Personality Identification using BERT Variates

Vijay Ram Enaganti (PES1UG20CS700),
Samyam Narayan (PES1UG20CS715),
Shal Ritvik Sinha (PES1UG20CS717)
Department of Computer Science Engineering
PES University, Ring Road Campus, Bengaluru.

ABSTRACT

Social media in recent times is growing to be an integral resource to understand people and learn more about the world at a faster pace. People from different parts of the world interact to know more about different cultures, trends, and lifestyles. There exist different online communities which are based on their common interests like gaming, food, travel, etc. Majority of these interactions are based on text. Large volumes of text data are created on the internet every second which are considered to be valuable sources of information, both good and bad. Data Scientists, Researchers, Developers and many more people are interested to gain useful insights and deduce important relations between different day-to-day parameters. These insights can be obtained using breakthrough technologies of recent times like, Machine Learning, Deep Learning, Neural Networks, etc which are mainly used to detect and learn from different patterns. Patterns in text reveal details regarding a person's mindset or personality which can be put to use in various fields. Tools like Tensorflow, Numpy, Pandas, Keras, Scipy, etc, help in accurately identifying patterns and predicting outcomes of different scenarios.

KEYWORDS

Myers Briggs Type Indicator, MBTI Personality Test, BERT Classifier, Feature Extraction, RoBERTa Classifier, XLM Classifier.

INTRODUCTION

Personality identification is a field of study that aims to understand and predict an individual's psychological traits and characteristics based on their behavior, attitudes, and beliefs. The Meyer Briggs Type Indicator (MBTI) is a commonly used tool for assessing and categorizing personalities into 16 different types, based on four dimensions of psychological preferences: extraversion-introversion, sensing-intuition, thinking-feeling, and judging-perceiving.

Recently, deep learning models such as BERT (Bidirectional Encoder Representations from Transformers) have been applied to the task of personality identification identification, using text data as input. BERT models have shown promising results in natural language processing tasks, and their ability to capture contextual information and relationships between words can make them well-suited for personality identification.

In this paper, we propose the use of BERT models for personality identification using the MBTI. We present a review of existing research on personality identification using BERT models, and discuss the potential advantages and challenges of using these models for the MBTI. We also propose future directions for research in this area, including fine-tuning BERT models on larger and more diverse datasets, and exploring the use of advanced techniques such as transfer learning and multi-task learning.

LITERATURE SURVEY

1. The Relationship between the Myers-Briggs Personality Types and Learning Styles.

Introduction:

The Myers-Briggs Type Indicator is the most widely used personality measure for non-psychiatric populations. The indicator involves four preferences, each of which has two sides. They include Extravert vs. Introvert, Sensing vs. Intuitive, Thinking vs Feeling, and Judgment vs. Perception. Through the analysis of the answers on the MBTI, a type is assessed for each individual. There are sixteen types, each being a combination of the four preferences.

Keywords:

Extravert, Introvert, Sensing, Intuitive, Thinking, Feeling, Learning Style,

Result:

The MBTI is used in counseling, in business and industry, in public schools, and at colleges and universities. It has specifically been proven to be useful in educational purposes.

They can help in parenting, working with others whether in paid or voluntary work, and generally in managing relationships with others.

2. EmoDet2: Emotion Detection in English Textual Dialogue using BERT and BiLSTM Models

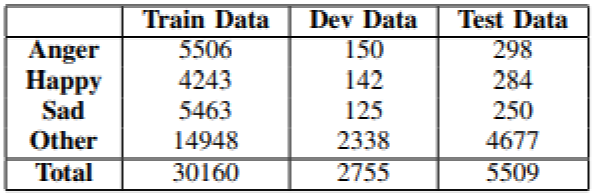
Introduction:

There exist many emotions and listing out the core / base emotions is a tough task as different people consider different emotions as core / base emotions. Performing emotion analysis on text is another challenge as mapping and analyzing with facial expressions is not possible. The rapid growth of Text communication and usage of social media calls for accurate emotion detection for various useful purposes. This can be done using different machine learning and deep learning methods. These methods not only help in detection but help in predicting emotions and sentiments. Features extraction for one of the methods used in this paper is done by using a combination of GloVe Word embeddings, BERT embeddings and psycholinguistic features from Affective Tweets and Weka Package. The proposed system combines a fully connected neural network architecture and BiLSTM neural network. EmoDet2, can determine the emotion and sentiment in English textual dialogue and classify it into four categories (Happy, Sad, Angry and Other).

Methodology:

Data and Pre-Processing

TRAINING AND TESTING DATASETS



	Train Data	Dev Data	Test Data
Anger	5506	150	298
Happy	4243	142	284
Sad	5463	125	250
Other	14948	2338	4677
Total	30160	2755	5509

No standard preprocessing techniques applied on the dataset for better performance of the BERT model. Emojis have been converted to text and spelling mistakes are handled using Ekphrasis.

Feature Vector Extraction

Step 1: A 300 dimensional vector using pre-trained word2vec embedding and a 300 dimensional vector using GloVe Embedding model along with a 173 dimensional vector from BERT Embedding have been extracted.

Step 2: Extracted the semantic features by converting the whole conversation to 145-dimensional vector using three vectors from the AffectiveTweets Wekapackage as follows:

- 43 features have been extracted using the TweetToLexiconFeatureVectorAttribute that calculates attributes for sentences using a variety of lexical resources.
- A two-dimensional vector using the SentimentStrength features from the same package.
- A 100-dimensional vector is obtained by vectorizing the sentence to embedding attributes.

Network Architecture

EmoDet2 was built using ensembling methods with different submodels:

- EmoDense
- EmoDet-BiLSTM
- EmoDet-BERT-BiLSTM (Cased)
- EmoDet-BERT-BiLSTM (Uncased)

Summary of each model is as below:

1. EmoDense
   - This submodel uses feed forward neural network
   - Number of hidden layers: 4
   - Number of neurons in layers: 512, 256, 128, 64
   - Activation function: ReLU
   - Optimizer: Adam
   - Learning Rate: 0.0001
   - Loss Function: MSE
   - Number of Epochs: 40
   - Batch size: 16
   - Validation split: 0.33

2. EmoDet-BiLSTM
   a. BiLSTM
   - The Bidirectional Long Short-Term Memory (BiLSTM) is the advanced form of LSTM in which the BiLSTM feeds the algorithm with the data once from beginning to the end, and once from the end to the beginning.
   - Number of layers: 2
   - Number of neurons in layers: 256, 256

   b. Neural Network
   - Number of hidden layers: 4
   - Number of neurons in layers: 512, 256, 128, 64

- Activation Function: ReLU
- Optimizer: Adam
- Learning Rate: 0.0001
- Loss Function: MSE
- Epochs: 100
- Batch size: 32
- Validation split: 0.33

3. EmoDet-BERT-BiLSTM
   - The BERT system can get better information from the original data than using the processed data. So first, we have extracted BERT embeddings then fed them into two BiLSTM layers.

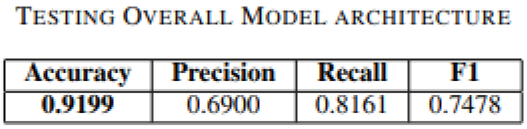
   a. BiLSTM
   - Number of layers: 2
   - Number of neurons in layers: 128, 128

   b. Neural Network
   - Number of hidden layers: 4
   - Number of neurons in layers: 512, 256, 128, 64
   - Activation function: ReLU
   - Optimizer: Adam
   - Learning rate: 0.0001
   - Loss function: MSE
   - Epochs: 100
   - Batch size: 128
   - Validation split: 0.2

All the models are ensembled with different combinations and tested for better performance in identifying and classifying emotions in text.

Result:

- The performance analysis of the overall ensembled model showed an accuracy approx. 92%.
- The model resulted in a F1 score of 0.7478 which is way higher than the F1 score of the baseline model considered in this paper



Testing Overall Model Architecture

Accuracy	Precision	Recall	F1
0.9199	0.6900	0.8161	0.7478

3. Emotion and sentiment analysis of tweets using BERT

Introduction

Emotion detection is one of the most challenging tasks
In the automated understanding of language. Understanding human emotions is a complicated task. Thanks to social networking sites , a large amount of publically available user generated data is available which can be analyzed in order to determine people's emotions and opinions. Performing emotion and sentiment analysis will help us in categorizing and predicting emotions and sentiments . We define two separate classifiers for the two tasks and we
evaluate the performance of the obtained models on real-world
tweet datasets.

Methodology:

1. Model creation

   The model is formed by fine tuning BERT on specific datasets of tweets developed for such tasks. Since tweets usually contain words that are irrelevant for text classification, a text preprocessing phase is needed in order to remove:

   - Mentions

- URLs
- Retweets

After the preprocessing phase the data can be used as input to train task specific BERT based models. The reference model used in this work is the BERT-Base, both in the uncased and the cased version. The uncased version implies that text is converted to lowercase before the word tokenization process and accents are ignored.

2. Experimental setting

   Parameters Observed:
   - Classification accuracy
   - F1 score

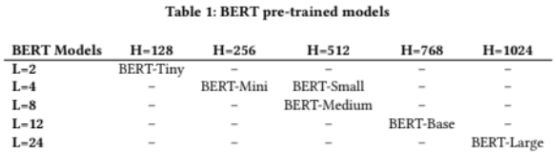


Table 1: BERT pre-trained models

BERT Models	H=128	H=256	H=512	H=768	H=1024
L=2	BERT-Tiny	–	–	–	–
L=4	–	BERT-Mini	BERT-Small	–	–
L=8	–	–	BERT-Medium	–	–
L=12	–	–	–	BERT-Base	–
L=24	–	–	–	–	BERT-Large

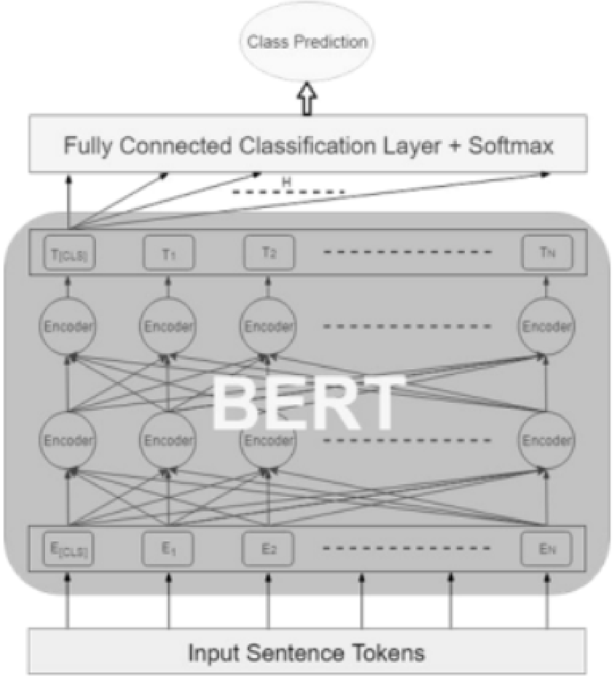
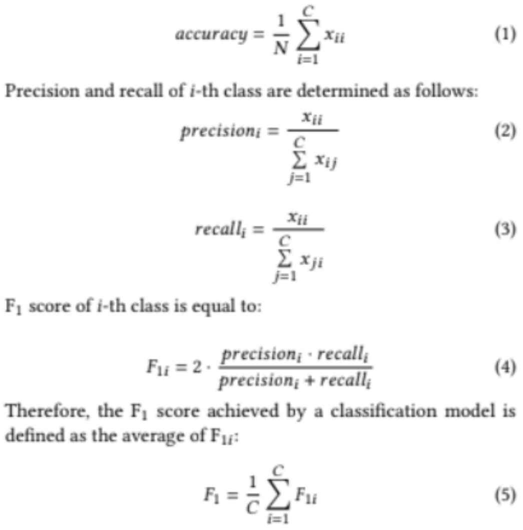


Figure 1: The architecture of the proposed classification model.



$$accuracy = \frac{1}{N}\sum_{i=1}^{C} x_{ii} \tag{1}$$

Precision and recall of *i*-th class are determined as follows:

$$precision_i = \frac{x_{ii}}{\sum\limits_{j=1}^{C} x_{ij}} \tag{2}$$

$$recall_i = \frac{x_{ii}}{\sum\limits_{j=1}^{C} x_{ji}} \tag{3}$$

F_1 score of *i*-th class is equal to:

$$F_{1i} = 2 \cdot \frac{precision_i \cdot recall_i}{precision_i + recall_i} \tag{4}$$

Therefore, the F_1 score achieved by a classification model is defined as the average of F_{1i}:

$$F_1 = \frac{1}{C}\sum_{i=1}^{C} F_{1i} \tag{5}$$

3. Emotion analysis
   - undersampling technique.

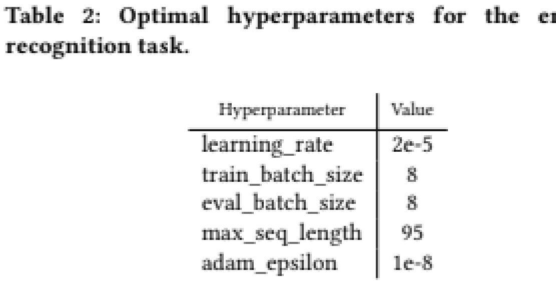


Table 2: Optimal hyperparameters for the em recognition task.

Hyperparameter	Value
learning_rate	2e-5
train_batch_size	8
eval_batch_size	8
max_seq_length	95
adam_epsilon	1e-8

4. Sentiment analysis
   - hyperparameter tuning through a grid

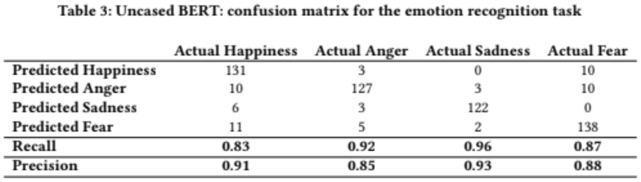


Table 3: Uncased BERT: confusion matrix for the emotion recognition task

	Actual Happiness	Actual Anger	Actual Sadness	Actual Fear
Predicted Happiness	131	3	0	10
Predicted Anger	10	127	3	10
Predicted Sadness	6	3	122	0
Predicted Fear	11	5	2	138
Recall	**0.83**	**0.92**	**0.96**	**0.87**
Precision	**0.91**	**0.85**	**0.93**	**0.88**

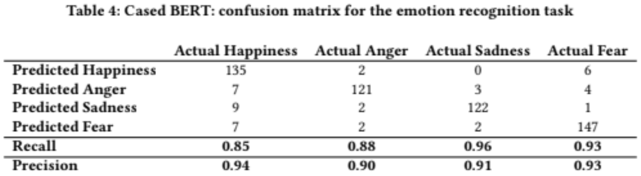


Table 4: Cased BERT: confusion matrix for the emotion recognition task

	Actual Happiness	Actual Anger	Actual Sadness	Actual Fear
Predicted Happiness	135	2	0	6
Predicted Anger	7	121	3	4
Predicted Sadness	9	2	122	1
Predicted Fear	7	2	2	147
Recall	**0.85**	**0.88**	**0.96**	**0.93**
Precision	**0.94**	**0.90**	**0.91**	**0.93**

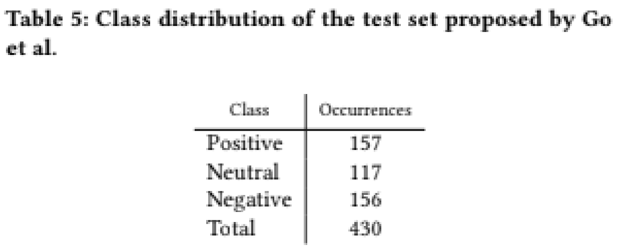


Table 5: Class distribution of the test set proposed by Go et al.

Class	Occurrences
Positive	157
Neutral	117
Negative	156
Total	430

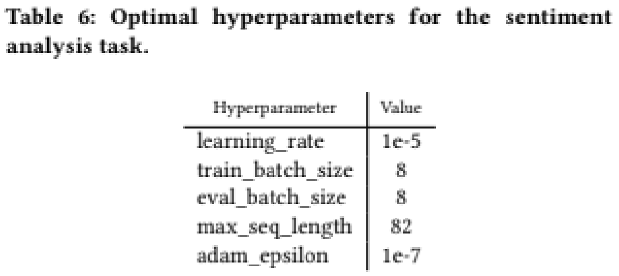


Table 6: Optimal hyperparameters for the sentiment analysis task.

Hyperparameter	Value
learning_rate	1e-5
train_batch_size	8
eval_batch_size	8
max_seq_length	82
adam_epsilon	1e-7

Result:

- 92% accuracy for sentiment analysis
- 90% accuracy for emotion analysis

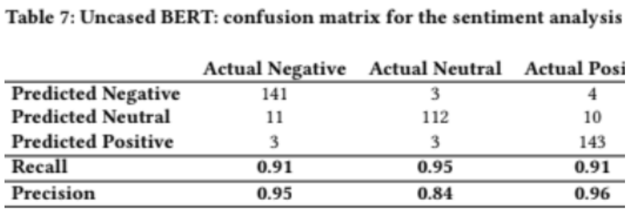


Table 7: Uncased BERT: confusion matrix for the sentiment analysis

	Actual Negative	Actual Neutral	Actual Posi
Predicted Negative	141	3	4
Predicted Neutral	11	112	10
Predicted Positive	3	3	143
Recall	**0.91**	**0.95**	**0.91**
Precision	**0.95**	**0.84**	**0.96**

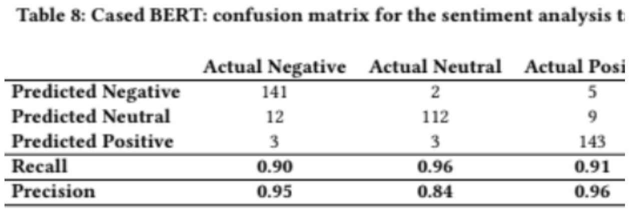


Table 8: Cased BERT: confusion matrix for the sentiment analysis t

	Actual Negative	Actual Neutral	Actual Posi
Predicted Negative	141	2	5
Predicted Neutral	12	112	9
Predicted Positive	3	3	143
Recall	**0.90**	**0.96**	**0.91**
Precision	**0.95**	**0.84**	**0.96**

4. COVID-Twitter-BERT: A Natural Language Processing Model to Analyze COVID-19 Content on Twitter

Introduction:

Twitter has been a valuable source of news and a public medium for expression during the COVID-19 pandemic. However, manually classifying, filtering and summarizing the large amount of information available on COVID-19 on Twitter is impossible and has also been a challenging task to solve with tools from the field of machine learning and natural language processing (NLP). Transformer-based models have changed the landscape of NLP. Models such as BERT, RoBERTa and ALBERT are all based on the same principle – training bidirectional transformer models on huge unlabelled text corpuses. This process is done using methods such as mask language modeling (MLM), next sentence prediction (NSP) and sentence order prediction (SOP). Different models vary slightly in how these methods are applied, but in general, all training is done in a fully unsupervised manner. The model is based on the BERT-LARGE (English, uncased, whole word masking) model. BERT-LARGE is trained mainly on raw text data from Wikipedia and a free book corpus.

Keywords:

Natural Language Processing, COVID-19, Language Model, BERT

Method:

The CT-BERT model is trained on a corpus of 160M tweets about the coronavirus collected through the Crowdbreaks platform during the period from January 12 to April 16, 2020. Crowdbreaks uses the Twitter filter stream API to listen to a set of COVID-19-related keywords in the English language. Prior to training, the original corpus was cleaned for retweet tags. Each tweet was pseudonymised by replacing all Twitter usernames with a common text token. A similar procedure was performed on all URLs to web pages. We also replaced all unicode emoticons with textual ASCII representations (e.g. :smile: for ,) using the Python emoji library. In the end, all retweets, duplicates and close duplicates were removed from the dataset, resulting in a final corpus of 22.5M tweets that comprise a total of 0.6B words.

The domain-specific pretraining dataset therefore consists of 1/7th the size of what is used for training the main base model. Tweets were treated as individual documents and segmented into sentences using the spaCy library. All input sequences to the BERT models are converted to a set of tokens from a 30000-word vocabulary. As all Twitter messages are limited to 280 characters, this allows us to reduce the sequence length to 96 tokens, thereby increasing the training batch sizes to 1024 examples.

We use a dupe factor of 10 on the dataset, resulting in 285M training examples and 2.5M validation examples. A constant learning rate of 2e-5, as recommended on the official BERT GitHub4 when doing domain-specific pre-training. Loss and accuracy were calculated through the pretraining procedure. For every 100,000 training steps, we therefore save a checkpoint and finetune this towards a variety of downstream classification tasks.

Result:

All metrics considered improve throughout the training process. The improvement on the MLM loss task is most notable and yields a final value of 1.48. The NSP task improves only marginally, as it already performs very well initially. Training was stopped at 500 000, an equivalent of 512M training examples, which we consider as our final model. This corresponds to roughly 1.8 training epochs. All metrics for the MLM and NLM tasks improve steadily throughout training. However, using loss/metrics for these tasks to evaluate the correct time to stop training is difficult.

Amongst runs on the same model and dataset, some degree of variance in performance was observed. This variance is mostly driven by runs with a particularly low performance. We observe that the variance is dataset dependent, but it does not increase throughout different pre-training checkpoints and is comparable to the variance observed on BERT-LARGE (pre-training step zero).

5. Fake News Classification using transformer based enhanced LSTM and BERT

Introduction:

Fake News is the misinformation disseminated among the public by mainstream sources like media outlets and social media and it is often misleading a large proportion of people in a society. It has been a topic of attention because this has been affecting our lives in various ways as there have been many incidents that have demonstrated the same very clearly. The concerns raised by it have only escalated with the constantly increasing time of people being spent on social media and thus being the main source of news for them. Classifying fake news is becoming increasingly difficult and in turn to assess the legitimacy of the news manually. Thus, recently computer researchers have been attempting to automate the process. This also motivated the authors to propose a model to automate the process and identify fake news patterns in news articles and media. To improvise upon this, the proposed research used contextual word embedding using the BERT model. BERT has performed well in the NLP tasks because of its intricate structure and excellent nonlinear representation learning capability.

Methodology:

Data Preprocessing:

Techniques applied:

- Case transformation
- Removed mentions
- Removed punctuations except “?”
- Removed special characters
- Removed stopwords
- Removed whitespaces

Model:

BERT

- BERT (Bidirectional Encoder Representations from Transformers) has been made up of a transformer attention mechanism that learns contextual relationships among words.
- BERT has many versions of pre-trained models for different use cases. Two of the most used models are-
  - BERT-base: 12 encoder stack layers + 768 hidden units + 12 multihead attention heads: 110M parameters.
  - BERT-large: 24 encoder stack layers + 1024 hidden units + 16 multihead attention heads: 340M parameters

- The input data needs to be converted into an appropriate format before using the pre-trained model. Relevant embeddings for each sentence has been obtained.

LSTM

- Long short-term memory is a type of RNN that can learn long-term dependencies.
- The chain-like structure of LSTMs is similar to RNNs, but the base module that makes up the LSTM is structurally distinct from other RNNs.

Model Architecture:

- The BERT-base-uncased model has been used with a feed-forward network of 768 hidden sizes.
- BERT tokenizer has been used to perform this task which takes input sequence as [CLS] and [SEP] concatenated to sentence at the beginning and end respectively which gives the input ids and attention masks as output.
- BERT provides contextualized sentence-level representations, which help LSTM to understand sentence semantics better.

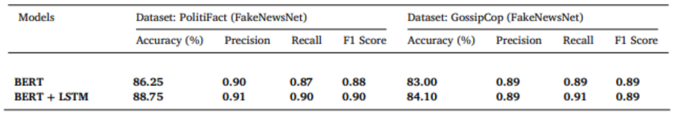
- A feed-forward linear layer with a size of 128 has been included in the classifier.

Result:

The model was experimented with 2 different datasets:

- PolitiFact
- GossipCop

The results for each of the datasets obtained are:



Models	Dataset: PolitiFact (FakeNewsNet)				Dataset: GossipCop (FakeNewsNet)			
	Accuracy (%)	Precision	Recall	F1 Score	Accuracy (%)	Precision	Recall	F1 Score
BERT	86.25	0.90	0.87	0.88	83.00	0.89	0.89	0.89
BERT + LSTM	88.75	0.91	0.90	0.90	84.10	0.89	0.91	0.89

6. Comparative Analysis of Transformer Based Pre-Trained NLP Models

Introduction

Recent advances in Transfer learning have revolutionized the Deep learning methods in the domain of Natural language processing (NLP). A Transformer is a simple network architecture that connects the encoder and decoder through an attention mechanism. Transformer based self-supervised pre-trained models have transformed the concept of Transfer learning in Natural language processing (NLP) using Deep learning approach.Self-attention mechanism made transformers more popular in transfer learning across a broad range of NLP tasks. The aim of this project is to identify the best pre-trained model for Sentiment analysis on a given dataset. We are considering BERT, RoBERTa, and ALBERT for this study.

Methodology

We used the Pytorch framework for building deep learning models with the help of Hugging face transformers.

Methods:

- BERT
  We fine-tuned the BERT model on pre-processed tweets data using a dropout layer, a hidden layer, a fully connected layer and a SoftMax layer for classification on top of BERT embeddings.

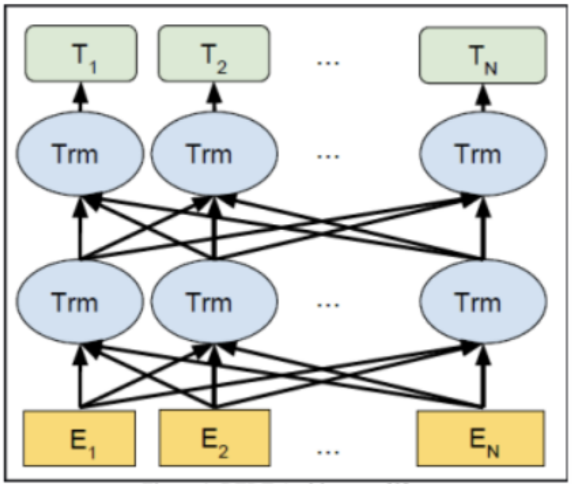


Figure 1. BERT Architecture [2]

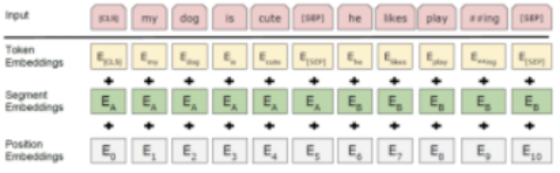


Figure 2. BERT Input Representation

- RoBERTa
  RoBERTa is an optimized BERT model. It uses a dynamic mask strategy where it generates a masking pattern every time it feeds a sequence to the model, but this is not the case in BERT, wherein masking was performed once during data pre-processing, resulting in a single static mask.

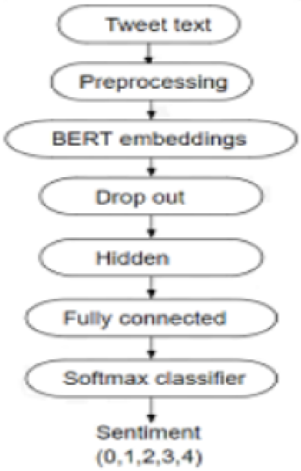


Figure 3. Proposed Method for BERT

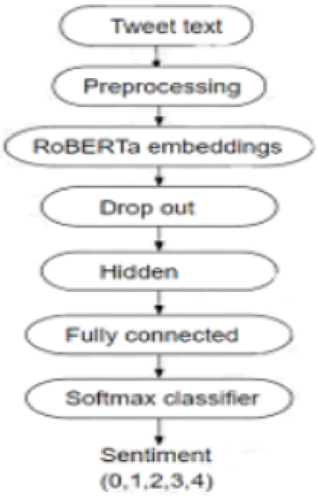


Figure 4. Proposed Method for RoBERTa

- ALBERT
  It is a light version of BERT. We have fine-tuned this model with proposed method on preprocessed tweets data using a dropout, a fully connected layer and finally a SoftMax on top of ALBERT embeddings

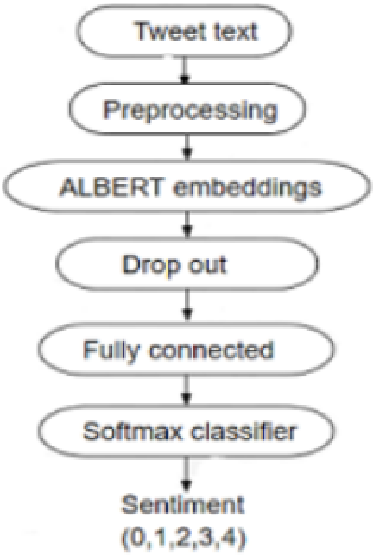
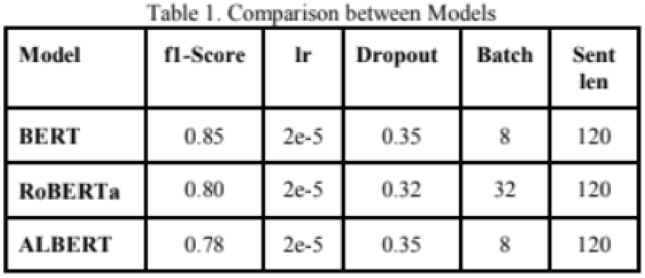


Figure 5. Proposed Method for ALBERT

Result:

Sentiment analysis :

Table 1. Comparison between Models



Model	f1-Score	lr	Dropout	Batch	Sent len
BERT	0.85	2e-5	0.35	8	120
RoBERTa	0.80	2e-5	0.32	32	120
ALBERT	0.78	2e-5	0.35	8	120

BERT

- Batch size : 8
- Drop out : 0.35
- Class 0 has high AUC compared to all other models .
- BERT has difficulty in classifying class 2 and 4 correctly.

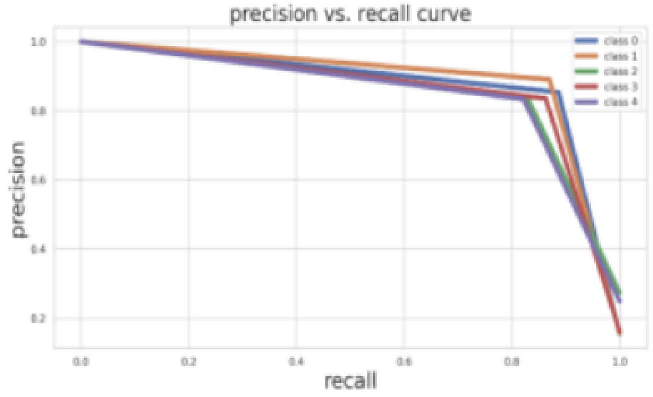


Figure 6. Precision-Recall Curve for BERT

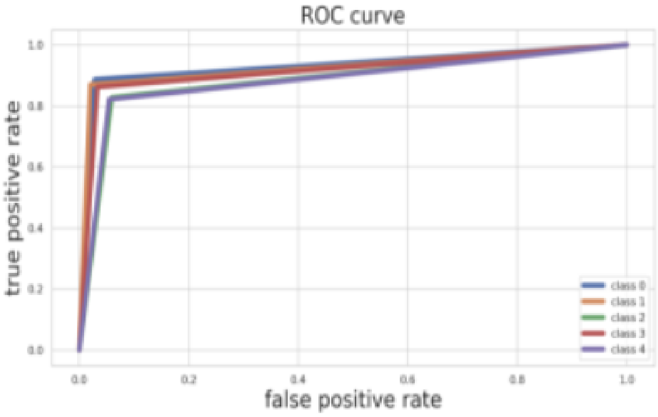


Figure 7. ROC Curve for BERT

RoBERTa

- Batch size : 32
- Drop out : 0.32
- Class 3 has high precision and low recall
- Class 0 has high precision and high recall. This means that class 0 classifies well with low misclassification error among all classes.
- The RoBERTa model also has difficulty in classifying class 2 and 4 correctly.

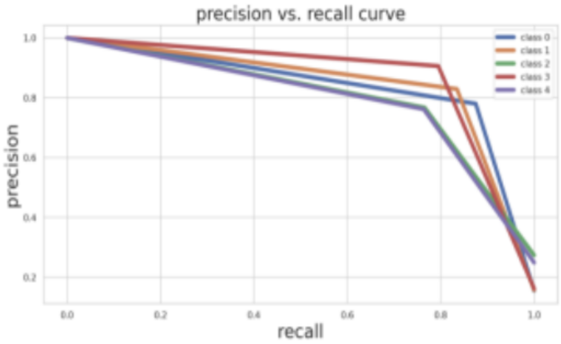


Figure 8. Precision-Recall Curve for RoBERTa

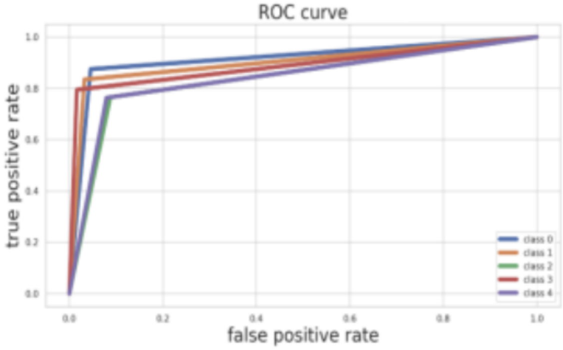


Figure 9. ROC Curve for RoBERTa

ALBERT

- class 3 has the highest f1-score and class 4 has lowest f1-score
- Class 3 has the highest AUC compared to all classes.
- The ALBERT model is not able to classify class 2 and 4 correctly.

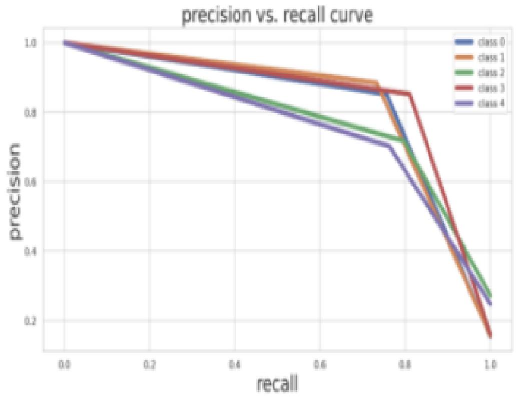


Figure 10. Precision-Recall Curve for ALBERT

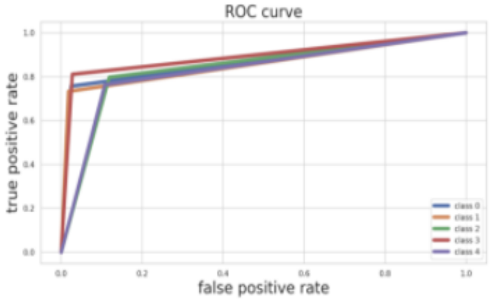


Figure 11. Recall Curve for ALBERT

We obtained the best results for BERT with a high training time (batch size=8).

The RoBERTa model achieves acceptable results with less training time (batch size=32). From the accuracy point of view the BERT model is the best for

Multiclass Sentiment classification on our dataset following the RoBERTa and ALBERT model. If speed is the main consideration, we recommend using RoBERTa

due to its speed of pretraining and fine-tuning with acceptable results. All models had difficulty in classifying class 2 (Negative) and 4 (Positive) correctly.

This work can be carried out in future to investigate how these models perform for different batch sizes and drop out Values. In future these models can be fine-tuned to enhance their performance. This work would help to choose the

best pre-trained models for Sentiment analysis based on accuracy and speed.

7. An ensemble model for idioms and literal text classification using knowledge-enabled BERT in deep learning

Introduction:

All languages and text genres utilize idioms often, yet idiomatic expressions continue to be a strange linguistic phenomena. Idioms are challenging to extract since there is no method that can exactly describe an idiom's structure. The fact that idiomatic phrases are an open collection and that new ones might always appear makes the process considerably more difficult. Literal refers to a phrase or sentence's fundamental or precise meaning. Natural language processing is greatly hampered by the peculiar lexical and syntactic features of non-literal statements. This work uses deep learning approaches to classify idioms and literals, which aids in improved categorization. Creating a knowledge base and knowledge graph for the literals and idioms along with their meanings help in easy identification and classification by the BERT Model. Using a stacking algorithm helps in putting the baseline models together to form an ensemble model with better performance.

Methodology:

Data Preprocessing

Methods used:

- Stopword removal
- Case transformation
- Special Character removal

Models

BERT

- The BERT Tokenizer is loaded from Keras.
- The BERT Model is loaded from TensorFlow

RoBERTa

- The RoBERTa Tokenizer is loaded from HuggingFace
- The RoBERTa Model is loaded from HuggingFace

K-BERT

- The architecture of the K-BERT model is composed of the four modules known as the embedding layer, knowledge layer, mask transformer and seeing layer.
- In addition to Bert, we have a knowledge layer.

Ensemble Model by stacking method

- Training a learning algorithm to integrate the predictions of various different learning algorithms is known as stacking.
- Two or more level 0 models make up the foundation of the stacking model.
- A level 1 model makes up the final prediction of the stacking model.

  In this paper,
  - BERT, RoBERTa and K-BERT are Level-0 Models (Base-Models)
  - Logistic Regression is a Level-1 Model (Meta-Model)

Result:

Parameters and Values:

Ensemble using weights and averages:

- Accuracy = 0.69
- Precision = 0.85
- Recall = 0.93
- F Score = 0.82
- Jaccard Score = 0.67
- Hamming Loss = 0.2
- Log Loss = 2.89

Knowledge based ensemble using stacking:

- Accuracy = 0.96
- Precision = 0.92
- Recall = 0.95
- F Score = 0.96
- Jaccard Score = 0.92
- Hamming Loss = 0.03
- Log Loss = 1.25

Other methods to visualize the results are:

- Confusion Matrix
- Knowledge Graphs

8. NLP-CUET@DravidianLangTech-EACL2021: Offensive Language Detection from Multilingual Code-Mixed Text using Transformers

Introduction

The increasing accessibility of the internet facilitated social media usage and encouraged individuals to express their opinions liberally. The exponential increase of offensive contents in social media has become a major concern to government organizations and tech companies.. Most of such offensive posts are written in a cross-lingual mannerand can easily evade the online surveillance systems. This paper presents an automated system that can identify offensive text from multilingual code-mixed data. In the task, datasets provided in three languages including Tamil, Malayalam and Kannada code-mixed with English where participants are asked to implement separate models for each language. In this paper two ML techniques, three deep learning techniques and three transformer based methods are used.

Methodology

ML techniques:

- LR
- SVM

Deep Learning:

- LSTM
- LSTM + Attention

Transformer models:

- XLM-R
- m-BERT
- Indic-BERT

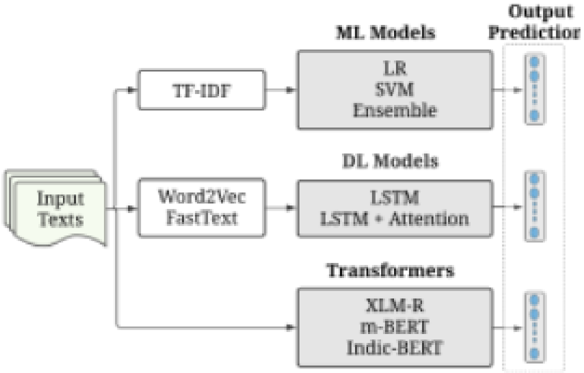


Figure 1: Abstract process of of offensive langu detection

Feature Extraction:

- For ML methods, tf-idf technique is applied to extract the unigram features.
- For DL methods, Word2Vec and FastText 2018) embeddings are used as feature extraction techniques
- Word2Vec is implemented using Keras embedding layer with embedding dimension 100 for all languages while pre-trained embedding matrix of each language is utilized for FastText embedding.

Machine Learning Models

- After using ML approaches like LR , SVM we take the ensemble of multiple ML classifiers to achieve better performance .
- Besides, Decision Tree (DT) and Random Forest (RF) classifiers are incorporated with LR and SVM to construct the ensemble approach.
- Majority voting technique is applied to get the prediction from the ensemble method.

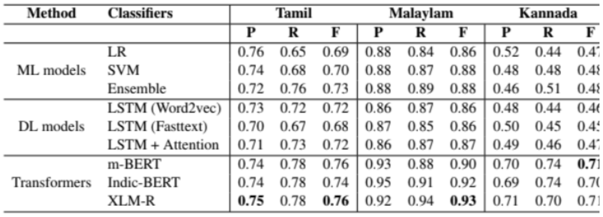
Deep Learning Models

- LSTM and combination of LSTM with Attention-based approach are employed to classify the offensive text to continue the investigations.
- To mitigate the chance of overfitting, a dropout technique utilized with a rate of 0.1.
- The output of the BiLSTM layer transferred to a softmax layer for the prediction.

Transformer Models

- Transformers used to develop our models include XLM-R, m-BERT, Indic-BERT .

- m-BERT is a transformer model pre-trained over 104 languages, and Indic-BERT is specifically pre-trained on Indian languages such as Kannada, Tamil, Telugu and Malayalam.
- We observed that each input text's average length is less than 50 words for Kannada and 70 words for Tamil and Malayalam languages. Therefore, to reduce the computational cost, the input texts' maximum size settles to 50 for Kannada and 70 for Tamil and Malayalam.



Method	Classifiers	Tamil			Malaylam			Kannada		
		P	R	F	P	R	F	P	R	F
ML models	LR	0.76	0.65	0.69	0.88	0.84	0.86	0.52	0.44	0.47
	SVM	0.74	0.68	0.70	0.88	0.87	0.88	0.48	0.48	0.48
	Ensemble	0.72	0.76	0.73	0.88	0.89	0.88	0.46	0.51	0.48
DL models	LSTM (Word2vec)	0.73	0.72	0.72	0.86	0.87	0.86	0.48	0.44	0.46
	LSTM (Fasttext)	0.70	0.67	0.68	0.87	0.85	0.86	0.50	0.45	0.45
	LSTM + Attention	0.71	0.73	0.72	0.86	0.87	0.87	0.49	0.46	0.47
Transformers	m-BERT	0.74	0.78	0.76	0.93	0.88	0.90	0.70	0.74	**0.71**
	Indic-BERT	0.74	0.78	0.74	0.95	0.91	0.92	0.69	0.74	0.70
	XLM-R	**0.75**	0.78	**0.76**	0.92	0.94	**0.93**	0.71	0.70	0.71

Result

Results indicate that ML ensemble achieved higher accuracy than DL methods.
However, the outcomes are not promising for the available datasets. Code-mixing of multilingual texts might be a reason behind this. Weighted f1 score increased from 0.73 to 0.76, 0.88 to 0.93 and 0.48 to 0.71 for Tamil, Malayalam and Kannada language respectively.

In future, the idea of ensemble technique could be adopted on transformer-based models to investigate the system's overall performance.

9. Comparative Analysis of BERT, RoBERTa, DistilBERT, and XLNet for Text-Based Emotion Recognition

Introduction:

The detection or recognition of emotions happens to be an extraction of finer-grained user sentiments. Text-based emotion recognition is a sub-branch of emotion recognition (ER) that focuses on extracting fine-grained emotions from texts. Though research in the field is fast gaining traction, the challenge of identifying appropriate embedding techniques for extracting the relationship between long term dependent texts and parallel processing of text sequence has for long inhibited the pace of attaining state-of-the-art results.

The proposal of transformers and the transformer language model provided a breakthrough in solving these limitations. The Bidirectional Encoder Representations from Transformers (BERT) pre-trained model, using the vanilla transformer language model released by Google in 2018 as a substructure, has been described as the rediscovery to the Natural Language Processing (NLP) pipeline due to the improved level of language understanding it offers. This paper aims to shed light on the efficacy of the BERT, RoBERTa, DistilBERT, and XLNet models in recognizing emotions from the International Survey on Emotion Antecedents and Reactions (ISEAR) dataset.

Keywords:

Natural Language Processing, Transfer Learning, Emotion Detection, BERT, DistilBERT, RoBERTa, XLNet

Method:

The dataset was acquired, preprocessed, and fed to the various candidate models. The candidate models were all fine-tuned on the data before final predictions were carried out. The ISEAR dataset is a publicly available dataset constructed through cross-culture questionnaire studies in 37 countries. It contains 7666 sentences classified into seven distinct emotion labels: joy, anger, sadness, shame, guilt, surprise, and fear. Its balanced class feature makes it ideal for making generalized predictive inferences; hence, its use for this study. The below figure presents the data distribution of the ISEAR dataset.

The obtained data contained several columns; the columns containing individuals' responses and the emotion labels were the columns of interest to this work. These two columns were, therefore, extracted for further processing. It was also realized that some columns contained emotion labels but no textual responses. These were again removed and the total amount of data reduced from 7666 to 7589. Special characters, double spacing, tags, and other irregular expressions found in the remaining data were removed. They were noticed to affect recognition performance negatively. Stop words were further removed, and the seven emotion labels encoded to a numerical scale. The training and test samples were tokenized to generate the tokens, which were then fed to the fine-tuning candidate models. The generated tokens were converted to vector representations and fed to the pre-trained models during the fine-tuning process. Thus, the models were trained on the input vector transformations and their outputs generated. The output was then evaluated using the designated test data, and results were obtained. Emotions were then classified into joy, sadness, fear, anger, guilt, disgust, and shame for each of the pre-trained models in the emotion classification process.

Result:

The confusion matrices obtained after the experiments are presented as follows:

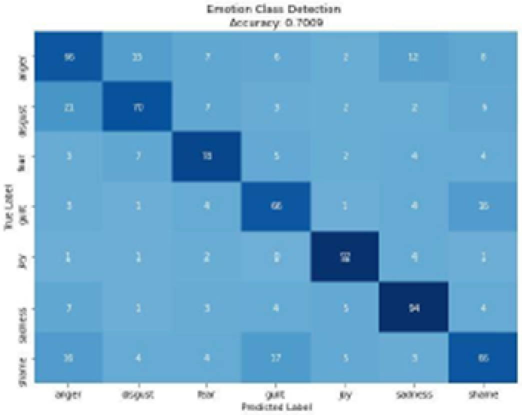


Fig.2 Confusion Matrix for BERT

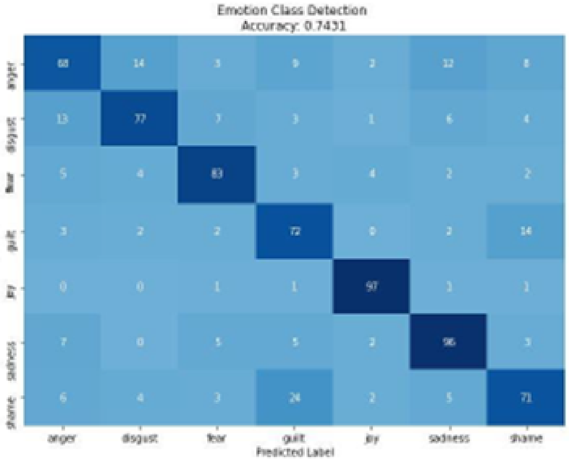


Fig.3 Confusion Matrix for RoBERTa

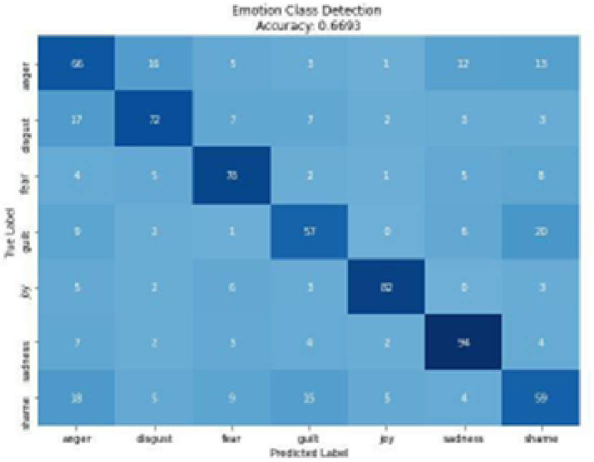


Fig.4 Confusion Matrix for DistilBERT

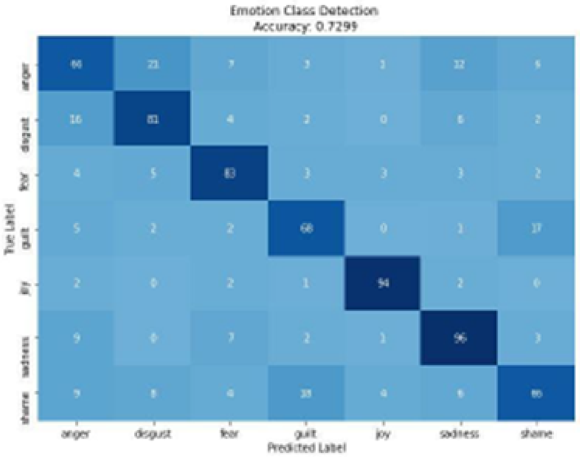


Fig.5 Confusion Matrix for XLNET

The classification report presented in Table 2 indicates the various precision, recalls, and F1-scores for the individual emotion classes after testing the various candidate models on the test data. RoBERTa, XLNet, BERT, and DistilBERT, with a recognition accuracy of 0.7431, 0.7299, 0.7009, and 0.6693, respectively.

Since all models were capable of recognizing emotions from the data, we posit that not only are these models efficient in other NLP tasks but are also efficient in recognizing emotions from texts. Secondly, from the results, we posit that under the same conditions, the RoBERTa pre-trained model outperforms the other pre-trained models under investigation in this work. Observations made during this work showed that even though the DistilBERT yielded the least accurate results, it was the fastest computationally. The XLNet model, on the other hand, was computationally the slowest. RoBERTa slightly outperformed the BERT model in speed.

The report further buttresses that RoBERTa is an optimal candidate for detecting emotions on the ISEAR dataset. XLNet also demonstrated some level of efficacy in some aspects of the seven classes. DistilBERT and BERT, on the other hand, could not achieve any high scores in any of the seven emotion classes for the precision, recall, and F1-score. The lower computational complexity of RoBERTa over XLNet also reinforces the recommendation of RoBERTa for emotion recognition in text.

AREA OF FOCUS

We focus on identifying the best classifier among the different high performance classifiers used in the above papers specific to Myers Briggs Type Indicator Personality Test.

Upon identifying the model, we would look forward to improving the model further by different techniques suitable to the model's architecture and parameters.