**Homework #2: Exploring Language Modeling (Statistical) and Spell Checking**

**1. Overview**

Your task in this assignment is first to explore how you can build your own language model using data that is publicly available, and also evaluate the performance of your work on some test sets. Then you will build your own Spell Checker for misspelled words only, i.e., errors that that are not in the dictionary, using the Noisy Channel approach.

**2. Explore Language Models: Build your own language model.**

The appended link is your starting point. It gives you all the information (data and code!!!) to build your own language model using Reuters data set which is rather small. I have uploaded the dataset split into training and testing data. (If you wish to build a realistic language model try this collection from the Gutenberg project <http://norvig.com/big.txt>. )

Starting point for your exploration are the links below. They also have a smaller version of the data so that you can quickly experiment on.

<https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-language-model-nlp-python-code/> (the code is courtesy of <https://nlpforhackers.io/language-models/> )

Your assignment is to understand the steps and the code provided and then to build a language model using the uploaded Reuters dataset and also evaluate it and report its perplexity for the different cases of smoothing that you will do. You will need to modify the way you input and process the files because they are not provided in the same format as the examples on the links above. The ideal would be for you to build a Trigram language model. However, if this is taking too much time to build and you encounter problems with the size of the data set, you can settle for a bigram language model.

In particular, you may also use the python sample codes uploaded in the assignment folder as scaffolding for your own implementation of:

1. Laplace Unigram Language Model
2. Laplace Bigram Language Model
3. Stupid Backoff Language Model

**3. Spell Checker for Misspelled Words**

Using the language model you created or any other publicly available (like the Google language model), create your own spell checker for misspelled words. Use the Noisy Channel Approach and propose candidates for correction by ranking them both based on Edit Distance and on the probability that the language model gives you.

You can use as scaffolding the uploaded python code for spell check, sentence and edit model.