In this blog post, we will introduce how we use BERT to analyse the text data collected from data_preparation.md.

# BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained natural language processing (NLP) model developed by Google. It is based on the Transformer architecture, which is a neural network architecture designed to handle sequential data, such as natural language. The model is pre-trained on a large corpus of text data and can be fine-tuned on specific NLP tasks, such as sentiment analysis, question-answering, and text classification. It has achieved state-of-the-art results on many NLP benchmarks and has become a popular choice for NLP tasks.

One advantage of BERT is that it is bidirectional. To be more specific, it processes text in both directions, allowing it to capture the full context of a word and the relationship between words in a sentence. This approach is different from previous models that processed words in one direction only. BERT's bidirectional nature has contributed to its success in various NLP tasks and has revolutionized the field of NLP by improving machines' understanding of human language.

In this project, we will use BERT to analyse Jim Cramer's attitude towards each stocks he mentioned, rating them on a scale from 1 (most negative) to 5 (most positive).

## Model selection

We chose nlptown/bert-base-multilingual-uncased-sentiment as our NLP model. It is designed to be language-agnostic, meaning that it can analyze sentiment in text from any language without the need for language-specific models. This is useful when sentiment analysis is performed across multiple languages, such as social media monitoring. Also, the model is "uncased," meaning that it does not differentiate between capitalized and lowercase letters, making it more flexible in its ability to analyze text in different formats.

## Tokenization

Tokenization is an essential step in NLP, as it allows the model to understand the structure and meaning of the text. The nlptown/bert-base-multilingual-uncased-sentiment model tokenizes text using a technique called WordPiece tokenization. WordPiece is a subword tokenization method in which the model learns to segment words into smaller subwords based on the frequency of those subwords in the training data.

For example, the word "unhappy" might be broken down into two subwords: "un" and "happy." By doing so, the model can capture the meaning of the word "unhappy" more accurately, as it can understand the relationship between the negative prefix "un" and the positive word "happy."

## Encoding

Encoding involves converting text into numerical vectors that can be understood and processed by NLP models. Our model uses a technique called transformer-based encoding. It feeds the tokens into a deep neural network. Such a network consists of multiple layers of transformers, which are capable of processing text in a bidirectional manner, capturing the context of words in both directions. Finally, a numerical vector representing the meaning of the token based on the context of the surrounding words is outputed.

nlptown/bert-base-multilingual-uncased-sentiment provides an easy way for us to encode the text in a few lines.

''' tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment') tokens = tokenizer.encode(row['Text'], return_tensors='pt') '''

### Classification

Classification in sentiment analysis means assigning sentiment labels to text, such as positive, negative, or neutral. Once the input text has been encoded, the encoded text vectors are fed into the neural network classifier. The classifier applies a set of matrix transformations and nonlinear functions to the encoded text vectors to generate a prediction of the sentiment label. Specifically, the neural network takes the sequence of encoded text vectors as input and applies a series of dense layers, followed by a final softmax layer, to generate a distribution over the possible sentiment labels.

### Output

The model outputs a score representing the sentiment expressed in the text, ranging from 1 (most negative) to 5 (most positive). For example, the model outputs a score of 1 for "To me the worst quarter of the season so far is $MMM. Just dismal with a 'dry' January included", indicating that it expresses highly negative sentiment.

## Conclusion and reflections

Through this experience, we have realized the importance of the tokenization and encoding process in capturing the true meaning and context of the text. We found the nlptown/bert-base-multilingual-uncased-sentiment model to be a versatile and effective tool for analyzing sentiment in text across multiple languages. Using WordPiece tokenization and transformer-based encoding, it can accurately predict sentiment labels for input text in a variety of formats.

Owing to time constraints, we were not able to fine-tune the model on our data. We believe that fine-tuning the model on our data would improve its performance, as it would learn to recognize the jargons used in financial commentary and the style of Jim Cramer.