The University of Texas at Dallas
CS 6320
Natural Language Processing
Spring 2020
Instructor: Dr. Sanda Harabagiu
Grader/ Teaching Assistant: Ramon Maldonado

Homework 1: 100 points (50 points extra-credit)
Issued February 5, 2020
Due March 9, 2020 before midnight

PROBLEM 1: Regular Expressions (**5 points**)
Write regular expressions for the following three languages. By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

1. Language 1: the set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug"); (1 point)
2. Language 2: all strings that start at the beginning of the line with an integer and that end at the end of the line with a word; (2 points)
3. Language 3: all strings that have both the word "grotto" and the word "raven" in them (but not, e.g., words like grottos that merely contain the word grotto); (2 points)

PROBLEM 2: N-Grams (**40 points**)
Compute the probability of the following two sentences:
S1: Sales of the company to return to normalcy.
S2: The new products and services contributed to increase revenue.

Using the bigram language model trained on the corpus that is provided, find out which of the two sentences is more probable. Compute the probability of each of the two sentences under the following two scenarios:
a) Use the bigram model without smoothing.
b) Use the bigram model with add-one (Laplace) smoothing.

Programming Assignment for Problem 2:
1. Write a program to compute the bigrams for any given input. **TOTAL: 5 points**

Apply your program to compute the bigrams you need for sentences S1 and S2.

2. Construct automatically (by the program) the tables with (a) the bigram counts (5 points) and the (b) bigram probabilities for the language model without smoothing. (5 points) **TOTAL: 10 points**

3. Construct automatically (by the program): (i) the Laplace-smoothed count tables; (5 points) (ii) the Laplace-smoothed probability tables (5 points); and (iii) the corresponding re-constituted counts (5 points) **TOTAL: 15 points**

4. Compute the total probabilities for each sentence S1 and S2, when (a) using the bigram model without smoothing; (5 points) and (b) when using the bigram model Laplace-smoothed. (5 points) **TOTAL: 10 points**

PROBLEM 3: Vector Semantics (**25 points**)

1. Considering the same corpus as in Problem 2, write a program to compute the *Positive Pointwise Mutual Information (PPMI)* of the following words and present the results in a term-context matrix. The context of a *context-word* is the "window" of words consisting of (i) 5 words to the left of the *context-word;* (ii) the *context-word*; and (iii) 5 words to the right of the *context-word*. IF there are fewer then 5 words to the right or the left of the *context-word* in the same sentence, the context will be padded with "NIL". Compute the PPMI and populate the term-context matrix for:
    a. The word "chairman" for the *context-word* "said";
    b. The word "chairman" in the *context-word* "of";
    c. The word "company" in the *context-word* "board";
    d. The word "company" in the *context-word* "said".

    TOTAL: 10 points
2. Use add-2 smoothing for the same words and contexts as in (1) and present the result in a term-context matrix. **TOTAL: 5 points**
3. Find which words are more similar among: [chairman, company], [company, sales] or [company, economy] when considering only the contexts provided by the *context-words* "said", "of", and "board"? Explain why. **TOTAL: 5 points**
4. Using the pre-trained Glove embeddings available at:
   https://nlp.stanford.edu/projects/glove/
   which words are more similar among: [chairman, company], [company, sales] or [company, economy]? Explain why. **TOTAL: 5 points**

PROBLEM 4: Part-of-speech tagging (**30 points**)

Use the Viterbi algorithm to assign POS tags to the following two sentences:

S1: The chairman of the board is completely bold.

S2: A chair was found in the middle of the road.

Use the following tag transition probability table A and observation likelihood array B. Both tables use the Penn Treebank POS tags.

A=

	DT	NN	VB	VBZ	VBN	JJ	RB	IN	</s>
<s>	0.38	0.32	0.05	0	0	0.11	0.1	0.23	0
DT	0	0.58	0	0	0	0.42	0	0	0
NN	0	0.12	0	0.05	0.32	0	0	0.25	0.11
VB	0.01	0.05	0	0	0	0	0.2	0.61	0.13
VBZ	0.2	0.3	0	0	0	0.25	0.15	0.1	0
VBN	0.18	0.22	0	0	0.2	0.07	0.16	0.11	0.06
JJ	0	0.85	0	0	0	0.12	0	0	0.03
RB	0	0	0	0.22	0.28	0.39	0.1	0	0.01
IN	0.57	0.28	0	0	0	0.15	0	0	0

B=

	a	the	chair	chairman	board	road	is	was	found	middle	bold	completely	in	of
DT	1	1	0	0	0	0	0	0	0	0	0	0	0	0
NN	0	0	0.69	1	0.88	1	0	0	0.01	0.66	0.38	0	0	0
VB	0	0	0.31	0	0.12	0	0	0	0	0	0	0	0	0
VBZ	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VBN	0	0	0	0	0	0	0	1	0.99	0	0	0	0	0
JJ	0	0	0	0	0	0	0	0	0	0.34	0.62	0	0	0
RB	0	0	0	0	0	0	0	0	0	0	0	1	0	0
IN	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Assignment for Problem 4:

1. Create the Hidden Markov Model (HMM) and show (a) the transition probabilities and (b) observation likelihoods in each state that will be reached by sentences S1 and S2 after 3 time-steps. Present only the transition and observation likelihoods in the states reached after three steps. **TOTAL: 5 points**
2. Create the Viterbi table for each sentence and populate it entirely. **TOTAL: 10 points**
3. What is the probability of assigning the tag sequence for each of the sentences.? **TOTAL: 10 points**
4. Execute the Stanford POS-tagger (available from: *https://nlp.stanford.edu/static/software/tagger.shtml*) on both sentences. Which POS tags were assigned by the Stanford POS-tagger more accurately? Explain. **TOTAL: 5 points**

Software Engineering (includes documentation for your programming assignments)

Your README file must include the following:

- Your name and email address.
- *Homework number* for this class (NLP CS6320), and the *number of the problem* it solves.
- A description of every file for your solution, the programming language used, supporting files, any NLP tools used, etc.
- How your code operates, in detail.
- A description of special features (or limitations) of your code.

Within Code Documentation:

- Methods/functions/procedures should be documented in a meaningful way. This can mean expressive function/variable names as well as explicit documentation.
- Informative method/procedure/function/variable names.
- Efficient implementation
- Don't hardcode variable values, etc

EXTRA-CREDIT PROBLEM 1 (20 points):

The main goal of the extra-credit problem 1 is to enable you to produce a simple question answering (QA) system using regular expressions to retrieve information from a single google news article which is provided to you.

Your code should be able to respond to a list of queries that you enter from a file, where each query is on a separate line. It should also be able to respond to a single query from interactive console input. When responding to queries from a file, your QA system should respond to each line, and terminate; when responding to console input, it should loop until the user quits, by entering the word "FINISH".

Your system should be able to correctly answer the following questions:

1. Who is the world's most famous pharaoh?
2. What was the cost for visiting the tomb of King Tut?
3. How old is King Tut's crypt?
4. When was the crypt sealed?
5. What do the paintings show?
6. Where did the fungus grow?
7. What reduces moisture and dust?
8. What happened to King Tut's tomb during most of the repairs?
9. What delayed the work on King Tut's tomb?
10. What do Egyptian officials hope?

Regular Expression templates (10 points)

You should have good regular expression templates for the questions and answers. Quality is more important than quantity. More general regular expressions, that can match multiple questions or multiple kinds of sentences for the answer, are better than rigid regular expressions that match only one string. You need to provide the code and README file, as specified in the "software engineering" section above.

Preciseness of answer (10 points)

Your answer should be as specific as possible. For example, for the first question above, the answer should be **"King Tut"** not **"Pharaoh, King Tut"**. You will obtain a point for each correctly answered question.

EXTRA-CREDIT PROBLEM 2 (30 points):

The main goal of the extra-credit problem 2 is to enable you to learn a neural language model on Google cloud. You should visit:
https://github.com/r-mal/utd-nlp/blob/master/neural_language_modeling_glove.ipynb
To open the notebook on Google's cloud, the students can just click the blue 'Open in Colab' button at the top of the webpage.

- Where we have prepared for you a framework for a simple feed-forward neural language model. You are provided with the Reuters newswire corpus that contains the text of 11,228 newswires from Reuters. These are split into 8,982 newswires for training and 2246 newswires for testing.
- You are instructed how to prepare your data, download the embeddings and build the neural model. You are asked to train and test the neural model as a feedforward network with two intermediate or "hidden" layers, between the input and output (**10 points**), which is provided, as well as with one hidden layer (**10 points**) and three hidden layers (**10 points**). This will enable you to use the sparse categorical cross entropy loss function, which is provided. To obtain full credit for each model, you are requested to (1) generate a validation set, (2) train and evaluate the model; (3) create a graph indicating the change of accuracy and loss of the model over time and (4) provide the perplexity values of the model.