Project 1 for CS421 – University of Illinois at Chicago

Mohit Haresh Adwani: madwan2@uic.edu

Shobhit Lamba: slamba4@uic.edu

---------------------------------------------------------Setup------------------------------------------------------------

Alternatively, you can give the following command (in either Windows or Linux):

Please install required libraries using the following commands:

*pip install nltk*

*pip install stanfordcorenlp*

The program requires Stanford coreNLP server running on port 9000. If your version of Stanford coreNLP doesn’t work properly. Please use version 3.8.0 from [here.](https://stanfordnlp.github.io/CoreNLP/history.html)

In the *execution* folder, give the following command on your command prompt:

*python main.py*

Since Stanford coreNLP is used, and the processing is done on the training data, the program takes significant amount of time to run. Because of this, we have included console print statement after it completes it’s processing on each essay.

--------------------------------------------------------Technique--------------------------------------------------------

1. Length of Essay:

* First, we get the sentences by applying nltk sent\_tokenize on the essay. Then we split all the sentences which has newline character (\n) into multiple sentences.
* We saw a pattern in the essays in which there wasn’t a space after period (which denotes the end of the sentence) and hence the sent\_tokenize didn’t split the sentence into two sentences. Hence, we split the sentences into two if the character after period is alpha and before the period isn’t period(.) since some essays had two or more consecutive periods to denote continuation. Also the sentence after the period should at least have 3 characters.
* While looking at multiple finite verbs in the sentence(hint given in project\_part1.pdf), if the sentence didn’t have coordinate or subordinate clause then we saw a pattern in the parse tree, denoting the finite verb phrase as ‘(SBAR (S’ – where SBAR denotes ‘clause introduced by a (possibly empty) subordinating conjunction’. If the sentence has a subordinating conjunction, then it is denoted by ‘(SBAR (IN that) (S’
* We noticed that splitting the sentence based on capitalization doesn’t work well and there are very few sentences in the whole training data which can be split using capitalization, hence this method wasn’t implemented.

Patterns of Errors:

* The period processing steps will mistakenly process the sentences which have abbreviation into multiple sentences. But since there just 2-3 instances of this in our training data, we ignored it.

1. Spelling Mistakes

* FILL THIS

1. Syntax/Grammar
2. Subject-Verb Agreement

* FILL this

1. Verb Tense

* A list of four grams, tri grams, bi grams and uni grams POS tags were created based on the English verb formation rules.
* The program extracts the longest sequence of verb forming tags and checks whether it is present it out lists. If not then it is counted as an error.

Patterns of Errors:

* Since the nltk POS tagger sometimes doesn’t tag the verbs correctly, some verb formations, even though correct are tagged as incorrect.
* Since the verb formation rule set isn’t exhaustive, some verb formations even though correct are tagged as incorrect. Ex: ‘’

A brief explanation of how you exploited POS tagging (and any other strategies used) to evaluate the essays. Also state some patterns of errors that you found.