Fall 2021 ACM Coding Challenge

Logan Jackson

September 4, 2021

1 Approach

In the Python solution the approach was originally going to be just to take the entire input.txt file and feed it into the model. But unfortuntely that didn't work. So on the second try I decided to split up the contents of input.txt and pass parts of the file to the model one at a time. This is what I decided to do this is really bad because transformers rely on context of past words in the document to be fed back into the transformer in order for it to learn. But since I am too lazy to write my own transformer I decided to take the laziest approach possible which was previouslly described. Now with this terrible approach the question was how to interpret the outputs in a way that was actually valid. Based on the architecture of a Transformer [1] we know that the output of our pretrained model is probabilities. in the case of distilbert this is the probability the model belives that it is correct, aka it's confidence in it's predictions. Now since we have a bunch of outputs from diffrent parts of the file we need to figure out the overall sentiment of the file. The way I did this I am not sure if it's correct but by summing the probabilities we are able to determine the tendency of the distrobution of confidence, negitive predictions are negitive, and positive predictions are positive, if across the file the model is more confident about positive prediction the value of

$$\bigcup P(x)$$

will be positive otherwise it will be negitive. The approach is potentially problematic due to the question of mutal exclusivity can a sentence be both positive and negitive or true nuetral? For the sake of this I assume the model will always predict one or the other never both making the options mutally exclusive. Now for the way that I calculated the confidence of the model

$$\bigcap P(x)$$

This I belive is an invalid approach for the following reasons: because the transformer relies on context from the rest of the file each individual batch sentiment could potentially be affected by previous context this means that the pobabilites of each batch are not independent of one another, this is a problem when dealing with an and operation. The product of all sentiments doesn't repersent the overall confidence of the model because each probability would have to be independent. This problem will most likely be addressed in the next model.

2 Numbers

There are two numbers associated with the model output that I previouslly discussed but as I explained earlier one of those numbers I belive to be invalid so there is only one number to talk about. that is:

$$\bigcup P(x) = 0.022$$

where $P(x)$ repersents the output of each batch. The tendency of the transformer is to have 2.2% more confidence in positive results than negitive results meaning that overall it belives the file to be slightly positive.

3 Potential Fixes

I think this approach is actual garbage, but if you can embed the output of the previous batch with the input of the next batch that would probably work and is probably how it's supposed to work. something like this:

```
previous_context = None
for i in range(len(strs)):
   # second arg optional passing None doesn't do anything
   res = clf(strs[i], previous_context)
   previous_context = res[0]['score']
return res

$ res = {'POSITIVE', 0.899999}
```

References

[1] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017.