CS 565: Assignment-1

Text-processing, N-Gram, Collocation and Morphological analysis

Due Date: 10^{th} **February,** 2019

- Your assignment and project group remain the same.
- Each group will submit a report and code. The report will include a brief write up of the analysis performed and its results along with the relevant plots. The code will include the complete code that was used to produce the results. In the report also mention individual contributions.
- If there is any query related to assignments, please post/message via Canvas only.
- Assignment submission guideline will be provided by Feb 7, 2019.

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 lets say corpus is *"His relationship with many western nations was troubled during his tenure as chief minister..."* then 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* as we discussed in the class. While non-contiguous *n-grams* can be useful for finding collocations, contiguous *n-gram* 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, Apache OpenNLP, Stanford CoreNLP and AllenNLP.

1 Analysis using existing NLP tools

Download a subset of English Wikipedia available at one of the following links: (0), (1), (2), (3). You need to complete the assignment only using one of the above text corpora. To determine the corpus to use, add the roll numbers of your group members and perform a modulus 4 operation on the sum.

1. Perform sentence segmentation and word tokenization on the downloaded corpus.
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.

Each group is expected to explore two tools. The tools can be any publicly available open source software (e.g. NLTK, Apache OpenNLP, AllenNLP, CoreNLP etc.).

In the report, summarize the options available for sentence segmentation as well as for word tokenization. In your report, you should also discuss the frequency distribution of the three different cases. Specifically, check if you can fit Zipf's law equation in the distribution. Describe the parameters after curve fitting.

2 Few Basic Questions

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. Repeat the above after performing lemmatization.
5. Compare the statistics of the two cases, with and without lemmatization.
6. Summarize the results in the report.

3 Writing some of your basic codes and comparing with results obtained using tools

1. Repeat section 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.
2. Implement the Likelihood Ratio Test for finding all bi-gram (contiguous) collocations in the corpus[1]. Do not use libraries. In your report, summarize the method and discuss if you have made any interesting observation.

4 Morphological parsing

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] If there are memory related issues while processing the provided corpus, consider using one-fourth of the corpus.