

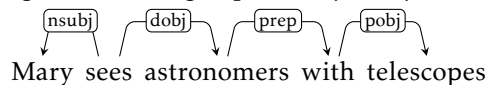
Dependency Parsing exercises:

Transition-based parsing II

Deadline: 29.05.2017. Please send the homework to petit.jean@phil.hhu.de and cranenburgh@phil.hhu.de with subject "dependency homework" and an attachment named "ex6_lastname1_lastname2.pdf".

1. Arc-eager parsing.

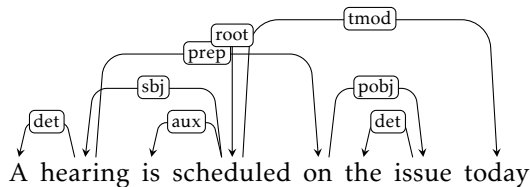
- (a) Enumerate the configurations an arc-eager transition-based parser goes through when producing the following dependency analysis:



A transition is a left-arc, right-arc, shift, or reduce operation (LA, RA, S, R). At each step, indicate the operation, the contents of the stack, the input buffer, and which dependency is added, if any:

- initial state: [Mary] [sees astronomers with telescopes] \emptyset
 - LA_{SBJ} [] [sees astronomers with telescopes] Mary \xleftarrow{SBJ} sees
 - ...
- (b) How does this compare to arc-standard parsing? In particular, do you think there could be the advantages or disadvantages for PP-attachments? In general, can you think of typological properties of languages that could make arc eager perform better than arc standard, or vice versa? Consider default word order (SVO, SOV, etc), free word order, head-final vs head-first, degree of inflection, etc.
- (c) After formulating a hypothesis in (b), compare it with some empirical results in the literature, e.g., the following paper contains results for two languages with both parsing strategies:
<http://aclweb.org/anthology/D14-1082.pdf>
 Is there a clear, consistent difference between the two strategies across languages, or is the difference not clear?
 Bonus points if you find a paper with more results.

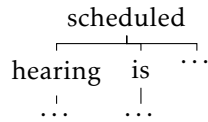
2. Non-projectivity.



- (a) Given a non-projective dependency tree, a projective version can be obtained by traversing the sentence with an in-order traversal of the tree.¹ Remember that a dependency tree can be drawn

¹See https://en.wikipedia.org/wiki/Tree_traversal#In-order

as a hierarchical tree structure, by starting with the head and drawing its dependents as children, and the dependents of those children etc. By drawing dependents of a head as children next to each other, you will get a different order than the non-projective order of the sentence:



What does the above sentence look like when it is re-ordered like this?

Draw the dependency tree to verify that there are no more crossing arcs with this sentence order.

- (b) The three methods for non-projective, transition-based parsing, pseudo-projectivity, online re-ordering, and k -planar parsing could be viewed on a scale of increasing elegance.

Is one of these approaches superior to the others when taking into account practical effectiveness, speed, coverage, precision, recall? Consider the results reported in the following papers:

- <http://aclweb.org/anthology/P05-1013.pdf>
- <http://aclweb.org/anthology/P09-1040.pdf>
- <http://aclweb.org/anthology/P10-1151.pdf>

3. For next week's lecture, prepare by reading the following exciting blog post:

<https://explosion.ai/blog/parsing-english-in-python>

Make note of any questions you may want to ask at the lecture. You don't have to write anything here for this exercise.