ML Researcher Assignment Report

By Paritosh Dahiya
(dahiyaparitosh@gmail.com)

Assignment - Semantic similarity

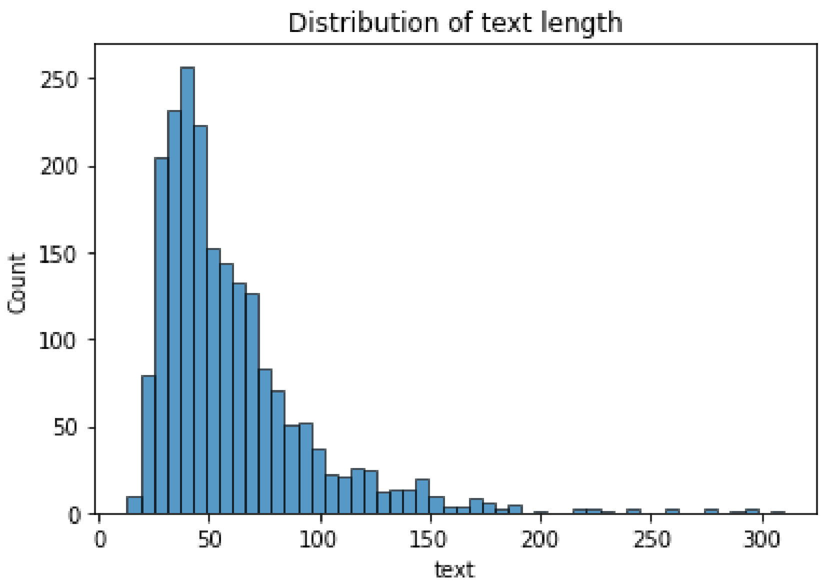
Problem Statement:

Given a text and a reason, predict if "text" satisfies the "reason".

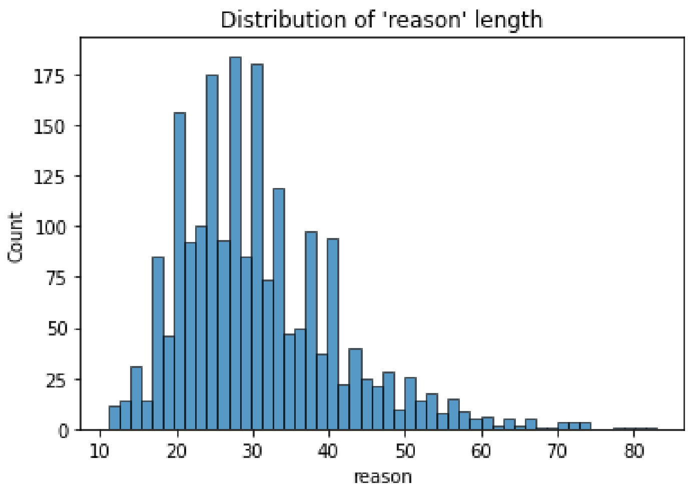
Data Insights:

First, I conducted some Exploratory Analysis of the Data and had the following insights:

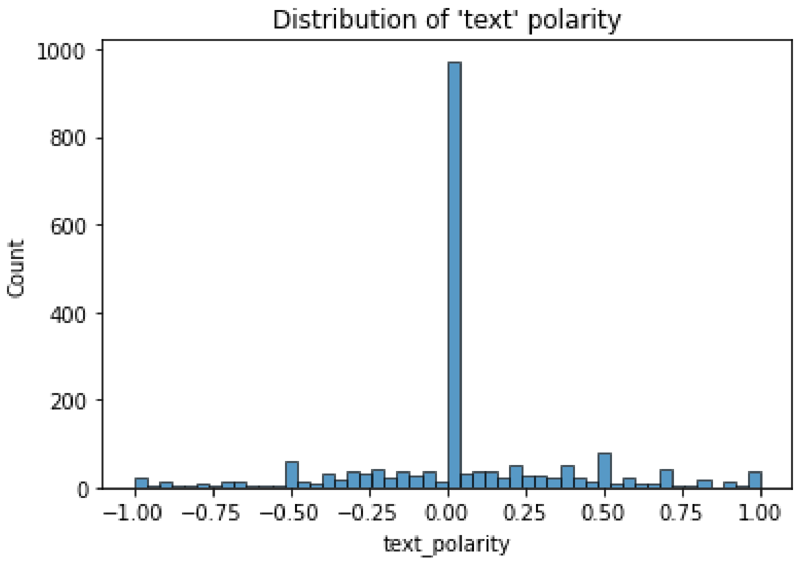
- The data does not contain any null values and any contractions.
- The maximum 'text' length of the train data was 66 and it was 186 for the test data.



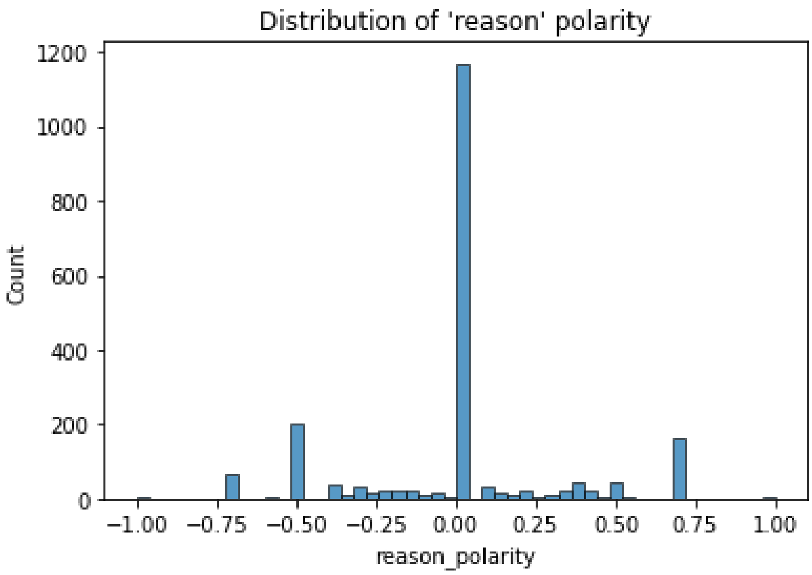
- And the maximum 'reason' length of the train data is 16 and it was 13 for test data.



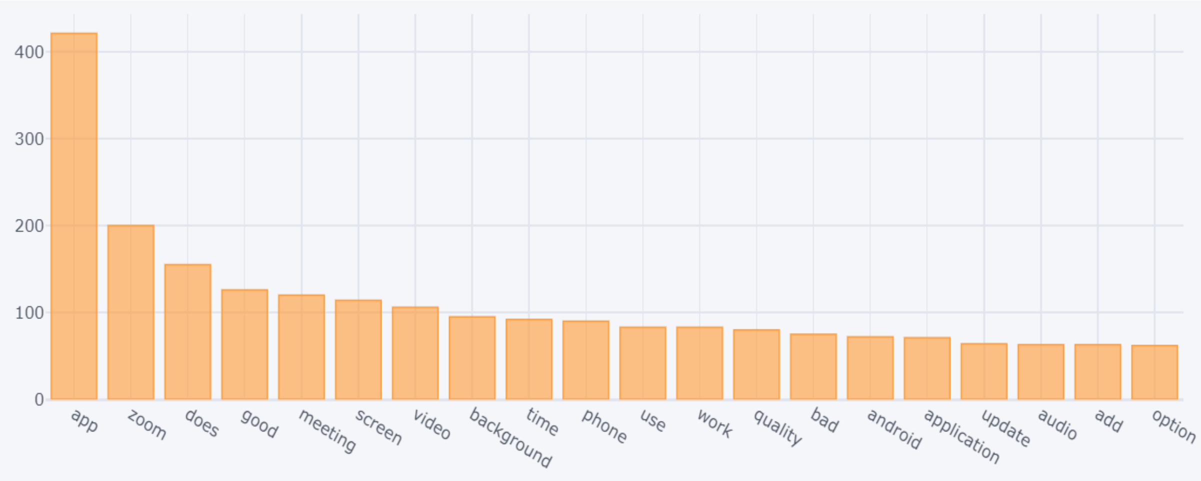
- Most ‘text’ sentences were neutral in polarity like any statement or a fact.

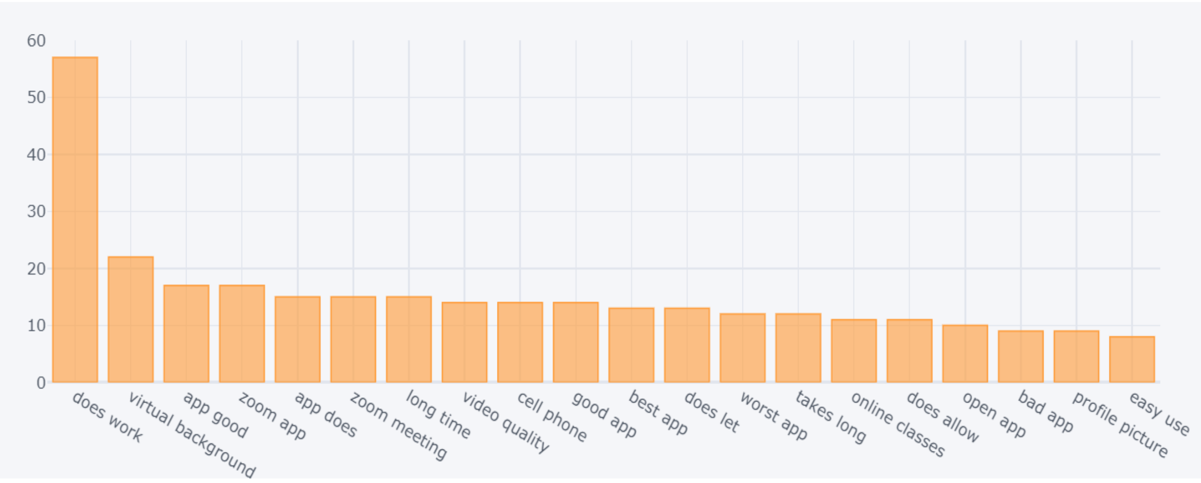


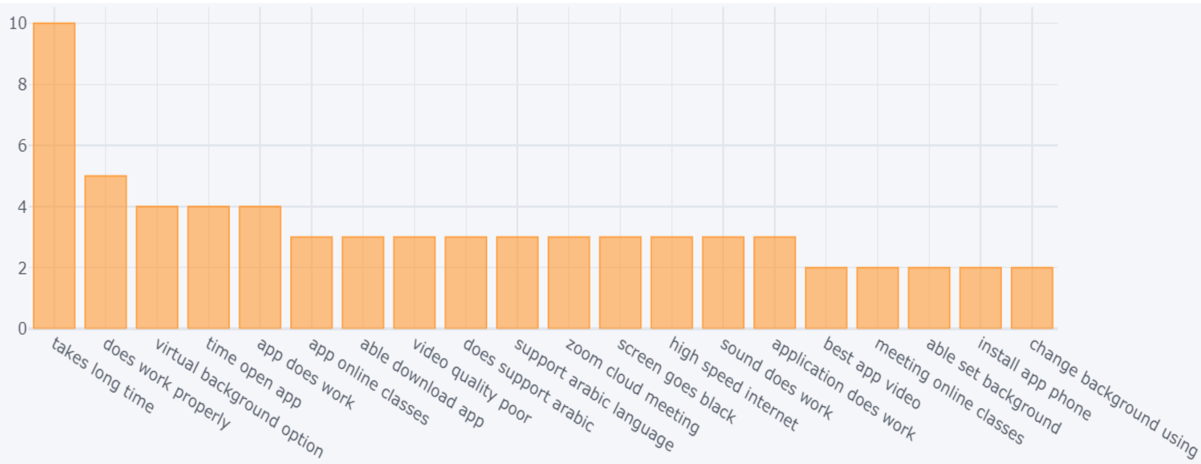
- Same was the case with the polarity of ‘reason’.



- Here is the top 20 **unigrams**, **bigrams** and **trigrams** in the 'text' field







Baseline approach:

1. **Tokenization:** Tokenize the text and reason features using a pre-trained tokenizer '**distilbert-base-uncased'**
2. **Encoding**: Encode the tokenized features using a pre-trained transformer-based language '**distilbert-base-uncased'**
3. **Concatenation:** Concatenate the pooled representations of the text and reason features to get a joint representation of the input.
4. **Classification:** Add a classification layer on top of the joint representation to predict the label.

- The baseline model was trained for 4 epochs on the '**distilbert-base-uncased**' model with
  batch_size = 32 and learning_rate = 1e-4.

The performance was as:

- Training Loss: 0.3133
- Training Accuracy: 1.0
- Test Loss: 0.9797
- Test Accuracy: 0.3334

Training Approach:

1. **Balance the dataset**: As the dataset contains **only positive** samples, we need to generate negative samples too.

   There can be many techniques to achieve the same but the one of the simplest and effective way is to **negate the 'text'** sentences, so that their meaning and/or **polarity** is **reversed**.

This was achieved using python module called '**negator**' which uses Spacy and transformers to negate the eligible sentences.

Out of 2061, **1854** sentences were able to be negated, increasing the dataset size to **3915**.

2. **Training:** After augmenting the dataset, the same '**distilbert-base-uncased**' was evaluated after 4 epochs of training.

With negative samples also added the final performance was:

- Training Loss: 0.5791
- Training Accuracy: 0.5264
- Test Loss: 0.8786
- Test Accuracy: 0.3334

Here the test accuracy is almost the same but the loss for the Neural Network classification layer has been **reduced** by amount of **0.1**.

<u>Proposed Techniques for further research:</u>

There are many more approaches which can be further implemented to increase the performance but are not tested yet such as:

- **Data augmentation** by replacing various words by there **synonyms** (or **antonyms** for negative samples).
- **Multimodal learning** by using various **numerical features** along with the sentences like word count/length, **polarity** of sentences, etc. More features can increase performance.
- If more time was permitted, above **proposed techniques** could be implemented and more models like **GPT** could also be tested.