Co-reference resolution

Yohan Chalier

Symbolic Natural Language Processing (SD213)

June 12, 2018

What is co-reference?

Co-reference occurs when two or more expressions refers to the same referent. Usually, one expression is in a full form (the *antecedent*) and the other one in a abbreviated form (a *proform*).

For example:
The **music** *was so loud that* **it** *couldn't be enjoyed.*

Co-reference resolution is needed to derive a correct interpretation of a text.

The problem of co-reference resolution

Naive algorithm

Look for the nearest preceding individual that is compatible with the referring expression.

It solves sentences like this:
The girl$_1$ *likes her*$_1$ *brother*$_2$ *and protects him*$_2$.

But it fails to differentiate those sentences:
He$_?$ *said that John*$_?$ *was coming.*
His$_1$ *sister said that John*$_1$ *was coming.*

Domination and c-command

Domination

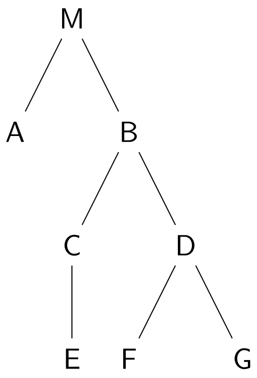
Node N_1 dominates node N_2 if N_1 is above N_2 in the tree and one can trace a path from N_1 to N_2 moving only downwards in the tree (never upwards).

c-command

Node N_1 c-commands node N_2 if

- N_1 does not dominate N_2
- N_2 does not dominate N_1
- The first (i.e. the lowest) branching node that dominates N_1 also dominates N_2

Domination and c-command



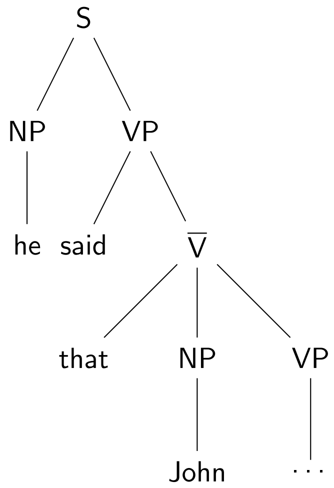
Co-reference and c-command

It was hypothesized that one restriction between proform and antecedent is that **the proform cannot appear in a position where it c-commands its antecedent**.

This is not trivial: Bouchard, Denis. (2010). *Une explication cognitive des effets attribués à la c-commande dans les contraintes sur la coréférence*. Corela. 10.4000/corela.965.

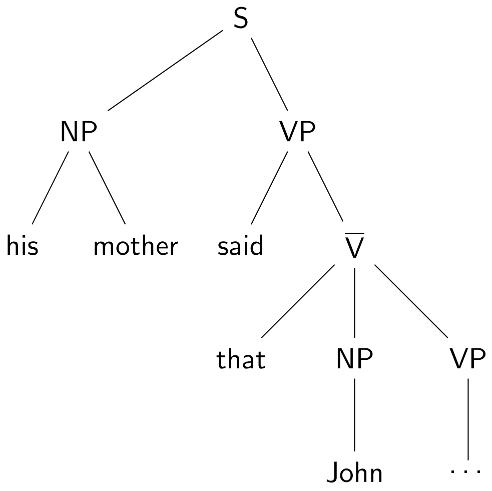
When John Comes Marching Home

he said that *john* was coming



When John Comes Marching Home

his mother said that *john* was coming



Current implementation

A Python script that contains:

- a small grammar and lexicon
- a very basic top-down parser in a class `Parser`
- a `Node` class to represent a syntax tree

Co-reference resolution

For each word recognized as a referring expression (for now, only pronouns), a method of the `Parser` finds a compatible word (*features agreement*), and if the proform does not c-command that word, it links them both.

See it in action (1/2)

```
Parsing: he said that John was_coming

 |_ s
   |_ np
     |_ pn : he
   |_ vp
     |_ v : said
     |_ sub
       |_ p : that
       |_ n : John
       |_ v : was_coming

he_0 said that John was_coming
```

See it in action (2/2)

```
Parsing: his sister said that John was_coming
his_0 sister said that John_0 was_coming

Parsing: the girl likes her brother and protects him
the girl_0 likes her_0 brother_1 and protects him_1
```

Difficulties and perspectives

- Parser improvement (if time, use CKY)
- Use more features to check compatibility (now just 2)
- Perform more tests to compute a precision score
- Ultimately, try parsing several sentences to find references across sentences

The script is available on GitHub:
`https://github.com/ychalier/coref/`

Thank you for your attention.