Gabriel Deml 11/28/2021

Code:
https://drive.google.com/file/d/1OmR32EIIG-miN68XCYhMAL1HNN_Vobq2/view?usp=sharing

Task 1:

```
  "exact": 49.65661185245436,
  "f1": 57.47739414990552,
  "total": 130319,
  "HasAns_exact": 74.31957706084934,
  "HasAns_f1": 86.05863245322598,
  "HasAns_total": 86821,
  "NoAns_exact": 0.4299048232102625,
  "NoAns_f1": 0.4299048232102625,
  "NoAns_total": 43498
```

Task 2:

The best way of solving this that I can think of is to have a special token at the end of the string. If the BERT model selects this token then it thinks the question is unanswerable. This would basically allow the model to put its “uncertainty” somewhere. With the BERT used in question one, we are forcing it to choose something even if choosing nothing is the correct answer.

The input vector would look something like this:
[CLS] What color is the sky? [SEP] The sky is usually blue on sunny days [No answer token]

If the model selects the no-answer token then the question is unanswerable.

I think a simple hack to implement this would be to just append the string “no answer” at the end of each sentence. The answer for the unanswerable questions would be no answer. It would be better to use a dedicated token for this, but as a test, I think the append method would work.