CSF429 Projects List

- Project selection will be done in FCFS manner.
- Each project has two parts/tasks. However, second part will be available after midsem exams only if the first part is completed.
- Each project must be selected by at least one group.
- Project cannot be changed after September 15th (once checked/approved by IC).

1. Grammar Check for English Language (**maximum two groups**)
   (Preprocessing, Parts of Speech, Dependency Parsing, Language Modelling)

Part 1: Build a grammar checker using the concepts taught in class, you will be given a corpus with some paragraphs labeled correct and unknown; correct paragraphs have no grammatical errors, while unknown may/may not have sentences which have grammatical errors in them. The task is to find all the sentences that have these errors. For example: "In the country there lived a fox. The quick brown fox jump over the fence. Farmer Shaun was terrified." In this paragraph, the sentence "The quick brown fox jump over the fence" is grammatically incorrect.

2. Literature Shelves and Relations (**maximum three groups. Datasets may be different**)
   (Preprocessing, Topic Modelling, Distributional Semantics, Information Extraction)

Part 1: Assume that you were given 10,000 Research papers to read in a bundle, completely un-assorted! Your first job would be to assort them in some hierarchical order, wherein these papers were separated in different sections/shelves. You are given some Computer Science Research papers with their metadata including metadata, The task is to assort them in shelves and sub-shelves. For example: You are given papers A,B,Z,1,2,-1,@,# At the level 1 the clusters could distinct based on character types, (A,B,Z), (1,2-1), (@,#), Further the cluster (1,2,-1) can be divided into ((1,2), (-1)) and so on.

3. Syntax Analysis in Source Code (**maximum two groups**)
   (Parsing, Language Models, N-grams)

Part 1: All programming languages have a distinct grammar that has to be adhered to for compilation and execution. This arrangement is then used by the parser to compile a program.
IF '(' expression ')' statement _ ELSE statement
statement _ ELSE statement
For the above 2 statements the 2nd line is syntactically incorrect because of the absence of a preceding 'if' statement. Design a parser which is able to determine if a code snippet is syntactically correct.

4. Stylometric Analysis (**maximum two groups**)
   (Preprocessing, Dependency parsing, Lexical and syntactic analysis.)

Part 1: Analyze the writing style of different authors, you will be given a corpus of books written by different authors, the idea is that different authors write in different manners. Using surface level, lexical, and syntactic features analyze the writing styles of different authors.

5. Aspect based sentiment analysis (**maximum two groups**)
   (Parsing, Information Extraction)

Part 1: Sentiment analysis is a widely used application of NLP. However pure classification still suffers from a lack of context and reasons for those particular sentiments. For example, a product may have a bad review because of the stock availability issues, but still, be a good product. The task is to analyze these sentiments based on certain aspects. Perform base sentiment analysis on sentences and look at the results, investigate the pitfalls of this format of classification.

6. Cross POS (**maximum three groups**)
   (Morphology/Lexical Analysis, POS tagging)

Part 1: POS tagging is based on the grammar, semantic and syntactic structure of the language. Some languages have higher similarities in terms of semantic and lexical similarity. Compare the performance of semantic rule-based, HMM and RNN models for POS tagging on any dataset

7. Query Formation (**maximum two groups**)
   (language models, preprocessing)

Part 1: Automatically correcting the queries by processing syntactic and semantic features of the words present in the query. Improving this correction by validating the search results.