Description of Project

In this project, we had the opportunity to incorporate the course’s discussion on Natural Language Processing and apply our theoretical knowledge to an actual application of grading ESL essays. Specifically, we provided an in-depth POS tagging mechanism to drive our essay grading program in order to simulate an actual human grader’s results. We had an opportunity to collaborate and work together effectively to produce a final result that explored the effectiveness of an essay in relation to its semantic and syntactic correctness.

What worked well

What worked well in our project was that we fully took advantage of the Stanford parser’s capability to effectively produce POS tags for the essays as well as a structure of its sentences. We took a layered approach to our program. We knew that an essay should be broken down into its constituent sentence structure. If this structure of sentences within the essay could be formed in a logical manner, then this behavior often mirrored a well put-together essay in terms of its semantic nature. Next, we worked to parse each sentence into its POS tags. We used this vital information to determine the syntactic well-formedness of the essay. For instance, to determine word order, we used POS tags to determine when words were most likely in wrong configurations. An example from our program is that we detected for the incorrect consecutive POS tags of “NN NN” found in the essays. When the program detected these consecutive tags, an error was raised for the word order because sub score because this is incorrect usage of English.

Whenever, errors were found within essays, the error count would be increased. We made the decision to directly correlate the error rate with the final sub scores for each essay. This was an intuitive decision because it logically follows that if Essay X has more errors than Essay Y, then Essay X will have lower sub scores than Essay Y in regards to whichever sub sections in which Essay X performed worse. We found that by taking this approach, our calculated scores had a relatively tight correlation with the actual sub scores from the human grader. For instance, across the sub sections, the average summed difference between all essays was about 0.05 points. This means that overall, our calculated scores did not vary too much from the actual scores provided in the training set.

What didn’t work well

During this project, we were also able to learn what strategies did not work well in our program and we learned from these to ultimately improve our final product. For instance, one initial speed bump in our design was how to handle run-on sentences. When run-on sentences occur (e.g. I like cheese Diego was there My brother came later) it is difficult to obtain a correct structural hierarchy for the essay. Our initial strategy was to ignore these difficulties and hope that they resolved themselves. However, we quickly found out that this was not a good way to handle these situations and we drastically improved our model. Once again, we utilized POS markers to give us hints as to the endings of sentences when they otherwise are not clearly evident. For example, in the run on sentence (I like cheese Diego was there My brother came later) we can use POS flags to detect that there are actually three separate sentences. We know this because each one of these sub sentences contains a main subject and a main verb. In some cases, there may also be indirect objects or other POS tags to alert us of end-of-sentences when they may not be automatically evident.

What we learned

This project offered some great learning experiences when it comes to NLP and software design. We learned that when theory is applied practice, the best strategy in software design is to supply a lot of test cases. The best way to improve the program is to start with intuitive decisions and constantly improve the performance of the program through using the results of test cases. When it comes to NLP, we learned that a lot of syntactic well-formedness data can be obtained from an effective POS tagger. By using the Stanford parser, while not perfect, still provided us with a wealth of information to determine the correctness levels of ESL essays. For instance, detecting subject-verb agreement is a relatively reasonable task for an automatic grader when POS information can be obtained for essays. With these POS tags, our program detected main subject of the sentence, used the clues to determine the plurality, and detected whether or not the following verb had agreement. Clearly, there are some nuances that must be added for special situations, but the invaluable tool of the Stanford parser provided a good base upon which to build or project.

What is needed to take our software to a higher level

In order to take our program to a higher level, we would initially like to work with a larger training set to improve the calculation of sub scores. With a larger corpus, we will become more aware of certain errors and we can further design our program to detect these errors and provide a well-informed score. Additionally, if we added an expert on GUI design to our team, we could provide a polished interface for the user to use, enhancing his or her experience with our program usage.

In our program, we extensively used POS tags to enhance the performance of our program. Additionally, we would like to explore other avenues to add onto the performance our program already has. For instance, we could utilize a database of common errors that ESL students make and detect these instances in our grader. This will allow us to take advantage of human graders’ experiences and incorporate this knowledge directly into our project.