# Group 2 Challenge 3

Twitter Sentiment Analysis

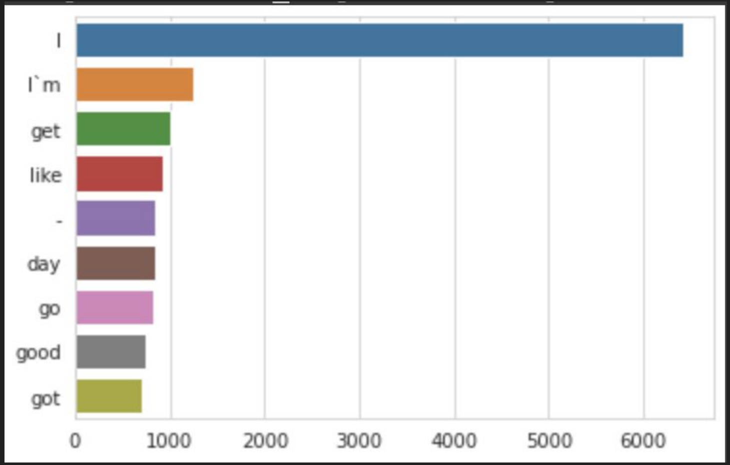
August Birkeland, Dario Dekovic, Santhosh Shanmugam & Tiril Syslak

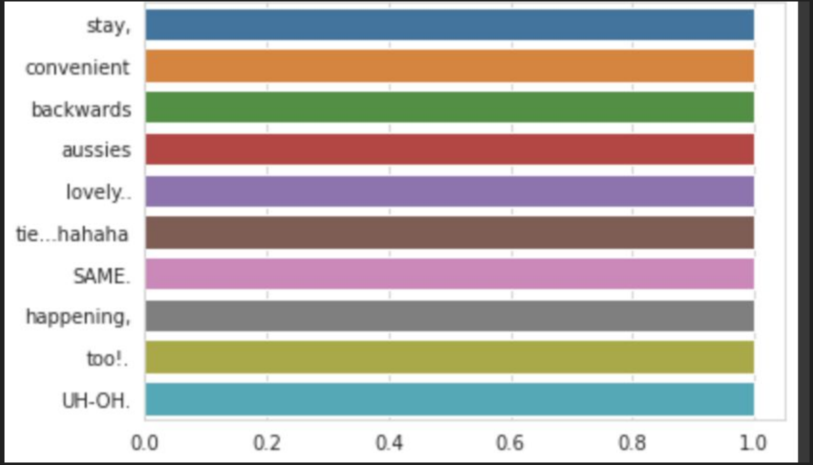
## Data Analysis Objectives

- Word Clouds
- Most and Least Common Words
- Words with Highest TF-IDF Score

# Data Analysis - Most and Least Common Words

- To analyse the dataset we tried to look at and plot the most and least common words. We can see from the plot that these words do not tell us much about the mood of the tweet.

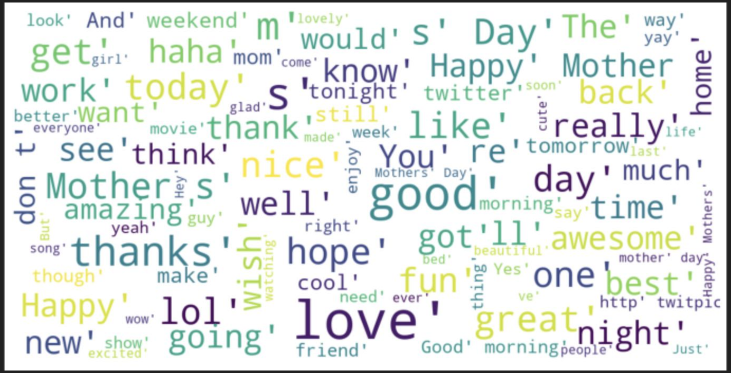




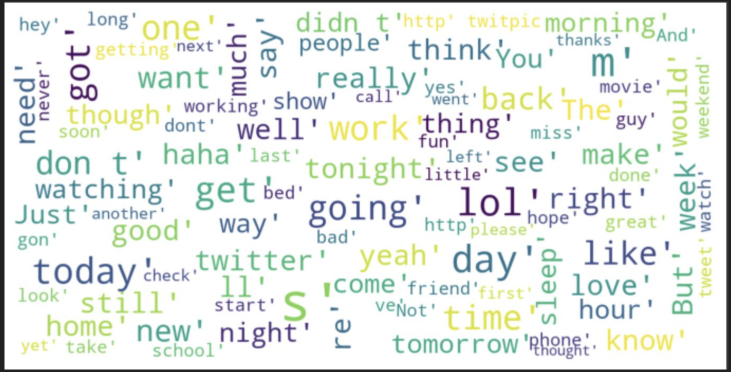
# Data Analysis - Word Clouds

- We also separated the datasets to one for each sentiment, to look at which words were most popular for each sentiment. The word cloud for each sentiment is given here, and we can observe quite a difference between the three (e.g. “get”, “today”, “going” for neutral, “love”, “hope”, “good” for positive and “miss”, “suck”, “tired” for negative)

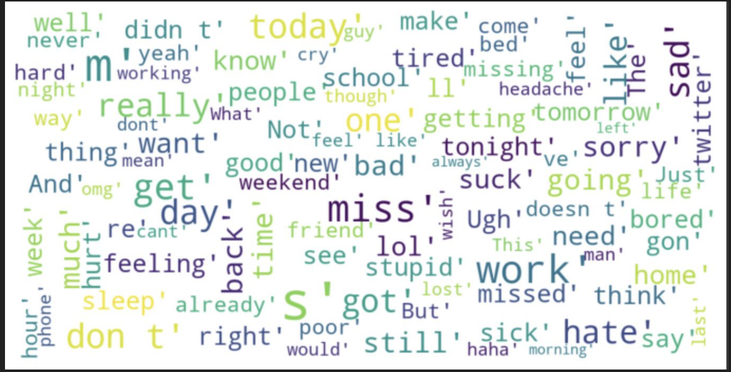
Positive



Neutral

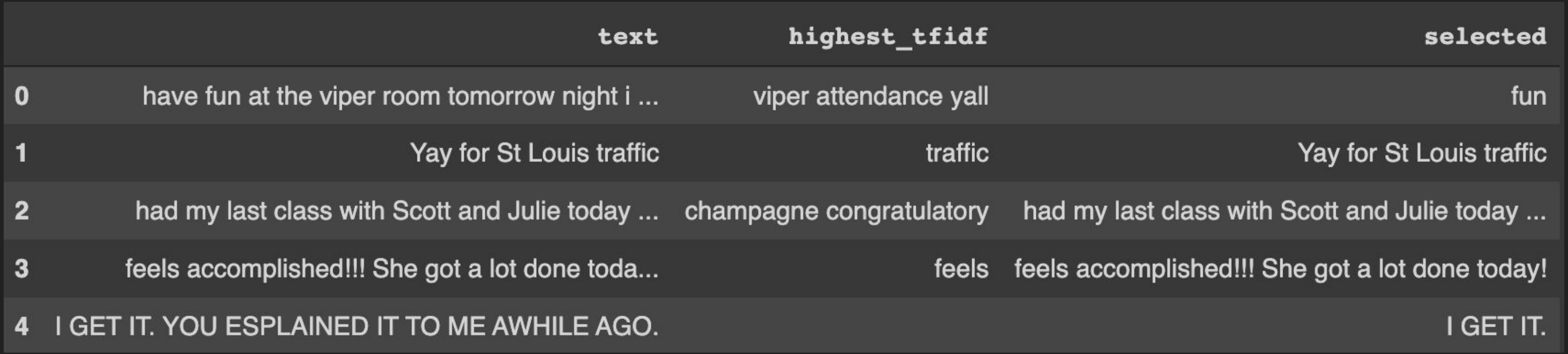


Negative



# Data Analysis - Words with Highest TF-IDF Score

- Another analysis we did was to find the most important words in each tweet according to the TF-IDF score (we chose the ones with score > 0.3)
- We also compared this to the “selected_text” column given in the dataset, but realized these were not much alike as the mean jaccard similarity between the two columns was only 0.4



| | text | highest_tfidf | selected |
|---|---|---|---|
| 0 | have fun at the viper room tomorrow night i ... | viper attendance yall | fun |
| 1 | Yay for St Louis traffic | traffic | Yay for St Louis traffic |
| 2 | had my last class with Scott and Julie today ... | champagne congratulatory | had my last class with Scott and Julie today ... |
| 3 | feels accomplished!!! She got a lot done toda... | feels | feels accomplished!!! She got a lot done today! |
| 4 | I GET IT. YOU ESPLAINED IT TO ME AWHILE AGO. | | I GET IT. |

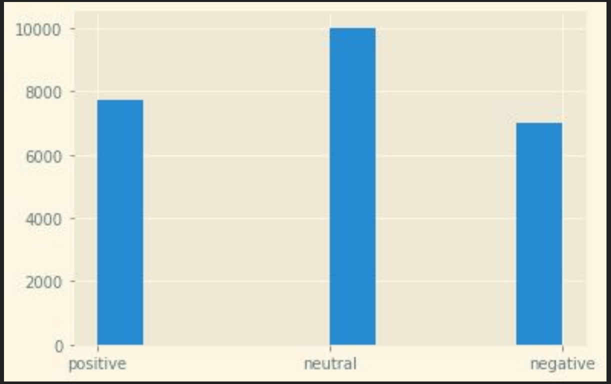
## Data Preprocessing Objectives

- Rebalancing Sentiments in Training Dataset
- Tokenization

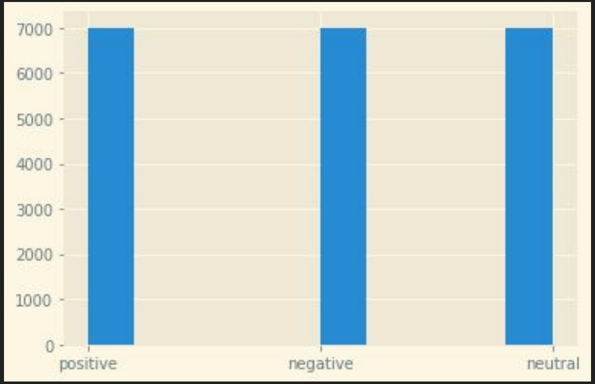
## Data Preprocessing - Target Rebalancing

- We suspected that the model was slightly biased towards neutral sentiments since there was about 2000 more rows of this sentiment than the other two
- We therefore selected a random sample of 7003 rows from each of the three sentiments in the training set and trained the model on all of these
- Unfortunately, this did not yield any better results

Before



After



## Data Preprocessing - Transformer Preprocessing

Transformers we used required text to be preprocessed in a certain way before we passed it to the tokenizers. We did the following things:

- Replaced all the links in the dataset with the token HTTPURL
- Replaced all the usernames in the dataset with the token @USER
- Translated all the emojis into text using the emoji package

## Data Preprocessing - Tokenization

- Tokenization is done by splitting the text into atomic elements and by doing this we are able to use BERT text embeddings as input to train the model which can prove to be lucrative
- To do this we used the automatic BERT tokenizer
- In general, tokenization is a method that converts strings of characters into transformer readable token ID integers

## Modelling Objectives

- Huggingface Transformers
    - Fine-Tune BERTweet Transformer
    - Fine-Tune Roberta Transformer
    - Fine-Tune Roberta Large Transformer

## Modelling Objectives- Huggingface and Transformers

- Huggingface provides a vast NLP library with various models. From what we’ve learned in previous courses we knew that transformers would perform the best
- Transformers are encoder-decoders with self-attention, meaning it is able to learn the input and output words in parallel
- We therefore tried to find a transformer that was appropriate for our problem.

## Modelling Objectives- BERTweet

- BERT (Bidirectional Encoders Representations from Transformers) predicts a word based both on the before and after contexts of surrounding words, unlike other language models that can only be trained left-to-right or the opposite
- BERTweet is a model that has the BERT-base model as architecture but uses the roBERTa pre-training procedure
- It was pre-trained on 850M Tweets and further details can be found in the original paper [1]

## Modelling Objectives- BERTweet Results

- We fine tuned BERTweet on our training data using the default learning rate from the Huggingface and obtained the following results on the testing set:

| No fine tuning | 2 epochs | 3 epochs | 4 epochs |
|---|---|---|---|
| 0.70447 | 0.78923 | 0.79597 | 0.78351 |

## Modelling Objectives - RoBERTa

- RoBERTa is a robustly optimized method made by Facebook that improves on the BERT model. It produced state-of-the-art results on NLP tasks at the time it was published (2019)
- We used a RoBERTa model pretrained on 124M tweets, with the same labels as our own task (negative, neutral and positive) [2].

## Modelling Objectives - Fine Tuning

- After downloading a pre trained transformer from Huggingface it is common to fine tune it to your own training set. In our case the RoBERTa model already had correct output-labels. Hence, we decided to first run it without fine tuning and then with different fine tuning parameters. The results are displayed in the table below and we can observe that fine tuning will improve the model up to around 2 epochs

| Fine tune? | Epochs | Learning rate | Submission Score |
|---|---|---|---|
| No | 0 | 0 | 0.70375 |
| Yes | 2 | 1e-4 | 0.79191 |
| Yes | 4 | 1e-4 | 0.77948 |

## Modelling Objectives - RoBERTa Large

- The last model we tried was roBERTa model described earlier with a larger base architecture. This architecture is roughly double the size of the original roBERTa model and is described in detail in the original paper [3]
- We fine-tuned the model and obtained the following results on the test set:

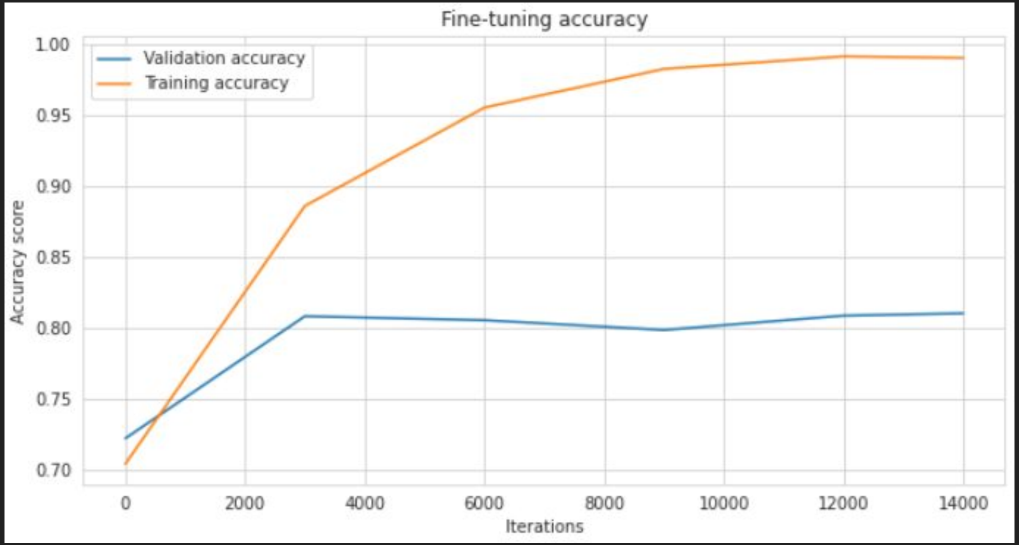
| 3 epochs | 10 epochs |
|---|---|
| 0.79350 | 0.78558 |

## Modelling Objectives - Testing of Overfitting

- From the previous results, it is clear that our models do not improve much through epochs of fine-tuning
- We decided to test that, so we separated the training dataset into two sets: the training set with 90% of the data and the validation set with 10% of the data
- We fine-tuned the roBERTa large model for ten epochs and calculated accuracy on training and validation sets every 2000 iterations.

## Modelling Objectives - Testing of Overfitting

- From the following graph, we can see that our model indeed overfits to training data thus one of the steps for further improvement would be to perform hyperparameter tuning to reduce overfitting



## Conclusions

After trying different models, we can see that transformers easily outperform the given baseline method. But transformers come with their limitations. The main limitation of transformers is the massive amount of data needed to pre-train them in order for them to achieve exceptional results. Without the Huggingface library, we couldn’t use the transformers simply because we don’t have the computational power to perform the pre-training process.

Also, we noticed through our experiments that transformers easily overfit training data because of their capacity, thus a careful selection of hyperparameters is needed to mitigate that problem.

## Future work

Future work would include the following steps:

- Testing different hyperparameters on the outlined models to mitigate the problem of overfitting
- Trying a few classical machine learning algorithms e.g. logistic regression, to compare performance versus speed between them and the transformers