PROBLEM 1: Language Models (**60 points**)
A] 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 trigram 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 trigram model without smoothing.
b) Use the trigram model with add-one (Laplace) smoothing.
c) Use the trigram model with Katz back-off smoothing.

Programming Assignment for Problem 1.A: (30 points)
1. Write a program to compute the trigrams for any given input. **TOTAL: 2 points**
   Apply your program to compute the trigrams you need for sentences S1 and S2.
2. Construct automatically (by the program) the tables with (a) the trigram counts (2 points) and the (b) trigram probabilities for the language model without smoothing. (3 points) **TOTAL: 5 points**
3. Construct automatically (by the program): (i) the Laplace-smoothed count tables; (2 points) (ii) the Laplace-smoothed probability tables (3 points); and (iii) the corresponding re-constituted counts (3 points) **TOTAL: 8 points**
4. Construct automatically (by the program) the smoothed trigram probabilities using the Katz back-off method. (8 points) How many times you had to also compute the smoothed trigram probabilities (2 points) and how many times you had to compute the smoothed unigram probabilities (2 points). **TOTAL: 12 points**

5. Compute the total probabilities for each sentence S1 and S2, when (a) using the trigram model without smoothing; (1 points) and (b) when using the trigram model Laplace-smoothed (1 points), as well when using the trigram probabilities resulting from the Katz back-off smoothing (1 points). **TOTAL: 3 points**

B] Neural Language Models (30 points): The main goal of the problem 1.B 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.

PROBLEM 2: Vector Semantics (**10 points**)
1. Considering the same corpus as in Problem 1, write a program to compute the *Positive Pointwise Mutual Information (PPMI)* of pairs [word, *context-word*]. The context of a *word* is the "window" of words consisting of (i) 5 words to the left of the *word;* and (ii) 5 words to the right of the *word*. If there are fewer then 5 words to the right or the left of the *word* in the same sentence, the context will be padded with "NIL". Compute the PPMI 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: 8 points
2. Find which words are more similar among: [chairman, company], [company, sales] or [company, economy] when considering only the contexts which contain *words* "said", "of", and "board"? Explain why. **TOTAL: 2 points**

PROBLEM 3: Part-of-speech tagging (**20 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	0.04	0	0	0.11	0.01	0.14	0
DT	0	0.58	0	0	0	0.42	0	0	0
NN	0	0.07	0	0.05	0.32	0	0	0.25	0.11
VB	0.07	0.08	0	0	0	0	0.2	0.61	0.13
VBZ	0.2	0.3	0	0	0	0.24	0.15	0.11	0
VBN	0.18	0.22	0	0	0.2	0.07	0.16	0.11	0.06
JJ	0	0.88	0	0	0	0.12	0	0	0
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 3:
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. (5 points)
2. Create the Viterbi table for each sentence and populate it entirely. (5 points)
3. What is the probability of assigning the tag sequence for each of the sentences
   S1 and S2? (5 points)

4. Execute the Stanford POS-tagger (available from:
   https://nlp.stanford.edu/software/tagger.shtml
   on both sentences. Which POS tags were assigned by the Stanford POS-tagger
   more accurately? Explain. (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
- Do not hardcode variable values, etc