# Dependency Analysis of Sentences

Introduction to Syntax & Parsing, ACS MPhil, Assignment 4 © Ted Briscoe (ejb@cl.cam.ac.uk)
GS18

## 1 Task

Choose 2 sentences from each of the 4 sets below (8 total) and draw or write out a typed dependency / grammatical relations (GRs) graph of bilexical head-dependent relations based on the PSTs you assigned to the sentences in previous assignments.

For instance the bilexical relations for:

My aunt's can opener can open a drum

should look something like this:

```
(|ncsubj| |open:7| |opener:5| _)
(|aux| |open:7| |can:6)
(|dobj| |open:7| |drum:9|)
(|det| |drum:9| |a:8)
(|ncmod| |poss| |opener:5| |aunt:2|)
(|ncmod| _ |opener:5| |can:4|)
(|det| |aunt:2| |My:1|)
```

using the GR scheme defined in section 7, p15f of
http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-662.pdf
(see also section 5.7 of Intro to Linguistics handout and the Parc DepBank examples on the course webpage)

Write up or draw your answers and submit them before the following session. Include BRIEF notes on any difficulties or issues you had with specific cases. It is more important to understand and be able to explain your reasoning than to get every GR right. Be prepared to discuss the difficult cases during the session. Please feel free to work on the task in groups, but the final selection of sentences and their analyses should be your own. I'd recommend trying some new sentences in this assignment, especially if you are finding the analyses easy. For the longer examples, you are also welcome to use a parser to get an initial set of GRs and then correct them – the RASP parser produces exactly this GR scheme, but the mapping to Stanford Dependencies, which are output by some of the parsers in the Stanford Core Tools, is fairly straightforward. As the sentences and their lists of dependencies gets longer, checking them becomes harder and the sentences become more structurally ambiguous, so another useful trick is to create shorter sentences from the longer ones and see how these are analysed.

## 2 Sentences

(1) a The old car broke down in the car park

b At least two men broke in and stole my TV

c The horses were broken in and ridden in two weeks

(2) a It was my aunt's car which we sold at auction last year in February

b The only rabbit that I ever liked was eaten by my parents one summer

c The veterans who I thought that we would meet at the reunion were dead

(3) a Natural disasters – storms, flooding, hurricanes – occur infrequently but cause devastation that strains resources to breaking point

b It won't rain but there might be snow on high ground if the temperature stays about the same over the next 24 hours

c My wildest dream is to build a POS tagger which processes 10K words per second and uses only 1MB of RAM, but it may prove too hard

(4) a English also has many words of more or less unique function, including interjections (oh, ah), negatives (no, not), politeness markers (please, thank you), and the existential 'there' (there are horses but not unicorns) among others.

b The Penn Treebank tagset was culled from the original 87-tag tagset for the Brown Corpus. For example the original Brown and C5 tagsets include a separate tag for each of the different forms of the verbs *do* (e.g. C5 tag VDD for *did* and VDG tag for *doing*), *be* and *have*.

c The slightly simplified version of the Viterbi algorithm that we present takes as input a single HMM and a sequence of observed words $O = (o_1, o_2, ...o_T)$ and returns the most probable state/tag sequence $Q = (q_1, q_2, q_T)$ together with its probability.