Test Accuracy: 94%

1. We learned the way parse trees helps to know if the sentence is a complete or incomplete. We explored POS taggers like NLTK POS tagger to know the order for identifying incorrect verb usage and/or pronoun usage. Developing a NLP system with many features was not a trivial task. We came across all sorts of ambiguities and had to deal with each one differently. There occurred rare cases, and issues while trying to capture unseen problems behind a body of text.

2. There are many things that worked and were a bit more straightforward. Getting the sentence length worked fairly well, since we separated the sentences using nltk's sentence tokenizer. Then the sentence was split with "," with respect to constraint of minimum 5 words. We had checked for various length and 5 was the best fit .

3. For getting the spelling mistakes , we counted the number of words that do not appear in the enchant dictionary of "en_US" and "en_UK". Also, since we observed that enchant outputs some correct spelled words as incorrect, we used GloVe to be doubly sure about the spell check of the words in the training corpus. For example, if there are certain words that were marked incorrect by enchant and correct by GloVe, then they were stored in the separate list so as to compare these words with the words that occur later in the essays of test corpus. If yes (i.e they occur in the test corpus), then they won't be considered as incorrect. This helped us to prevent from the detection of incorrect spelling mistakes. Few other words like e.g. Mrs., Mr. etc. are correct and shouldn't be marked as errors, thus we have separately handled the cases where if either of such words occur, they are left untouched. *Using GloVe ensures correct spelling mistakes to be encountered and not incorrect ones.*

   We also gathered all possible lists of ProperNouns such as Countries, Cities, Brands, Companies, Languages, Nationalities, Days and Months. For future, we can add the lists of other noun forms so that it cannot misclassify the proper spelling.

4. We faced problems to calculate the sub score for subject - verb agreement. We used the parse tree given by Stanford Core NLP. Parse Tree helped us to map a list of all violating rules after traversing the tree to the pairs of incorrect POS tags in each sentence. We then traversed the parse tree to find the head noun and its

corresponding verb to check if they disagree with the rules mentioned for subject and verb agreement.*The most important feature of using a ParseTree for this part is that it not only find the set of subject and the verb that are adjacent to each other (and violates the rule) but also those that are separated by many POS tags and violates the subject-verb agreement policy. This increases the chances of finding errors that are hard to catch otherwise.*

5. For verb agreement, we used POS tagging technique to create bigrams and trigrams of only those words that matches the Verb list (verb list includes POS tags like “VB”, “VBZhas", “VBZis", “MD", “VBD", “VBG", “VBN” and “VBP"). The resulted chunks of bigrams and trigrams are checked against the rules (hand crafted verb disagreement criteria). If either of the two (bigrams or trigrams) matches any rule defined for verb disagreement then, that accounts for an error. Also, we have checked for the verb tense agreement by creating chunks of the sentences with each chunk having a verb associated with it. We created set of rules that check for the chunks that have occurrences of improper tense. Looking at agreement between verb usage with number as well as tense, worked fairly well in capturing a lot of the basic verb usage errors, but for some more complicated structures it can be difficult to just use a POS tagger to look for errors.

6. For Sentence Formation error, parse trees were exploited. Parse tree was seen for many examples to find pattern of errors in the sentence. It was found that trees where FRAG was child of non "S" or "SBAR" node, it was an error. Also "SBAR" where its parent was not NP, VP and S was seen as an error. So the parse tree was recursively traversed to find such FRAG and SBAR to get the errors.

7. For Text Coherence, we collected all pronouns and possessive objective and eliminated the ones that are not third person. Then, for every singular third person pronoun and possessives, we checked if there exists a male or female antecedents. If not, it was counted as an error. And finally, for plural third person pronouns and possessives we checked for one case of checking an antecedent where if there is no plural antecedent, or no singular antecedent with compatible number then it was counted as an error.

8. For Topic Coherence**,** we used an electronic dictionary called “WordNet” . We created the ‘synsets’ of each token in the essay and the words in corresponding topic. For each ‘synset’ for an essay word and a topic word, a similarity score is

calculated. Then for that particular pair of words, maximum similarity score (depends on maximum similarity in the meaning of the words) is calculated and added to the total score for that essay. In this way, we get the maximum score for the essay that have achieved the objective of addressing the actual topic of the essay. Once the similarity score for each essay is retrieved, the score is assigned on the basis of different thresholds. But it is very hard to capture different styles of the way each writer writes using this approach.

9. Certain things were complicated and didn't work so well. As mentioned earlier, using just the POS tagger to look for errors on more complicated structures can be difficult. The NLTK tagger should probably include a more diverse set of tags in order to improve the calculation of verb usage errors. Of course one issue the NLTK tagger has is when a sentence's verb usage is incorrect the tagger doesn't perform as accurate as it should, since the tagger is trained on formal English more or less. Similar issues can occur with the Stanford parser. The sentence "It is unless." for example returns a parse tree in which there is nothing wrong with the sentence.

10. Finally, for calculating the range for each sub score, we plotted the essays against the errors we received and decided different threshold for each component. Using, the weighted formula that we found using Linear Regression, we fitted the overall formula and found the total score. Same technique of histogram analysis was used to find the threshold and classifying the essay into low and high type.

11. We could increase the performance of the system by adding a spell corrector function which would replace the misspelled words. This would (potentially) prevent double penalizing in the rest of the grading. We could also add another category that checks for proper use of punctuation i.e. some essays didn't have any periods, didn't use periods etc. We could also check for errors using the POS tagger, but instead of just limiting it to verbs and nouns we could check for incorrect usages (i.e. a transition from PRP to TO).