Discourse

COMP90042

Natural Language Processing

Lecture 12

THE UNIVERSITY OF MELBOURNE

Discourse

- Most tasks/models we learned operate at word or sentence level:
  - POS tagging
  - Language models
  - Lexical/distributional semantics
- But NLP often deals with documents
- **Discourse**: understanding how sentences relate to each other in a document

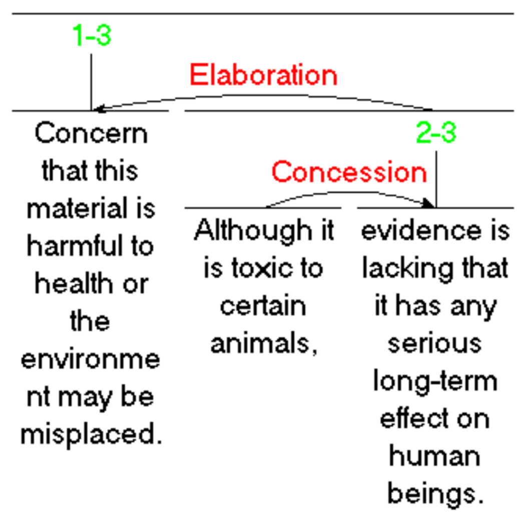
Three Key Discourse Tasks

- Discourse segmentation

partition.
create cuts

- Discourse parsing

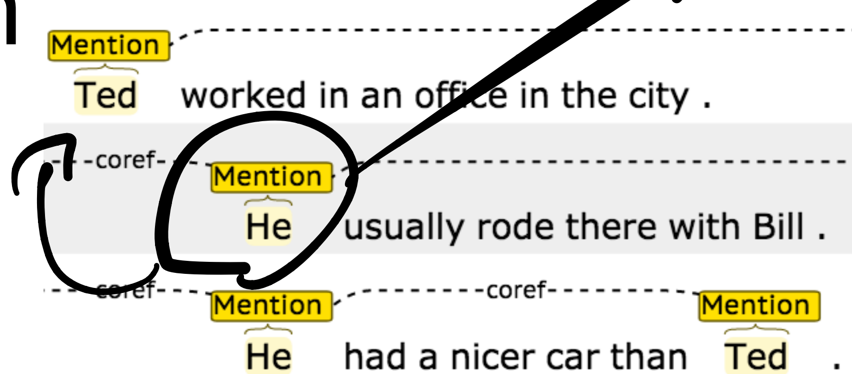
hierarchical structure
(tree)



- Anaphora resolution

resolve ambiguity.

resolve "he"



Discourse Segmentation

Discourse Segmentation

- A document can be viewed as a sequence of segments
- A segment: a span of cohesive text
- Cohesion:
  - organised around a particular **topic** or **function**
    - Wikipedia biographies: early years, major events, impact on others
    - Scientific articles: introduction, related work, experiments

Unsupervised Approaches

gap between sentences

- TextTiling algorithm: looking for points of low lexical cohesion between sentences
- For each sentence gap:

  k sentences before the gap

  - Create two BOW vectors consisting of words from *k* sentences on either side of gap

    after the gap.

  - Use cosine to get a similarity score (*sim*) for two vectors
  - For gap *i*, calculate a depth score, insert boundaries when depth is greater than some threshold *t*

$$depth(gap_i) = (sim_{i-1} - sim_i) + (sim_{i+1} - sim_i)$$

either side of gap.

Text Tiling Example (k=1, t=0.9)

depth score · # ~~words~~ sentences before, after threshold.

He walked 15 minutes to the tram stop.

d=0.7-0.9=-0.2 sim: 0.9

~~(0.9-0.7)~~+(0.7-0.9))=-0.2

Then he waited for another 20 minutes, but the tram didn't come.

~~d=(0.9-0.7)+(0.1-0.7)=-0.4~~ sim: 0.7

The tram drivers were on strike that morning.

d=(0.7-0.1)+(0.5-0.1)=1.0 sim: 0.1

So he walked home and got his bike out of the garage.

d=(0.1-0.5)+(0.8-0.5)=-0.1 sim: 0.5

He started riding but quickly discovered he had a flat tire

d=(0.1-0.5)+(0.8-0.5)=-0.6 sim: 0.8

He walked his bike back home.

d=0.8-0.5=0.3 sim: 0.5

He looked around but his wife had cleaned the garage and he couldn't find the bike pump.

depth really deep $\Rightarrow d > t$.

$$depth(gap_i) = (sim_{i-1} - sim_i) + (sim_{i+1} - sim_i)$$

Supervised Approaches

- Get labelled data from easy sources
  - Scientific publications
  - Wikipedia articles

boundaries of paragraph.

Abstract

Language models are typically applied at the sentence level, without access to the broader document context. We present a neural language model that incorporates document context in the form of a topic model-like architecture, thus providing a succinct representation of the broader document context outside of the current sentence. Experiments over a range of datasets demonstrate that our model outperforms a pure sentence-based model in terms of language model perplexity, and leads to topics that are potentially more coherent than those produced by a standard LDA topic model. Our model also has the ability to generate related sentences for a topic, providing another way to interpret topics.

possible cut.

1 Introduction

Topic models provide a powerful tool for extracting the macro-level content structure of a document collection in the form of the latent topics (usually in the form of multinomial distributions over terms), with a plethora of applications in NLP (Hall et al., 2008; Newman et al., 2010a; Wang and McCallum, 2006). A myriad of variants of the classical LDA method (Blei et al., 2003) have been proposed, including recent work on neural topic models (Cao et al., 2015; Wan et al., 2012; Larochelle and Lauly, 2012; Hinton and Salakhutdinov, 2009).

Separately, language models have long been a foundational component of any NLP task involving generation or textual normalisation of a noisy input (including speech, OCR and the processing of social media text). The primary purpose of a language model is to predict the probability of a span of text, traditionally at the sentence level, under the assumption that sentences are independent of one another, although recent work has started using broader local context such as the preceding sentences (Wang and Cho, 2016; Ji et al., 2016).

no jump in section.

In this paper, we combine the benefits of a topic model and language model in proposing a topically-driven language model, whereby we jointly learn topics and word sequence information. This allows us to both sensitise the predictions of the language model to the larger document narrative using topics, and to generate topics which are better sensitised to local context and are hence more coherent and interpretable.

Our model has two components: a language model and a topic model. We implement both components using neural networks, and train them jointly by treating each component as a sub-task in a multi-task learning setting. We show that our model is superior to other language models that leverage additional context, and that the generated topics are potentially more coherent than LDA topics. The architecture of the model provides an extra dimensionality of topic interpretability, in supporting the generation of sentences from a topic (or mix of topics). It is also highly flexible, in its ability to be supervised and incorporate side information, which we show to further improve language model performance. An open source implementation of our model is available at: https://github.com/jhlau/topically-driven-language-model.

2 Related Work

Griffiths et al. (2004) propose a model that learns topics and word dependencies using a Bayesian framework. Word generation is driven by either LDA or an HMM. For LDA, a word is generated based on a sampled topic in the document. For the

Supervised Discourse Segmenter

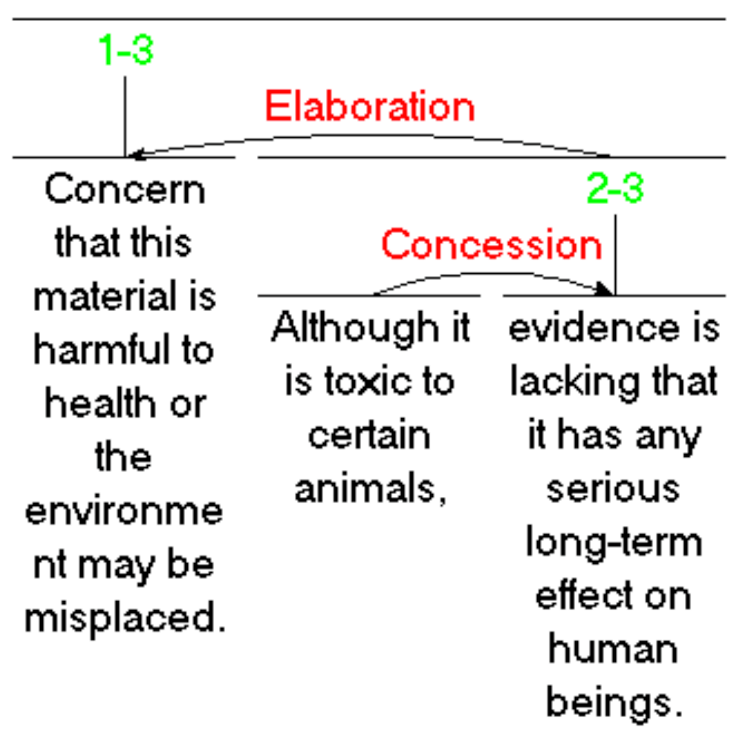
- Apply a binary classifier to identify boundaries
- Or use sequential classifiers
- Potentially include classification of section types (introduction, conclusion, etc.)
- Integrate a wider range of features, including
  - distributional semantics
  - discourse markers (*therefore*, *and*, etc)

Discourse Parsing

Discourse Parsing

- Identify **discourse units**, and the **relations** that hold between them
- **Rhetorical Structure Theory (RST)**, is a framework to do hierarchical analysis of discourse structure in documents

RST tries to understand relations



RST

- Basic element: **elementary discourse units (EDUs)**
  - Typically clauses of a sentence
  - EDUs do not cross sentence boundary
  - *[It does have beautiful scenery,] [some of the best since Lord of the Rings.]*
- **RST relations** between discourse units:
  - *conjuction*, *justify*, *concession*, *elaboration*, etc
  - *[It does have beautiful scenery,]*
    ↑ *(elaboration)*
    [some of the best since Lord of the Rings.]

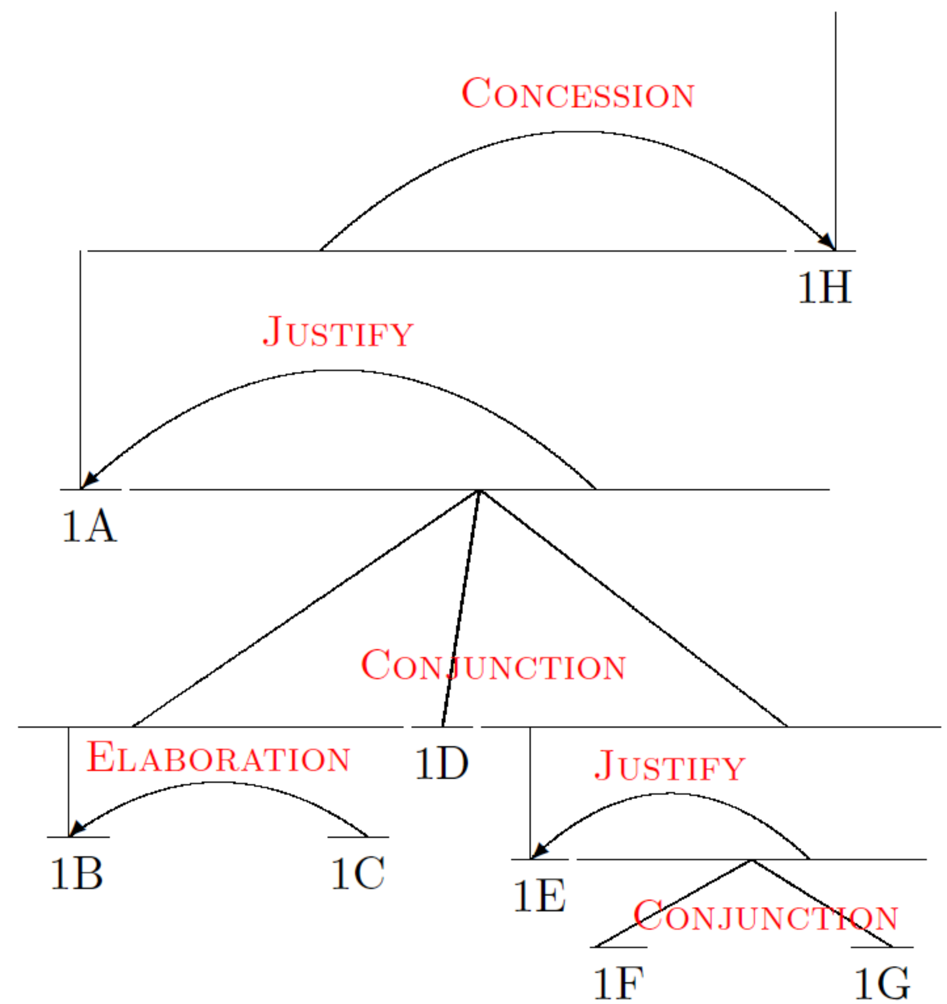
Nucleus vs. Satellite

- Within a discourse relation, one argument is the **nucleus** (the primary argument)
- The supporting argument is the **satellite**
  - *[It does have beautiful scenery,]$_{nucleus}$*
    ↑ *(elaboration)*
    [some of the best since Lord of the Rings.]$_{satellite}$
- Some relations are equal (e.g. conjunction), and so both arguments are nuclei
  - *[He was a likable chap,]$_{nucleus}$*
    ↑ *(conjunction)*
    [and I hated to see him die.]$_{nucleus}$

equal

RST Tree

- An RST relation combines two or more DUs into composite DUs
- Process of combining DUs is repeated to create an RST tree



[It could have been a great movie][1A] [It does have beautiful scenery,][1B] [some of the best since Lord of the Rings.][1C] [The acting is well done,][1D] [and I really liked the son of the leader of the Samurai.][1E] [He was a likable chap,][1F] [and I hated to see him die.][1G] [But, other than all that, this movie is nothing more than hidden rip-offs.][1H]

Merge EDUs to get Tree

Parsing Using Discourse Markers

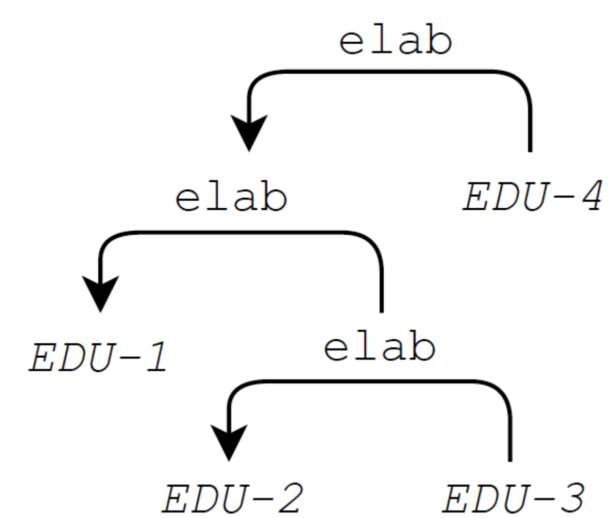
- Some discourse markers (cue phrases) explicitly indicate relations
  - Some examples: *although*, *but*, *for example*, *in other words*, *so*, *because, in conclusion*,…
- Can be used to build a simple rule-based parser
- However
  - Many relations are not marked by discourse marker at all
  - Many important discourse markers (e.g. *and*) ambiguous
    - Sometimes not a discourse marker
    - Can signal multiple relations

reason or justification

not marking discourse sometimes.

Parsing Using Machine Learning

- RST Discourse Treebank
  - 300+ documents annotated with RST trees
- Basic idea:
  - Segment document into EDUs
  - Combine adjacent DUs into composite DUs iteratively to create the full RST tree



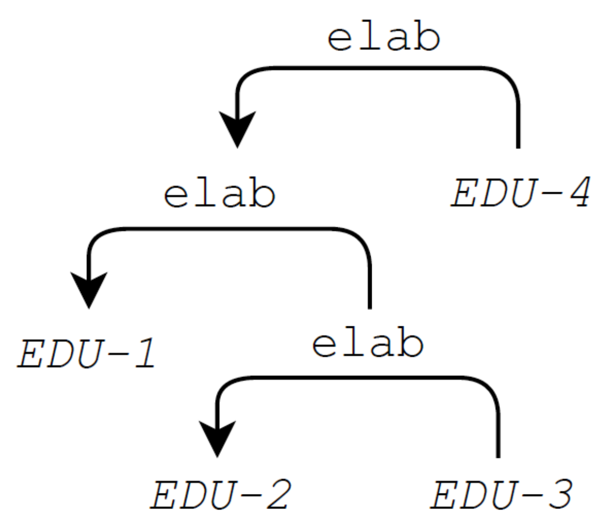
EDU-1: Roy E. Parrott, the company's president and chief operating officer since Sept. 1, was named to its board.
EDU-2: The appointment increased the number of directors to 10,
EDU-3: three of whom are company employees.
EDU-4: Simpson is an auto parts maker.

Parsing Using Machine Learning

- Transition-based parsing (lecture 16):
  - Bottom-up
  - Greedy, uses shift-reduce algorithm
- CYK/chart parsing algorithm (lecture 14)
  - Bottom-up
  - Global, but some constraints prevent CYK from finding globally optimal tree for discourse parsing

Parsing Using Machine Learning

- Top-down parsing
  - Sequence labelling problem
  - BERT



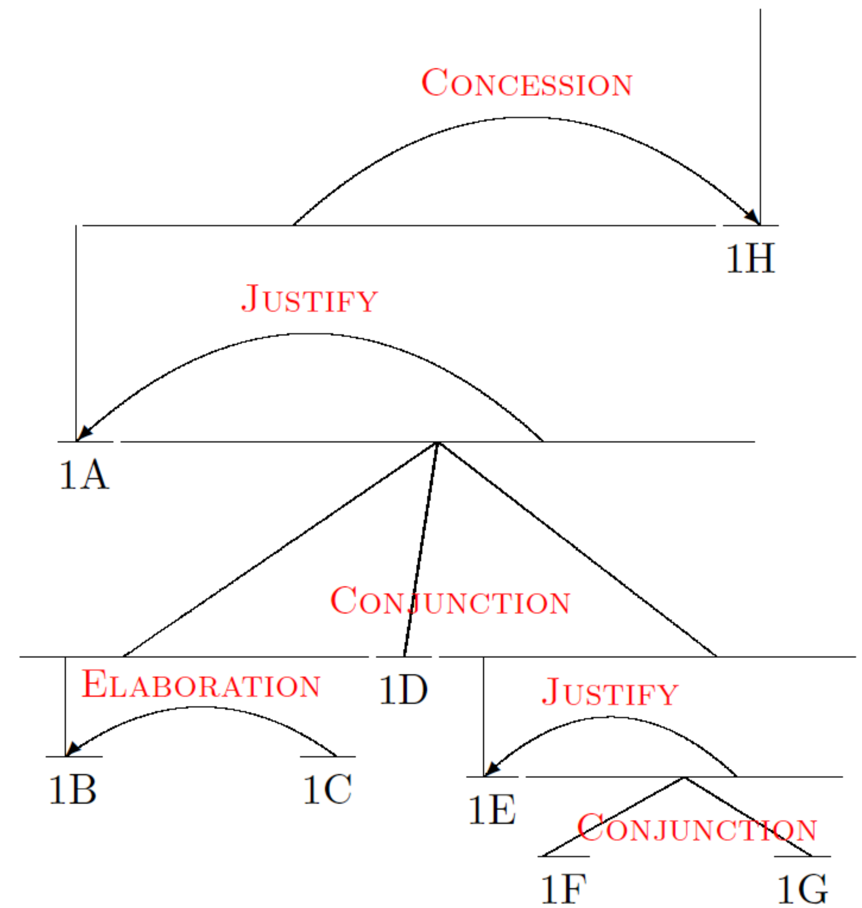
EDU-1: Roy E. Parrott, the company's president and chief operating officer since Sept. 1, was named to its board.
EDU-2: The appointment increased the number of directors to 10,
EDU-3: three of whom are company employees.
EDU-4: Simpson is an auto parts maker.

Discourse Parsing Features

- Bag of words
- Discourse markers
- Starting/ending *n*-grams
- Location in the text
- Syntax features
- Lexical and distributional similarities

Why Discourse Parsing?

- Summarisation
- Sentiment analysis
- Argumentation
- Authorship attribution
- Essay scoring



[It could have been a great movie]^{1A} [It does have beautiful scenery,]^{1B} [some of the best since Lord of the Rings.]^{1C} [The acting is well done,]^{1D} [and I really liked the son of the leader of the Samurai.]^{1E} [He was a likable chap,]^{1F} [and I hated to see him die.]^{1G} [But, other than all that, this movie is nothing more than hidden rip-offs.]^{1H}

Anaphora Resolution

Anaphors

- **Anaphor**: linguistic expressions that refer back to earlier elements in the text
- Anaphors have a **antecedent** in the discourse, often but not always a noun phrase
  - *Yesterday, Ted was late for work. It all started when his car wouldn't start.*
- Pronouns are the most common anaphor
- But there are various others
  - Demonstratives (*that problem*)

Antecedent Restrictions

- Pronouns must agree in *number* with their antecedents
  - *His coworkers were leaving for lunch when Ted arrived. They invited him, but he said no.*
- Pronouns must agree in *gender* with their antecedents
  - *Sue was leaving for lunch when Ted arrived. She invited him, but he said no.*
- Pronouns whose antecedents are the subject of the same syntactic clause must be *reflexive* *(…self)*
  - *Ted was angry at him.* [*him* ≠ *Ted*]
  - *Ted was angry at himself.* [*himself* = *Ted*]

Antecedent Preferences

- The antecedents of pronouns should be recent
  - *He waited for another 20 minutes, but the tram didn't come. So he walked home and got his bike out of the garage. He started riding it to work.*
- The antecedent should be salient, as determined by grammatical position

  major.
  - Subject > object > argument of preposition
  - *Ted usually rode to work with Bill. He was never late.*

Entities and Reference

(16.1)
a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day

(16.2)
a. John went to his favorite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

- Discourse 16.1 (left) more coherent
- Pronouns all refer to John consistently, the protagonist

Centering Theory

- A unified account of relationship between discourse structure and entity reference
- Every utterance in the discourse is characterised by a set of entities, known as **centers**
- Explains preference of certain entities for ambiguous pronouns

For an Utterance U_n

- **Forward-looking centers**:
  - All entities in U_n: $C_f(U_n) = [e_1, e_2, \ldots]$
  - C_f(16.1a) = [John, music store, piano]
  - Ordered by syntactic prominence: subjects > objects
- **Backward-looking center**:
  - Highest ranked entity in previous utterance's ($C_f(U_{n-1})$) forward-looking centers that is also in current utterance (U_n)
  - C_b(16.b) = [John]

(16.1) a. John went to his favorite music store to buy a piano.

b. He had frequented the store for many years.

Centering Algorithm

- When resolving entity for anaphora resolution, choose the entity such that the **top foward-looking center** matches with the **backward-looking center**

(16.1)
a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day

(16.2)
a. John went to his favorite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

The Centering Algorithm

1. John saw a Ford in the dealership

$C_f(U_1)$ = [John, Ford, dealership]

$C_b(U_1)$ = None

2. He showed it to Bob

$C_f(U_2)$ = [John, Ford, Bob]

$C_b(U_2)$ = John

top rank in previous utterance.

3. <u>He</u> bought it

If he = John:

$C_f(U_3)$ = [John, Ford]

$C_b(U_3)$ = John

If he = Bob:

$C_f(U_3)$ = [Bob, Ford]

$C_b(U_3)$ = Ford

Bob ≠ Ford.

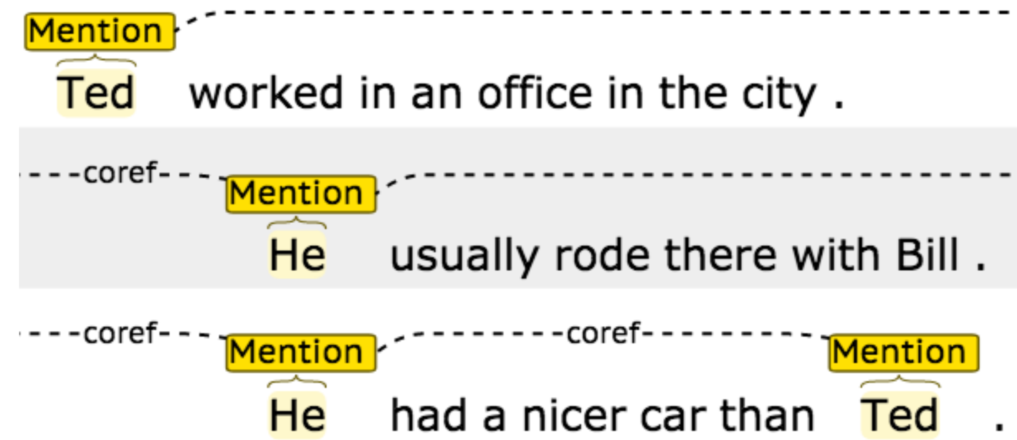
top forward-looking center = backward-looking center

Supervised Anaphor Resolution

- Build a binary classifier for anaphor/antecedent pairs
- Convert restrictions and preferences into features
  - Binary features for number/gender compatibility
  - Position of antecedent in text
  - Include features about type of antecedent
- With enough data, can approximate the centering algorithm
- But also easy to include features which indicate tendencies, rather than rules
  - Like repetition, parallelism

Anaphora Resolution Tools

- Stanford CoreNLP includes pronoun coreference models
  - rule-based system does very well
  - considerably faster than learned models



SYSTEM	LANGUAGE	PREPROCESSING TIME	COREF TIME	TOTAL TIME	F1 SCORE
Deterministic	English	3.87s	0.11s	3.98s	49.5
Statistical	English	0.48s	1.23s	1.71s	56.2
Neural	English	3.22s	4.96s	8.18s	60.0
Deterministic	Chinese	0.39s	0.16s	0.55s	47.5
Neural	Chinese	0.42s	7.02s	7.44s	53.9

Source: https://stanfordnlp.github.io/CoreNLP/coref.html
Evaluated on CoNLL 2012 task.

Motivation for Anaphor Resolution

- Essential for deep semantic analysis
  - Very useful for QA, e.g., reading comprehension

Ted's car broke down. So he went over to Bill's house to borrow his car. Bill said that was fine.

Whose car is borrowed?

A Final Word

- For many tasks, it is important to consider context larger than sentences
- Traditionally many popular NLP applications has been sentence-focused (e.g. machine translation), but that is beginning to change…

Further Reading

- E18, Ch 16