ECS763P/U Natural Language Processing Late Summer Resit Assignment

Complete both parts 1 and 2.

PART 1: Quiz (worth 20% of grade)

Instructions: Please submit a pdf file with your answers to the below questions clearly indicated.

Each question is worth 2%.

1. Which of these are properties of natural language as a data type (select all that apply)?

   Select one or more:
   a. Unambiguous
   b. Free
   c. Restrictive
   d. Creative
   e. Ambiguous
   f. Formal

2. Which of these are methods to reduce the number of features that a text classifier might use?

   Select one or more:
   a. Use bigrams (two word sequences) as features instead of unigrams (single words).
   b. Use stems of words in place of the words.
   c. Use minimum document frequency to define the feature set.
   d. Parse the sentences in the text.
   e. Using maximum document frequency to define the feature set.
   f. Increase the size of the vocabulary.

3. Consider the training data below for a language model consisting of three sentences. Note the padding around the words- you should consider both the beginning and end of sentence markers as words. To obtain the counts on the sentences each one will be scanned through and each word counted as per the two functions C in the formula for a bigram model shown below. Remember that in the count in the denominator the word $\langle/s\rangle$ for the end of sentence will never be counted. Note that the size of vocabulary V will not include the beginning padding marker $\langle s\rangle$ but can contain the end of sentence marker $\langle/s\rangle$.

   After the counts have been collected, according to a Lidstone (add-k) smoothed bigram model where k = 0.5, what is the raw (i.e. not log) probability value for $p(\langle/s\rangle|Mohammed)$. Give your answer to 2 DECIMAL PLACES.

$$p(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + k}{C(w_{i-1}) + kV}$$

(note beginning (<s>) and end-of-sentence (</s>) markers, treat them as words):

<s> John likes Mary and Bill </s>
<s> Mary likes John and Mohammed </s>
<s> Mary and John like Mohammed and Bill </s>

4. In the below sentence showing the word/POS pair, select the appropriate Penn Treebank style POS tag in the missing position indicated by ?:

   Bill/NN wanted/VBD to/? go/VB home/NN.

   Select one:
   a. TO
   b. RB
   c. IN
   d. MD
   e. JJ

5. In this question refer to the Context Free Grammar lexicon and production rules below. According to these rules, which of the trees (a) or (b) is the correct syntax tree for the sentence “I want a non-stop trip to Chicago”?

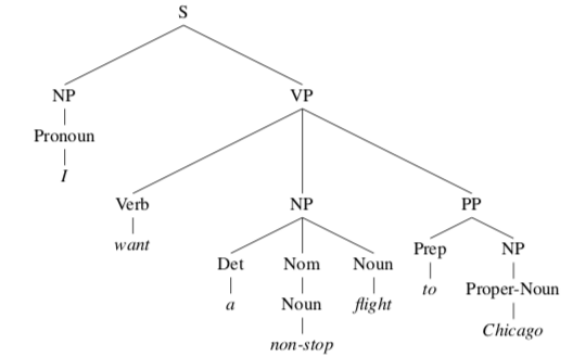
Lexicon:

Noun →	flight | flights | breeze | trip | morning
Verb →	is | prefer | like | need | want | fly | have
Adjective →	cheapest | non-stop | first | latest | other | direct
Prounoun →	me | I | you | it
Proper-Noun →	Alaska | Baltimore | Los Angeles | Chicago | United | American
Determiner →	the | a | an | this | these | that
Preposition →	from | to | on| near | in
Conjunction →	and | or | but

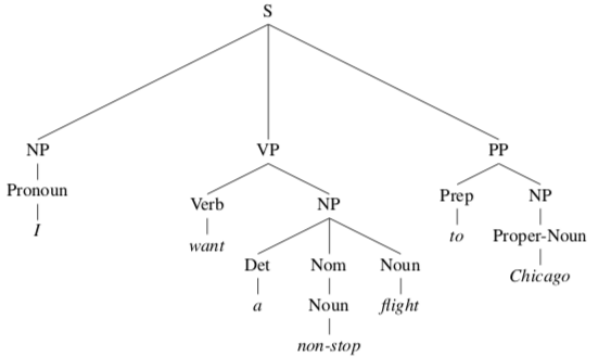
Production Rules:

Grammar Rules		Examples
S →	*NP VP*	I + want a morning flight
NP →	*Pronoun*	I
|	*Proper-Noun*	Los Angeles
|	*Det Nominal*	a + flight
Nominal →	*Nominal Noun*	morning + flight
|	*Noun*	flights
VP →	*Verb*	do
|	*Verb NP*	want + a flight
|	*Verb NP PP*	leave + Boston + in the morning
|	*Verb PP*	leaving + on Thursday
PP →	*Preposition NP*	from + Los Angeles

(a)



(b)



6. The creation of a treebank principally involves _____.

   Select one:
   a. parsing
   b. human expert annotation

7. Select which of the below is a semantic representation language.

   Select one or more:
   a. Abstract Meaning Representation
   b. First Order Logic
   c. Syntax
   d. Logical Grammar
   e. Frames
   f. Context Free Grammar

8. Select all of the following statements which are consistent with the distributional hypothesis of language meaning.

   Select one or more:
   a. Meaning is determined by a logical form attached to each word or constituent.
   b. The meaning of a sentence is determined by its denotation according to a formal model of the world.
   c. A word's meaning can be approximated by the distribution of the contexts it occurs in.
   d. Words that appear in the same contexts can be assumed to be synonymous.
   e. You shall know a word by the company it keeps.
   f. The syntax of a language determines the meaning of sentences.

9. Select all of the following boolean features which are often used to determine whether a pronoun is an anaphor/co-referent of another phrase which is its candidate antecedent in the text. The anaphor candidate:

   Select one or more:
   a. has the same grammatical role as the candidate antecedent.
   b. appears in a parallel grammatical structure to the candidate antecedent.
   c. has gender agreement with the candidate antecedent.
   d. has the same number of characters as the candidate antecedent.
   e. begins with the same character as the candidate antecedent.
   f. has the same number of words as the candidate antecedent.

10. A simple 3-node net with no bias nodes has two input nodes x1 and x2, which in practice receive either 0s or 1s for possible inputs, and two weights going to an output node Z with the ReLU activation: one weight w1 from node x1 to node Z with a weight of 2, and one weight w2 from node x2 to node Z with a weight of -1. The input to the net on a given example pair of inputs is x1=1 and x2=0 to the two input nodes. What is the activation value of node Z with these two inputs?

    Select one:
    a. 2
    b. 1

PART 2: Programming (worth 80% of grade)- Fake Review Classification

Background: Consumers tend to rely heavily on reviews when making decisions about what to buy online. For a company like Amazon, which depends on this process, it is therefore particularly important that these reviews can be trusted. Your task will therefore be to develop a method for automatically classifying Amazon reviews as *real* or *fake*, to explore how plausible it is to automate this task. You will be working with a recently released corpus of Amazon reviews which have been manually analysed and annotated by the company itself (see https://s3.amazonaws.com/amazon-reviews-pds/readme.html)[1]. Along with the review texts, which are labelled as either fake (_*label1*_) or real (_*label2*_), the data set contains a series of other features for each review (*rating*, *verified purchase*, *product category*, *product ID*, *product title*, *review title*). The corpus is made up of 21,000 reviews, equally distributed across product categories, which have been identified as 'non-compliant' with respect to Amazon policies.

In this coursework, you will implement a Support Vector Machine classifier (or SVM) that classifies the reviews as real or fake. You will use both the review text and the additional features contained in the data set to build and train the classifier on part of the data set. You will then test the accuracy of your classifier on an unseen portion of the corpus. Much of the background for this part is in **Unit 2 on Text Classification**, though you should use all of your knowledge across the module.

Instructions: Follow the below instructions, and submit WELL DOCUMENTED code as one or more IPython files (.ipynb) building on the template file NLP_Resit.ipynb as your starting point (Python 3.7+). No separate report is required. You have the data in the file amazon_reviews.txt. Ensure your code runs from top to bottom without errors before submission. If you do use more than one IPython file, it must be clear which file corresponds to which questions.

The template file contains some functions to load in the dataset, but there are some missing parts that you are going to fill in as per the questions below.

1. (10 points) Start by implementing the *parseReview* and the *preProcess* functions. Given a line of a tab-separated text file, *parseReview* should return a triple containing the identifier of the review (as an integer), the review text itself, and the label (either 'fake' or 'real'). The *preProcess* function should turn a review text (a string) into a list of tokens.

Hint: you can start by tokenising on white space; but you might want to think about some simple normalisation too.

2. (20 points) The next step is to implement the *toFeatureVector* function. Given a preprocessed review (that is, a list of tokens), it will return a Python dictionary that has as its keys the tokens, and as values the weight of those tokens in the preprocessed reviews. The weight could be simply the number of occurrences of a token in the preprocessed review, or it could give more weight to specific words. While building up this feature vector, you may want to incrementally build up a global *featureDict*, which should be a list or dictionary that keeps track of all the tokens in the whole review dataset. While a global feature dictionary is not strictly required for this coursework, it will help you understand which features (and how many!) you are using to train your classifier and can help understand possible performance issues you encounter on the way.

[1] See https://s3.amazonaws.com/amazon-reviews-pds/LICENSE.txt for the licensing information and terms and conditions for the use of the dataset.

Hint: start by using binary feature values; 1 if the feature is present, 0 if it's not.

3. (20 points) Using the *loadData* function already present in the template file, you are now ready to process the review data from *amazon_reviews.txt*. In order to train a good classifier, finish the implementation of the *crossValidate* function to do a 10-fold cross validation on the training data. Make use of the given functions *trainClassifier* and *predictLabels* to do the cross-validation. Make sure that your program stores the (average) precision, recall, f1 score, and accuracy of your classifier in a variable *cv_results*.

Hint: the package sklearn.metrics contains many utilities for evaluation metrics - you could try precision, recall, fscore, support to start with.

4. (15 points) Now that you have the numbers for accuracy of your classifier, think of ways to improve this score. Things to consider:

- Improve the preprocessing. Which tokens might you want to throw out or preserve?
- What about punctuation? Do not forget normalisation and lemmatising - what aspects of this might be useful?
- Think about the features: what could you use other than unigram tokens from the review texts? It may be useful to look beyond single words to combinations of words or characters. Also the feature weighting scheme: what could you do other than using binary values?
- You could consider playing with the parameters of the SVM (cost parameter? per-class weighting?)

Report what methods you tried and what the effect was on the classifier performance in the notebook.

5. (15 points) Now look beyond textual features of the review. The data set contains a number of other features for each review (*rating, verified purchase, product category, product ID, product title, review title*). How can the inclusion of these features improve your classifier's performance? Pick three of these metadata types to use as additional features and report in the notebook how they improve the classifier performance.