Individual Report

Aihan Liu

Overview of Dataset

The data can be extracted using Reddit API, sourced from Pushshift[1]. Since the Pushshift API limits the number of 1000 data per request, it is necessary to apply multi-threaded downloads. PMAW (Pushshift Multithread API Wrapper) [2] is a wrapper for the Pushshift API which uses multithreading to retrieve Reddit comments and submissions.

```
def get_pushshift(subreddit, limit, before, after, filter_attr, datat_type):
    api = PushshiftAPI()
    if datat_type =='submission':
        submission_result= api.search_submissions(subreddit=subreddit, limit=limit, before=before, after=after, mem_safe=True, filter=filter_attr)
        print(f'Retrieved {len(submission_result)} posts from Pushshift')
        submission_df = pd.DataFrame(submission_result)
        return submission_df
    if datat_type == 'comment':
        # return the comments that match the submission id
        comment_result = api.search_comments(subreddit=subreddit, limit=limit, before=before, after=after, mem_safe=True)
        print(f'Retrieved {len(comment_result)} comments from Pushshift')
        comment_df = pd.DataFrame(comment_result)
        return comment_df
# cite: https://pypi.org/project/pmaw/
# https://medium.com/swlh/how-to-scrape-large-amounts-of-reddit-data-using-pushshift-1d33bde9286
```

Figure 1. Data Download

There are nearly 100k observations from January 2021 through the end of October 2022. Some data preprocessing methods are applied to the dataset for EDA, including changing all words to lowercase, tokenization, removing stopwords, and lemmatization using the NLTK package.

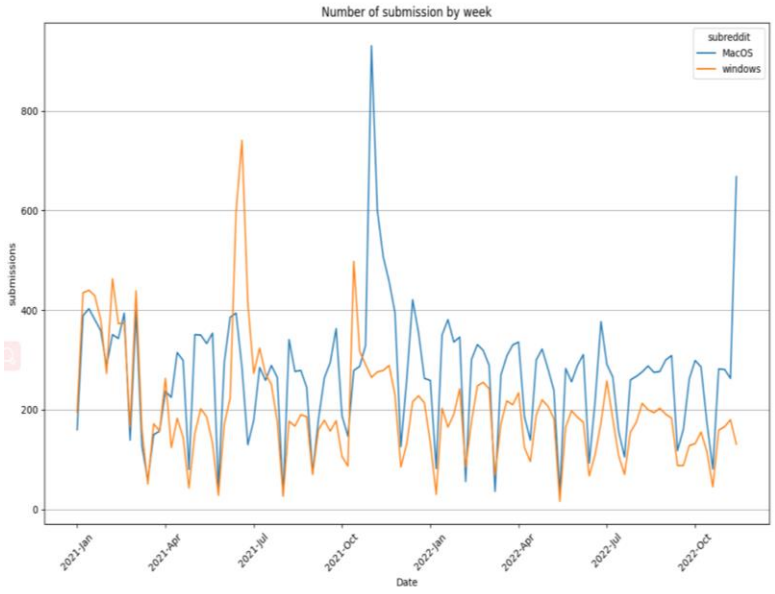

Figure 2. The number of submissions from 2021 Jan to 2022 Oct

We first took nearly one and half weeks to decide which problem we wanted to solve. Since I worked for the Pushshift data in the other project, I recommended to use this data. Sentiment Analysis is the easiest and most general task for NLP. We decided to generate a sentiment analysis model. Because the reddit data doesn't have any label for sentiment, we decided to work for the Interpretability and Explainability for the model.

Individual Work

My work for this project is based on data downloading, model generation and problem solving.

I had a problem downloading the data using the original API, and no matter how we changed the amount of data we needed to download, we only ended up with about 200 results. Luckily, I found the PMAW package that could use for downloading unlimited dataset.

For the model generation, my group partner had trouble in install git-lfs, which using for uploading the model to hugging face. I tried to solve it and uploaded the model. I also generate the IG model and the result too. Also, I kept critical thinking and raised many questions during this project, and we tried to solve these questions together.

Portion of Work

The DistilBERT model is trained with all the IMDB training data from the dataset package. The training loop is performed under the parameters shown in Table 1.

The results were outstanding: we achieved 89% accuracy on the test set (IMDB testing). On one hand, it proves our model architecture was correct and useful, on the other we were concerned with overfitting and model robustness.

In addition to fine-tuning a new model, there are over 1000 models in hugging face for the sentiment analysis task. We picked one of the most downloaded models (Twitter-RoBERTa-base model) to verify and compare the output with our trained model.

```
model = TFAutoModelForSequenceClassification.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)
```

Parameter	Value
Epochs	1
Batch Size	16
tokenizer	Same as model
Maximum Input Length	256
Learning Rate	Linear learning schedules
Loss Function	SparseCategoricalCrossentropy
Optimizer	Adam
Evaluation Metrics	SparseCategoricalAccuracy

Table 1. Training Parameters.

The team used these models to test the outcome of Reddit data. Since the Reddit data doesn't have any label for sentiment, we use the integrated gradients method to explain the relationship between model prediction and input characteristics.

The alibi implementation of the integrated gradients method is specific to TensorFlow and Keras models[3]. In this experiment, the embedding layer is involved in the gradient calculation as a baseline, which means we ensure that the attribution for tokens will be 0 if we use the embedding layer. The 0 attribution is due to integration between which is 0.

```
# calculate the attributions with respect to the
# first embedding layer of the (distil)BERT
elif model_name == 'distilbert-base-uncased':
    layer = auto_model.layers[0].layers[0].embeddings
```

```
# define Integrated Gradients explainer
ig = IntegratedGradients(auto_model,
                         layer=layer,
                         n_steps=n_steps,
                         method=method,
                         internal_batch_size=internal_batch_size)
```

```
# get explanation
explanation = ig.explain(X_test,
                         forward_kwargs=kwargs,
                         baselines=baselines,
                         target=predictions)

# Get attributions values from the explanation object
attrs = explanation.attributions[0]
print('Attributions shape:', attrs.shape)
```

Figure 3. Usage of IntegratedGradients

The RoBERT base model is better than the Distilling Bert base model because compared with movie reviews, the comments from social media are more similar to our data. In the meantime, the model performance is different too. Since we use the same model in tokenization as the training model, they do not have the same results in their tokens.

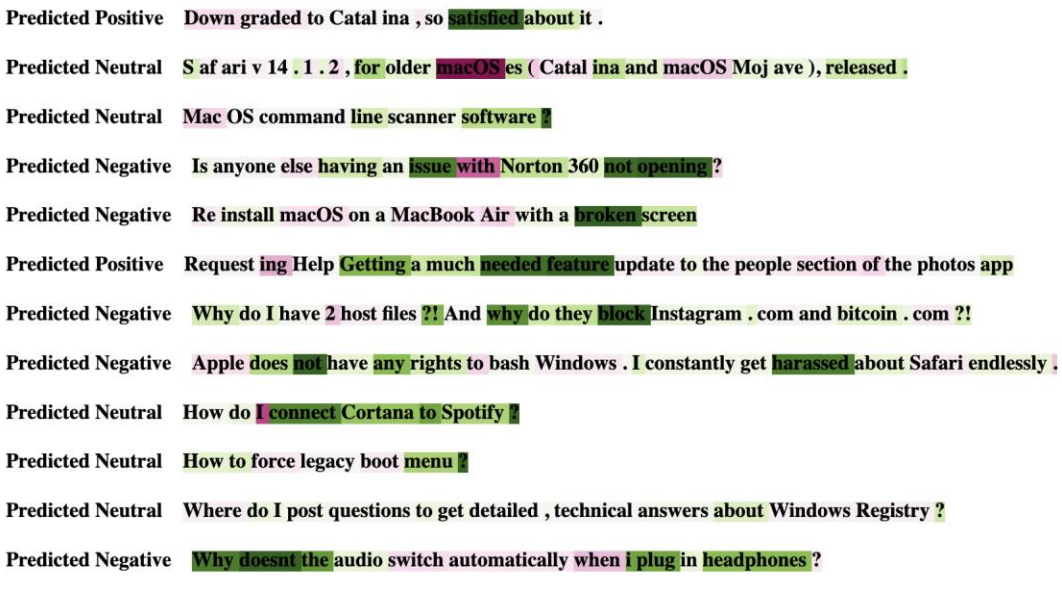

Figure 4. Predicted sentiments from RoBERTa-base model.

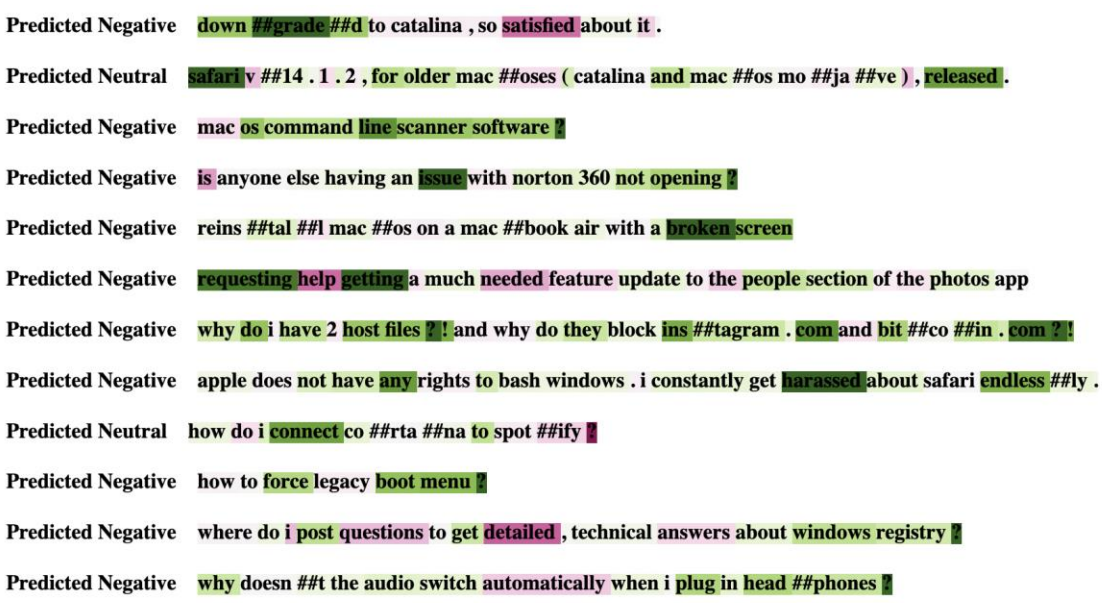

Figure 5. Predicted Sentiments from Distilling BERT base model.

This study is not perfect and is more of an attempt at a new field that we are not familiar with. There are a few points that were not improved due to time issues.

- As mentioned above, the tokenization method did not match, it probably affects the output of IG results.

- Due to a large amount of data and relatively long training time, the model was trained after only one epoch, and although it achieved good results on the testing set, it was not the best though.
- Methods for the integral approximation: we only applied a gausslegendre method, there are four more methods we haven't tried.
- By comparing the two we can see that the RoBERTa-base model gives more reasonable results. However because the training sets are different, it does not completely indicate that the RoBERTa model is better than the Distilling BERT. The difference in the focus of the two models has also been explained in Model Descriptions. In the future, if we can download the Twitter data and train different models, we can better explain whether there is a difference between these two models in using tweet data fine-tuning to Reddit.
- The study did not address well the question we posed: what are the factors that affect the quality of both windows and macOS subreddits reviews and the difference in user perception? A deeper study of the IG results could better answer the questions raised above so that we could explain why our result is reasonable.

In this project, I learned the Integrated Gradient, including the techniques and coding, and how to use hugging face (including upload model and using other's model). I also learned a little bit CSS coding when generate the result into HTML format.

Codes

There are several files related to this project that I wrote:

- DataDownload.py: Download Reddit data with Pushshift API
- Preprocessing.py: Tokenization and preprocessing the text data
- EDA.py: Exploratory data analysis
- Sentiment Analysis.py: An attemptation for sentiment analysis
- IntegratedGradients.py: Intergrated Gradient and generate results

Sentiment Analysis and Integrated Gradients are mostly online code, I made changes to their inputs, outputs, and some of the models to ensure that our data would run smoothly on this model.

References

[1] Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., & Blackburn, J. (2020, May). The pushshift Reddit dataset. In *Proceedings of the international AAAI conference on web and social media* (Vol. 14, pp. 830-839).

[2] PMAW: https://github.com/mattpodolak/pmaw

[3] Sundararajan, M., Taly, A., & Yan, Q. (2017, July). Axiomatic attribution for deep networks. In *International conference on machine learning* (pp. 3319-3328). PMLR.

[4] Sentiment analysis: https://huggingface.co/blog/sentiment-analysis-python

[5] Integrated Gradients: https://docs.seldon.io/projects/alibi/en/stable/examples/integrated_gradients_transformers.html

[6] Bert model Comparision: https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8

[7] Twitter-Roberta-base-sentiment: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

[8] alibi document: https://readthedocs.org/projects/alibi/downloads/pdf/stable/#page=99&zoom=100,96,609