CSE 556: Natural Language Processing **(Due: 10/09/19 11:59 PM)**

Assignment #2

Assignment Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- Python is the only allowed programming language for the assignment.
- A seperate PDF needs to be submitted where all the answers to the questions should be mentioned.
- **Report a sample answer using your own custom input for each task.**
- Any assumptions taken should also be mentioned in that PDF.
- All log values need to be taken to the **base 10**.
- NLTK is the only allowed inbuilt library for the assignment and it can only be used for preprocessing steps. All other functions need to be implemented from scratch.
- Make sure to use **Pickle** or any other library to save all your trained models. There will not be enough time during the demo to retrain your model. This is a strict requirement. Use this link to understand more about how to use Pickle.
- Plagiarism Check will be performed on all codes. Please cite any sources if used.
- Upload a zip file consisting of all the code files, pickled files and a report in the format: **Assignment2_Rollno.zip**

Problem 1: Multinomial Naive Bayes Model

(10+15+4=29 points)

Implement a Multinomial Naive Bayes Model from scratch for the 20 newsgroup dataset. For **task-1** perform the 2-class classfication on the classes **rec.motorcycles** and **rec.sport.baseball**. For **task-2** perform the multiclass classification over all the **20 classes** in the dataset. Perform the steps in the following order and save the models for each in a seperate pickle file. **During the demo a user input test case will be provided for each part below**. **For each test case, you must print the value of the log(probability) of each class and the name of the class with the highest log(probability)**. (You can remove the headers from all files during preprocessing). (You might also want to do stop word removal as a preprocessing step if it improves results. Mention it in your pdf.)

1. Using add-1 smoothing to handle OOV words, implement both the tasks mentioned above.
2. Use add-k smoothing instead to handle OOV words, and implement both the tasks mentioned above (k=5, 10, 100).
3. What differences do you see in the two techniques above? Which one should produce better results? What happens on increasing the value of k?

Problem 2: Language Modelling

(3+6+6+6=21 points)

Task-1: Train a unigram, bigram, trigram model using all files of **rec.sport.baseball** folder.
Task-2: Train a unigram, bigram, trigram model using all files of **rec.motorcycles** folder.

(You can remove the headers from all files during preprocessing)

1. Using **LaPlace Smoothing**, report one sentence generated from each category along with the log(probability) value of the sentence. (You should report 6 sentences in this case). **The returned sentence should have a length >5 words and <10 words and should have maximum log(probability)** for the specified conditions.

2. Using **LaPlace Smoothing**, for each of tasks and models above, given a sentence as input, output the **log(probability)** of the sentence.
3. Using **LaPlace Smoothing**, for each of tasks and models above, given a sentence as input, output the **perplexity** of the sentence.
4. Using any **advanced smoothing** technique of your choice, for each of tasks and models above, given a sentence as input, output the **log(probability)** of the sentence. (Name the technique used in the report)