Jank Chaudhari (jchaud23) and Giridhar Patri (gbpatri2)

Introduction:

The second part of our term project for CS 421 focuses on enhancing our understanding and implementation of syntactic well-formedness and rudimentary notions of coherence. This phase builds upon work done in the first part, introducing a more comprehensive analysis of sentence formation and coherence. The goal is to effectively map these analyses to a qualitative score categorizing essays as High or Low, based on their syntactic/semantic integrity.

Syntactic Well-Formedness Analysis:

The primary objective in this section is to utilize a syntactic and dependency parser to assess the grammatical well-formedness of essays, particularly focusing on sentence formation. This involves several subcriteria such as the proper formation of main sentences, correct constitution of sentence constituents, use of subordinating conjunctions. For example, main sentences should start with a noun phrase, prepositional phrase, adverb, or a subordinate clause depending on the sentence type (declarative, interrogative, etc.). Constituents should be correctly formed with no missing elements like prepositions or determiners, and subordinating conjunctions should correctly complement the main verb or gerund in a sentence. Utilizing parse trees, both constituent and dependency, we are able to identify common syntactic errors. Incorrect sentence constructions often reveal themselves through anomalies in these trees. For instance, the absence of a finite verb where one is expected or misplacement of pronouns within noun phrases are typical indicators of syntactic inaccuracies. By analyzing patterns in parse trees generated from sample sentences, we can identify and count syntactic errors, providing a quantitative measure of syntactic well-formedness which contributes to the overall essay score.

Semantic Analysis and Pragmatics:

The coherence of an essay, a critical part of its overall quality, is assessed through the use of word embeddings and cosine similarity techniques. This involves the computation of word embeddings using pre-trained models and evaluating the semantic proximity of sentences within an essay to gauge how “on topic” it is. The method entails averaging the embeddings of words within a sentence and then calculating the cosine similarity between these averaged embeddings across consecutive sentences. To address the essay topic relevance (d.i), we compute the cosine similarity between the embedding of the essay prompt and the averaged embeddings of the essay content. A higher similarity indicates better alignment with the topic. Coherence (d.ii), on the other hand, is evaluated by examining the cosine similarities between adjacent sentences. Essays with high coherence should show minimal abrupt shifts in topic or style, reflected by consistent cosine similarity scores across sentences.

Integration and Conclusion:

In integrating these analyses, each essay is processed to yield scores for syntactic well-formedness, topic relevance, and coherence. These scores are then combined according to a predefined formula to produce a final numeric score, which is subsequently mapped to a High or Low qualitative assessment. The threshold for a “high” or “low” was determined by averaging all the low and high scores, and tweaking the number for best results.

The project has been a great learning experience, emphasizing the complexity of linguistics and the challenges inherent in processing and understanding it computationally. While the results are relatively promising, there is room for improvement, particularly in refining the sensitivity of syntactic and semantic analyses to capture more nuanced aspects of language use. Future work could explore deeper syntactic structures and more dynamic models of semantic context to enhance the accuracy and reliability of our NLP applications.

Reflections:

This project has not only reinforced our understanding of NLP concepts but has also highlighted the practical challenges of applying these techniques effectively. As we worked on this project longer, we realized how many details we aren’t accounting for (if we were to actually implement a production level model). The relationship between different types of linguistic analysis and their respective implementation with code showed a new intersection of CS we haven’t seen before.