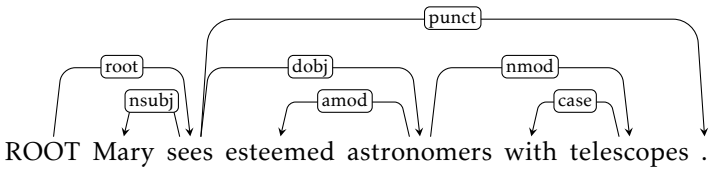
Dependency Parsing exercises: Transition-based arc-eager parsing, non-projective parsing

May 25, 2021

1. Arc-eager parsing.

   (a) Enumerate the configurations an arc-eager transition-based parser goes through when parsing the sentence:



   A transition is a left-arc, right-arc, shift, or reduce operation (LA, RA, SH, RE). At each step, indicate the operation, the contents of the stack, the input buffer, and which dependency is added, if any:[1]

   Solution: Here's one possible sequence (there may be others, due to spurious ambiguity):

TRANSITION	STACK	BUFFER	ARCS
	[ROOT]	[Mary sees esteemed . . .]	∅
SH	[ROOT Mary]	[sees esteemed astronomers . . .]	
LA_{SBJ}	[ROOT]	[sees esteemed astronomers . . .]	+(Mary $\overset{\text{SBJ}}{\leftarrow}$ sees)
RA_{ROOT}	[ROOT sees]	[esteemed astronomers with . . .]	+(ROOT $\overset{\text{ROOT}}{\rightarrow}$ sees)
SH	[ROOT sees esteemed]	[astronomers with telescopes .]	
LA_{AMOD}	[ROOT sees]	[astronomers with telescopes .]	+(esteemed $\overset{\text{AMOD}}{\leftarrow}$ astronomers)
RA_{DOBJ}	[. . . sees astronomers]	[with telescopes .]	+(sees $\overset{\text{DOBJ}}{\rightarrow}$ astronomers)
SH	[. . . astronomers with]	[telescopes .]	
LA_{CASE}	[. . . sees astronomers]	[telescopes .]	+(with $\overset{\text{CASE}}{\leftarrow}$ telescopes)
RA_{NMOD}	[. . . astronomers telescopes]	[.]	+(astronomers $\overset{\text{NMOD}}{\rightarrow}$ telescopes)
RE	[ROOT sees astronomers]	[.]	
RE	[ROOT sees]	[.]	
RA_{PUNCT}	[ROOT sees .]	[]	+(sees $\overset{\text{PUNCT}}{\rightarrow}$.)

   (b) How does this compare to arc-standard parsing? In particular, do you think there could be disadvantages (or advantages) for PP-attachments when using the arc-eager parser? 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.

   Solution:

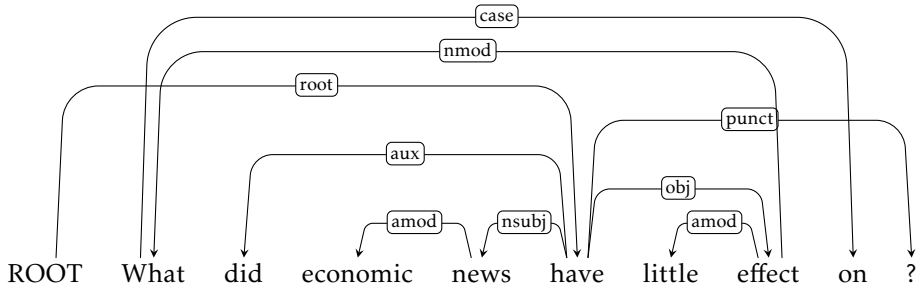
   - The arc-eager system handles PP attachments (in English!) more naturally in that it can attach the direct object to the corresponding verb without committing to any PP-attachment decision. Consider the sentence *ROOT Mary sees astronomers with big white telescopes*. The arc-standard parser has to decide about the PP-attachment in the context

[1] A configuration is *terminal* if its buffer is empty, just as in the arc-standard system.

of ([ROOT sees], [astronomers with big white telescopes], ...), where *telescopes* is placed in the buffer rather far from the top. The arc-eager system makes the PP-attachment decision locally ([ROOT sees astronomers], [telescopes], ...), where all the words involved (*sees*, *astronomers*, and *telescopes*) are directly accessible at the top of the stack and buffer, and the intervening words (*with big white*) are already attached to the final noun.[2]

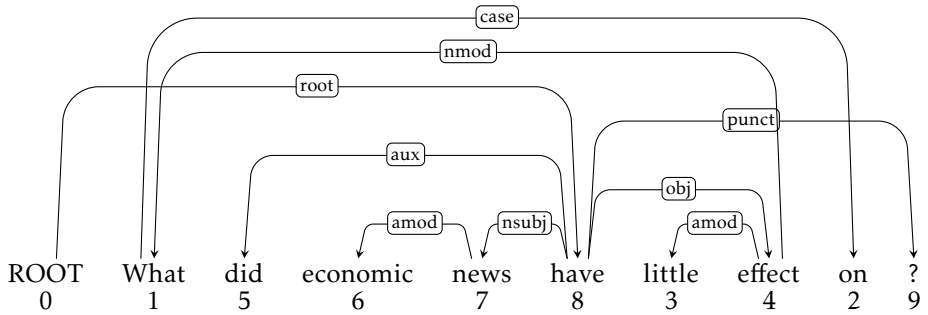
- Consider the SOV sentence in German: *ROOT Mary hat Astronomen mit Teleskopen, die groß und weiß waren, beobachtet*. The arc-eager parser makes the PP-attachment decision in the context of ([ROOT Mary hat Astronomen], [Teleskopen, die groß und weiß waren, beobachtet], ...), with no local access to the verb. The task for the arc-standard parser is easier in this case, since it takes the PP-attachment decision in the context of ([ROOT Mary hat Astronomen], [Teleskopen beobachtet], ...), where the right dependents of *Teleskopen* are already processed.
- In general (tentative answer):
  - if there is a lot of left arcs (head-initial language): the choice between eager and standard doesn't make much of a difference, the two systems handle left arcs similarly
  - if there is a lot of right arcs (head-final language): arc-eager may be more effective (as in the PP-attachment ambiguity example in English)
  - if there's a balanced mix between left and right arcs: arc-standard may be more effective (as in the PP-attachment ambiguity example in German)
- In the paper by Chen and Manning (2014)[3] the arc-standard system is reported to give better results on Chinese (but not on English)

2. Consider the following dependency tree:



(a) Determine a *projective order* for the tree above. It can be obtained by traversing the sentence with an in-order traversal of the tree.[4] Then re-order the words in the sentence accordingly (i.e. place each word on its position according to the projective order) and draw the resulting dependency tree to verify that there are no crossing arcs with this word order.

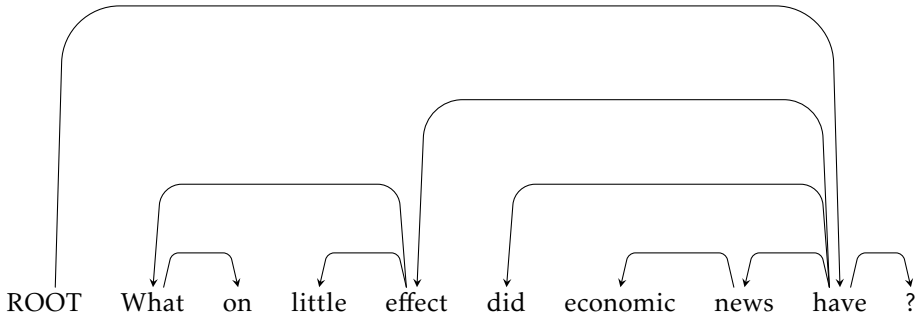
Solution: One possible solution:



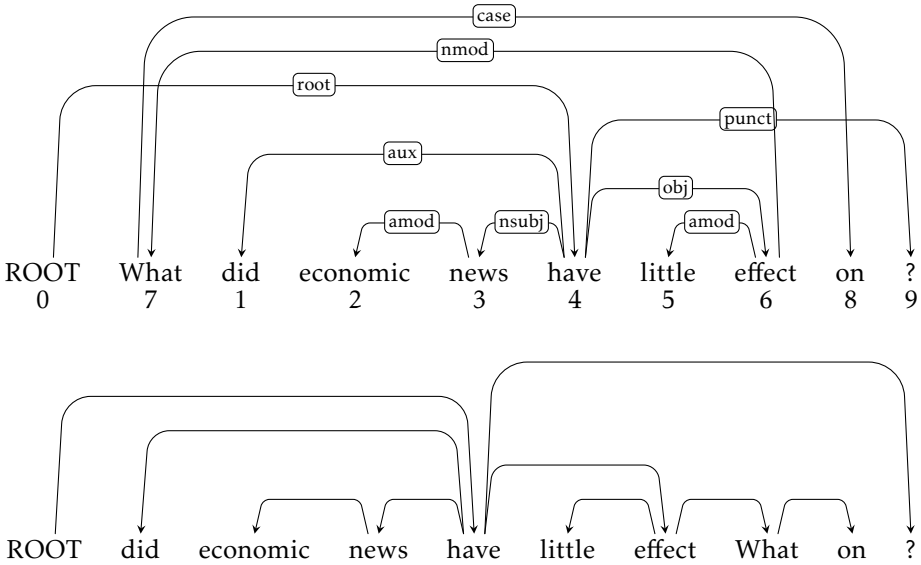
[2] However, this reasoning applies to UD-style dependencies! If the preposition were the head of the PP, an arc-eager parser would have to make the PP-attachment decision without necessarily having local access to the complement of the preposition (i.e. the noun in the PP).

[3] https://www.emnlp2014.org/papers/pdf/EMNLP2014082.pdf

[4] See https://en.wikipedia.org/wiki/Tree_traversal#In-order.



Another one:



(b) Propose a transition sequence that allows to reconstruct the non-projective dependency tree above using the online reordering parser. A transition is left-arc, right-arc, shift, or swap (LA, RA, SH, SW). At each step, indicate the operation, the contents of the stack, the input buffer, and which dependency is added, if any:

Solution: Note that there are other possible transition sequences which yield the same tree. In particular, the derivation below can be shortened: instead of doing all the swaps as soon as possible, one can wait for some of the words to group into subtrees (e.g. *economic news*) and perform the swaps after that.

TRANSITION	STACK	BUFFER	ARCS
	[ROOT]	[What did economic news . . .]	∅
SH	[ROOT What]	[did economic news . . .]	
SH	[ROOT What did]	[economic news . . .]	
SW	[ROOT did]	[What economic news . . .]	
SH	[ROOT did What]	[economic news have . . .]	
SH	[ROOT did What economic]	[news have . . .]	
SW	[ROOT did economic]	[What news have . . .]	
SH	[ROOT did economic What]	[news have . . .]	
SH	[ROOT did economic What news]	[have little . . .]	
SW	[ROOT did economic news]	[What have little . . .]	
LA_{AMOD}	[ROOT did news]	[What have little . . .]	+(economic $\overset{\text{AMOD}}{\leftarrow}$ news)
SH	[ROOT did news What]	[have little effect on ?]	
SH	[ROOT did news What have]	[little effect on ?]	
SW	[ROOT did news have]	[What little effect on ?]	
LA_{NSUBJ}	[ROOT did have]	[What little effect on ?]	+(news $\overset{\text{NSUBJ}}{\leftarrow}$ have)
SH	[ROOT did have What]	[little effect on ?]	
SH	[ROOT did have What little]	[effect on ?]	
SW	[ROOT did have little]	[What effect on ?]	
SH	[ROOT did have little What]	[effect on ?]	
SH	[ROOT did have little What effect]	[on ?]	
SW	[ROOT did have little effect]	[What on ?]	
LA_{AMOD}	[ROOT did have effect]	[What on ?]	+(little $\overset{\text{AMOD}}{\leftarrow}$ effect)
SH	[ROOT did have effect What]	[on ?]	
SH	[ROOT did have effect What on]	[?]	
RA_{CASE}	[ROOT did have effect What]	[?]	+(What $\overset{\text{CASE}}{\rightarrow}$ on)
RA_{NMOD}	[ROOT did have effect]	[?]	+(effect $\overset{\text{NMOD}}{\rightarrow}$ What)
RA_{OBJ}	[ROOT did have]	[?]	+(have $\overset{\text{OBJ}}{\rightarrow}$ effect)
LA_{AUX}	[ROOT have]	[?]	+(did $\overset{\text{AUX}}{\leftarrow}$ have)
SH	[ROOT have ?]	[]	
RA_{PUNCT}	[ROOT have]	[]	+(have $\overset{\text{PUNCT}}{\rightarrow}$?)
RA_{ROOT}	[ROOT]	[]	+(ROOT $\overset{\text{ROOT}}{\rightarrow}$ have)