**Computational Linguistics 1 - Fall 2011**
**CMSC/LING723, LBSC 744**
**Homework 7 - Due 13 December 2011**

# Mini-Projects

This homework is intended to be a mini-project. Choose one of the four options listed below, and email `compling723.fall2011@gmail.com` *by November 29* with the project that you have chosen.

## Machine Learning

### Option 1: Unsupervised Learning (EM Algorithm)

Learn a POS tagger without any labeled data. Train a bi-tag POS tagging model using the Expectation Maximization algorithm. You may learn this tagger on any data you like, including our Wall St Journal corpus, so long as you are not using any annotations on the data. Let's say that we are looking for 10 tag classes (rather than the 45 that we know to be in the the WSJ corpus), and let's call them C0, C1, C2, C3, C4, C5, C6, C7, C8, and C9. We'll initialize the classes according to some simple orthographic rules so that you all have the same starting point. (These details to be provided later.)

Run EM for 200 iterations. For this option, you will not be able to report accuracy (since it's an unsupervised task), so you will report the log likelihood of your training corpus. Additionally, we want to see if the tag classes we're learning make any sense, so you'll report the five words in each class with the highest $P(w|C_j)$, as representatives of the class, at iterations 1, 50, 100, and 200. Discuss the quality of these classes, and do they improve with higher iterations?

### Option 2: Perceptron Algorithm

Instead of using an HMM model for our POS tagger, implement a perceptron model. Use just the features that we used for our POS tagger (i.e., bigram tag features: $\phi(\tau_i, \tau_{i-1})$, and tag-word features: $\phi(\tau_i, w_i)$).

Iterate over the training set 10 times. (Normally we would use a heldout set to determine when to stop training.) Report tagging accuracy on the test set, and discuss whether it's different than the HMM model. Show an instance where the perceptron model made a different prediction than the HMM model. Discuss the pro's and con's of each (HMM and perceptron).

## Feature Engineering

### Option 3: POS Improvement

We implemented a really basic OOV model for our POS tagger. Try implementing something better; report your POS tagger's accuracy on our test set.

### Option 4: Parsing Improvement

Our parser is fairly basic. Can you introduce some new annotations for the parse labels that will help with things like lexical semantics? One very popular approach has been to label the nodes of a parse tree with the label of that node's parent. Thus, S $\rightarrow$ NP VP would become S $\rightarrow$ NP-S VP-S. You might also label each node with its head child, following the Collins rules for determining heads of a parse node. Thus, S $\rightarrow$ NP VP might become S-bit $\rightarrow$ NP-dog VP-bit for a sentence like "That dog bit the policeman".

Report your parser's accuracy on the test set.