Multi Topic Classification using Transformers

Sumukh Nitundila
nitun001@umn.edu

Venkata Sai Krishna Abbaraju
abbar005@umn.edu

***Abstract*—With the amount of user-generated content on the internet, categorizing all the content on these websites is becoming increasingly difficult. Self-tagging has proven to be an unreliable method for users. As a solution, there is a need for automatic categorization of data based on its content. This project aims to achieve that using transformer-based models.**

***Index Terms*—AI, Machine Learning, Transformers, Natural Language Processing, BERT**

I. Introduction

The internet has revolutionized the amount of information available at our fingertips. To give an idea about the scale, more than 200 Million reviews are written on Yelp every single year and more than 3B pictures and videos are uploaded on Instagram every year [4]. This scale of data would have been unimaginable just a couple of decades ago. The biggest contributor to this explosion of data is the amount of user-generated content floating around on the internet. Be it political opinions on Twitter or cat videos on Instagram, uploading content has never been easier. But with all this data, finding relevant data that piques your interest can get very hard. One way to make it easier for people to find what they want is to categorize data. But every day categorizing data gets harder because of the massive scale.

One of the more popular websites for user-generated content was Yahoo Answers where any user can ask a question. And other users try to answer these. Our project attempts to categorize this data using sophisticated NLP methods and some of the most popular transformer models. Dataset consisting of this content was available on Kaggle [6] on which we tried to perform analysis.

In the next section, we will some of the literature about transformers and transformer-based models in our project

II. Literature Survey

In the last few years, there has been an explosion of generative AI products and companies. OpenAI's newest offering, ChatGPT, was the fastest product to reach 10M active users. Why has there been a sudden explosion in these AI products? The answer to this question is the concept of *Transformers*

A. Transformers

In a paper written by A.Vaswani, Noam Shazeer et.al, titled "Attention Is All You Need", the authors noted that most of the dominance sequence transduction models were based on RNNs or CNNs in an encoder-decoder combination [1]. But all the best-performing models also connect the encoder and decoder through an attention mechanism. So they propose the use of transformers, which make use of attention mechanisms and completely abandons recurrence and convolutions.

The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder.

In a transformer, attention is used in 3 different places [4]:

- Self-attention in the Encoder — the input sequence pays attention to itself
- Self-attention in the Decoder — the target sequence pays attention to itself
- Encoder-Decoder-attention in the Decoder — the target sequence pays attention to the input sequence

Why focus on attention? What does it solve? The problem with other *seqtoseq* models was that they were processing each word in sequence. The problem with this kind of processing is that by the time we reach the final word in the sequence, we might find something which adds context to a word we saw in the beginning of the sequence. Since this was the case, these AI models struggled to understand speech correctly. But with transformers, the whole sequence is processed in parallel at the same time. So context from each part of the sequence is available during the processing of every word. This helps the model understand and mimic human speech patterns much better.

B. BERT

The number of language tasks humans do is not varied. So researchers at Google proposed an idea to train a transformer model which they called Bidirectional Encoder Representations from Transformers (BERT) [2]. What makes BERT unique is the fact that it is designed to pre-train deep bidirectional representations from unlabeled text. It does so by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is trained on 2 things - Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM is a concept where in each sentence, a random word is masked and the AI learns to guess that random word. NSP is where you train the AI to guess the next sentence based on the previous text. To fine-tune this, we just need to swap out the input and output layers. This flexibility makes it an extremely popular tool used for a wide variety of Natural Language Processing (NLP) tasks.

C. RoBERTa

No matter how great a project is, human tendency is to challenge its greatness by attempting something even better. RoBERTa is a project by Meta (then Facebook) that was borne out of a similar ambition [3]. Liu et.al felt that BERT is severely undertrained for any complex NLP tasks. During training of BERT, researchers used MLM. This was scrutinized since the word that was masked was for a sentence. RoBERTa is trained using Dynamic Language Masking where during each epoch, a different word was picked to mask. For NSP, authors felt that having all sentences in the sequence being sourced from the same document is important since it helps the model understand additional context. They also increased training batch size and the increased the size of training dataset. These lead to improved results in SQuAD test compared to its predecessor.

In the next section, we will explain details about the data we used in our project

III. Data

The original dataset consists of 10 classes of different varieties like culture, education, sports, finance, business, etc. Each class here consists of 150k questions that users posted on the Yahoo Answers forum. we are solving a multi-class classification problem for 4 specific categories culture, computers, Sports, and politics with a sample of 5,000 questions from each topic.

A. Data preprocessing

We obtained the dataset from Kaggle and stored it in Postgres to extract a balanced sample of 5,000 questions from each of the classes. The question titles and question contents had a lot of web content like HTML tags, contractions, and math symbols to be handled. The other challenge was to filter non-English sentences since the dataset was multi-lingual. We used a hybrid approach by leveraging langid, langdetect and XLNet models to detect the language of the text and use a majority vote to select the final label. There were 15k English question titles, and 13k question titles and content that will be used for model training purposes.

- What is the average length of a question title by class (Fig.1).
  Questions related to computers tend to be lengthy followed by political questions. Whereas, questions on sports are short when compared to the other three categories.
- What is the average length of question titles combined with content by classes? (Fig.2)
  Questions on culture and computers tend to have lengthy titles and content while sports and politics have relatively smaller lengths.
- What are the most frequent words per class? (Fig.3, Tab:I)

In the next section, we discuss some of the methods we used to classify the data

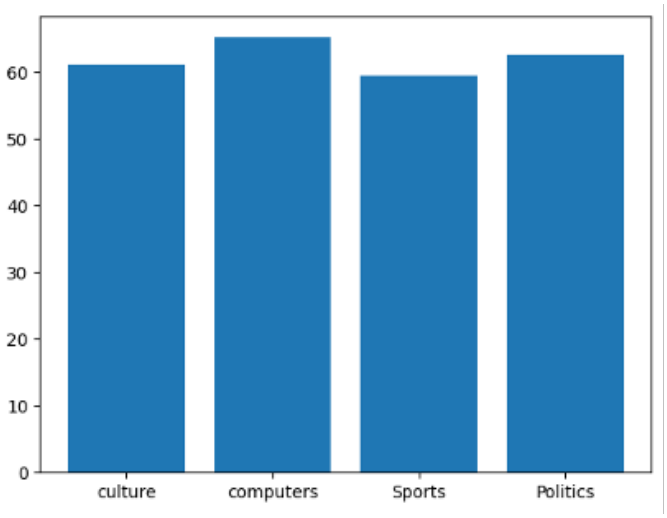

Fig. 1. Average Length of Questions

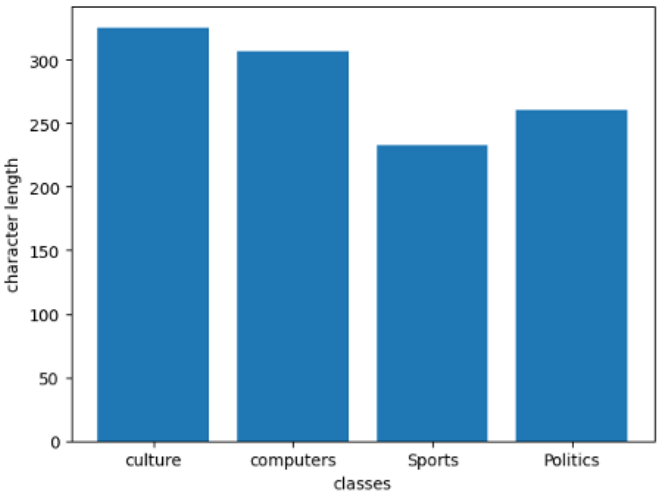

Fig. 2. Average Length of Content+Questions

Fig. 3. Word Cloud for classes 1) Culture (Top Left) 2) Computing (Top Right) 3) Sports (Bottom Left) 4) Politics (Bottom Right

Class	*Common Words in the Title*
Culture	Church, Christians, heaven
Computers	Software, testing, internet
Sports	Team, wrestling, game
Politics	License, global warming, scientist

TABLE I
MOST FREQUENTLY USED WORDS BY TOPIC

```
English text: where can i find stuff for truckers for their myspace page?
Word-piece:['[CLS]', 'where', 'can', 'i', 'find', 'stuff', 'for', 'truck', '##ers', 'for', 'their', 'myspace', 'page', '?', '[SEP]']
Byte-pair:['<s>', 'where', 'Ġcan', 'Ġi', 'Ġfind', 'Ġstuff', 'Ġfor', 'Ġtruck', 'ers', 'Ġfor', 'Ġtheir', 'Ġmy', 'space', 'Ġpage', '?', '</s>']
```

Fig. 4. Tokenizer breakdown

IV. METHODS

We are using NLP transformers to solve this multi-class text classification problem. Here is the outline of this section. We first understand how BERT was pre-trained and how to fine-tune it for a classification task.

A. Model Architecture

1) Tokenizer: NLP transformers take the input text in a defined format. To convert the input text to that format we use tokenizers. BERT and RoBERTa use a sub-word tokenization mechanism called Word-piece and byte-pair encoding respectively. For example, Figure 4 has sample text in English as perceived by both the algorithms,

Later these tokens are converted into numerical vectors using a vocabulary matrix, for BERT the size is 30522 X 768, which means each word has a 768-word embedding. Then to this segment and positional embeddings are added so that the transformer can preserve the temporal sequence of text. The positional embeddings in BERT are formed by applying sine and cosine on word embedding matrix. In our case, our embedding layer looks like Fig.5,

2) Encoder units in BERT: Now the embeddings are ready to be passed to the encoder block of BERT. BERT uses a multi-head attention layer where the attention is calculated multiple times (optimized for parallel execution) with varying weight matrices and concatenated. In Fig.6, we can observe in our fine-tuned BERT model the weights query, key, and value of each of 768 X 768 dimensions and there are 12 encoder units.

3) Final Layer in BERT: The output from the 12^{th} encoder block is a 768 X 768 vector, which has a classification head, the [CLS] special token. It is a 768-dimensional vector that has the entire contextual representation of the input sentence (Fig.7). Note that, BERT and RoBERTa have the same architecture except that, RoBERTa has a larger vocabulary size of 50265. The researchers at Facebook tweaked the pre-training process of BERT by excluding the next sentence prediction task and including a dynamic masking mechanism to optimize BERT.

```
(bert): BertModel(
  (embeddings): BertEmbeddings(
    (word_embeddings): Embedding(30522, 768, padding_idx=0)
    (position_embeddings): Embedding(512, 768)
    (token_type_embeddings): Embedding(2, 768)
    (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (dropout): Dropout(p=0.1, inplace=False)
  )
```

Fig. 5. BERT word embedding layer

```
(0-11): 12 x BertLayer(
  (attention): BertAttention(
    (self): BertSelfAttention(
      (query): Linear(in_features=768, out_features=768, bias=True)
      (key): Linear(in_features=768, out_features=768, bias=True)
      (value): Linear(in_features=768, out_features=768, bias=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
```

Fig. 6. BERT attention layer

```
  (pooler): BertPooler(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (activation): Tanh()
  )
)
(dropout): Dropout(p=0.1, inplace=False)
(classifier): Linear(in_features=768, out_features=4, bias=True)
```

Fig. 7. BERT pooling layer

B. Working

1) Fine Tuning Phase: Our hypothesis was to decide, if a question title is enough to decide the topic of a sentence or do we need the title of the question and its content. To approach this solution, we used BERT and RoBERTa for the sequence classification task in huggingface which has pre-trained BERT, RoBERTa. We fine-tuned it on Yahoo-Answers multi-class classification dataset. As stated in the above architecture, the [CLS] token layer is customizable for training purposes. In our project, we have used a fully connected linear layer of 768 X 4 dimensions where 4 represents the number of classes. Refer the Figure.8 to visualize it. The outputs are logits and we can either use argmax on logits or use a softmax to normalize as probabilities for prediction. We used cross entropy as loss function and Adam as optimizer which are used to adjust the weights of query, key, value, and fully connected layer during the backpropagation phase in the encoder network. We built 4 models of BERT and RoBERTa where each of the models takes - a title and a concatenation of title & context as inputs for this experiment. Table 2 shows the hyper-parameters we used to train the models.

Fig.9 illustrates the train and validation losses when models are trained on the title alone. In the case of BERT, Validation accuracy started to decrease after epoch 2. While RoBERTa performed well in the first epoch itself

Hyper parameter	*Value*
Learning rate	1e-4, 1e-5
Epochs	3
Loss function	Cross entropy
Optimizer	Adam

TABLE II
HYPERPARAMETERS

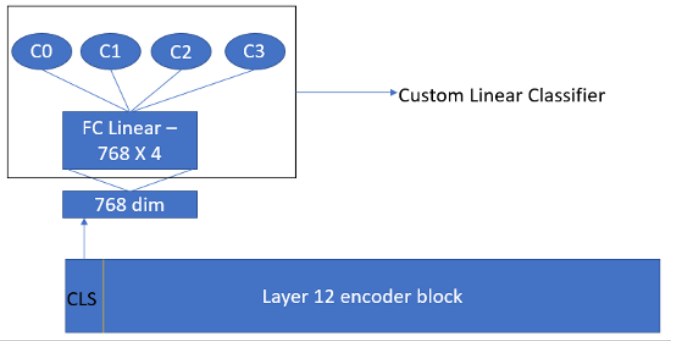

Fig. 8. BERT and RoBERTa architecture of classifier on CLS token

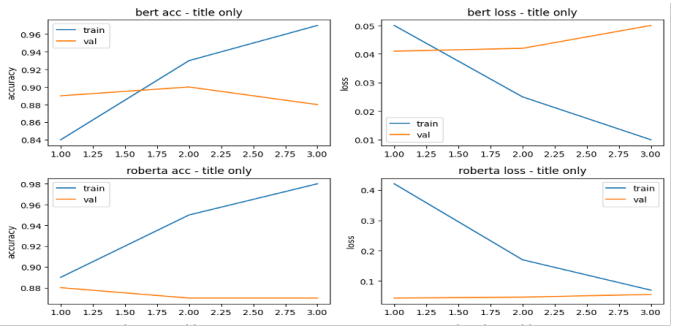

Fig. 9. Training and Validation accuracy when only Title is used

Fig.10 shows that both BERT and RoBERTa have done quite well on validation set till epoch 2. Another thing to be noted is that the accuracies of title and content models is more compared to title-only models.

V. Results

Our observation was that both BERT and RoBERTa performed better when the context is given compared to when only the title is given. Fig.11 shows the train and test set accuracies of the models. The best-performing model was BERT when it is trained on title and content. It was able to generalize well with training and test accuracies of 96% and 93% respectively. Also, we observed that BERT when trained on titles had a train accuracy of 93% but dropped to 87% on the test set. This could be because the title only has short sentences and they were not enough for the model to learn

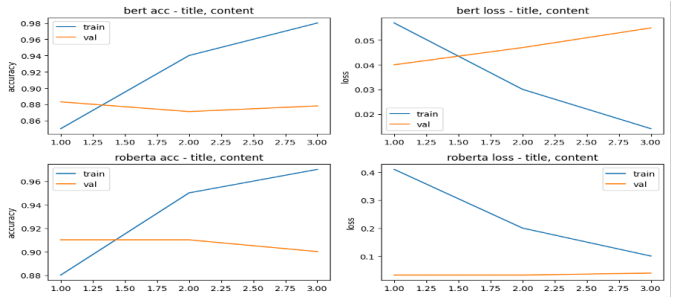

Fig. 10. Training and Validation accuracy when both Title+Content is used

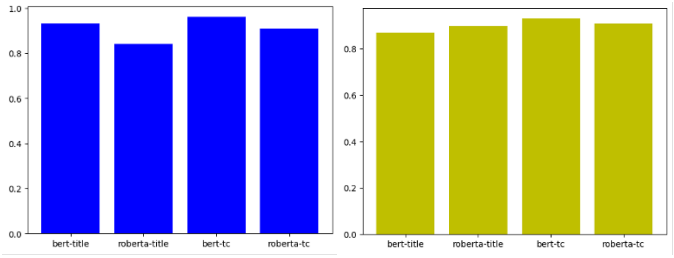

Fig. 11. Training vs Testing accuracy

Models	***BERT- title***		***RoBERTa-title***	
Topics	*accuracy*	*F1*	*accuracy*	*F1*
Culture	82	0.81	86	0.87
Computers	87	0.90	92	0.92
Sports	91	0.90	84	0.89
Politics	87	0.87	85	0.85
Overall Test set accuracy	87		87	

TABLE III
ACCURACY AND F1 ON TEST SET AT CLASS LEVEL BY MODELS TRAINED ON TITLE

and generalize well on unseen data. Although, 87% might not seem like a bad number for accuracy the point here was to understand how much giving question context improves the model.

From Table.IV, the BERT model with title and content has performed well on all classes where accuracy is more than 90%. While its counterpart, the baseline BERT trained on titles in Table.III, has relatively under performed. Similar is the case with RoBERTa model. Also, the F1 scores are equivalent to the accuracy here because the classes are balanced.

To understand the model predictions even better, we used explainable AI tools like Integrated gradient. Thankfully, this is readily available in the transformers-interpret package [7]. We were able to answer the following questions with its help. *Note: The True label is one-hot encoded where 0: 'culture'; 1: 'computers'; 2: 'sports'; 3: 'political'.*

1) How to perceive the outputs of the two models?

If we understand what words the model pays attention to, we can understand how well the model has learned. For example, Where did BERT do well and ROBERTA fail?

In Fig.12 we can observe that, BERT perceived the text correctly and concentrated on word 'won' which is closely related to sports. However, in Fig.13, RoBERTa incorrectly

Models	***BERT***		***RoBERTa***	
Topics	*accuracy*	*F1*	*accuracy*	*F1*
Culture	91	0.88	85	0.87
Computers	97	0.96	96	0.96
Sports	93	0.94	93	0.92
Politics	91	0.92	91	0.90
Overall Test set accuracy	93		91	

TABLE IV
ACCURACY AND F1 ON TEST SET AT CLASS LEVEL BY TITLE + CONTENT MODELS

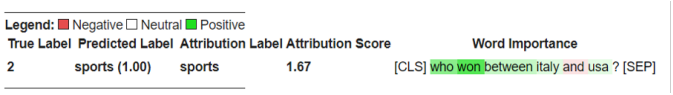

Fig. 12. BERT-title correct prediction

Fig. 13. RoBERTa title incorrect prediction

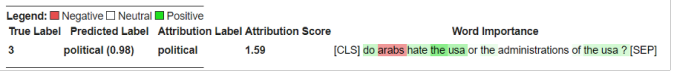

Fig. 14. BERT- title with content model

classified it as political by concentrating on countries.

2) Is there any bias detected?

In the example in Fig.14, it can be clearly seen that, Arabs is attributed negatively and the USA impacts the label positively. This could be due to inherent bias in the pre-trained corpus.

VI. Conclusions

Based on the current transformer's advancements in the field of NLP we decided to use them to solve this 4-class classification problem. After our analysis, we conclude that having both question title and question content together helps BERT and RoBERTa to perform better than only training on the title. It should also be worth noting that these statistics are on 20,000 samples on 4 categories. If we retrain the models on a larger dataset with a few more categories, we may be able to derive more conclusions that can help address this classification problem.

VII. Future Work

In the future, we can try out models that tokenize at sentence level like ALBERTA, or XLNet, a regressive model to solve this problem, and compare the performances on larger datasets with a few more categories

VIII. Ethics

The advent of Large Language Models, that are trained on large text datasets, has transformed the way problems are solved on text data. However, one should always note that there will be an inherent bias in almost every LLM as seen in Fig14. This bias could exist because the model was only trained on a particular set of data and did not have diversity in terms of country, gender, political beliefs, etc. These factors affect the task at hand. Hence, there needs to be extensive testing to understand the behavior of any model.

IX. Acknowledgments

Special thanks to Prof. Jaideep Srivastava for the guidance and his valuable input.

References

[1] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[2] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[3] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

[4] https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853

[5] https://www.statista.com/topics/1716/user-generated-content/

[6] https://www.kaggle.com/datasets/jarupula/yahoo-answers-dataset

[7] Kokhlikyan, Narine, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov et al. "Captum: A unified and generic model interpretability library for pytorch." arXiv preprint arXiv:2009.07896 (2020).