COMP90042 Web Search & Text Analysis

Workshop Week 11

Zenan Zhai

May 21, 2019

University of Melbourne

Roadmap

Features

1. Word Semantics
   - Lexicon semantics
   - Distributional semantics
2. Sequence Labeling
   - Part-of-speech tagging
   - Named entity recognition
3. Parsing
   - Dependency parsing
   - Phrase-structure parsing

Roadmap

Applications

1. Text classification
2. Question answering
3. Discourse tasks
4. Machine translation
5. Summarization

   . . .

Outlines

Dependency parsing

- Dependency grammar
- Projectivity
- Parsing
  - Transition-based
  - Graph-based

Discourse

- Discourse segmentation
- Discourse parsing
- Anaphor resolution

Dependency grammar

For each word we have:

- A head word which this word depends on.
- A dependency label of the connection.

Phrase-structure parsing

- Elements: words at leaves, otherwise phrases
- Link: CFGs, no labels
- Results: Constituent tree

Dependency parsing

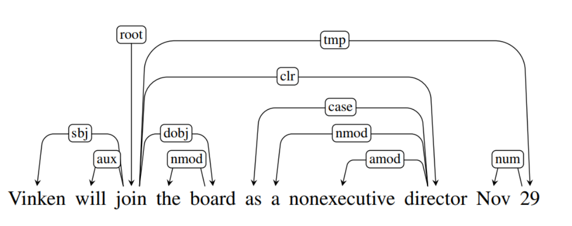
- Elements: pair of words
- Link: dependencies with labels
- Result: Dependency tree

All use part-of-speech tags as ”features”.

Projectivity

Condition:

- A tree is projective if, for **all arcs** from head to dependent, there **is a path from the head to every word that lies between the head and the dependent**


root
tmp
clr
case
sbj
dobj
nmod
aux
nmod
amod
num
Vinken will join the board as a nonexecutive director Nov 29

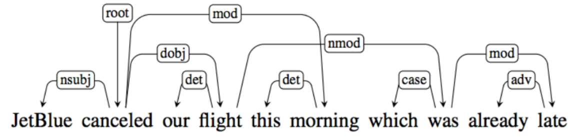

root
mod
nmod
dobj
mod
nsubj
det
det
case
adv
JetBlue canceled our flight this morning which was already late

Figure JM3, Ch 13

Transition-based parsing

2 data structures:

- input buffer: words to process
- stack: words being processed currently

transitions:

- shift: add new word from buffer to stack
- arc: left or right, combining **2 words on the top** of the stack and remove dependent.

Exercise:

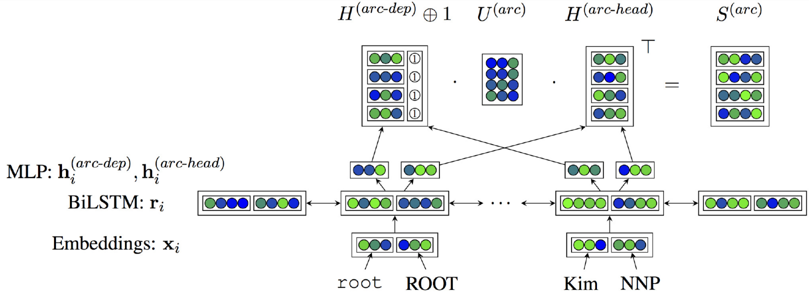
- *Yesterday, I shot an elephant in my pyjamas.*

Graph-based parsing

CYK algorithm

- Recall CYK for pharse-structure parsing runs in $O(N^3G)$ where G is size of CFGs.
- For dependency parsing, $|G| = N^2$. (Pair of words, arrow can be left or right.)
- Trivial CYK runs in $O(N^5)$.

Neural methods (Dozat and Manning, 2017)



$H^{(arc\text{-}dep)} \oplus 1$
$U^{(arc)}$
$H^{(arc\text{-}head)}$
$\top$
$S^{(arc)}$
1
·
·
=
MLP: $\mathbf{h}_i^{(arc\text{-}dep)}, \mathbf{h}_i^{(arc\text{-}head)}$
BiLSTM: $\mathbf{r}_i$
Embeddings: $\mathbf{x}_i$
…
root
ROOT
Kim
NNP

Outlines

Dependency parsing

- Dependency grammar
- Projectivity
- Parsing
  - Transition-based
  - Graph-based

Discourse

- Discourse segmentation
- Discourse parsing
- Anaphor resolution

Discourse segmentation

A task for finding the sections in documents.

TextTiling algorithm

1. BOW k sentences at both sides of all gaps.
2. Calculate similarity between neighbor BOW vectors.
3. Calculate $depth(gap_i) = (sim_{i-1} - sim_i) + (sim_{i+1} - sim_i)$ (Note that i is the id of gaps.)

Supervised methods

1. Encode sentences/sections.
2. Perform classification on presence of boundary/type of sections

Discourse parsing

Rhetorical structure theory (RST) parsing:

- Similar to dependency parsing/pharse structure parsing, except we swap words with discourse units (DUs).

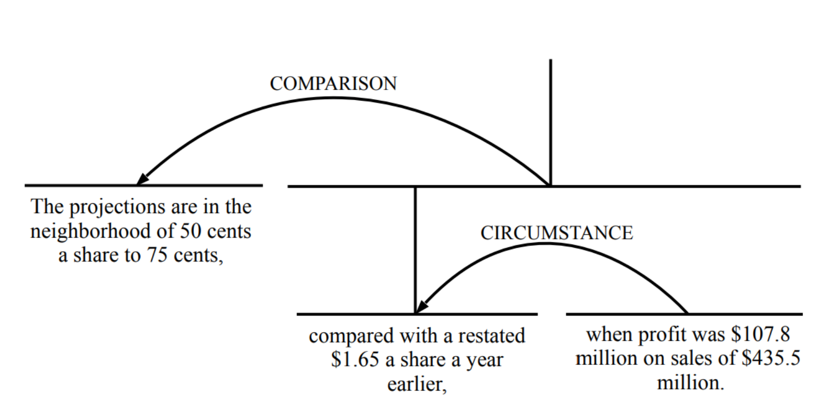

COMPARISON
CIRCUMSTANCE
The projections are in the neighborhood of 50 cents a share to 75 cents,
compared with a restated $1.65 a share a year earlier,
when profit was $107.8 million on sales of $435.5 million.

Figure from Ji and Eisenstein, 2014

Anaphor resolution

Concept:

- Anaphor: Linguistic expressions that refer back to earlier elements in the text.
- Antecedent: The element an anaphor refers to.
  - Pronouns: easy case, repetition of some previous mentions.
  - Demonstrative: that guy
  - Definites: the guy

Restrictions:

- Number. (e.g. *rats* ↔ *they*)
- Gender. (e.g. *Agirl* ↔ *she*)
- Reflexive pronoun (if as subject) (e.g. *Aboy* ↔ *himself*)

Unsupervised methods

The centering algorithm

- Assumption: One discourse focus on only one entity.
- Goal: Avoid rough shift of antecedent from that entity.

Definition:

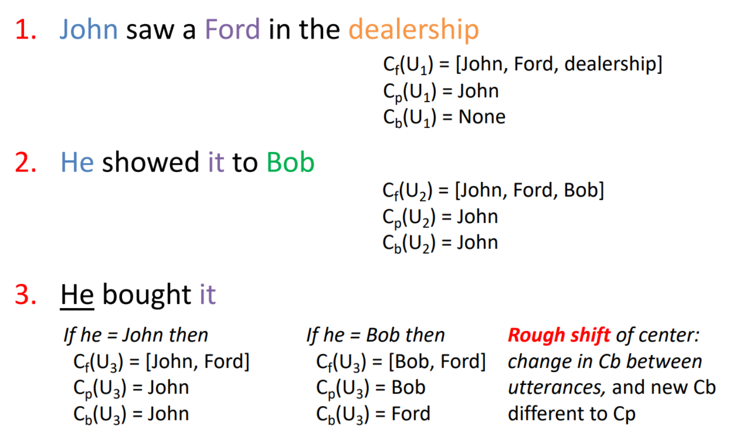
- U: a sentence in discourse
- C_f: list of entities in the current sentence, ordered by salience.
- C_b: backward center of current sentence.
- C_p: preferred forward center of current sentence.

Rules:

- $C_b(U_i)$: Entity **in** $C_f(U_i)$ with **highest order** in $C_f(U_{i-1})$
- $C_p(U_i)$: Entity with **highest order** in $C_f(U_i)$

The centering algorithm - Example

What is a ”rough” shift?



1. John saw a Ford in the dealership

$C_f(U_1)$ = [John, Ford, dealership]
$C_p(U_1)$ = John
$C_b(U_1)$ = None

2. He showed it to Bob

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

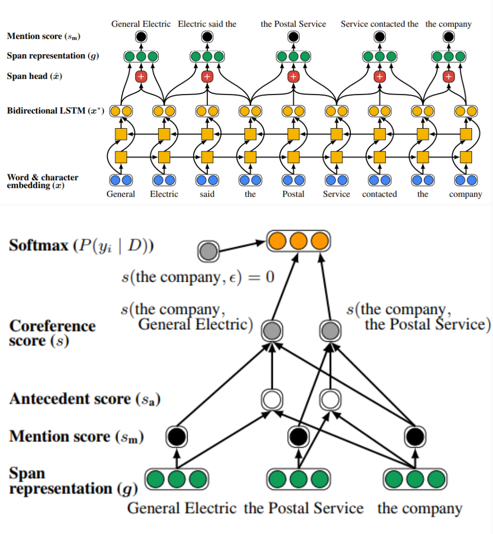
3. He bought it

If he = John then
$C_f(U_3)$ = [John, Ford]
$C_p(U_3)$ = John
$C_b(U_3)$ = John

If he = Bob then
$C_f(U_3)$ = [Bob, Ford]
$C_p(U_3)$ = Bob
$C_b(U_3)$ = Ford

***Rough shift** of center: change in Cb between utterances,* and new Cb different to Cp

Supervised anaphor resolution



Lee et al. 2017