Data Selection Proposal

Groupmates: Sarah Santoso, Sindbad Walter, Tiffany Yong

1. Choice of dataset: https://www.kaggle.com/danofer/sarcasm

   We had 2 other options, a news headlines dataset and a Twitter dataset. However, we decided to go with the Reddit dataset for the following reasons:

   - Lots of data (1.3M+), due to the high volume of comments on Reddit as compared to a news headlines dataset.
   - The way that the dataset is curated is that it considers a comment sarcastic if it has the "/s" tag, which is determined by the user. On Twitter, there is no such convention to mark sarcastic tweets. Thus, the determination of whether a tweet is sarcastic is done by an external party, which may have judged it wrongly.
   - There are two options, a balanced dataset with an equal amount of sarcastic tweets to not sarcastic tweets, and a representative dataset that represents the ratio of sarcastic and non-sarcastic tweets on Reddit. This is compared to both other datasets, which only had an unbalanced option.
2. Methodology
   a. Data Preprocessing

      The data already has the most important indicator, which is whether the comment is sarcastic or not, and the body text of the comment. We will not need any other information like author or number of upvotes, since this information is not available when someone copy-pastes the text into our input field.

   b. Machine learning model

      Long-Short Term Memory (LSTM) type of RNN:

      - Pros
        - Many people have used this model (based on code posted on Kaggle)
        - Good for sequenced data, in this case we would be inputting sentences, being seen in our case as a sequence of words
        - The big advantage of using LSTM over other RNN's models is that the LSTM remembers long-term dependencies, for example if we analyze a text with a very important word at the beginning of it, which we need to understand the rest of the text then LSTM do it very well
        - Has a forget gate, which removes useless information
      - Cons
        - Uses more memory and executes slower than other types of RNN such as GRU
        - Can easily overfit
        - More complex than GRU, hence harder to implement, or adapt

   c. Evaluation Metric

- We aim to hit 70% accuracy in total, with a balanced number of false positives and false negatives.
- Since the output of our program is binary, meaning it will either say that the comment is sarcastic or that it is not, we can measure the accuracy using a confusion matrix. A confusion matrix stores how many data points the program calculated as 0 (false) correctly, how many it calculated as 1 (true) correctly, how many it calculated as 0 incorrectly, and how many it calculated as 1 incorrectly. This is useful for calculating accuracy as it allows us to take into account false positives and negatives, instead of simply calculating the accuracy based on how many values the program got correctly.
- Using the values in the confusion matrix, we can calculate precision and sensitivity, possibly using logistic loss, accuracy equation, or area under the curve.

3. Application

   We will be building a simple webapp, with an input field where people can copy and paste a Reddit comment to test if it is sarcastic. They can paste text that is not from a Reddit comment as well, but we will be optimizing for precision on Reddit comments. Once they click a button to submit the input, the model will determine if the input text is sarcastic or not, and return the result to the user.