DATA130030.01: Natural Language Processing **03/23/2022**

Lecture 4: Assignment-01

Lecturer: Baojian Zhou *The School of Data Science, Fudan University*

In this assignment, you will explore a wiki-text data and build language models and classifiers. The corpus is downloaded and extracted from wiki dumps, `https://dumps.wikimedia.org/enwiki/20220201/`. You have the following tasks

- Task-0. Download the preprocessed data from `https://pan.baidu.com/s/1l2btCoYUN6QCfJZkqra3Ag?pwd=coxe` and the example code `https://pan.baidu.com/s/1fjupt1YSewRE5iKEgzsxCA?pwd=h6w4`. In the data file (json format), each line represents a Wikipedia page with attributes, "title", "label", and "text". There are 10,000 records in total with 10 categories. You can use the example code to load these records.
- Task-1. Data exploring and preprocessing: 1) Print out how many documents in each of these classes; 2) Print out the average number of sentences in each class. (You may need to use sentence tokenization of nltk.); 3) Print out the average number of tokens in each class (You may need to use word tokenization of nltk.); and 4) For each of sentence in the document, remove punctuations and other special characters so that each sentence only contains English words and numbers. To make your life easier, you can make all words as lower cases.
- Task-2. Build language models: 1) Based on processed text, build unigram, bigram, and trigram language models using Add-one smoothing and Kneser-Ney smoothing on training data set; 2) Report the perplexity of these six trained models on testing text set and explain these numbers. 3) Use each of built model to generate 5 sentences and explain these generated sentences.

  For this task, you can randomly shuffle data set and use 90% (9,000 records) sentences as training and 10% (1,000 records) as testing. You may need to take a look at related tools `https://www.nltk.org/api/nltk.lm.html`. But it is encouraged to implement models by yourself. For Kneser-Ney smoothing, you can choose the discount parameter as $d = 0.1$.
- Task-3. Build Naive Bayes classifiers: 1) Build Naive Bayes classifiers (with Laplace smoothing) by using 30%,50%,70%, and 90% of documents as training samples and 10% for testing dataset (fixed), respectively. 2) Report Micro-F1 score and Macro-F1 score for these classifiers; explain and analysis our result.

Code format: It is encouraged to use Jupyter Notebook (`https://jupyter.org/`). You do not have to write a report just explain your results in comment sections.

Submission deadline: **6:00pm, 04/06/2022**.