School of Computing and Information Systems
The University of Melbourne
COMP90042
WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2018)

Workshop exercises: Week 6

Discussion

1. What differentiates **probabilistic parsing** from **chart parsing**? Why is this important? How does this affect the algorithms used for parsing?
2. What is a **probabilistic grammar** and what problem does it attempt to solve?
3. A hidden Markov model assigns each word in a sentence with a tag, e.g.,

   Donald/NNP has/VBZ small/JJ hands/NNS

   The probability of the sequence is based on the tag-word pairs, and the pairs of adjacent tags. Show how this process can be framed as a CFG, and how the various probabilities (e.g., observation, transition, and initial state) can be assigned to productions. What are the similarities and differences between CYK parsing with this grammar, and the HMM's Viterbi algorithm for finding the best scoring state sequence?
4. Using typical dependency types, construct (by hand) a dependency parse for the following sentence: *Yesterday, I shot an elephant in my pyjamas.* Check your work against the output of the online GUI for the Stanford Parser (`http://nlp.stanford.edu:8080/parser/index.jsp`).
5. In what ways is (transition–based, probabilistic) dependency parsing similar to (probabilistic) CYK parsing? In what ways is it different?

Programming

1. NLTK doesn't have much dependency parsing support — there is a little MaltParser interface (`http://maltparser.org/`), but it can be unreliable.

   One popular dependency parsing platform is the Stanford Parser (`https://nlp.stanford.edu/software/lex-parser.shtml`) — the entire package is a somewhat large and requires Java. There are some Python bindings, however. (For example, `http://projects.csail.mit.edu/spatial/Stanford_Parser`)
2. Parse the (tokenised) sentences in `nltk.corpus.treebank_raw.sents()`
3. What proportion of the resulting dependency trees are non–projective? Why do you suppose this is?

Catch-up

- How can a **prior** probability be estimated from a collection of data, using a **maximum likelihood estimate** approach? What about a **posterior** probability?
- Why are we often concerned by a model where some events have a probability equal to 0?
- What is a **head**? A **dependency**? How do dependencies differ from **constituents**?
- What are the major differences between the (tree) structure produced by parsing using a context–free grammar, and that produced by parsing using a dependency grammar?
- What are some common dependency arc labels, and what grammatical notions do they encode?

Get ahead

- Revise how to train a PCFG parser. Read up on how to use the Stanford parser as a "vanilla" PCFG parser — compare its output on some selected sentences, when using training sets of different sizes (for example, slices of the Penn Treebank).
- Adapt the CYK code in `WSTA_N8_context-free_grammars` to work for a PCFG, and devise a means of estimating PCFG production probabilities from corpora. You will need a way to deal with low count events, such as unseen words.