ASSIGNMENT 1

COMP 550, Fall 2017

Due: Friday, September 29th, 2017, 11:59pm.

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:	20 points
Question 3:	50 points
	100 points total

Assignment

Question 1: Identify the Ambiguity (30 points)

Analyze the following passages by identifying the most salient instances of linguistic ambiguity that they exhibit. Write a *short* paragraph for each that answers the following questions. What is the ambiguity, and what are the different possible interpretations? What in the passage specifically causes this ambiguity? What domain does this ambiguity operate over (phonological, lexical, syntactic, orthographic, etc.)? What sort of knowledge is needed for a natural language understanding system to disambiguate the passage, whether the system is human or machine? Be more specific than simply saying "contextual knowledge."

1. *Every student read a book.*
2. *The lion is a majestic animal.*
3. *Use of this sidewalk is prohibited by police officers.*
4. *My English teacher recently recovered from a bowel cancer operation... and he tried to show me a semi colon.* (Source: The 2016 UK Pun Championship)
5. *She is my ex-mother-in-law-to-be.*

Question 2: FST for German Verbal Conjugation (20 points)

Develop a FST to perform morphological analysis for the following German verbal conjugation table, which shows verbs conjugated in the present tense:

Infinitive	**1 Sg**	**2 Sg**	**3 Sg**	**1 Pl**	**2 Pl**	**3 Pl**
Regular verbs						
spielen (to play)	spiele	spielst	spielt	spielen	spielt	spielen
warten (to wait)	warte	wartest	wartet	warten	wartet	warten
gehen (to go)	gehe	gehst	geht	gehen	geht	gehen
arbeiten (to work)	arbeite	arbeitest	arbeitet	arbeiten	arbeitet	arbeiten
Verbs with a stem change						
sprechen (to speak)	spreche	sprichst	spricht	sprechen	sprecht	sprechen
backen (to bake)	backe	bäckst	bäckt	backen	backt	backen
Irregular verbs						
sein (to be)	bin	bist	ist	sind	seid	sind
haben (to have)	habe	hast	hat	haben	habt	haben

The morphological analyzer should provide the infinitive form of the verb, which we will take to be its lemma, along with its POS, person and number agreement. For example, feeding "*habe#*" as input to the final FST should result in the output "*haben* +V +1 +Sg".

Your response should include three components:

- A schematic transducer in the style of Figure 3.13 in J&M (page 61)
- A lexicon table as in the top half of Figure 3.14 in J&M (page 62)
- A "fleshed-out" FST in the format of the bottom half of Figure 3.14 for the lexical items presented above

Question 3: Sentiment Analysis (50 points)

In this question, you will train a simple classifier that classifies a sentence into either a positive or negative sentiment. These sentences come from a movie review dataset constructed by the authors of this paper:

Bo Pang and Lillian Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, *Proceedings of ACL 2005.*

The goal of this question is to give you experience in using existing tools for machine learning and natural language processing to solve a classification task. Before you attempt this question, you will need to install Python 2 on the machine you plan to work on, as well as the following Python packages and their dependencies:

- NLTK: `http://www.nltk.org/`
- NumPy: `http://www.numpy.org/`
- scikit-learn: `http://scikit-learn.org/stable/`

Download the corpus of text available on the course website. This corpus is a collection of movie review sentences that are separated into positive and negative polarity. Your task is to train a sentence classifier to distinguish them.

Data storage and format

The raw text files are stored in *rt-polarity.neg* for the negative cases, and *rt-polarity.pos* for the positive cases.

Preprocessing and feature extraction

Preprocess the input documents to extract feature vector representations of them. Your features should be N-gram counts, for N $\leq$ 2. You may also use scikit-learn's feature extraction module. You should experiment with the complexity of the N-gram features (i.e., unigrams, or unigrams and bigrams), and whether to remove stop words. NLTK contains a list of stop words in English. Also, remove infrequently occurring words and bigrams as features. You may tune the threshold at which to remove infrequent words and bigrams. You can also experiment with the amount of smoothing/regularization in training the models to achieve better results. Read scikit-learn's documentation for more information on how to do this.

Setting up the experiments

Design and implement an experiment that correctly compares the model variants, so that you can draw reasonable conclusions about which model is the best for generalizing to similar unseen data. Compare the logistic regression, support vector machine (with a linear kernel), and Naive Bayes algorithms. Also, compare against the expected performance of a random baseline, which just guesses positive or negative with equal probability.

Report

Write a *short* report on your method and results, carefully document i) the problem setup, ii) your experimental procedure, iii) the range of parameter settings that you tried, and iv) the results and conclusions. It should be no more than one page long. Report on the performance in terms of accuracy, and speculate on the successes and failures of the models. Which machine learning classifier produced the best performance? For the overall best performing model, include a confusion matrix as a form of error analysis.

What To Submit

Submit your solutions to Questions 1 to 2, as well as the report part of Question 3 in class as a pdf. For the programming part of Question 3, you should submit one zip file with your source code. All work should be submitted to MyCourses under the Assignment 1 folder.