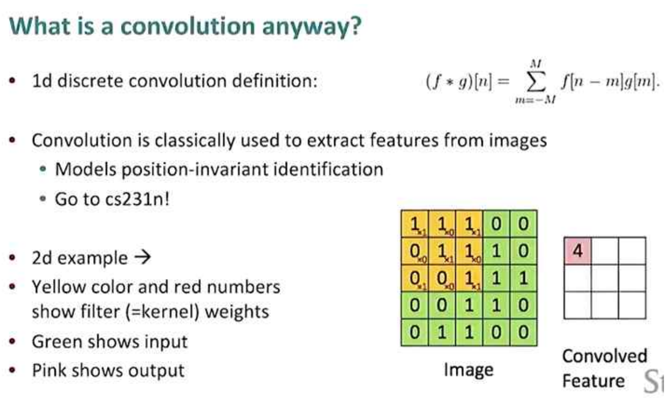
->
: How do we compute the probabilities?
-> Neural Coref Model -> standard feed-forward neural network

- Input layer: word embeddings and a few categorical features



-> Embeddings: previous

7. Interlude: ConvNets for language
: Main CNN/ConvNet idea: what if we compute vectors for every possible word subsequence of a certain length?



-> regardless of whether phrases is grammatical

A 1D convolution for text

tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3

t,d,r	−1.0	0.0	0.50
d,r,t	−0.5	0.5	0.38
r,t,k	−3.6	-2.6	0.93
t,k,g	−0.2	0.8	0.31
k,g,o	0.3	1.3	0.21

Apply a **filter** (or **kernel**) of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

+ bias

➔ non-linearity

1D convolution for text with padding

∅	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

∅,t,d	−0.6
t,d,r	−1.0
d,r,t	−0.5
r,t,k	−3.6
t,k,g	−0.2
k,g,o	0.3
g,o,∅	−0.5

Apply a **filter** (or **kernel**) of size 3

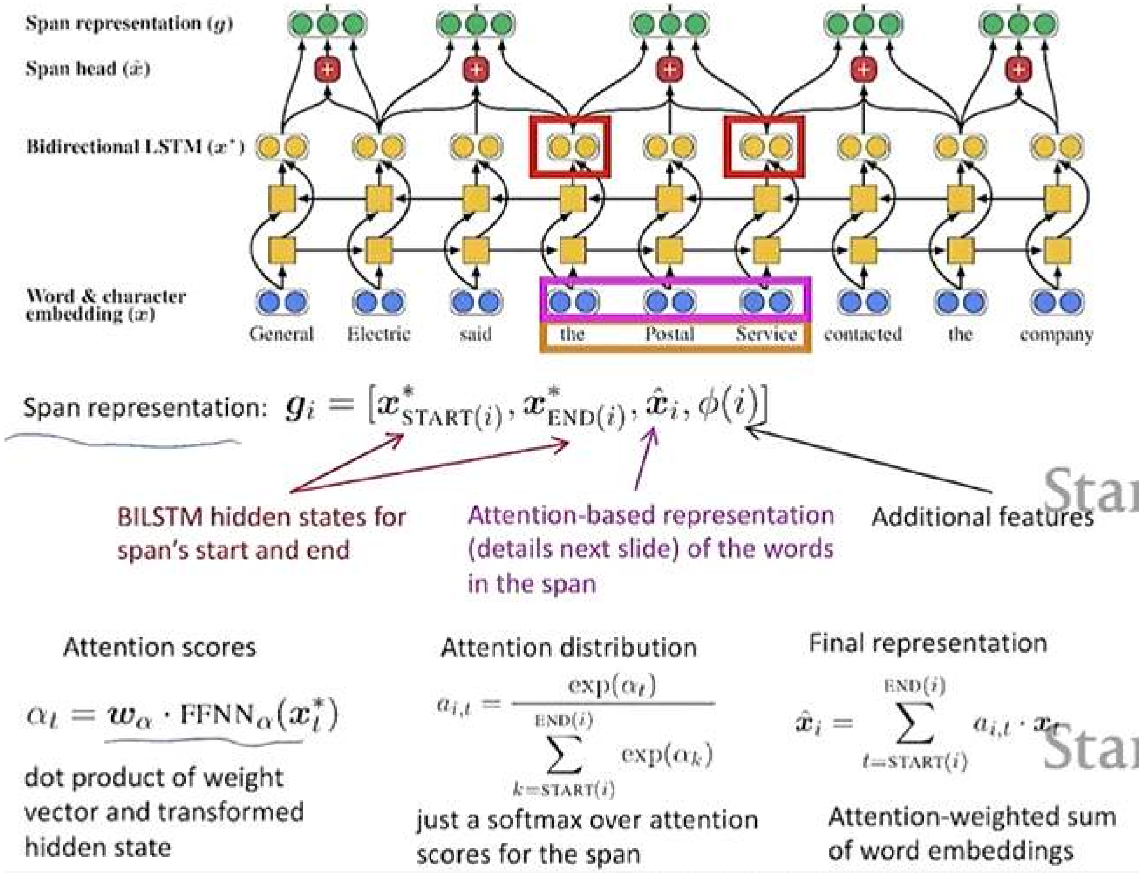
3	1	2	−3
−1	2	1	−3
1	1	−1	1

8. Current state-of-the-the-art neural coreference systems

: Mention ranking model, improvements over simple feed-forward NN

-> Use an LSTM, attention, do mention detection and coreference end-to-end

: End-to-end Model



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