Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.046963

ARTICLE

Classification of Conversational Sentences Using an Ensemble Pre-Trained Language Model with the Fine-Tuned Parameter

R. Sujatha and K. Nimala*

Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamilnadu, 603203, India

*Corresponding Author: K. Nimala. Email: nimalak@srmist.edu.in

Received: 20 October 2023 Accepted: 07 December 2023 Published: 27 February 2024

ABSTRACT

Sentence classification is the process of categorizing a sentence based on the context of the sentence. Sentence categorization requires more semantic highlights than other tasks, such as dependence parsing, which requires more syntactic elements. Most existing strategies focus on the general semantics of a conversation without involving the context of the sentence, recognizing the progress and comparing impacts. An ensemble pre-trained language model was taken up here to classify the conversation sentences from the conversation corpus. The conversational sentences are classified into four categories: information, question, directive, and commission. These classification label sequences are for analyzing the conversation progress and predicting the pecking order of the conversation. Ensemble of Bidirectional Encoder for Representation of Transformer (BERT), Robustly Optimized BERT pretraining Approach (RoBERTa), Generative Pre-Trained Transformer (GPT), DistilBERT and Generalized Autoregressive Pretraining for Language Understanding (XLNet) models are trained on conversation corpus with hyperparameters. Hyperparameter tuning approach is carried out for better performance on sentence classification. This Ensemble of Pre-trained Language Models with a Hyperparameter Tuning (EPLM-HT) system is trained on an annotated conversation dataset. The proposed approach outperformed compared to the base BERT, GPT, DistilBERT and XLNet transformer models. The proposed ensemble model with the fine-tuned parameters achieved an F1_score of 0.88.

KEYWORDS

Bidirectional encoder for representation of transformer; conversation; ensemble model; fine-tuning; generalized autoregressive pretraining for language understanding; generative pre-trained transformer; hyperparameter tuning; natural language processing; robustly optimized BERT pretraining approach; sentence classification; transformer models

1 Introduction

Sentence classification is one of the most challenging area of interest in Natural Language Processing (NLP). Its applications are diverse, including sentiment analysis, topic modelling, question-answering, and more. Over the years, NLP researchers and practitioners have employed various techniques to categorize sentences, aiming to understand and organize textual data effectively. The evolution of sentence classification methods, from conventional deep learning models like Convolutional

Neural Networks (CNN) and Recurrent Neural Networks (RNN) to recent innovations involving Transformer models like BERT, RoBERTa, GPT, and XLNet.

1.1 Traditional Approaches: CNNs and RNNs

In the earlier days of NLP, deep learning models like CNNs and RNNs served as the baseline for sentence and text classification. These models leveraged their ability to process data sequences, enabling them to analyze the context of a given sentence and classify it into predefined categories. RNNs, in particular, exhibited the capacity to store information about the context of sentences, thanks to the use of Long and Short-Term Memory networks (LSTM) [1] and Gated Recurrent Units (GRU). These traditional models paid close attention to the semantic features of sentences, aiming to grasp the nuances and underlying meanings in text. By considering the order and arrangement of words within a sentence, they could effectively perform classification tasks.

1.2 The Advent of Bidirectional Models

The evolution of sentence classification models led to the introduction of bidirectional processing. Models such as Bidirectional LSTM (Bi-LSTM) [2] and Bidirectional GRU (BiGRU) [3] came into play, enabling sentences to be analyzed in both forward and backward directions. This bi-directional approach empowered these models to capture a richer and more comprehensive context of sentences, thus improving classification accuracy. In these bidirectional models, information flows not only from the beginning to the end of a sentence but also from the end to the beginning. This bidirectional processing allowed for a deeper understanding of the content within a sentence, ultimately resulting in more accurate classifications.

1.3 Sequence-to-Sequence Models

With advancements in deep learning, the NLP community transitioned from single-sentence classification to more complex tasks that involved sequence-to-sequence processing [4]. These models, equipped with both an encoder and a decoder, could handle various applications such as summarization and translation. The encoder takes an input sequence and converts it into word embeddings. This transformation ensures that the model captures the semantics and context of the sentence effectively [5]. Subsequently, the decoder generates a summary or translation of the input, striving to preserve the core content while presenting it more concisely and coherently. These sequence-to-sequence models marked a significant leap forward in NLP, as they could handle the intricate relationships between sentences in a document. They played a crucial role in automating tasks that required summarizing long texts or translating between languages.

1.4 The Emergence of Attention Mechanisms

While sequence-to-sequence models represented a significant advancement, they still faced some challenges. Memory constraints and the vanishing gradient problem were among the issues that researchers sought to address. This quest for improvement led to incorporating attention mechanisms [6] in NLP models. Attention mechanisms allow models to focus on specific parts of the input sequence when processing a given word or token. This selective attention ensures that the model gives more weight to the words that are most relevant in the context. As a result, attention mechanisms mitigated memory issues and contributed to more precise and context-aware sentence classification.

1.5 The Reign of Transformer Models

In recent years, the NLP landscape has witnessed a revolution with the advent of Transformer models. These models, including BERT [7], RoBERTa [8], GPT [9], and XLNet [10], have redefined the field by introducing novel architecture and training techniques. What sets Transformer models apart is their ability to process data more efficiently than their predecessors. Transformer models are characterized by their self-attention mechanisms, which allow them to assign varying levels of importance to each component within a sequence. Unlike traditional models that treat each word equally, Transformers adapt their attention weights based on the significance of each word in the context. This adaptive attention mechanism has proved to be highly effective in capturing the nuances of language, making these models state-of-the-art in various NLP tasks. Transformer models have pushed the boundaries of language understanding, achieving remarkable results in sentence classification. By considering the context and semantics of sentences, they outperform previous models and set new standards in NLP research and applications.

The additional work-related details are organized as follows. Section 2 presents detailed reviews of the related works. Section 3 describes the clear view of the Ensemble pre-trained language Model for sentence classification. Followed by Section 4 explains how the experiments are conducted and the dataset collection. Section 5 evaluates the results. Section 6 concludes the approach and future work.

2 Related Works

The sentence classification aims to analyze the conversation with minimal semantic loss. Researchers proposed various sentence classification approaches based on deep learning and ensemble models [11]. This section explains and presents a few publications on categorizing sentences and texts.

2.1 Neural Network-Based Representation

Johnson et al. put up a Deep Pyramid CNN (DPCNN) model that is simple to reduce the error rate by expanding the network layer [12]. The computational cost for this model will be high if the network layer count is increased. Sun et al. proposed an Inner attention Multi-channel CNN model to extract sentence features and classify the relations [13]. In this work, the model needs to be modified according to semantic percepts. Hassan et al. proposed a joint CNN-RNN model for sentence classification [14]. This framework integrates the CNN and RNN over the unsupervised, pre-trained word vectors to reduce the number of parameters. Bangyal et al. proposed a model for text classification using deep learning approaches. This model classified the fake news on COVID-19 [15].

2.2 Gated and Attention-Based Representation

Shen et al. suggested the Directional Self-Attention Network (DiSAN), which is based purely on the proposed attention and without the basis of Neural Network structure, to learn sentence embedding. DiSAN comprises a multi-dimensional self-attention that reduces the sequences into a vector representation after a directional self-attention that encodes worldly expectations. DiSAN, despite its simple design, outperforms the complicated RNN models in terms of accuracy and timeliness [16]. Wang et al. suggested Hierarchical Attention Networks (HAN) for sentence ordering [17]. This model has two essential characteristics: one is a hierarchical structure and the next one is the attention mechanism for document classification. Zhang et al. suggested the CNN-BiGRU model, which is a simple one. CNN and BiGRU are combined into this model [3]. BiGRU is used to get both the contextual representations and the semantic distribution, which is restricted to a Gaussian distribution. Dahiya et al. analyzed the responsiveness of single-layer convolutional neural

networks and offered valuable suggestions for improving the model's display [18]. Using CNN's Gated Fusion and the Universal Sentence Encoder or Bi-LSTM, Nagar et al. suggested a model for classifying text that took n-grams and semantic data into account [19]. Lan et al. presented the Stacked Residual Recurrent Neural Networks with Cross-Layer Attention (SRCLA) model for text categorization, which stacks additional RNN with a cross-layer attention model to filter other semantic data [20]. Dai et al. conducted correlation studies on tasks requiring categorizing sentences and semantic relatedness. To use diverse masks, they provide a Positional Self-Attention Layer for generating various Masked-Self-Attentions and a subsequent Position-Fusion Layer where fused positional data duplicates the Masked-Self-Attentions to create sentence embedding [21].

2.3 Pre-Trained Language Model Representations

Even though several cutting-edge findings have been made, there are still many restrictions on the models and how the model can improve performance because of vague and task-explicit construction-dependent difficulties. Indeed, even with the help of recent developments in a context, such as Contextualized word Vectors (CoVe) and Embeddings from Language Model (ELMo), which are practical applications of those developments, the model designs still require careful planning and training. Several NLP issues, such as standard language interpretation, text characterization and textual repercussions, have recently seen advancements, including the technique for calibrating trained language models on an extensive network with much nonspecific input. Howard et al. [22] gave the Universal Language Model Fine-tuning (ULMFiT) suggestion and completed the text characterization with cutting-edge innovations. According to Devlin et al., they claimed that this is done by mutually altering both left and right side variables in all layers. BERT is intended to pre-train profound bidirectional depictions from unsupervised data. By adding just a single extra result layer, the pre-trained BERT model can be customized to create state-of-the-art models for various applications. Sun et al. looked at various BERT [23] adjustment techniques for text categorization problems, including pre-processing significant texts, layer selection, layer-wise learning-rate, disastrous neglecting and low-shot learning problems. However, they have taken datasets with long texts, disregarding the context of datasets with short texts and hidden vector selection. Yu et al. expanded the Work to text classification tasks. They showed that for binary classification problems, post-training BERT using a credible domain-related corpus for binary classification problems would help solve the domain problem and improve the classifier's display [24]. Plaza-del-Arco et al. proposed a model which uses the pre-trained language models BERT, Cross Lingual Language Model (XLM) and BETO. This model is used for identifying hate speech [25]. The authors have trained the model on COVID-19 fake news dataset and COVID-19 English news dataset [26]. The model is for binary text classification.

3 Ensemble Pre-Trained Language Model for Sentence Classification

The proposed model for sentence classification on the unstructured conversation corpus has two major phases. In phase I, the input is fed into selective language models [27] with the default hyperparameter values and the results are observed. Several trainings were carried out with a specific combination of values on hyperparameters [28]. They compared the results and identified the hyperparameter, which produced a better result than other combinations. These parameter values are considered fine-tuned parameters. This set of language models has been trained again with the fine-tuned parameter. The hyperparameters for the transformer model are *batch_size, learning-rate, sequence_length, warmup*, etc., which are gathered with the help of fine-tuning approach.

In phase II, each of the individual models' results is compared and the better language model for sentence classification is identified. In this phase, the max voting ensemble approach is used to choose the improved model for sentence classification. The proposed framework is shown in Fig. 1.

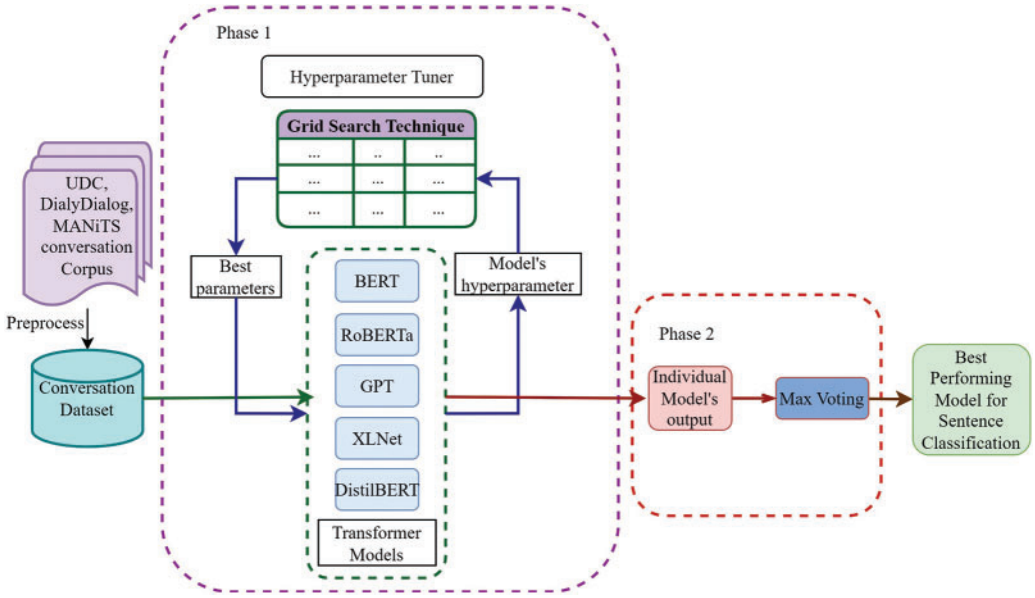


Figure 1: Ensemble of pre-trained language models coupled with hyperparameter tuning (EPLM-HT) approach

In this section, first, the fine-tuning approach is formally described in Section 3.1, followed by transformer models. The transformer models are Bi-directional Encoder Representations of Transformer (BERT) described in Section 3.2, the Robustly Optimized BERT Pretraining Approach (RoBERTa) in Section 3.3, followed by the Generative Pretraining Model in Section 3.4 and Generalized Auto regression Pretraining for Language Understanding (XLNet) in Section 3.5, finally DistilBERT in Section 3.6. The ensemble approach is described in Section 3.7.

3.1 Fine-Tuning Approach

Fine-tuning a pre-trained Transformer model is an effective technique in transfer learning [29], enabling pre-trained models for similar tasks and significantly reducing the data required for training. Here are the steps involved in fine-tuning a Transformer model:

- Select a pre-trained Transformer model: Choose one that has already been trained on a vast dataset, like BERT or GPT-2.
- Prepare the dataset: Prepare a smaller dataset specific to the task, such as sentiment analysis or question answering. The dataset should be labelled or annotated with the target labels or answers.
- Fine-tune the model: Fine-tune the pre-trained Transformer model by training it on the task-specific dataset. The process involves updating the model weights using backpropagation and gradient descent to minimize the loss function between the predicted and target labels or answers. The fine-tuning process can be done using various techniques, such as sequence classification, token classification, or question answering.

- Evaluate the model: Evaluate the performance of the fine-tuned model on a validation set and test set using appropriate evaluation metrics, such as accuracy, *F1_score*, or mean squared error.
- Tune hyperparameters: Tune the hyperparameters of the fine-tuned model, such as the learning rate, batch size, or dropout rate, to further improve its performance on the task-specific dataset [30].

Fine-tuning a Transformer model can be a time-consuming and resource-intensive process, but it can significantly improve the model's performance on a specific task.

3.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model [7,31] introduced by Google in 2018. It is designed to perform various natural language processing (NLP) tasks, including Question Answering, Text Summarization and Natural Language Inference. BERT has two models—$\text{BERT}_{\text{large}}$ and $\text{BERT}_{\text{base}}$—with 24 and 12 layers, respectively. BERT uses a unique classification embedding always contained in the sequence's first token, [CLS] and a second unique token, [SEP], separates sentences. The last hidden state h of the first token, [CLS] is used to represent the entire sequence and a softmax classifier is used to predict the probability of label c.

$$p\,(c|\boldsymbol{h}) = softmax(W\boldsymbol{h}) \tag{1}$$

where W stands for the particular problem's parameter matrix, change each parameter from BERT and W simultaneously by maximizing the log-likelihood of the proper label. The BERT model trains with a default hyperparameter (*learning-rate 2e-5, batch_size 16 or 32 epochs 3 or 5, sequence_length 256 or 512*). Then, tune the parameters with the following default values and repeat the training process. Based on the evaluation results, the optimal hyperparameter can be identified with the hyperparameter optimization approach. This research uses the grid search optimization technique to determine the optimal hyperparameter for validation.

3.3 Robustly Optimized BERT Pretraining Approach (RoBERTa)

RoBERTa (Robustly Optimized BERT pretraining Approach) [8] is a variant of BERT that aims to optimize the initial generation of BERT by adjusting several procedural parameters [32]. RoBERTa investigates crucial hyper-parameters, including larger pretraining datasets, static and dynamic-MLM, batch sizes, text encoding and the (Next Sentence Prediction) NSP technique. The RoBERTa tokenizer is used to tokenize input texts and input ids are then assigned. These ids are padded to a set-length to prevent variations per row. These tokens then extract features based on which sentence pairs are classified.

3.4 Generative Pre-Trained Transformer (GPT)

GPT (Generative Pre-Trained Transformer) is a left-to-right transformer model that uses transformer decoders to force the semi-supervised learning strategy to cope with model language. GPT primarily uses the Books Corpus dataset [33] for pretraining and fine-tuning, which includes 12 transformer layers and 12 attention heads transformer decoders. The GPT model's primary responsibility is to foresee the next token in the order.

3.5 XLNet: Generalized Autoregressive Pretraining for Language Understanding

XLNet (Generalized Autoregressive Pretraining for Language Understanding) and BERT have comparable architectures, with twelve transformer layers and 768 hidden layers in the XLNet-base-cased model. However, XLNet is an auto-regressive model (AR), whereas BERT is an autoencoding (AE) based model. This disparity is made clear in the MLM challenge, where language words are to be predicted by the model using random masking.

3.6 DistilBERT

DistilBERT is a smaller and faster version of BERT that keeps the basic structure of the original BERT but ditches token embeddings [34] and poolers. And it trims down the number of layers in the BERT-base model by half for a speedier performance. Despite using 40% fewer parameters, DistilBERT specifically kept 97% of BERT's performance.

3.7 Ensemble Approach

The ensemble approach with Transformer models [35] improves the models' performance robustness. Train the multiple Transformer models on the same task and dataset with different hyperparameters, architectures, or initialization. This method can combine the models' predictions by averaging them or using more complex techniques such as weighted averaging or stacking. This approach can improve the performance and robustness of the models, especially for tasks that are difficult or have high variability. The chosen ensemble approach for this proposed work is max voting. The basic idea behind max voting is to combine the predictions of multiple models by taking the mode, or most frequently predicted class, as the final prediction. Max voting can improve the accuracy and reduce the variance of classification models by averaging the diversity of the ensemble.

4 Experimental Setup and Dataset Collection

4.1 Experimental Framework

The proposed framework for sentence classification has two main vital phases: a fine-tuning approach is for extracting well-favoured parameters and the following phase is an ensemble approach for choosing an optimal model for sentence classification on the conversation dataset, as depicted in Fig. 2.

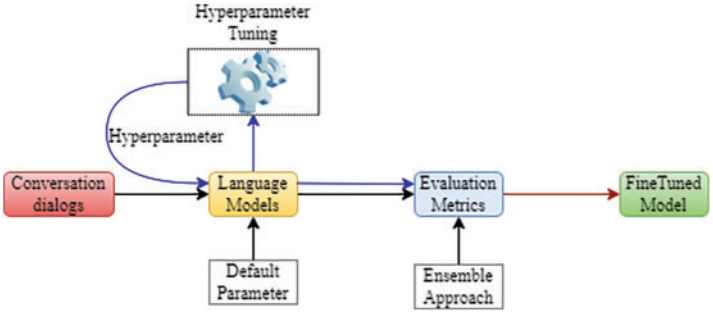


Figure 2: The model structure of the ensemble of pre-trained language models for sentence classification

4.2 Algorithm Implementation and Fine-Tuning

As per the framework discussed, implement the algorithm based on the modules. The modules such as Train the Model, Hyperparameter tuning and ensemble approach. The base transformer model is trained on the annotated dataset and with various combinations of hyperparameters. This hyperparameter can significantly affect the accuracy of the prediction. The grid search method is used here for hyperparameter tuning. This approach is used to identify the optimal hyperparameter for each model before taking the ensemble approach. Max voting has been accepted for the ensemble approach. Each model's results are considered in this approach and the model with the highest prediction result is selected. The Algorithm 1 describes the step-by-step procedure for the proposed model.

Algorithm 1: Procedure for Identifying language model for sentence classification with the fine-tuning approach

Input: $Data_{ls}$ one batch of domain knowledge training dataset
Output: *F1_score*, Training Loss, Validation Loss

1. *//initialize no. of models, list_of_hyperparameter values, results-> accuracy, F1_score, precision, recall*
2. **for i in no_of_models:**
3. **outcome←train_model (list_of_hyperparameters);** *//train the model with the list of Hyperparameters*
4. **results = store_result (outcome);** *//store the outcomes*
5. **fine_tune_params ←grid_search (results);** *//fine-tuning*
6. **outcome←train_model (fine_tune_params);** *//train the model with fine-tuned parameter*
7. **store_result (outcome);** *//store the outcome from the model with a fine_tuned parameter*
8. **end for**
9. **Model←voting (results);** *//ensemble approach*
10. //return the reliable model for sentence classification
11.
12. **Function: train_model (parameters):** *//function for train the model*
13. **load_tokenizer ();** *//tokenization*
14. **preprocess_data ();** *//preprocessing*
15. **build_model ();** //build the model
16. **Evaluate_model ();** *//evaluate the model*
17. **//return output_values (***accuracy, F1_score, precision, recall)*
18.
19. **Function: store_result (outcome):** *//function for storing the outcomes*
20. ***//record the list of results (accuracy, F1_score, precision, recall)***
21.
22. **Function: grid_search (results):** ***//****Fine Tuning*
23. ***//identify the optimal parameter with the grid search technique***
24. ***//return the fine_tuned parameter***
25.
26. **Function**: **voting (results)** *//voting approach to select the model*
27. ***//****return a more desirable model for sentence classification using the max voting approach*

$Data_{ls}$ = Labeled Dataset

An ensemble of pre-trained language with fine-tuning models was implemented using python programming with the help of Google Colab Pro. For the proposed model, the transformer models were trained with the help of Scikit-learn, transformers, TensorFlow, Keras, pandas, numpy and torch libraries. Graphical representations of results are obtained with the use of the matplotlib library. Each pre-trained model specified in this proposed work is trained on an annotated dataset separately. Then, accuracies and training loss for the individual models with the hyperparameter specifications are analyzed. Finally, compare the results with other models using the voting method and identify the model for sentence classification in the conversation.

4.3 Dataset

Datasets for the experimental analysis can be collected in two ways. One is from pre-processed data sources available from social media or data servers. Another way is to collect, pre-process manually and annotate by the experts. The dataset for this work is collected from DailyDialog [36] dataset. This dataset consists of conversations with utterances. This collection is a semi-supervised dataset. After extracting utterances from the corpora, the dataset is preprocessed: dialogue de-duplication and filtering out meaningless sentences. The preprocessed dataset contains 5812 sentences and expressions distributed under four dialogues_act categories shown in Table 1. The class labels are *Information, Question, Directive* and *Commissive*. The dataset's size allows us to train our most significant models without ever updating on the same sequence again. The dataset is splited into 85% training and 15% evaluation sets. A few examples of conversation sentences are given below:

1. *Thank you for your advice! I'll try it. - Commissive*
2. *May I turn on the radio then? - Directive*
3. *Really? I hope to taste it. Remember to tell me. - Commissive*
4. *Did you see May today? - Question*
5. *I think that showbiz stars have an effortless life. - Information*

Table 1: Dataset descriptions

	Training	Test	train
Total no. of sentences	5812	872	4940
Information	3222	484	2738
Question	1714	257	1457
Directive	509	76	433
Commissive	367	55	312

4.4 Evaluation Metric

The performance metrics for validating any models are *Recall, Specificity, FalsePositiveRate, FalseNegativeRate, Error, Precision, Accuracy and F1_score*, shown in Eqs. (2) to (9).

$$Recall = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \tag{2}$$

$$Specificity = \frac{\sum_{i=1}^{l} \frac{tn_i}{tn_i + fp_i}}{l} \tag{3}$$

$$FalsePositiveRate = \frac{\sum_{i=1}^{l} \frac{fp_i}{fp_i + tn_i}}{l} \tag{4}$$

$$FalseNegativeRate = \frac{\sum_{i=1}^{l} \frac{fn_i}{fn_i + tp_i}}{l} \tag{5}$$

$$Error = \frac{\sum_{i=1}^{l} \frac{fp_i + fn_i}{tp_i + fn_i + tn_i + fp_i}}{l} \tag{6}$$

$$Precision = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \tag{7}$$

$$Accuracy = \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + tn_i + fp_i}}{l} \tag{8}$$

where i - index of the class

l - class labels count

tp_i - correctly predicted sentence count in class i

th_e - correctly predicted sentences count not in class i

fp_i - sentences count which does not belong to class i, but the classifier predicted it to be in class i

fn_i - sentences count which belongs to class i, but the classifier predicted it not to be in class i

$tp_i + fn_i$ - total no. of sentences labelled in class i

$tn_i + fp_i$ - total no. of sentences not marked in class i

$tn_i + tp_i + fp_i + fn_i$ - total no. of sentences in class i

$tp_i + fp_i$ - total no. of sentences labelled in class i

The *F1_score* is a different metric that combines the *precision* and *recall* rates into a single measurement. This metric's value spans from 0 to 1 and if the evaluated classifier correctly categorizes every text, it will take the value 1. Eq. (9) is used to compute the *F1_score* for multiclass classification. $Precision_i$ is calculated using Eq. (7) and $Recall_i$ is added using Eq. (2).

$$F1\_score = \frac{\sum_{i=1}^{l} \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i}}{l} \tag{9}$$

The testing and validation results are also assessed to evaluate the validation performance. The remaining test dataset was used to calculate the test performance to see if over-fitting occurred.

Every hyperparameter (*loss-function, optimizer, epochs, validation-split rate* and *sequence_length*) is adjusted by assessing the cross-entropy loss's performance [37] as a loss function, which entails changing a categorical-cross-entropy (CCE) loss objective from a one-hot vector to an integer. The following definition of the *CCE Loss* applies to tasks requiring multiclass classification as in Eq. (10):

$$CCE\ Loss = -\sum_{i=1}^{output\ size} y_i . \log \widehat{y_i} \tag{10}$$

where $\widehat{y_i}$ is the expected model output and y_i is the desired outcome. Both are referred to as one-hot vectors. Three epochs are used to train our models with a 15% split validation data rate. Across all pre-trained models, the sequence length for our model was constant at 256. The model is trained with *CCE Loss* using Adam optimizer. Adam optimizer is an optimization algorithm. The network weights are updated iteratively using this optimizer rather than the conventional stochastic gradient descent method based on training data.

4.5 Hyperparameters

Hyperparameters in Transformer models are basic settings that affect how the model learns and generalizes from data during training. Table 2 provides commonly used hyperparameters in Transformer models.

Table 2: Hyperparameters

Hyperparameters	Values
loss_function	Categorical cross entropy
optimizer	Adam
learning-rate	$5e^{-5}$, $3e^{-5}$, $2e^{-5}$, $1e^{-5}$
batch_size	32, 16
epochs	2, 3, 5
sequence_length	128, 256, 512
validation_split_rate	15%
warm_up	6%
decay	0

Default parameters can be varied depending on the implementation, dataset size and computational resources. The Hyperparameter combinations of Table 2 values are used for training. *learning-rate, batch_size* and *epochs* have some list of values, which are combined with other values, and then train the model with these parameter combinations and observe the outcomes. The epoch size is limited to 2 to 5 due to GPU size and training time extension. Subsequent epochs do not yield further improvements in model performance as per the expected level. As far as the conversation benchmark dataset chosen for this work is concerned, the longest sequence span is 120 tokens. So, for the experimental analysis, the sequence length has defaulted to 256 tokens as designated for this task. In this work, the grid search technique is used for hyperparameter optimization.

4.6 Experimental Results

The Experimental Results for the proposed model are shown in Tables 3 and 4. Training loss and accuracy results for pre-trained models with listed hyperparameter values (*learning-rates* ∈ {1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}} and *batch_sizes* ∈ {16, 32}) are shown in Tables 3 and 4.

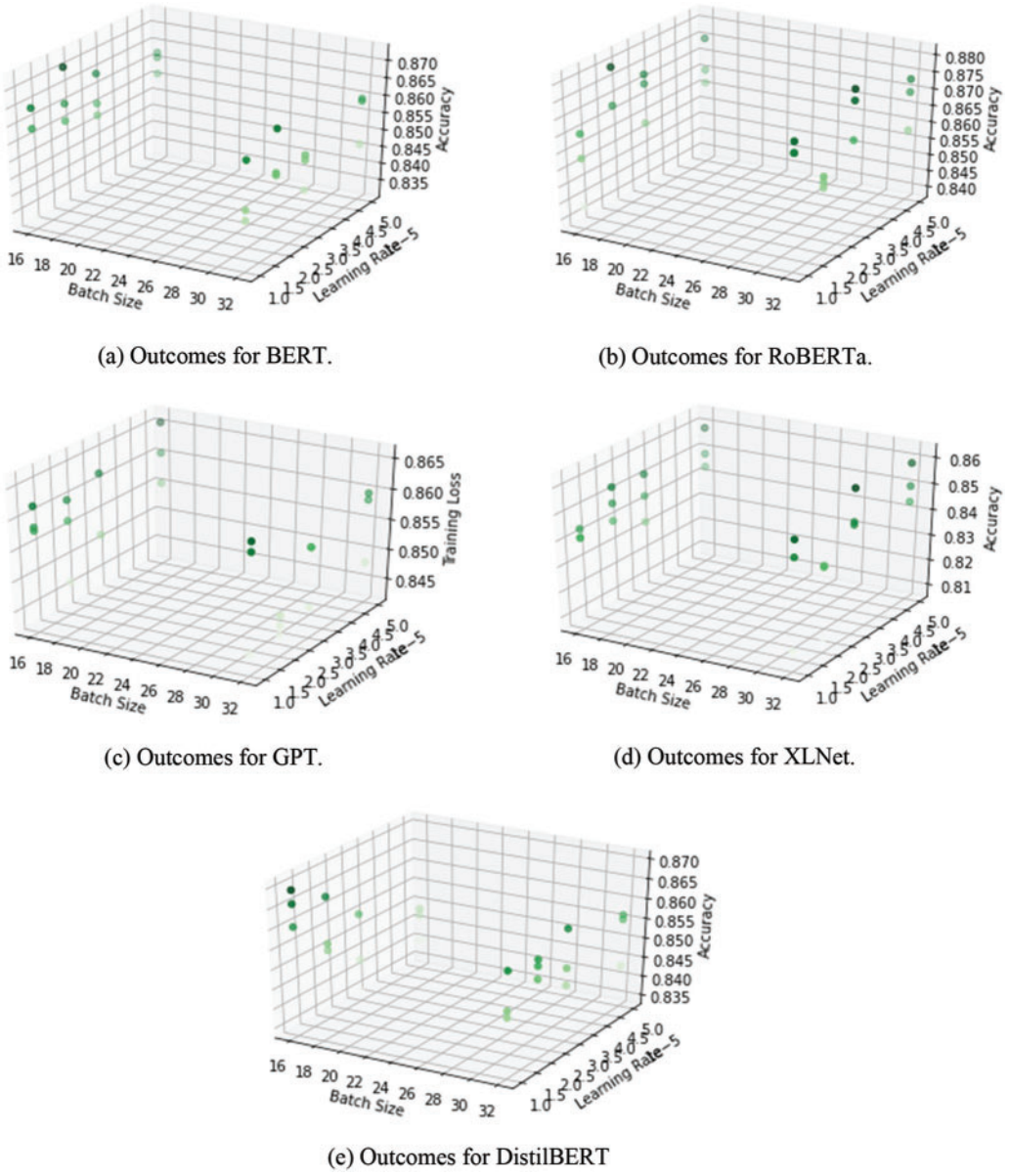
Table 3: Effect of different hyperparameter setting combinations-Training Loss

Batch size	*Learning-rate*	BERT	RoBERTa	GPT	XLNet	DistilBERT
16	1e^{-5}	0.329 ± 0.21	0.305 ± 0.16	0.560 ± 0.37	0.5243 ± 0.21	0.298 ± 0.07
	2e^{-5}	0.127 ± 0.04	0.148 ± 0.12	0.371 ± 0.03	0.3831 ± 0.11	0.133 ± 0.04
	3e^{-5}	0.055 ± 0.02	0.076 ± 0.03	0.220 ± 0.03	0.2588 ± 0.10	0.062 ± 0.03
	5e^{-5}	0.028 ± 0.02	0.064 ± 0.03	0.110 ± 0.03	0.1642 ± 0.08	0.043 ± 0.01
32	1e^{-5}	0.018 ± 0.01	0.036 ± 0.01	0.076 ± 0.03	0.0943 ± 0.03	0.031 ± 0.01
	2e^{-5}	0.013 ± 0.01	0.031 ± 0.01	0.061 ± 0.03	0.0638 ± 0.02	0.026 ± 0.01
	3e^{-5}	0.013 ± 0.01	**0.022 ± 0.01**	0.039 ± 0.02	0.0567 ± 0.01	0.020 ± 0.01
	5e^{-5}	**0.012 ± 0.01**	0.025 ± 0.01	**0.035 ± 0.02**	**0.0500 ± 0.01**	**0.020 ± 0.01**

Table 4: Effect of different hyperparameter setting combinations-Validation Accuracy

Batch size	*Learning-rate*	BERT	RoBERTa	GPT	XLNet	DistilBERT
16	1e^{-5}	86.47 ± 0.45	86.54 ± 0.73	85.16 ± 0.55	84.06 ± 0.40	**86.69 ± 0.24**
	2e^{-5}	**87.16 ± 0.23**	**87.52 ± 0.60**	86.71 ± 0.47	84.72 ± 0.61	85.52 ± 0.82
	3e^{-5}	86.53 ± 0.37	87.33 ± 0.26	87.02 ± 0.29	84.48 ± 0.84	85.53 ± 0.87
	5e^{-5}	86.25 ± 0.68	87.05 ± 0.45	85.98 ± 0.45	84.87 ± 0.96	84.97 ± 0.92
32	1e^{-5}	86.16 ± 0.53	87.05 ± 0.43	85.36 ± 0.26	84.36 ± 0.91	85.47 ± 0.49
	2e^{-5}	86.53 ± 0.22	85.96 ± 0.67	**87.03 ± 0.33**	83.88 ± 0.82	85.09 ± 0.95
	3e^{-5}	85.29 ± 0.74	87.22 ± 0.44	86.79 ± 0.31	**85.19 ± 0.92**	86.29 ± 0.23
	5e^{-5}	85.95 ± 0.62	87.13 ± 0.36	87.16 ± 0.14	84.97 ± 0.74	85.42 ± 0.27

The grid search approach outcome graphs are shown in Fig. 3. The grid search method explicitly identifies the peak point and the respective coordinates. The fine-tuned parameter was determined from the peak point, which has given the highest accuracy. The coordinates of this point are considered as a fine-tuned parameter.



(a) Outcomes for BERT.

(b) Outcomes for RoBERTa.

(c) Outcomes for GPT.

(d) Outcomes for XLNet.

(e) Outcomes for DistilBERT

Figure 3: Grid search method outcomes for listed pre-trained models

Before making final predictions, each pre-trained model was adjusted using the data. Additional pre-trained models have undergone various phases and then changed for the text classification tasks with hyperparameters. Fine-tune the further steps on the corresponding datasets with a reduced learning-rate. After fine-tuning, the subtasks are combined to improve classification results. The Grid Search method involves defining a grid of hyperparameter values and then systematically testing each combination of these hyperparameters to find the optimal set of values that yield the best performance of the model on a given validation set. Grid search identifies a model's hyperparameters, producing the most valid inferences. Table 5 displays the fine-tuned parameter values for the proposed model.

Table 5: Hyperparameter combinations after fine-tuning

Hyperparameter	XLNet	BERT	RoBERTa	DistilBERT	GPT
learning_rate	$2e^{-5}$	$1e^{-5}$	$3e^{-5}$	$3e^{-5}$	$5e^{-5}$
num_train_epochs	3	3	3	3	3
seed	35	31	21	6	30
batch_size	32	16	32	32	32

The proposed approach requires fewer fine-tuned parameters to boost computational effectiveness. The proposed method can classify sentences with limited fine-tuned hyperparameters. The proposed task is solved by combining multiple models in an ensemble approach. The wellness/generalizability of the model is enhanced via ensemble approaches. In this case, the ensemble model's predictions are computed using the max voting method, which generally differs from the soft voting approach. In contrast to a single model, the latter takes the findings and combines the forecasts from each model to outperform it overall. From the above analysis, the BERT and RoBERTa model gives better accuracy and less loss than other transformer models, as in Table 6.

Table 6: Evaluation metrics for the models with fine-tuned hyperparameters

Pretrained models	Training loss	Validation loss	*F1_score*
bert_base_uncased	0.2993	0.3703	87.11
roberta-base	**0.3060**	**0.3688**	**87.69**
xlnet-base-cased	0.5204	0.4947	77.37
gpt2	0.3119	0.3649	85.74
distilbert-base-uncased	0.3476	0.3805	85.96

5 Result Analysis

The BERT model for sentence classification obtained 0.03% loss and 87% accuracy after training with the dropout set to 0.5 and the *learning_rate* set to 1e-5. The maximum *sentence_length* was set to 256 and the *batch_size* was set to 16.

With the dropout set to 0.5, the *learning_rate* set to 3e-5, the maximum *sentence_length* to 256 and the *batch_size* to 16, the RoBERTa model provides 0.03% loss and 88% accuracy.

The Report leads us to conclude that the RoBERTa pre-trained model produces better results for sentence classification on the conversation dataset when wholly optimized for the task. The classification report shown in Table 6 indicates that RoBERTa handled the sentence categorization better with BERT, DistilBERT, GPT and XLNet.

This approach proves that RoBERTa is the best model for identifying the sentences on the conversation dataset. BERT also showed some effectiveness in several facets of the four classes. Contrarily, DistilBERT and XLNet failed to attain any high *precision, recall, or F1_scores* in any of the four types of sentences. RoBERTa is recommended for conversation sentence classification since it has lower computational complexity than BERT.

Table 6 compares RoBERTa to other BERT variations. Using more extensive pretraining data improves performance across various tasks, which is one of RoBERTa's benefits. In downstream NLP tasks, RoBERTa also performs better than XLNet and BERT. The drawbacks of RoBERTa, however, include its resource-intensive nature due to its enormous data requirements and increasing computing complexity. The performance improves with increasing the pretraining step, but it gets computationally expensive and takes longer.

6 Conclusions and Future Work

This paper has presented the proposed model, an Ensemble of Pre-trained Language Models with a Hyperparameter Tuning (EPLM-HT) model for sentence classification on conversation datasets. Fine-tuning approach incorporated and identified optimal parameters and then trained the model with these fine-tuned parameters. Each transformer model is evaluated individually with fine-tuned parameter fusion. The output metrics, such as *F1_score* and accuracy, are recorded. The proposed framework combines the results of five different transformer models and identifies the more acceptable outcome with the max voting approach. Finally, a better model is identified and fine-tuned parameters are specified for sentence classification, producing better accuracy and lower loss than the base model. The proposed method is tested with an annotated conversation dataset. In this work, The BERT and RoBERTa model achieved 87.16%, 87.52% for accuracy and 87.11%, and 87.69% for *F1_score*, respectively. The suggested method surpassed the performance of the foundational transformer models such as BERT, GPT, DistilBERT, and XLNet. The proposed approach will be applied to numerous NLP applications, including multiclass classifications. Future can be used for text, sentence and sentiment type categories in a better way with an appropriate transformer model. This framework will give a better lead for multiclass classification with fine-tuned parameters and an ensemble approach.

Acknowledgement: The authors would like to acknowledge the anonymous reviewers for their helpful comments and suggestions.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: Study conception and design: Sujatha R.; data collection: Sujatha R.; analysis and interpretation of results: Sujatha R., Nimala K.; draft manuscript preparation: Sujatha R. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data and materials used in this research are available upon reasonable request to the corresponding author.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] G. Van Houdt, C. Mosquera and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10462-020-09838-1 (accessed on 06/12/2023).

[2] R. Anhar, T. B. Adji and N. Akhmad Setiawan, "Question classification on question-answer system using bidirectional-LSTM," in *Proc. of 5th Int. Conf. on Science and Technology (ICST)*, IEEE, Yogyakarta, Indonesia, vol. 1, pp. 1–5, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9166190 (accessed on 06/12/2023).

[3] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren *et al.,* "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, vol. 6, pp. 73750–73759, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8543213 (accessed on 06/12/2023).

[4] Z. He, M. Pan, Y. Wang, G. Xu, W. Su *et al.,* "The influence of embedding size and hidden units parameters changes on the sequence2sequence model," in *Proc. of 5th Int. Conf. on Artificial Intelligence and Big Data (ICAIBD)*, IEEE, Chengdu, China, pp. 150–155, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9820494 (accessed on 06/12/2023).

[5] J. Shobana and M. Murali, "Abstractive review summarization based on improved attention mechanism with pointer generator network model," *Webology*, vol. 22, no. 1, pp. 77–91, 2021. [Online]. Available: https://www.researchgate.net/profile/Shobana-Jay;akumar/publication/350585197_Abstractive_Review_Summarization_based_on_Improved_Attention_Mechanism_with_Pointer_Generator_Network_Model/links/60a68d6045851522bc3d2581/Abstractive-Review-Summarization-based-on-Improved-Attention-Mechanism-with-Pointer-Generator-Network-Model.pdf (accessed on 06/12/2023).

[6] A. Chowanda and A. D. Chowanda, "Generative Indonesian conversation model using recurrent neural network with attention mechanism," *Procedia Computer Science*, vol. 135, pp. 433–440, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918314844 (accessed on 06/12/2023).

[7] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi *et al.,* "RoBERTa: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[9] M. Samadi, M. Mousavian and S. Momtazi, "Deep contextualized text representation and learning for fake news detection," *Information Processing & Management*, vol. 58, no. 6, pp. 102723, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457321002077 (accessed on 06/12/2023).

[10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov *et al.,* "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. of 33rd Conf. on Advances in Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf (accessed on 06/12/2023).

[11] R. Sujatha and K. Nimala, "Text-based conversation analysis techniques on social media using statistical methods," in *Proc. of Int. Conf. on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Chennai, India, IEEE, pp. 1–11, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9752562 (accessed on 06/12/2023).

[12] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. of 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, vol. 1, pp. 562–570, 2017. [Online]. Available: https://aclanthology.org/P17-1052/ (accessed on 06/12/2023).

[13] K. Sun, Y. Li, D. Deng and Y. Li, "Multi-channel CNN based inner-attention for compound sentence relation classification," *IEEE Access*, vol. 7, pp. 141801–141809, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8847427 (accessed on 06/12/2023).

[14] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8314136 (accessed on 06/12/2023).

[15] W. H. Bangyal, R. Qasim, N. U. Rehman, Z. Ahmad, H. Dar *et al.,* "Detection of fake news text classification on COVID-19 using deep learning approaches," *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 1–14, 2021. [Online]. Available: https://downloads.hindawi.com/journals/cmmm/2021/5514220.pdf (accessed on 06/12/2023).

[16] T. Shen, J. Jiang, T. Zhou, S. Pan, G. Long *et al.,* "DiSAN: Directional self-attention network for RnN/CNN-free language understanding," in *Proc. of AAAI Conf. on Artificial Intelligence*, New Orleans, Lousiana, USA, vol. 32, no. 1, pp. 5446–5455, 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11941 (accessed on 06/12/2023).

[17] T. Wang and X. Wan, "Hierarchical attention networks for sentence ordering," in *Proc. of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, vol. 33, no. 1, pp. 7184–7191, 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4702 (accessed on 06/12/2023).

[18] S. Dahiya, A. Mohta and A. Jain, "Text classification based behavioural analysis of WhatsApp chats," in *Proc. of 5th Int. Conf. on Communication and Electronics Systems (ICCES)*, IEEE, Coimbatore, India, pp. 717–724, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9137911 (accessed on 06/12/2023).

[19] A. Nagar, A. Bhasin and G. Mathur, "Text classification using gated fusion of n-gram features and semantic features," *Computación y Sistemas*, vol. 23, no. 3, pp. 1015–1020, 2019. [Online]. Available: https://www.scielo.org.mx/scielo.php?pid=S1405-55462019000301015&script=sci_arttext&tlng=en (accessed on 17/11/2023).

[20] Y. Lan, Y. Hao, K. Xia, B. Qian and C. Li, "Stacked residual recurrent neural networks with cross-layer attention for text classification," *IEEE Access*, vol. 8, pp. 70401–70410, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9063530 (accessed on 06/12/2023).

[21] B. Dai, J. Li and R. Xu, "Multiple positional self-attention network for text classification," in *Proc. of AAAI Conf. on Artificial Intelligence*, New York Hilton Midtown, New York, USA, vol. 34, no. 5, pp. 7610–7617, 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6261 (accessed on 06/12/2023).

[22] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," arXiv preprint arXiv:1801.06146, 2018.

[23] C. Sun, X. Qiu, Y. Xu and X. Huang, "How to fine-tune bert for text classification?," in *Proc. of 18th China National Conf. Chinese Computational Linguistics(CCL)*, Kunming, China, pp. 194–206, 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-32381-3_16 (accessed on 06/12/2023).

[24] S. Yu, J. Su and D. Luo, "Improving BERT-based text classification with auxiliary sentence and domain knowledge," *IEEE Access*, vol. 7, pp. 176600–176612, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8903313 (accessed on 06/12/2023).

[25] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López and M. T. Martín-Valdivia, "Comparing pre-trained language models for Spanish hate speech detection," *Expert Systems with Applications*, vol. 166, pp. 114120, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742030868X (accessed on 06/12/2023).

[26] R. Qasim, W. H. Bangyal, M. A. Alqarni and A. Ali Almazroi, "A fine-tuned BERT-based transfer learning approach for text classification," *Journal of Healthcare Engineering*, vol. 2022, pp. 17, 2022. [Online]. Available: https://www.hindawi.com/journals/jhe/2022/3498123/ (accessed on 06/12/2023).

[27] J. Briskilal and C. N. Subalalitha, "An ensemble model for classifying idioms and literal texts using BERT and RoBERTa," *Information Processing & Management*, vol. 59, no. 1, pp. 102756, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457321002375 (accessed on 06/12/2023).

[28] D. Vidyabharathi and V. Mohanraj, "Hyperparameter tuning for deep neural networks based optimization algorithm," *Intelligent Automation & Soft Computing*, vol. 36, no. 3, pp. 2559–2573, 2023. [Online]. Available: https://www.techscience.com/iasc/v36n3/51874 (accessed on 06/12/2023).

[29] I. Alghanmi, L. Espinosa Anke and S. Schockaert, "Combining BERT with static word embeddings for categorizing social media," in *Proc. of 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-Generated Text*, Online, Association for Computational Linguistics, pp. 28–33, 2020. [Online]. Available: https://aclanthology.org/2020.wnut-1.5/ (accessed on 06/12/2023).

[30] M. Al Duhayyim, H. G. Mohamed, S. S. Alotaibi, H. Mahgoub, A. Mohamed *et al.,* "Hyperparameter tuned deep learning enabled cyberbullying classification in social media," *Computers, Materials & Continua*, vol. 73, no. 3, pp. 5011–5024, 2022. [Online]. Available: https://www.techscience.com/cmc/v73n3/49079 (accessed on 06/12/2023).

[31] M. A. H. Wadud, M. F. Mridha, J. Shin, K. Nur and A. K. Saha, "Deep-BERT: Transfer learning for classifying multilingual offensive texts on social media," *Computer Systems Science and Engineering*, vol. 44, no. 2, pp. 1775–1791, 2023. [Online]. Available: https://www.techscience.com/csse/v44n2/48287 (accessed on 06/12/2023).

[32] F. A. Acheampong, H. Nunoo-Mensah and W. Chen, "Transformer models for text-based emotion detection: A review of BERT-based approaches," *Artificial Intelligence Review*, vol. 54, pp. 5789–5829, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s10462-021-09958-2 (accessed on 06/12/2023).

[33] J. Zheng, F. Cai, H. Chen and M. de Rijke, "Pre-train, interact, fine-tune: A novel interaction representation for text classification," *Information Processing & Management*, vol. 57, no. 6, pp. 102215, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457319311227 (accessed on 06/12/2023).

[34] R. Tang, Y. Lu, L. Mou, O. Vechtomova and J. Lin, "Distilling task-specific knowledge from BERT into simple neural networks," arXiv preprint arXiv:1903.12136, 2019.

[35] A. Alsayat, "Improving sentiment analysis for social media applications using an ensemble deep learning language model," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 2499–2511, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s13369-021-06227-w (accessed on 06/12/2023).

[36] Y. Li, H. Su, X. Shen, W. Li, Z. Cao *et al.,* "DailyDialog: A manually labelled multi-turn dialogue dataset," arXiv preprint arXiv:1710.03957, 2017.

[37] E. Lee, C. Lee and S. Ahn, "Comparative study of multiclass text classification in research proposals using pretrained language models," *Applied Sciences*, vol. 12, no. 9, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/9/4522 (accessed on 06/12/2023).