CS565: Intelligent Systems and Interfaces
Assignment 1
Topics: Tokenization, N-Grams, and Morphological Analysis
Marks: 50 **Due Date: September 20, 2021**

1 Assignment

1.1 Instructions

- Each student will require to submit the assignment individually. No group submission will be allowed.
- Each student will submit a report. The report will include brief write up of the analysis performed and its results along with the relevant plots.
- You are expected to write the code in Google Colab Notebooks and include the notebook link in your report. The code must reproduce the results discussed in your report. Add appropriate comments and instructions in the code wherever required.

1.2 Introduction

An *n-gram* is a sequence of n items from a given sequence of word or text. The items can be letters, words or akshara according to the application, for our case it would be words only. For instance let's say corpus is *"His relationship with many western nations was troubled during his tenure as chief minister..."* The list of uni-grams would be *{ His, relationship, with, many, western, ...}*, bi-grams would be *{ His relationship, relationship with, with many, ...}* and similarly tri-grams would be *{ His relationship with, relationship with many, with many western, ... }*. The given example of *n-grams* is contiguous, whereas we can also create a list of non-contiguous *n-grams*. While non-contiguous *n-grams* can be useful for finding **collocations**, contiguous *n-grams* analysis is more common for downstream applications such as language modeling, machine translation, text categorization etc.
The objective of this assignment is to get started with basic NLP tasks, exploring available tools and choosing the one which you like the most. Some of the famous tools are NLTK, spaCy, Apache OpenNLP, Stanford CoreNLP and AllenNLP.

1.3 Tasks

1.3.1 Analysis using existing NLP tools [10 marks]

Download a subset of English and Hindi Wikipedia available at the following links: [English] and [Hindi]. You need to complete the assignment using the two text corpora given above.

1. Perform sentence segmentation and word tokenization on the downloaded corpus. For both the tasks, try to explore at least two different methods across the tools and compare the results.

2. Find all possible uni-grams, calculate their frequencies and plot the frequency distribution.

3. Find all possible bi-grams, calculate their frequencies and plot the frequency distribution.

4. Similarly, find all possible tri-grams, calculate their frequencies and plot the frequency distribution.

You may explore a few tools (e.g. NLTK, Apache OpenNLP, AllenNLP, CoreNLP etc.) first and choose one to do the assigned tasks. In the report

- summarize the options available for sentence segmentation as well as for word tokenization.
- Discuss if you see any difference in the results obtained by the chosen different methods for both the tasks.
- You should also discuss the frequency distribution of the three different cases. What are your observations regarding most frequent words and least frequent words? Can you characterize them or you find mixed kind of words are present in the two categories (most and least frequent). Specifically, check if you can fit Zipf's law equation in the distribution. Describe the parameters after curve fitting.

1.3.2 Few Basic Questions [5 marks]

Often we do not consider all **types** identified in the corpus to reduce the vocabulary size and also take care of the **out of vocabulary (OOV)** or **rare** words. In one of the such methods, we consider **tokens** contributing to most frequent **types**. Threshold is determined using frequency analysis. Here, the coverage implies, which **types** should be considered as part of the vocabulary that it covers 90% of the tokens in the given corpus.

1. How many (most frequent) uni-grams are required for 90% coverage of the selected corpus?

2. How many (most frequent) bi-grams are required for 80% coverage of the corpus?

3. How many (most frequent) tri-grams are required for 70% coverage of the corpus?

4. In your report, include relevant plots and discuss your observation. You should also discuss if there are certain frequent patterns observed in the left out tokens. If yes, how many groups you would like to make based on the distinct patterns.

1.3.3 Writing some of your basic codes and comparing with results obtained using tools [10 marks]

1. Repeat section 1.3.2 after implementing discussed heuristics in the class for sentence segmentation and word tokenization. If you want to improvise on the discussed heuristics, you can do that but you should describe your heuristics in the report. Also, summarize your findings by comparing the results obtained using your heuristics and tools.

1.3.4 Morphological parsing [5 marks]

1. Perform a morphological analysis of 5 words randomly sampled from 100 frequent words and 5 words randomly sampled from 100 least frequent words.

2. For this analysis, use any available morphological analyzer with the tool. In the report along with the analysis describe the model of the used morphological analyzer.

1.3.5 Sub-word tokenization [20 marks]

Sub-word tokenization helps to deal with unknown words or missing words without requiring infinite vocabulary. This tokenization scheme uses simple words in the vocabulary to create compound words. **Byte-pair-encoding(BPE)**[1] is one of the popular algorithm for sub-word tokenization. In this task, you are expected to write your own code of BPE algorithm for sub-word tokenization. **Do not use libraries.** Set desired vocabulary size and learn tokenization for the two corpora. Report 50 most frequent and least frequent tokens from the two corpora.

Create a list of 10 words in both English and Hindi that are not present in the given corpora. Find tokenization for them.

Further compare BPE tokenization of the 10 words selected during Morphological analysis with their morphemes.

[1] https://www.aclweb.org/anthology/P16-1162.pdf