Data Preprocessing and Project Updates

MAIS 202 Deliverable 2

Sarvasv Arora, Kaiwen Xu

February 22, 2021

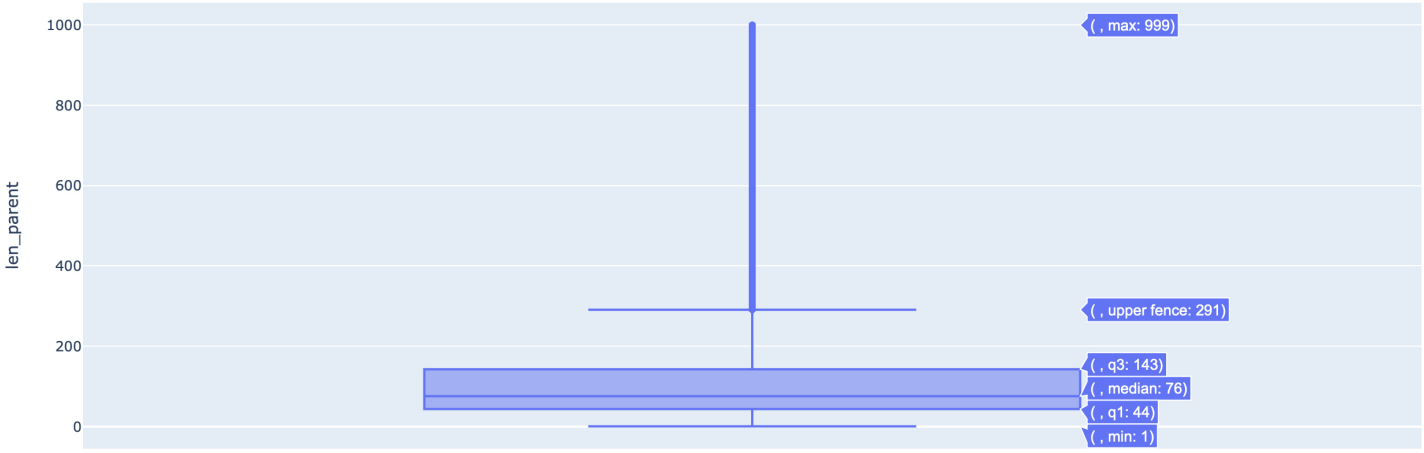
Project Idea

We plan to make a discord/reddit bot that replies to specified comment/post with a Deep Learning generated sarcastic — but fun — comment.

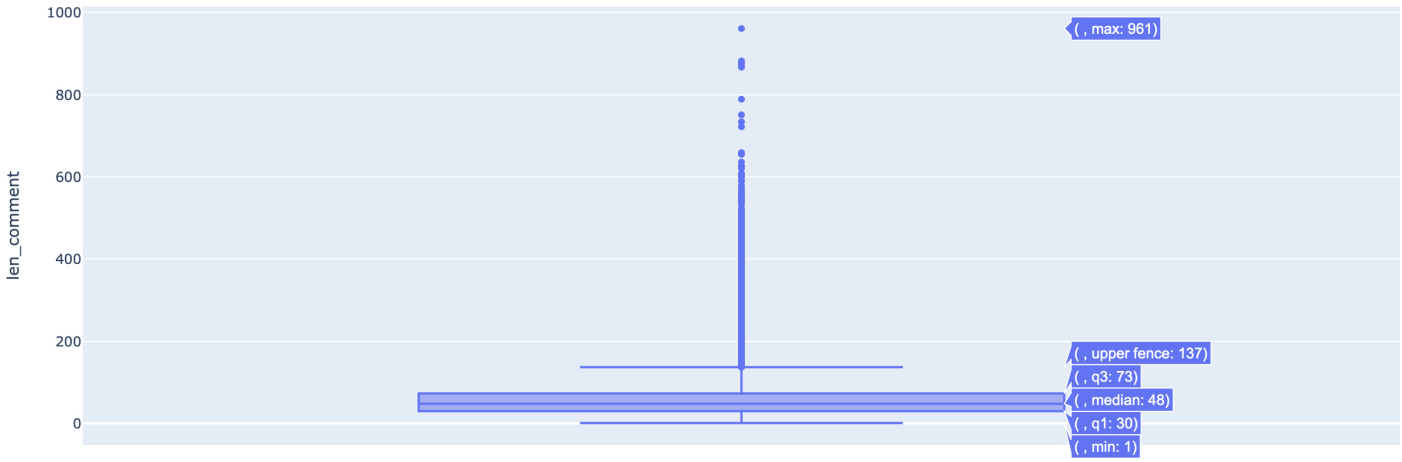
Data Preprocessing

We are using a Kaggle dataset containing around 1M comment-reply pairs, with the replies marked as sarcastic (with a label of 1) or non-sarcastic (with a label of 0). Since our goal is to generate sarcastic comments, we will be using only those with label 1. This amounts to roughly 500k raw comments.

Next, we have a look at the length of the replies and the parent comments. This is important since it will allow us to pad the tokenized data accordingly (described later), as well as select only those comments which actually are suitable for our model type.
The parent comments have a mean length of 133 words with a standard deviation of 214. Since we are fine tuning our model on the GPT 2 pre-trained model, which accepts a maximum of 1024 tokens as input, this seams pretty reasonable. On further inspection, we find that majority of parent comments have length below 1000 words. To delete outliers and clean the data, we simply delete all the rows in the dataset where the length of parent comment is greater than 1000. This only removes a few thousand rows. Further cleaning will be done when we actually tokenize the sentences.

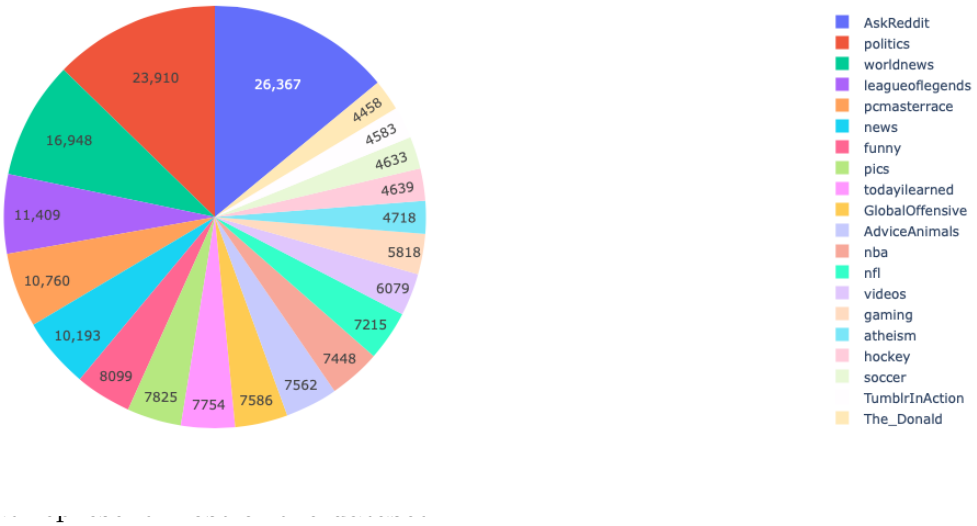


Next, we remove the outliers (based on length) from the replies. The replies have a mean length of 56 words with a standard deviation of 41. In this case, the outliers, having length more than 1000, are only a handful (in order ¡ 100), and we remove them as before. Although not necessary, we do this step because we don't want to produce very long replies.



Another option we thought of was to only choose those replies which have a score (upvotes - downvotes) above a certain threshold, based on the thinking that a good score implies better quality reply. However, doing so would have two problems. First, that it will greately reduce the size of our text corpus. Second, that it is not really an indicator of the quality of the reply, but is more so based on how the user perceives that comment and the score on that comment prior to the user seeing it [1].

Finally, we explore the Subreddits from where the majority of the comments come from. Here are the top



Finally, the data will be tokenized and exported into JSON file format so that it can be used with our choice of ML framework.

Machine Learning Model

Contrary to the first deliverable, we aim to fine-tune the Hugging Face GPT 2 model or an EncoderDecoder Transformer to produce the replies. We came to a conclusion that the data we have is not enough to train a Machine Language model from scratch, and it will produce far from human-like responses.

GPT 2 is a decoder-only transformer, and have been used before to summarize text [2]. We can work on those lines to generate replies.

Another option is to use a BERT-to-BERT EncoderDecoder architecture which seems like a more natural way to produce replies.

We still need to finalize on this, and hope to do so in KC's office hours post midterms. We will then work on training the model in the reading break.

Our validation metrics still remain the same i.e., calculating the BLEU score on generated replies to parent comments in the test set as compared with the replies in the test set. We have also found out about the shortcomings of this metric, and thought of training a classifier that can predict whether the machine generated reply is human-like or not. This is an ambitious idea though, and we might work on it only if we complete the main idea on time.

References

[1] Maria Glenski and Tim Weninger. *Predicting User-Interactions on Reddit*. 2017. arXiv: 1707.00195 [cs.SI].

[2] Urvashi Khandelwal et al. *Sample Efficient Text Summarization Using a Single Pre-Trained Transformer*. 2019. arXiv: 1905.08836 [cs.CL].