ASSIGNMENT 3

COMP 599, Fall 2016

Due: November 16^{th}, 2016 in class. No late assignments accepted.

You must do this assignment individually. You may consult with other students orally, but may not take notes or share code, and you must complete the final submission on your own.

Question 1:	30 points
Question 2:	40 points
Question 3:	30 points
	100 points total

Assignment

Question 1: Lambda Calculus and Compositional Semantics (30 points)

a) Simplify the following lambda calculus expressions by applying beta-reduction as much as possible. Show each step of the derivation.

- (λa.a b a)(λc.c)
- ((λv.v v) (λu.u)) (λv.v)
- (λt.t) (λt.t t) (λt.t b)

b) Augment the grammar given in Lecture 16 to account for the quantifier *no*, including the corresponding lexical rule and semantic attachment. Show a derivation of the sentence *No student hates COMP-599*. (You'll also need to add an entry for the verb *hates*.) Use explicit event variables, and be sure to show the intermediate lambda expressions at each node of the parse tree.

c) Show how to construct an underspecified representation of the sentence *No student wants an exam* using the Cooper storage scheme presented in class. Show how to recover both interpretations of the sentence (and explain what the interpretations are).

Question 2: Lesk's Algorithm (40 points)

Implement and apply Lesk's algorithm to the publicly available data set of SemEval 2013 Shared Task #12 (Navigli and Jurgens, 2013), using NLTK's interface to WordNet v3.0 as your lexical resource. (Be sure you are using WordNet v3.0!) The relevant files are available on the course website. Starter code is also provided to help you load the data. More information on the data set can be found at `https://www.cs.york.ac.uk/semeval-2013/task12/`.

The provided code will load all of the cases that you are to resolve, along with their sentential context. Apply word tokenization and lemmatization (you have code to do this from A1) as necessary, and remove stop words.

As a first step, compare the following two methods for WSD:

1. The most frequent sense baseline: this is the sense indicated as #1 in the synset according to WordNet

2. NLTK's implementation of Lesk's algorithm (`nltk.wsd.lesk`)

You need to implement the evaluation measures of precision, recall, and F1. There is sometimes more than one correct sense annotated in the key. If that is the case, you may consider an automatic system correct if it resolves the word to any one of those senses. What do you observe about the results?

Next, develop a third method that combines distributional information about the frequency of word senses, and the standard Lesk's algorithm. Make and justify decisions about any other parameters to the algorithm, such as what exactly to include in the sense and context representations, how to compute overlap, and how to trade off the distributional and the Lesk signal, with the use of the development set, which the starter code will load for you. You may use any heuristic, probabilistic model, or other statistical method that we have discussed in class in order to combine these two sources of information. It is beyond the scope of the assignment to use external corpora or lexical resources (e.g., thesauri, or WordNet in other languages, etc.), so do not do so. Given these constraints, feel free to use your creativity to find ways to improve performance!

Some issues and points to watch out for:

- The gold standard key presents solutions using lemma sense keys, which are distinct from the synset numbers that we have seen in class. You will need to convert between them to perform the evaluation. This webpage `https://wordnet.princeton.edu/man/senseidx.5WN.html` explains what lemma sense keys are.
- The data set contains multi-word phrases, which should be resolved as one entity (e.g., latin_america). Make sure that you are converting between underscores and spaces correctly, and check that you are dealing with upper- vs lower-case appropriately.
- We are using instances with id beginning with d001 as the dev set, and the remaining cases as the test set, for simplicity. This is different from the setting in the original SemEval evaluation, so the results are not directly comparable.

Discuss the results of your experiments with the three models. Also include a discussion of the successes and difficulties faced by the models. Include sample output, some analysis, and suggestions for improvements. The entire report, including the description of your model, must be no longer than **two pages**. Going beyond this length will result in a deduction of marks.

Your grade will depend on whether you adequately followed the guidelines above, whether you followed standard model design and experimental procedure during the development of your method, and on the quality of your report (both linguistic, and content).

Question 3: Reading Assignment — Narrative Event Chains (30 points)

Read the following paper:

Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Event Chains. *ACL 2008*. `http://aclweb.org/anthology/P/P08/P08-1090.pdf`

Write a max. one-page (c. 500 words) discussion on this paper, including the following points:

1. A brief summary of the contents of the paper, including the theoretical framework and the experiments.
2. An evaluation and synthesis of what you learned in the paper. What are the advantages and limitations of this work? How does it relate to the topics that we have discussed in class on unsupervised learning and frame semantics?
3. Three questions related to the paper. These can be clarification questions, or questions about potential extensions of the paper, or its relationship to other work.

What To Submit

On paper: Submit a hard copy of your solutions to Question 1 and 3 as well as the report part of Question 2 in class.

Electronically: For the programming part of Question 2, you should submit one zip file with your source code to MyCourses under Assignment 3.