Linguistics 165, Homework 6
due 13 March 2015

7 March 2015

Please write up your solutions to these problems as a single PDF file and send it to lign165-homework@ling.ucsd.edu as usual. Please format any code you include in the writeup using a monospace font. You don’t need to submit any code directly to us for this homework assignment.

1. **Tree searches with Tregex.** In the ditransitive construction, there are two different syntactic frames, as illustrated by (nearly) meaning-equivalent alternants like:

   **Double-Object Construction**
   Pat gave Kim the book.

   **Prepositional-Dative Construction**
   Pat gave the book to Kim.

   The **Prepositional-Dative** is so called because the “goal” argument of the verb (the entity that gets given something) is expressed as a prepositional phrase. The **Double-Object** construction is so called because both the goal and the theme (the entity that is being given) are expressed as object NPs, without being marked using a preposition.

   (a) What is the difference in Penn Treebank syntactic structure between the double-object and prepositional-dative construction in terms of the VP rewrite rule used for each construction? You can answer this by, for example, using the Tregex GUI to browse trees containing `give` or other such verbs.

   (b) There are many different verbs besides `give` that allow both the double-object and prepositional-dative variants, including `sell`, `award`, and `tell`. First, use your intuition to guess the order of preference strength for each of these verbs, ranging from most double-object-preferring to most prepositional-dative-preferring. Report these intuitions. Now, use Tregex to compute relative-frequency estimates of

   \[
P(\text{double-object}|\text{verb with ditransitive meaning})
   + P(\text{prepositional-dative}|\text{verb with ditransitive meaning})
   = 1
\]
—that is, that **double-object** and **prepositional-dative** are the only two ways of realizing a ditransitive meaning. How do the relative frequency estimates line up with your intuitions? Comment on what you’ve discovered. **Hint:** Tregex patterns are left-associative: for example, @VP < @NP < @NNS means a VP that has both an NP daughter and an NNS daughter, not a VP that has an NP that has an NNS daughter. You can use parentheses to disambiguate. For example, @VP < @NP < @NNS and (@VP < @NP) < @NNS mean the same thing, but @VP < (@NP < @NNS) means something different (it means a VP that has an NP that has an NNS daughter).

2. **PCFGs and tree probabilities.** Consider the four possible parses of the sentence $s =$*the old harbor burns*:

Rank them intuitively in order of their likelihood as the intended meaning of the sentence, explaining your intuitions as much as possible.

Let’s assume that these are the only four parses for this sentence. For each parse $t$, compute $P(t)$ by using “vanilla” relative frequency estimation on the Penn Treebank. Then, use Bayes’ Rule to compute $P(t|s)$ for each of the four trees. (**Note:** in the Penn Treebank, sentence-final punctuation is grouped under $S$, but we’re ignoring sentence-final punctuation in this problem specification, so you might want to allow your Tregex search for $S \rightarrow NP \ VP$ rules to be flexible in whether you include instances of $S \rightarrow NP \ VP$ to count in the estimation of $P(S \rightarrow NP \ VP)$. However, you will get reasonable estimates of $P(t|s)$ whether or not you do this, so it’s not required.)
3. **Extra Credit:** implement a CKY parser in Python and show that for the following grammar:

\[
\begin{align*}
S & \rightarrow NP \ VP & N & \rightarrow \text{mud} \\
NP & \rightarrow N & N & \rightarrow \text{tables} \\
NP & \rightarrow NP \ Conj \ N & N & \rightarrow \text{chairs} \\
VP & \rightarrow V \ NP & V & \rightarrow \text{covered} \\
VP & \rightarrow VP \ Conj \ VP & & \\
\end{align*}
\]

it recognizes the following sentences:

- mud covered tables and chairs
- mud covered tables and covered chairs

but doesn’t recognize the following sentences:

- mud covered tables and mud covered chairs

Show the parse table resulting from your parser for each of these sentences.

**Remember**, for a CKY parser you need to use a Chomsky Normal Form (CNF) grammar. You can either hand-convert the grammar to CNF, or include code in your program that automates the conversion process.