Natural Language Systems Coursework: POS-tagging, POS tag disambiguation, Constituent Parsing and Distributional Semantics

1. Part-of-speech (POS) tagging:

a) Use a POS-tagger of your choice (e.g. NLTK tagger, Stanford, TreeTagger etc. to tag corpus A. Indicate in your report what kind of tagger is the one you have used. What are the 5 most frequent tags in this corpus?

b) Let's assume that the POS tagging you have generated for the corpus is a gold standard. Use the annotated corpus to estimate word likelihood and tag transition probabilities you would need to be able to disambiguate which of the following two POS tagging results is more likely.

(1) People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN
(2) People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/VB for/IN outer/JJ space/NN

Explain what you have done and comment/explain the results in the report.

2. Constituent parsing:

Use Stanford parser (you may use an online demo) to generate the constituent trees for the following two sentences:

(1) She stood by the door covered in tears.
(2) She stood by the door covered in ivy.

Provide screenshots with parse trees and explain the results in the report. What kind of parser is Stanford parser?

3. Distributional semantics:

a) Implement a program to cluster a given list of target words into n groups based on their distributional co-occurrence patterns. You may first want to construct a word-by-word matrix that captures co-occurrence patterns of the given target words using a given corpus. Define your context and features flexibly so that you can analyse the results in part c below. Your program should take as input a list of words to cluster and a number of clusters. Use any available machine-learning framework for clustering (e.g. sckikit-learn (http://scikit-learn.org/) or Weka (http://www.cs.waikato.ac.nz/ml/weka/)). Explain briefly what you have done in the report (1⁄2 page).

b) Use corpus B (available in Blackboard) and target list D (with 50 words, also available in Blackboard) to evaluate the results of your clustering. Use the following pseudoword disambiguation approach: for each target word, randomly substitute half of its occurrences in the corpus with its reverse (e.g., "procedure" will be transformed into "erudecorp"). Now, apply your clustering algorithm to the list of 100 target words, which contains original words and their reverses, producing 50 clusters. If you generate 50 clusters, how many of them will contain "correct" pairs (i.e., a word and its reverse)? Repeat this process 5 times and give the average accuracy. Explain the result in the report (1⁄2page).

c) Analyse the impact of (1) the size of context, (2) type of features and (3) training data on the quality of generated clusters.

To analyse the contribution of contextual representation, consider different ways of constructing a word-by-word matrix (i.e. vary the dimensions of the context window) and experiment with different definitions of context (stems vs. words).

To analyse the impact of training data, in addition to corpus B, use also corpus C (available in

Blackboard) and "train" your system on each corpora separately, and also on their combination. Comment the results and report any difference (1 page).

What other type(s) of feature you may consider using (you don't need to implement or analyse the impact of the additionally proposed feature(s))?