Assignment 4

L445 / L545

Due Wednesday, March 1

1. Using your knowledge of n-grams, explain to a layperson what n-grams models are, what they do, what they don't do well, and why smoothing is necessary. You should address whether linguistic knowledge is necessary in NLP. You are writing not to me, but to a reasonably educated person who knows nothing about NLP. Write no more than one page (points will be deducted from longer answers) – for "one page", I'm assuming 12-point font, double-spaced, one-inch margins.

2. Do question 5.2 on page 171 in Jurafsky & Martin, i.e., annotate some selected sentences by hand with POS labels from the Penn Treebank. (The tagset is located on page 131, and the guidelines are referenced in the next question.)

3. (a) Using the guidelines for the Penn Treebank tagging found at: `ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz` and your own linguistic intuitions, hand-write some rules to distinguish the following tags IN and RP. I would prefer several high-precision rules over a smaller number of high-recall (but low-precision) rules. Include a brief description of how your rule is derived from the guidelines.

   (b) The following are some rules learned by a Brill tagger. Explain what they do and why they work.

      i. `VB VBP PREVTAG NNS`

      ii. `IN WDT NEXT1OR2TAG VB`

      iii. `JJ NNP NEXTTAG NNP`

4. (a) The tagging systems we talked about in class use the preceding context (e.g., the preceding tag) to disambiguate a tag for a given word. Why? And would you expect this to work for all languages? Why or why not?

   (b) Some tagging decisions seem to depend upon the following word; for example, *I can/VBP tuna* vs. *I can/MD help*. HMM taggers condition on the previous tags, e.g., $P(w_i|w_{i-2}w_{i-1})$. How is it, then, that HMM taggers can often tag these cases accurately? In other words, how do they capture cases where the tag of w_i depends upon w_{i+1}?

5. Prep for the next assignment ...

   (a) Read this 1990 paper by Fred Karlsson on the basics of Constraint Grammar: `http://www.aclweb.org/anthology/C90-3030`

   (b) On the next assignment, I'm going to have you work in groups of 2–4 using: i) a POS tagger, `TreeTagger`, and ii) a constraint grammar compiler, `vislcg3`. Figure out your group.

   (c) At least one of you in the group should be able to successfully install `TreeTagger` (`http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`) and `vislcg3` (`http://beta.visl.sdu.dk/cg3.html`) . . . For this assignment, confirm that someone(s) has/have them installed.