Ling 193

Instructor: Andrew McInnerney

Daniel Wang

Dec. 14, 2022

Final Project: Code Description and Outline

As we talked about using *"minimum edit distance"* to find the word in the dictionary that best matches with the typo in our sentence, I was thinking of using a couple of concepts we covered in the class to improve this autocorrect system. In the example given in the code, we see that our current autocorrect system does not differentiate singular determiners and plural determiners. It only corrects the typo to be a determiner when needed without checking whether the determiner it generates would fit into the whole sentence. In this project, I focused on using *"word frequency"* and *"sentence semantic structure"* to correct determiners before any noun phrase.

My code is consisted of 6 parts in general: *"download dictionary"*, *"1st layer of correction"*, *"2nd layer of correction"*, *"preparing for the 3rd layer of correction"*, *"3rd layer of correction"*, and *"driver function"*. I'll briefly go over the ideas behind each part in this write-up. Further implementation details can be found in the ".ipynb" file I provided.

"Minimum edit distance" is implemented in *"1st layer of correction". Compared to the code you showed to us, I modified the functions a little bit so that it no longer returns a single matching word. Now we will have the top 100 candidates from this section. It will be much easier for me to find the matching word from a range of factors, not just its "minimum edit distance" to the typo.*

"2nd layer of correction" takes word frequency into account. When we examine the output from last section, we see there are many distinctive words that have the same "minimum edit distance" to the typo. If we just randomly choose a certain word over

others, it will probably generate more trouble for our latter calculation. Therefore, we take in a dictionary that records the frequency of each word in it. We'll pick the word with both minimum edit distance to the typo and the most frequencies among other candidates. This word will be the one that replaces the typo in the sentence.

The idea behind "determiner & noun phrase correction" is that we will analyze the constituent tag for each word in the sentence. When we run into a combination of "DT_NN" or "DT_NNS", we will check if the determiner matches the noun phrase behind it. If it doesn't, we'll check if any other determiners are shown in our 100-candidates list from the first part. If it fits into the sentence better than our current word, we will replace our current word with it. For example, after the 1st and 2nd level of correction, we turned "anu tomatoes" into "an tomatoes". In *"3rd layer of correction"*, we can pinpoint the mismatching determiner "an" with its plural noun phrase "tomatoes". After this layer of correction, we'll have "any tomatoes" by our hand. Noticed that we also use brown corpus from *nltk* to determine what are the common determiners for plural noun and singular noun in this section.

After my version of refinement, this autocorrect system can fix the first 2 examples shown in the class:

s1 = "The manager hired ths employees"

and

s2 = "John will not buy anu tomatoes"

However, the system will only perform as expected if the 1st and 2nd layer of correction would correct the typo into the constituent tag, in this case "DT". If we try to modify s2 a little bit:

s3 = "John will not buy som tomatoes"

Our intuition tell us that "som" should be corrected as "some" in this sentence. Unfortunately, without going deeper into the tree structure of the sentence, our autocorrect system does not know what constituent is missing for this typo. Therefore, it will change “som” to be the most frequent word with a “*minimum edit distance”*, which is “so”. Since “so” is not tagged with “DT” and “so” is a legal word, our 3rd layer of correction does not pay any attention to it. Eventually, we will end up with a wiggly sentence.

Further improvement may focus on parsing the sentence into a tree structure. In that way, the correction can be dependent on the semantic structure of the sentence and thus more accurate in general.