TOPIC ANALYSIS

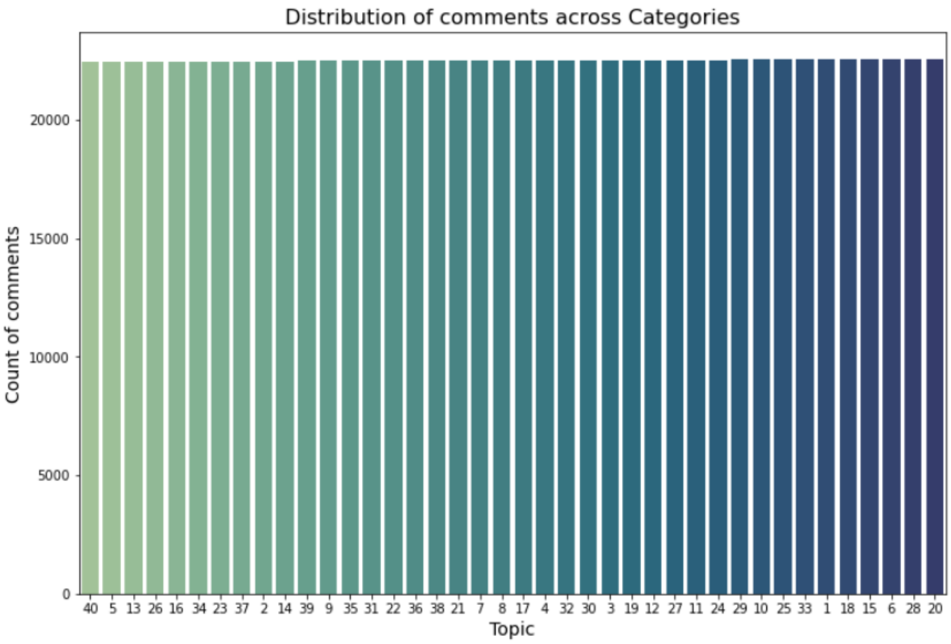
1.1 Introduction

The problem highlights the use of machine learning algorithms to categorize different comments scraped from an online platform and make relevant predictions about the topics associated with those comments. There are a total of 40 topics to classify these comments. Even though the problem seems like a simple classification problem, as we dive deeper to understand the data, we realize that the real problem asks us to make sense of the comments mentioned in the dataset and then assign categories. Since the number of topics/classes is much greater than any common classification problem, the expected accuracy won't be too high. These days, Topic Modeling and Classification have received tremendous popularity when analyzing products and services for various brands, during election times to measure popularity, discover public sentiments around multiple issues, etc. Primarily deriving meaningful topics from these comments is incredibly challenging because of variations in language, insertion of emojis, and use of partial and profane comments. It is essential to choose a scheme that translates the comments to word embeddings to calculate some similarity between those comments to assign relevant topics; it is also imperative to translate the context and meaning of those comments and cluster them to relevant topics.

There are multiple approaches to Topic Modeling, such as Latent Dirichlet Analysis (LDA) and Probabilistic Latent Semantic Analysis (LSA). These benchmark techniques utilized for such problems seem to provide viable results. The initial approach was to use Tf-Idf and Word2Vec to vectorize the comments and then use state-of-the-art classification techniques to assign topics to these vectors. When utilized, bag-of-Words with Tf-Idf and Word Embedding with Word2Vec would pose a significant hidden problem. The main problem with these approaches is that they treat the exact words with different meanings identically without adding any context to them. For example, the term "bank" in "Peter is fishing near the bank." and "Two people robbed the state bank on Monday." would have the same vectors in this representation. This approach would give us misleading results, and therefore, to improve the performance of our prediction mechanisms, it is essential to switch to a process that finds a way to translate the context of the words. Transformers: a reasonably new modeling technique, presented by Google's research professionals in their seminal paper "Attention is All You Need," tackles the exact problem. Google's BERT (Bidirectional Encoder Representations from Transformers) combines ELMO context embedding and several Transformers, plus it's bidirectional (which was a big novelty for Transformers). The vector assigned to a word using BERT is a function of the entire sentence; therefore, a word can have different vectors based on the context. ELMO is a word embedding technique that utilizes LSTMs to look at each sentence and then assigns those embeddings.

1.2 Problem Analysis

When dealing with textual data, the initial analysis began with understanding the type of comments provided. We were working with 900,000 comments in total, which belonged to 40 categories, replaced with numerical labels. We commenced our EDA by thinking of techniques that would perform the initial task of converting the words in each comment into their numerical representation. Understanding the distribution of the comments across the 40 categories was crucial too.



The nearly even distribution of comments encouraged us to proceed with the tokenizing strategies.

The initial strategy we followed for tokenizing was to implement NLTK's word tokenizer, word_tokenize(). This is often popularly used to segregate words in a sentence into a list of words, also known as 'tokens.' Cleaning up these tokens and purging them off of punctuation, numbers, special symbols, and stopwords (articles such as 'a', 'the') is necessary in order to get the crux of the sentence by capturing the words. Upon cleaning the data and tokenizing the important words, we obtained the following WordCloud:

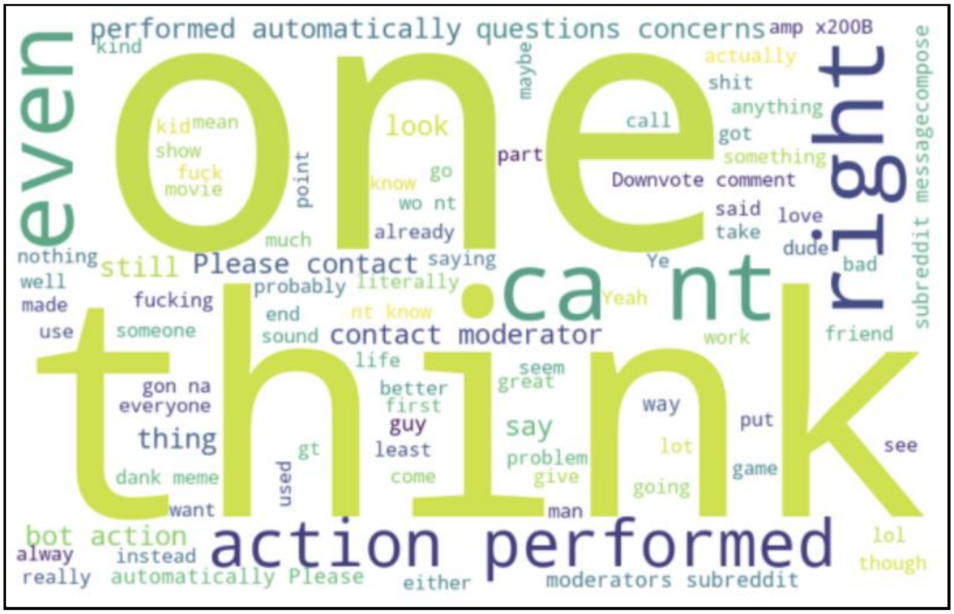


Fig: Word Cloud

Implementing an embedding strategy to prepare the text for training seemed the next logical step. We performed word embedding using Word2Vec as it is one of the most popular embedding techniques that attempts to find semantic and syntactic similarities and relations with other words. When transforming the generated text into vectors using Term Frequency-Inverse Document Frequency (TFIDF) to evaluate how important a particular word is in the given collection of comments, we did encounter computational and memory issues.

Consequently, we utilized a subset of the 900,000 comments as our dataset to perform the initial training analysis. We observed that models such as Linear Support Vector Machines (SVM), which usually perform well on smaller datasets, seemed to perform poorly on our dataset since it consisted of several categories. Owing to this, we explored a transformer-based approach for training, such as BERT.

In layman's terms, a transformer can be described as a model that uses the concept of self-attention to boost the speed with which the models can be trained, which makes it suitable for language understanding. For instance, when processing a text, "The animal didn't cross the street because it was too tired," it may be effortless for us to understand that the "it" in the given sentence stands for the animal; it is not so for our model. When the model is processing the word "it," self-attention allows it to associate "it" with "animal." Apart from this, another conspicuous advantage of using BERT is that it leverages bidirectional training. Bidirectional training simultaneously considers the previous and next tokens and learns text representation, which is crucial in our problem as we aim to classify the given comment in a specific category based on some specific words used in it. For tokenization, we first split the given dataset into an 80% train set and 20% validation set and used the BERT tokenizer based on WordPress. Encoding was performed to convert all the comments from their tokenized form to a corresponding encoded format using "batch_encode_plus". Setting the "return_attention_mask" parameter to True ensures that the attention mask is returned according to the specific tokenizer defined by the max_length attribute. We performed padding on the text and limited each incoming comment to a maximum length of 300. The results of these operations were tensors which were then split into input_ids, attention_masks, and labels, preparing them for model training. These similar steps were also replicated for the incoming test dataset.

The training protocol is interesting because unlike other recent language models BERT is trained to take into account language context from both directions rather than just things to the left of the word. In pretraining BERT masks out random words in a given sentence and uses the rest of the sentence to predict that missing word. Once a BERT pre-trained model is initialized using the BertModel class, additional layers can be added to act as classifier heads as needed. This is the same way other custom Pytorch architectures are created. For BERT we need to be able to tokenize strings and convert them into IDs that map to words in BERT's vocabulary. The mechanics for applying this come in the list of dictionaries the learning rates are specified to be applied to different parts of the network within the optimizer, in this case, an Adam optimizer.

Following is the model architecture followed for the BERT transformer used:

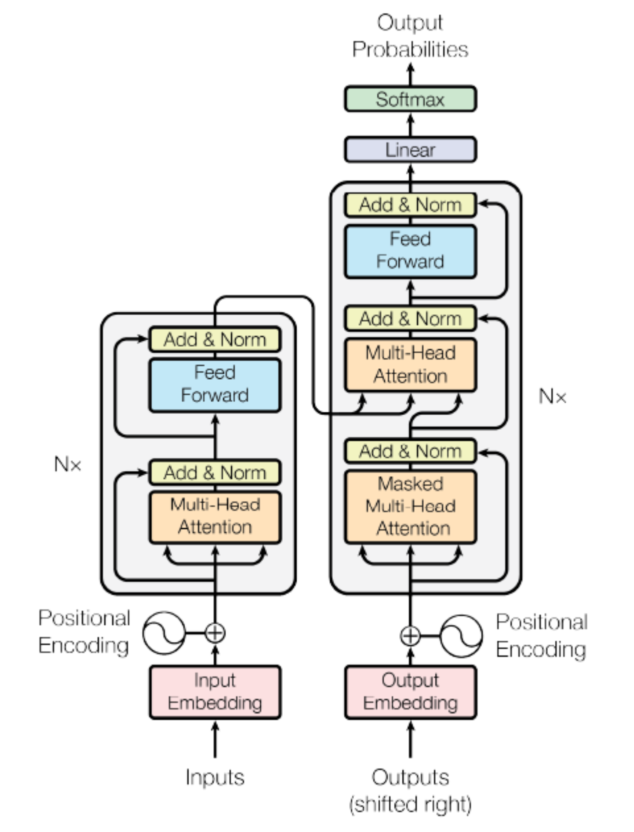
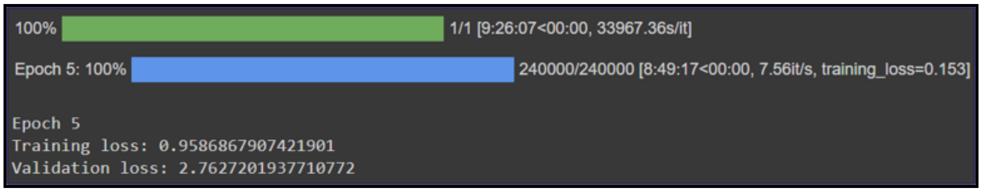


Figure 1: The Transformer - model architecture.

Fig: Transformer Architecture

We instantiate the DataLoaders for our train, test, and validation set using the tensor datasets created at the end of encoding and sample the data using RandomSampler. With the batch size set to 3, the model is let run for 5 epochs. We instantiate a model with from_pretrained(), with pre-trained encoder weights and model configuration copied from the bert-base-uncased model. This is useful because it allows us to make use of the pre-trained BERT encoder and easily train it on our sequence classification text dataset. We use the AdamW() optimizer and set the learning rate to 0.00001 and epsilon value to 0.00000001 which implements gradient bias correction as well as weight decay. The optimizer allows us to apply different model hyperparameters for specific parameter groups. We call the classification model within the evaluate function defined, for each batch of data with the input_ids, attention_mask, and labels argument. We get the logits and calculate the training loss between the predictions and the passed labels. Having already set up our optimizer, we can then do a backward pass and update the weights every time. We also provide a few learning rate scheduling tools. We set up a scheduler that warms up for num_warmup_steps and then linearly decays to 0 by the end of training. Finally, the model is run for 5 epochs and the predictions for each epoch are saved while calculating the validation and training losses in each step. Finally, the f1 score and accuracy are calculated for the predicted labels and we achieve a test accuracy of 50.84% for the test data passed in the trained classification model.

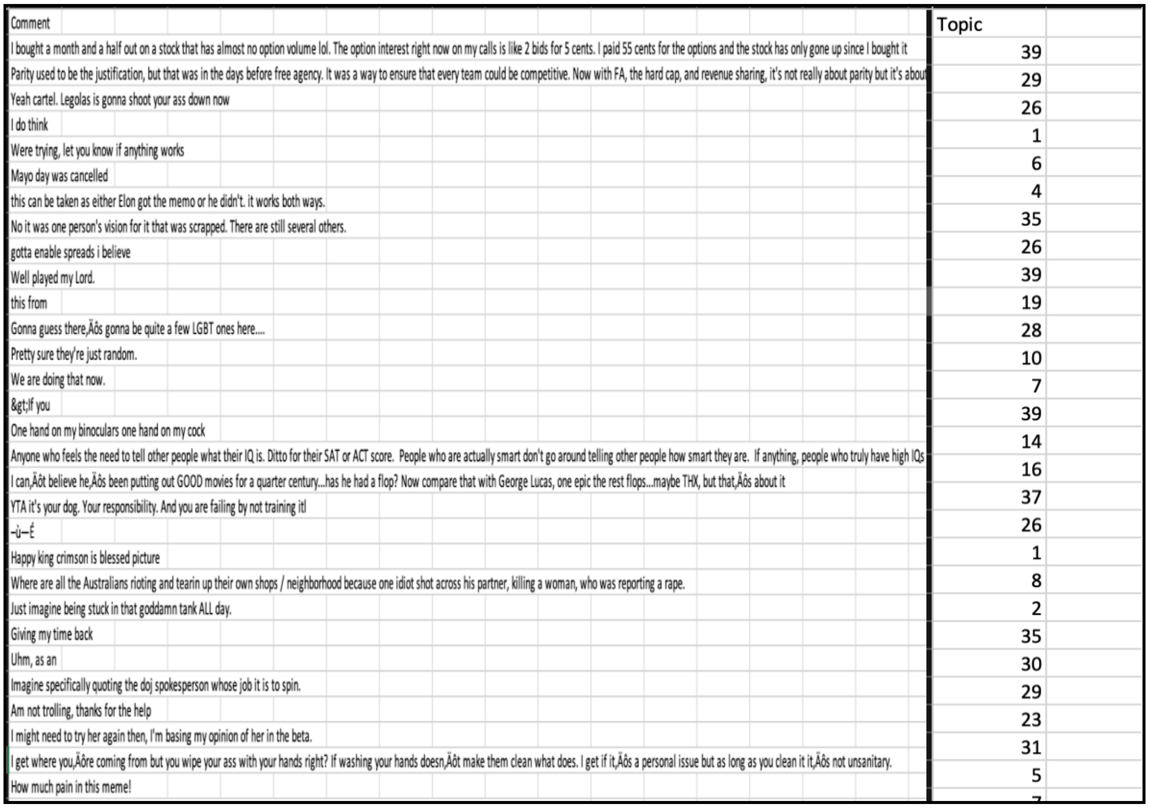


```
100%  1/1 [9:26:07<00:00, 33967.36s/it]
Epoch 5: 100%  240000/240000 [8:49:17<00:00, 7.56it/s, training_loss=0.153]

Epoch 5
Training loss: 0.9586867907421901
Validation loss: 2.7627201937710772
```

1.3 Conclusion

Attention guided BERT model performs relatively better as compared to the previously employed schemes when it comes to the classification of these comments into multiple topics. As stated, we get a test accuracy of 50.84% which can be considered great for these types of problems just because of the presence of a large number of categories. It is difficult to achieve very high accuracy because of the same and hence we can say that the model performs well in this case. Since it has only been trained for 5 epochs, we believe that increasing the number of epochs can lead to better results. Also, a test involving the removal of stopwords, links, and other redundant text and emojis from the text would make the data being fed to the model cleaner and thus we can expect better results in that case. We have used a smaller BERT model because it takes a lot of time to train the model according to the data we have, thus if we have access to the larger pre-trained model, we expect the accuracy may improve.



Comment	Topic
I bought a month and a half out on a stock that has almost no option volume lol. The option interest right now on my calls is like 2 bids for 5 cents. I paid 55 cents for the options and the stock has only gone up since I bought it	39
Parity used to be the justification, but that was in the days before free agency. It was a way to ensure that every team could be competitive. Now with FA, the hard cap, and revenue sharing, it's not really about parity but it's about	29
Yeah cartel. Legolas is gonna shoot your ass down now	26
I do think	1
Were trying, let you know if anything works	6
Mayo day was cancelled	4
this can be taken as either Elon got the memo or he didn't. it works both ways.	35
No it was one person's vision for it that was scrapped. There are still several others.	26
gotta enable spreads i believe	39
Well played my Lord.	19
this from	28
Gonna guess there,Äôs gonna be quite a few LGBT ones here....	10
Pretty sure they're just random.	7
We are doing that now.	39
&gt;If you	14
One hand on my binoculars one hand on my cock	16
Anyone who feels the need to tell other people what their IQ is. Ditto for their SAT or ACT score. People who are actually smart don't go around telling other people how smart they are. If anything, people who truly have high IQs	37
I can,Äôt believe he,Äôs been putting out GOOD movies for a quarter century...has he had a flop? Now compare that with George Lucas, one epic the rest flops...maybe THX, but that,Äôs about it	26
YTA it's your dog. Your responsibility. And you are failing by not training it!	1
-ù-É	8
Happy king crimson is blessed picture	2
Where are all the Australians rioting and tearin up their own shops / neighborhood because one idiot shot across his partner, killing a woman, who was reporting a rape.	35
Just imagine being stuck in that goddamn tank ALL day.	30
Giving my time back	29
Uhm, as an	23
Imagine specifically quoting the doj spokesperson whose job it is to spin.	31
Am not trolling, thanks for the help	5
I might need to try her again then, I'm basing my opinion of her in the beta.	7
I get where you,Äôre coming from but you wipe your ass with your hands right? If washing your hands doesn,Äôt make them clean what does. I get if it,Äôs a personal issue but as long as you clean it it,Äôs not unsanitary.	
How much pain in this meme!	