IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 August 2022, accepted 15 September 2022, date of publication 5 October 2022, date of current version 13 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212367

RESEARCH ARTICLE

Sentiment Analysis Using Pre-Trained Language Model With No Fine-Tuning and Less Resource

YUHENG KIT AND MUSA MOHD MOKJI
Faculty of Engineering, School of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
Corresponding author: Yuheng Kit (yhkit2@graduate.utm.my)

This work was supported by the Ministry of Education Malaysia and Universiti Teknologi Malaysia (UTM) under the Fundamental Research Grant Scheme under Grant R.J130000.7851.5F179.

ABSTRACT Sentiment analysis has become popular when Natural Language Processing algorithms were proven to be able to process complex sentences with good accuracy. Recently, pre-trained language models such as BERT and mBERT, have been shown to be effective for improving language tasks. Most of the work in implementing the models focuses on fine-tuning BERT to achieve desirable results. However, this approach is resource-intensive and requires a long training time, up to a few hours on a GPU, depending on the dataset. Hence, this paper proposes a less complex system with less training time using the BERT model without the fine-tuning process and adopting a feature reduction algorithm to reduce sentence embeddings. The experimental results show that with 50% fewer sentence embeddings, the proposed system improves the accuracy by 1-2% with 71% less training time and 89% less memory usage. The proposed approach has also been proven to work for multilingual tasks by using a single mBERT model.

INDEX TERMS Sentiment analysis, natural language processing.

I. INTRODUCTION

Recently, researchers have shown considerable interest in pre-trained language models such as BERT for Natural Language Processing (NLP) tasks due to their promising performance. Among others, sentiment analysis has been shown to be successful using these models. Sentiment analysis is also known as opinion mining. Sentiment analysis aims to automatically identify the sentiment polarity of textual data [1]. Applications benefiting from this technology include movie reviews, customer service reviews, product reviews, and others where customer feedback and suggestions are collected and analyzed to improve the product.

Natural language processing began in the 1940s [2] using rule-based methods, statistical learning methods, and deep learning methods. Natural language processing continues to evolve and has gained increasing interest. Pre-trained language models have been trained with enormous corpora, such as Wikipedia. The model is then fine-tuned with specific downstream tasks without training from scratch. Some of the earlier models based on Long Short-Term Memory architecture (LSTMs) are ULMFIT [3] and ELMO [4]. Since Transformer [5] was introduced, the later models rely on attention mechanisms such as GPT [6], GPT-2 [7], GPT-3 [8], BERT [9], XLNET [10], Megatron-LM [11], and T5 [12].

There is plenty of work on fine-tuning the BERT model in training the BERT model for a specific downstream task. Some of the examples are the fine-tuning BERT with SST-5 dataset [13], aspect-based sentiment analysis using BERT [14], fine-tuning BERT for multi-label sentiment analysis in unbalanced code-switching text [15], BERT-BiLSTM for sentiment orientation prediction of investors and consumers in the energy market [16], white supremacist classification using BERT [17], news text classifier by fine-tuning BERT model [18], fine-tuning the BERT model to obtain contextual word embeddings for Persian rumor verification [19], web page classification by fine-tuning the BERT with DRIMN [20], and others. Fine-tuning is required to update the BERT pre-trained parameters with an additional output layer to improve the accuracy of the model for the a specific purpose. However, fine-tuning is a resource-intensive process requiring an expensive GPU with high memory, which causes some

The associate editor coordinating the review of this manuscript and approving it for publication was Kah Phooi (Jasmine) Seng.

work not to benefit from the pre-trained language model [21]. In this case, researchers require high investment to enjoy the good performance of the pre-trained language model.

Thus, some works have attempted to compress the pre-trained BERT [9] using a few methods, such as quantization, knowledge distillation, pruning, matrix decomposition, and dynamic inference acceleration [21]. These works show improvement in the training time but involve complex processes in changing the algorithm or model.

In contrast, fine-tuning is not the only option, because the sentence embeddings from the pre-trained BERT can be used as features representing contextualized sentences. These features can then be trained using conventional machine learning algorithms, such as kNN and SVM, which use much less memory and GPU resources than the fine-tuning process.

In addition, relying on cloud computing consumes a lot of network transmission resources and result in delays, which could, for example, endanger patients' lives in the medical industry. [22] Besides, the amount of time and processing power required to train deep learning networks to recognize and react to data patterns that are important to their applications is one of the largest problems confronting the creation of new AI technologies. [23]. This is one of the main reasons why it is crucial to reduce the training time and resource consumption for pre-trained language models, as this will enable local devices to run the training on their own with less expensive resources and without the need for cloud computing. This concept is called edge computing, which shifts computation and communication resources from the cloud to the edge of networks to deliver services and execute calculations, eliminating unnecessary communication latency and enabling faster responses for end users [24]. With fewer resources needed for training, it is possible to deploy the model and allow the training to occur in edge devices so that the models can be updated with the latest data.

There is some interest in using sentence embeddings from BERT, such as ColBERT, which uses sentence embeddings from BERT as inputs for parallel lines of hidden layers in a neural network for humor detection [25]. Other examples are rumor detection on Twitter using sentence embedding classification with supervised learning techniques [26], citation intent classification using BERT as word embedding with kMeans and HDBSCAN [27], BERT as article feature extraction with CNN as the classifier for sentiment classification [28], FakeBERT using BERT as a word embedding model to detect fake news in social media [29], and BERT-ACNNs that concatenate BERT sentence embedding and word2vec sentence embedding to obtain a new sentence embedding and utilize it for classification [30]. However, none of these studies have focused on how to reduce GPU memory usage for long sentences. A ''sentence'' can be an arbitrary span of contiguous text rather than an actual linguistic sentence [9]. In this work, ''long sentence'' refers to a sentence with a token length of at least 512 tokens, the limit set for BERT tokenization.

A multilingual system is also the focus of this paper as many countries commonly use two or more different input languages for daily communication. With the release of multilingual BERT or mBERT [9], a single mBERT model can process multilingual input with promising results. Thus, this paper will evaluate the mBERT model for different classification approaches to study the extent of its capability when multiple languages are used as input sentences.

mBERT was pre-trained with concatenated Wikipedia data for 104 languages without cross-lingual alignment. mBERT performs well on zero-shot cross-lingual transfer when the source and target languages are similar in Document Classification, NLI, NER, POS tagging, and Dependency Parsing [31]. Zero-shot cross-lingual transfer refers to training and selecting a model in a resourceful source language and transferring it directly to the target language [31]. Other studies have focused on alignment methods to induce cross-lingual signals in contextual embeddings, based on two methods [32], which are Rotation Alignment [33] and Fine-tuning Alignment [34]. However, none of them address the concern of a multilingual input system with more than two languages as input.

In this regard, this paper proposes a multilingual sentiment analysis system using mBERT for long sentences that avoids the fine-tuning process of the pre-trained mBERT model while simultaneously maintaining the process at low complexity and less resource usage. The solution is to feed the sentence embeddings to multiple conventional classifiers and adopt a feature reduction algorithm to reduce the length of the sentence embeddings, resulting in reduced training time and GPU memory usage owing to the modification of the training process.

II. METHODOLOGY

This section discusses the proposed multi-language sentiment analysis system using a pre-trained language model with no fine-tuning, as shown in Figure 1. First, the input representation from the mBERT tokenizer is fed into the mBERT model to obtain a vector of length 768 called sentence embeddings. The sentence embeddings are then reduced to a smaller vector using a feature selector. Finally, the reduced sentence embeddings are used to classify the input sentence into sentiment categories using machine learning classifiers after the language of the input sentence is identified using a language detector. In this paper, two languages are considered, namely English and Malay.

A. BERT & mBERT

As shown Figure 1, the input sentences are first fed into the mBERT tokenizer, splitting the words in the input sentences into smaller subwords and characters, and finally converting them into a sequence of numerical representation according to the WordPiece [35] vocabulary (30k for BERT, 110k for mBERT) as shown in Figure 2. This numerical representation is called token embeddings, which also consists of special tokens, namely [CLS] and [SEP]. The [CLS] token always

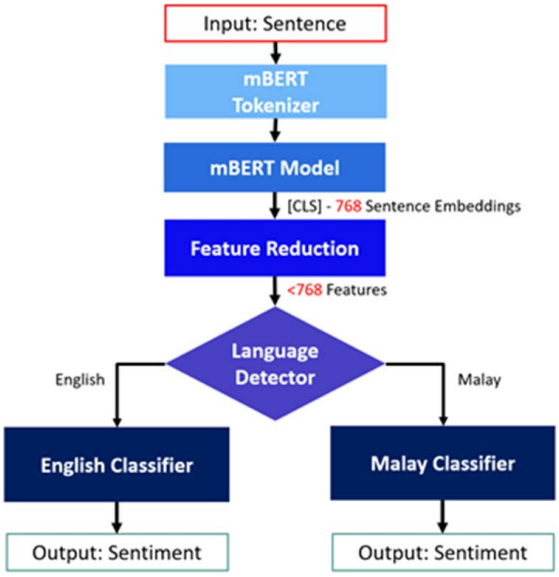

FIGURE 1. Proposed multilingual sentiment analysis system using mBERT.

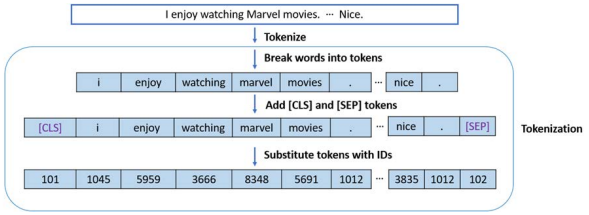

FIGURE 2. Tokenization by BERT tokenizer.

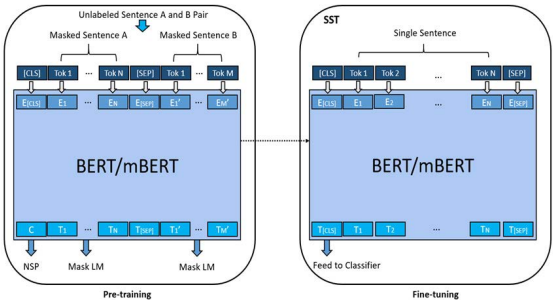

FIGURE 3. Illustration of BERT on pre-training and fine-tuning.

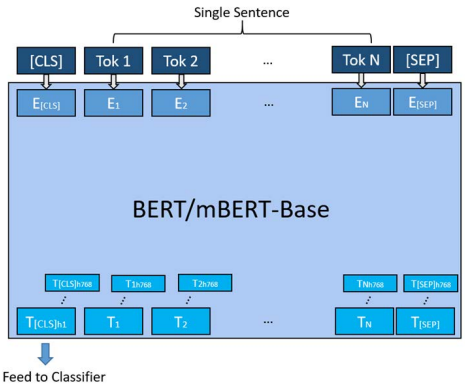

FIGURE 4. Illustration of Single Sentence Classification task.

appears at the start of the token embeddings, representing the sequence of words used for classification tasks. For the input considered to have two sentences, the [SEP] token separates the sentences.

Besides the token embeddings, segment embeddings are created to indicate whether the token belongs to a specific sentence. Subsequently, position embeddings are created to label the sequence number of the tokens. For a given token, its input representation is constructed by summing the token embeddings, segment embeddings, and position embeddings. This input representation is then fed into the pre-trained mBERT model to obtain a vector of sentence embeddings. In total, there is a maximum of 512 input tokens where the first token is set with special [CLS] token. Any long sentence with more than 512 tokens, which is approximately 100+ words, will be truncated because the limit of the BERT/mBERT is 512 tokens. In contrast, sentences with fewer than 512 tokens can be padded to have longer tokens. For example, to generate 512 tokens from "I enjoy watching Marvel movies.", the sentence is first broken into words in the format of ['[CLS]', 'i', 'enjoy', 'watching', 'marvel', 'movies', '.', '[SEP]', '[PAD]', '[PAD]', '[PAD]', ..., '[PAD]'] and then converted to 512 numerical tokens as [101, 1045, 5959, 3666, 8348, 5691, 1012, 3835, 0, 0, 0, ..., 0]. Zero padding is added to the resulting 8 tokens generated for the sentence to complete the 512 tokens.

For the following process, the critical component of the BERT/mBERT model is the bidirectional training of the Transformer, which is a model that learns the contextual relations between words in sentences. The bidirectional encoder trains an unlabeled sentence by jointly conditioning both the left and right contexts of the sentence in all layers with the Masked Language Model and Next Sentence Prediction (NSP) as the pre-training objectives [9]. With 768 hidden units, the mBERT model outputs an encoded vector with a length of 768 for each input token. BERT/mBERT is pre-trained with a large corpus of training data and can be fine-tuned to fit downstream tasks, such as sentiment analysis, allowing users to enjoy the excellent performance of the pre-trained BERT/mBERT, as shown in Figure 3. For classification tasks, the first position vector output where the [CLS] token is placed can be used as the input to the classifier, as shown in Figure 4. Since this paper implements the mBERT model with no fine-tuning, the 768 sentence embedding vector obtained from the [CLS] token position is utilized. Note that the 768 hidden units are the size of BERT-base, while the BERT-large model has 1024 hidden units.

B. FEATURE REDUCTION

Since the BERT-base model has 768 hidden units, it produces 768 sentence embeddings for every input sentence. When the dataset is extensive, millions of unique numbers are generated. For example, a single dataset consisting of 10k

sentences will produce approximately 7.68 million unique numbers, which would significantly increase memory usage during classifier training. The proposed solution is to adopt feature reduction algorithms for the 768 sentence embeddings to retain only the essential and informative features, which will significantly reduce the number of features to be fed to the classifier compared to the features represented by the original sentence embeddings. In general, feature reduction algorithms reduce features by removing irrelevant, redundant, or noisy features [36], where many established algorithms currently exist. The model's performance degraded when the features are irrelevant or redundant [37]. In this case, nine feature reduction algorithms were selected for analysis in this paper, and a detailed discussion can be found in Section III-B.

C. SENTENCE EMBEDDING CLASSIFICATION

For the BERT-base model, which consists of 12 layers of Transformer encoders, fine-tuning the model's parameters by training a new dataset requires high GPU memory to perform the task. Therefore, this paper proposes a system without a fine-tuning process. In contrast, the proposed method applies the mBERT sentence embedding classification.

The pre-trained mBERT model used in this paper can generate sentence embeddings for 104 languages as it has been pre-trained with them. mBERT can be used to fine-tune a single language and perform tasks with the input of another language. However, the mixed pre-trained language of mBERT could still produce different sentence embeddings for two different languages with the exact meaning of the input sentence. This leads to misclassification. To avoid this error, separate classifiers trained for each language have been proposed. As shown in Figure 1, the reduced sentence embeddings of the mBERT model are classified by either the trained English classifier or the Malay classifier, depending on the detected language of the input sentence. In this work, the classifiers will be trained using a low-complexity conventional classifier, and the Google Translate API will be used to detect the language of the input sentence.

III. EXPERIMENTAL RESULT

This section discusses the experimental results for the proposed sentiment analysis system based on the Stanford Sentiment Treebank (SST) dataset [38]. The discussion in Section III-B sets the experiments on implementing the feature reduction methods where the redundancy of the sentence embeddings is explored. Then, with knowledge of the applicability of the feature reduction methods, the discussion in Section III-C focuses on the performance of the sentence embedding classification approach compared to the fine-tuning approach. Several classifiers were tested, and their accuracies are analyzed. Other than the accuracy, another crucial measure for the proposed sentence embedding classification approach is memory usage and runtime, which are discussed in Section III-D. Finally, the discussion in Section III-E explores the performance of mBERT in handling two languages, English and Malay, when implemented as a single system, as depicted in Figure 1.

TABLE 1. SST-3 dataset.

Dataset	English SST-3	Malay SST-3
Classes	3 Classes	3 Classes
Language	English	Malay
Data size	8000	8000
Class Distribution	Negative: 3036 Neutral: 1521 Positive: 3473	Negative: 3036 Neutral: 1521 Positive: 3473

A. EXPERIMENTAL SETUP

The datasets used for the experiment are three variants of the Stanford Sentiment Treebank (SST) dataset, namely SST-2, SST-3, and SST-5. Accordingly, SST-2, SST-3, and SST-5 consist of two, three, and five sentiment levels, respectively. While SST-2 and SST-5 are the original datasets, SST-3 is a simplified version of the SST-5 dataset. In SST-3, SST-5 levels 1-2 are grouped as Negative (0), level 3 as Neutral (1), and levels 4-5 are grouped as Positive (2). Since the proposed multilingual system handles the English and Malay languages, the English version of the SST-3 dataset was translated to Malay using the Google Translate API and used as the main dataset for testing. SST-2 and SST5 are then used as the references for comparison with the SST-3 dataset. Table 1 details the SST-3 dataset used in this paper.

The pre-trained BERT used in the experiments is uncased BERT-Base, while the pre-trained mBERT used is cased mBERT-Base with 12 layers of Transformer blocks, 768 hidden layer size, 12 attention heads, and 110M parameters [9]. 8000 samples are used to train the classifiers for the classification process, of which 75% are used for training, 12.5% for validation, and 12.5% for testing. In fine-tuning the BERT/mBERT model, the output layer is set as a linear layer, the epoch number is set at 3, the base learning rate is set at 2e-5, the batch size equals 16, the dropout probability is maintained at 0.2, and the model is run with the AdamW optimizer.

The processor used in the experiments is an AMD Ryzen 7-3700X with 32GB RAM and 512GB SSD, while the GPU used is an Nvidia RTX3090 with 24GB. The experimental environment is Python 3.7 with related libraries.

B. FEATURE REDUCTION

In this section, the classification performance of the English SST-3 dataset is analyzed when feature reduction algorithms are applied to 768 sentence embeddings. In total, nine feature reduction algorithms were tested. These nine algorithms can be categorized as component-based, filter-based, and other methods. Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are categorized as component-based, where both methods find a new set of basis vectors for the data. Filter-based techniques, such as

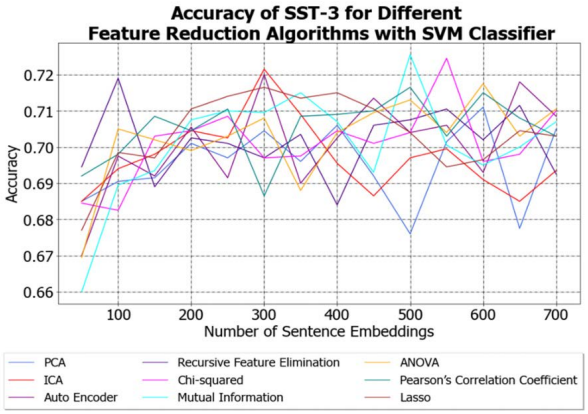

FIGURE 5. Accuracy of SST-3 for different feature reduction algorithms.

ANOVA (Analysis of variance), Pearson's coefficient correlation, mutual information, and chi-square select features based on statistical tests for their correlation with the outcome variable.

Figure 5 shows the results of the nine feature reduction algorithms with a Support Vector Machine (SVM) used as the classifier and the accuracies measured at 50 feature intervals. The results show that even with fewer features, some classification accuracies are higher by 1% to 2% compared with the 70.65% classification accuracy when using the original 768 sentence embeddings. At 200 features, which is almost 75% less than the original sentence embeddings, the accuracy across the nine feature reduction algorithms is consistent at approximately 70%. In contrast, the classification accuracy separation is wider for feature lengths greater than 200. This is evidence of overfitting, where the information redundancy is high when more than 200 features are used.

Figure 6 explains the overfitting situation with the PCA's explained variance ratio, a metric that measures the percentage of variance attributed to each of the selected principal components. It can be observed that the first 200 principal components explained the majority of the variance by 88.32%, an indication of sufficient information represented compared to the original 768 sentence embeddings. With additional principal components at 350, the explained variance ratio is close to 95%, showing that more than half of the original sentence embeddings are insignificant.

The insignificance of most features in sentence embeddings is also evident in the filter-based method. While PCA converts sentence embeddings into a new set of features, the filter-based feature reduction method reduces the original features based on a measure called feature importance. For ANOVA, a variant of the filter-based method, the feature importance is measured based on the F-value, which compares the variance between all features and the variance within the feature. The higher the value, the more important the feature. Figure 7 shows the cumulative feature importance obtained using the ANOVA method after normalizing the feature importance. The 350 best-selected features contribute to 89% of the feature importance. The plot also clearly indicates

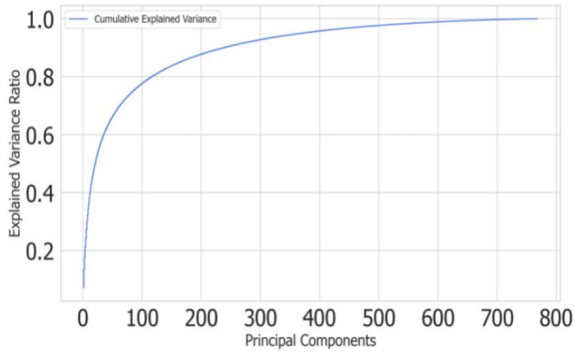

FIGURE 6. Explained variation ratio of PCA.

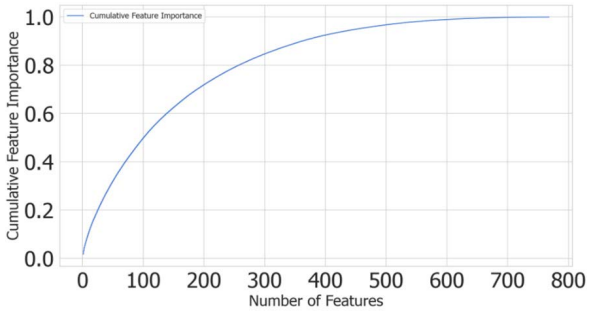

FIGURE 7. Cumulative feature importance of ANOVA.

that more than half of the original sentence embeddings are insignificant, which is the same finding as that of the plot of the explained variation ratio of PCA.

C. SENTENCE EMBEDDING CLASSIFICATION

Section III-B demonstrates that using PCA, the number of features can be reduced to approximately a quarter of the 768 sentence embeddings and still output comparable classification performance to the fine-tuning approach. While SVM is the only classifier used in Section III-B experiments, this section explores the performance of other classifiers to demonstrate the applicability of the sentence embedding classification approach as opposed to the fine-tuning approach. This section also discusses the memory usage and processing time of both the fine-tuning approach and the sentence embedding classification approach, where the purpose of implementing sentence embedding classification is to achieve less memory usage and faster processing time.

The machine learning algorithms used in the experiments are Support Vector Machine (SVM), Logistic Regression, Multilayer Perceptron (MLP), Gaussian Naive Bayes, Gaussian Process, k-nearest, decision tree, Adaboost, Gradient Boosting, Histogram Gradient Boosting, and Random Forest. From the SST-3 data set experiments run on the original 768 sentence embeddings, SVM was identified as the best-performing classifier with 70.65% accuracy for sentence embedding classification as shown in Figure 8 because of its nature and effectiveness in high-dimensional space. The result is a mere 1% difference compared to the fine-tuning

approach, indicating that the sentence embedding classification approach is as effective as fine-tuning. Other classifiers with comparable accuracy are logistic regression, Gaussian Process, and histogram gradient boosting, whereas the rest of the classifiers recorded accuracies below 68%. Note that these classifiers are implemented using the default parameters set by the Scikit-learn library. It is a possible that, with proper parameter settings, the performance of these classifiers can be improved. However, this is not the focus of the present paper.

Interestingly, the classification performance improved slightly for the seven tested classifiers when the experiments were conducted with a reduced set of 200 PCA features (PCA-200), as shown in Figure 8. This strengthens the finding that most sentence embedding features are redundant. Thus, the sentence embedding classification approach is an excellent option for performing the NLP classification task compared to the resource-intensive fine-tuning approach, where 768 sentence embeddings have been proven to be more than sufficient to represent the input sentence.

Referring to Figure 8, the Gaussian Process provides the best performance with PCA-200. However, even with reduced features, the Gaussian Process classifier requires an extended processing time of 1070.96s to complete the training, which deviates from the purpose of implementing the sentence embedding classification approach. The Gaussian process is a neural network type classifier, which explains the poor runtime that scales with the number of samples. The complexity of the Gaussian process is a result the matrix inversion of large covariance matrices. Thus, a low complexity classifier should be chosen for sentence embedding classification. In this case, SVM, the second performed classifier for PCA-200 shown in Figure 8, is superior in that it only requires 4.34s to complete the training process. Therefore, the following discussion focuses on using SVM as the classifier for the sentence embedding classification approach.

D. MEMORY USAGE AND RUNTIME

Memory in BERT is required to store the input data and model parameters when the input data propagates through the network. With the huge 110M model's parameters, BERT-base, like the other deep learning models, requires huge memory in training the samples. For the larger variant, the BERT-large model, tripled to 345M parameters.

Table 2 shows the usage of 15.5GB of GPU memory for the BERT-base fine-tuning process on the SST-3 English dataset carried out with 3 epochs. For comparison, sentence embedding classification using the original sentence embeddings (SEC SE-768) only consumes 1.7GB of GPU memory, which is almost 89% less memory. With a comparable classification accuracy of approximately 71%, the SEC SE-768 approach has a clear advantage over the fine-tuning process where over-provisioned and costly hardware resources for the training process can be avoided. Note that GPU memory consumption is consistent with the SEC approach even when the reduced features of PCA-200 are used. This is due to the

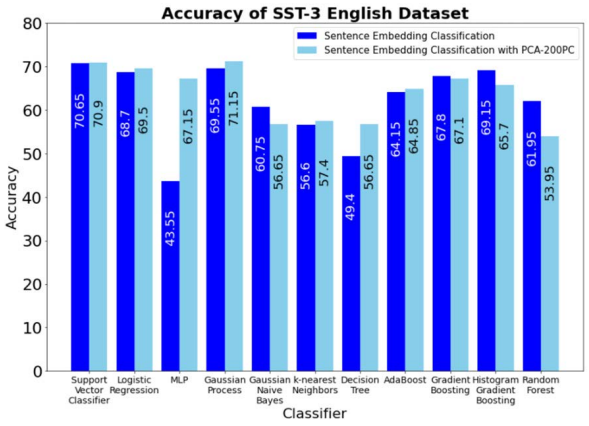

FIGURE 8. Accuracy of SST-3 English dataset with different classifiers.

TABLE 2. Memory usage comparison between fine-tuning and sentence embedding classification.

Method	GPU Memory Usage	Runtime	Accuracy	Throughput
Fine tuning	15.5GB	361.87s	72.40%	22.11
SEC SE-768	1.7GB	106.16s	70.65%	75.36
SEC PCA-200	1.7GB	97.34s	70.90%	82.19

fact that the GPU is only used to handle the BERT processing in encoding the input sentence into the numerical sentence embedding vector while the low complexity SVM training process is run on the computer's CPU.

Since the SEC approach requires heavy processing only to generate the sentence embeddings, the runtime to complete the training process (from feeding the samples to the BERT model until the testing phase) is fairly fast at 106.16s when compared to the 361.87s of training using the fine-tuning approach as listed in Table 2. This is expected because the SEC approach does not require updating the 110M parameters of the BERT model. For the SEC SE-768 method, more than 85% of the runtime is used by BERT to generate sentence embedding, which requires an average runtime of 93s. The SVM classifier then requires no more than 20s to complete the training process. Thus, the runtime did not deviate much for the SEC PCA-200, which clocked merely approximately 9 seconds faster than the SEC SE-768 that used the original sentence embeddings as the input feature to the SVM classifier. Oppositely aligned to this outcome, the throughput, which refers to the number of input samples processed in one second, results in 22.11, 75.36, and 82.19 samples/sec for fine-tuning, SEC SE-768 and SEC PCA-200 approaches, respectively. For this experiment, 8000 samples were used.

Apart from the 12 Transformer encoders, 768 hidden units, and 12 attention heads, the length of the input token adds to the memory usage of the BERT model. Previously, Table 2 clearly shows that fine-tuning a long sentence task that uses a maximum of 512 tokens requires an over-provisioned and expensive GPU to accommodate the high 15.5 GB GPU memory usage. The higher the token

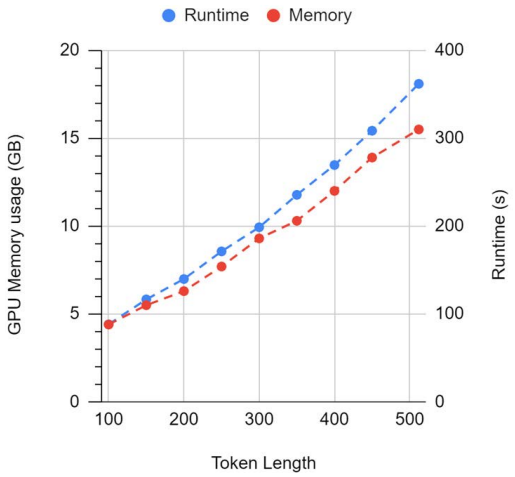

FIGURE 9. GPU memory usage and runtime for different token lengths.

length, the higher the GPU memory usage as the GPU needs to hold a larger dimension of the input vectors for computation. As the token length decreased, the GPU memory usage and training runtime decreased linearly, as depicted in Figure 9. Although a shorter token length means less computational effort, it is only applicable to short sentence processing. Therefore, the proposed sentence embedding classification approach is preferable. Even with the maximum of 512 tokens, the GPU memory usage of 1.7 GB for the SEC method shown in Table 2 is still lower than the memory usage of 4.4 GB for the 100 tokens of the fine-tuning approach shown in Figure 9.

Table 3 evaluates the performance of the proposed method for different datasets, which are IMDB and SemEval Task 4A. The SemEvalTask 4A dataset is manually balanced with 2667, 2666, and 2667 for the negative, neutral, and positive classes, respectively. The accuracy and training time for the IMDB dataset show improvement for SEC-768 and SEC-PCA-200 compared to the fine-tuning method. However, the accuracies for SemEval Task 4A for SEC-768 and SEC-PCA-200 slightly decline when compared to the fine-tuning method, but both SEC-768 and SEC-PCA-200 show a significant reduction in training time across all datasets, indicating that training can be carried out more frequently and with fewer resources, so that the models can handle the most recent challenge.

Table 4 studies the performance of the proposed method with other pre-trained language models such as ALBERT and DistilBERT. Although ALBERT-base (12M parameter, 12 layers and 768 hidden units) and DistilBERT-base (40% lesser parameter than BERT-base) are smaller than the BERT-base, 14.7GB and 9.0GB GPU memory usage are needed to do the fine-tuning process respectively which is expensive, indicating that SEC-768 and SEC-PCA-200 are needed to perform the task in a low resource and lesser training time

TABLE 3. Performance comparison between fine-tuning and sentence embedding classification for different datasets.

Dataset	Method	Accuracy	F1	Recall	Precision	Training Time
IMDB	Fine-tuning	91.60%	91.48%	91.85%	91.11%	428.42s
	SEC-768	100.00%	100.00%	100.00%	100.00%	93.14s
	SEC-PCA-200	99.00%	99.00%	99.00%	99.00%	93.66s
SemEval Task 4A	Fine-tuning	73.10%	73.10%	73.10%	73.10%	350.27s
	SEC-768	66.20%	66.20%	66.20%	66.20%	108.90s
	SEC-PCA-200	66.40%	66.40%	66.40%	66.40%	97.27s
SST-3	Fine-tuning	72.40%	72.40%	72.40%	72.40%	366.90s
	SEC-768	70.65%	70.65%	70.65%	70.65%	108.01s
	SEC-PCA-200	70.90%	70.90%	70.90%	70.90%	97.10s

TABLE 4. Performance comparison between fine-tuning and sentence embedding classification for different models.

Model	Method	GPU Memory Usage	Accuracy	Training Time
ALBERT-base	Fine-tuning	14.7GB	69.10%	374.09s
	SEC-768	**1.4GB**	70.85%	107.36s
	SEC-PCA-200	**1.4GB**	70.05%	**97.04s**
DistillBERT-base	Fine-tuning	9.0GB	72.20%	179.77s
	SEC-768	**1.7GB**	69.60%	108.30s
	SEC-PCA-200	**1.7GB**	67.45%	**97.22s**

E. MULTILINGUAL CLASSIFICATION TASK

This section evaluates the performance of the mBERT model for multilingual system. To provide a basis for the discussion, Tables 5 and 6 list the classification accuracy when four SST dataset variants are translated using the Google Translate API from English to the targeted language, trained, and tested individually based on the BERT and mBERT models, respectively. The experiments were conducted using a fine-tuning classification approach and two sentence embedding classification (SEC) approaches that use different feature lengths (SE-768 and PCA-200). Table 7 shows the performance of the proposed multilingual system, which handles several languages in a single system.

Among the results in Table 6, the 27.50% fine-tuning approach accuracy of the SST-5 English dataset, which is lower than the accuracy recorded by SEC PCA-200 at 40.85%, can be considered an anomaly. This occurs because the mBERT model experiences underfitting from the trained dataset and is unable to learn the relationship between the sentence embeddings and the output well. By changing the training dropout value from 0.2 (refer to Section III-A) to 0, the accuracy for fine-tuning mBERT with English SST-5 obtained an improved classification accuracy of 42.70%. This indicates that different data sets have different configurations for achieving the best accuracy. Nevertheless, the 0.2 dropout value is maintained for all experiments to obtain unbiased results.

In terms of the GPU memory usage, the fine-tuning of the mBERT model used a consistent 16.5 GB for all SST

TABLE 5. Classification performance on different SST datasets using the BERT model.

Dataset	Fine-Tuning	SEC SE-768	SEC PCA-200
SST-2 English	91.40%	86.10%	86.25%
SST-3 English	72.40%	70.65%	70.90%
SST-3 Malay	**58.40%**	**51.05%**	**52.50%**
SST-5 English	51.40%	47.60%	48.35%

TABLE 6. Classification performance on different SST datasets using the mBERT model.

Dataset	Fine-Tuning	SEC SE-768	SEC PCA-200
SST-2 English	83.90%	74.80%	75.50%
SST-3 English	67.70%	61.60%	62.65%
SST-3 Malay	**58.80%**	**57.55%**	**58.65%**
SST-5 English	27.50%	40.35%	40.95%

TABLE 7. Average accuracy for the English+Malay multilingual sentiment analysis system.

Input Feature	Classifier	Runtime	Accuracy
SE-768	Single	603.47s	60.07%
SE-768	Multiple	379.78s	59.14%
PCA-200	Multiple	229.01s	59.00%

TABLE 8. Average accuracy for the 5-language multilingual sentiment analysis system.

Input Feature	Classifier	Runtime	Accuracy
SE-768	Single	4200.32s	57.08%
SE-768	Multiple	960.94s	57.26%
PCA-200	Multiple	573.56s	57.77%

dataset variants listed in Table 4, which is 1 GB higher than the 15.5 GB used by the BERT model. For the sentence embedding classification (SEC) approach, GPU memory usage was consistent at 1.7 GB for all datasets regardless of whether the BERT or mBERT model was used. Since the mBERT model does not increase the GPU memory usage when implementing the SEC approach, the GPU memory usage of the proposed method can be kept low, as discussed in Section III-D.

Tables 5 and 6 show the classification accuracy gap between the BERT and mBERT models. Overall, the accuracy is higher for BERT because BERT was pre-trained in English only, while mBERT is more complex with 104 languages pre-training. However, mBERT can handle other languages, which in this case, the Malay language is better. Evidently, Tables 3 and 4 show that the SST-3 Malay dataset classification using mBERT has higher accuracy across the three classification approaches than the BERT, where an approximately 5% accuracy increase was recorded for both SEC approaches. Compared to the decreasing accuracies for the other three English datasets, this opposite, increasing trend shows that the mBERT model works for languages other than English, particularly the Malay language.

Next, experiments were conducted to investigate the performance of mBERT for multi-input language sentiment classification. Two modes of systems were considered, single classifier and multi-classifier, with the latter being the multilingual system proposed in this paper. Table 7 shows the average accuracy of the multilingual tests with 5-fold cross validation. The 5-fold cross validation is implemented to have more confidence in the performance and reduce bias on the test set.

In Table 7, the single classifier mode is where the 8k SST-3 English dataset and 8k SST-3 Malay dataset are combined to train a single SVM classifier with the original mBERT sentence embeddings SE-768. On the other hand, multiple classifiers refer to the individual training of SST-3 English and SST-3 Malay, in which these trained classifiers are then combined into a single system, as shown in Figure 1. The accuracy of the language detector depicted in Figure 1, which uses the Google Translate API, which has the ability to detect a sentence's language, is approximately 99%. Thus, the language detector does not significantly affect the proposed system.

From Table 7, the average accuracy of the single classifier is 60.07%, and it takes 603.47s to complete the process, with 93s used to obtain the mBERT sentence embeddings for a single language and 417.47s for training the SVM classifier. For the multiple classifiers, the two tested features of SE-768 and PCA-200 resulted in almost similar accuracies compared to the single classifier, with a slight decrease of approximately 1%. However, the training time is shorter at 379.78s for SE-768 and better for the PCA-200 at 229.01s. This shows that when a system must handle more than one language, the multi-classifier approach is better than the single classifier, particularly at runtime.

While this paper focuses on English+Malay multilingual input, Table 8 presents the results extended to five languages to validate the results in Table 7. The languages used are English, Malay, Chinese, Arabic, and Indonesian. The training time for 5 languages (40k dataset) with a single classifier is 4200.32s. However, with 5 individual classifiers, the total training time for all classifiers is just 960.94s. Furthermore, feature reduction using the PCA-200 does its job when the training time is reduced by more than 38% with almost similar accuracy when compared to the full features of SE-768. Observing Table 7 and Table 8, the superior performance of the proposed multi-classifier can be seen by the linear increase in runtime when additional languages are considered compared to an exponential increase in runtime for a single classifier. Thus, the multi-classifier approach can keep the complexity of the system low, particularly when adopting the reduced PCA-200 feature. In terms of classification accuracy, both single and multiple classifiers give comparable results for the five language tests, similar to the English+Malay tests shown in Table 7.

IV. CONCLUSION

In this paper, it has been shown that the sentence embedding classification (SEC) approach with several classifiers has comparable performance to the fine-tuning approach of the BERT/mBERT model. Among the tested classifiers, robust and low complexity classifiers such as SVM can benefit from the SEC approach with a shorter training time because the 110M updating parameter in BERT/mBERT is skipped. Furthermore, the proposed method is effective when the feature reduction process applied to BERT/mBERT sentence embeddings reduces the complexity of the system and avoids data overfitting by removing redundant or irrelevant features. More notably, the SEC approach has proven superior to fine-tuning GPU memory usage, which requires only 1.7GB GPU memory usage. This approach allows anyone to enjoy the excellent performance of the pre-trained BERT/mBERT language models without the over-provisioned and expensive GPU, and the training can be conducted more frequently because fewer resources are required.

The GPU memory usage is also not affected by the token length when using the SEC approach because no BERT/mBERT parameter update is needed, giving an advantage, especially for long sentence applications. In contrast, the GPU memory usage for fine tuning the BERT/mBERT models increases depending on the token length, which from the experiments presented in this paper can increase up to 16.5GB. Finally, as part of the SEC approach, this paper has suggested that a multi-classifier in a multilingual sentiment analysis system can keep the system's complexity low without significantly trading off the classification accuracy.

To further benefit from the idea of implementing the multi-classifier SEC approach, methods to restructure long sentences should be explored to improve classification accuracy with low memory usage using pre-trained language models. For instance, filtered words or sentences can be analyzed to check for any information loss and reduce the complexity of the input words or sentences. Another direction is to enhance sentence embeddings to obtain better features by concatenating a few layers of [CLS] from BERT or concatenating them with other sentence embedding methods. In addition, further research can be conducted using datasets from different domain to study its effectiveness with low resource usage.

Except for DistilBERT, there are other knowledge distillation (teacher-student learning) models such as TinyBERT, which performs Transformer distillation at both the pre-training and task-specific learning stages. To train the student model, it is necessary to use BERT that has been fine-tuned for downstream tasks in task-specific learning as a teacher. This increases the training time and complexity, and the distillation process requires more than 22.0GB of GPU memory to run the training with 8000 datasets. This proves that the proposed method uses the least resources and is less expensive for training the model for downstream tasks. Although TinyBERT has an advantage in deployment, that is, low resource usage, the training process requires high resources. For future work, simplification of the computation intensive training process for pre-trained language models can be further studied to allow training to be performed on less resource devices such as edge devices. In addition, the research can be further studied with larger pre-trained language models, such as Longformer on how the training process can run on a low resource device for edge computing with less training time.

REFERENCES

[1] C. Du, H. Sun, J. Wang, Q. Qi, and J. Liao, "Adversarial and domain-aware BERT for cross-domain sentiment analysis," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4019–4028, doi: 10.18653/v1/2020.acl-main.370.

[2] (Jul. 9, 2022). *NLP Overview*. [Online]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/200405/nlp/overview_history.html#:~:text=NLP%20%2D%20overview&text=The%20field%20of%20natural%20language,this%20sort%20of%20translation%20automatically

[3] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339.

[4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2018, pp. 2227–2237.

[5] A. Vaswani, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[6] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018. [Online]. Available: https://openai.com/blog/language-unsupervised/

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, 2019. [Online]. Available: https://github.com/openai/gpt-2

[8] T. B. Brown, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5753–5763.

[11] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-LM: Training multi-billion parameter language models using model parallelism," 2019, *arXiv:1909.08053*.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, *arXiv:1910.10683*.

[13] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained sentiment classification using BERT," in *Proc. Artif. Intell. Transforming Bus. Soc. (AITB)*, Nov. 2019, pp. 1–5, doi: 10.1109/AITB48515.2019.8947435.

[14] M. Hoang, O. A. Bihorac, and J. Rouces, "Aspect-based sentiment analysis using BERT," in *Proc. 22nd Nordic Conf. Comput. Linguistics*, 2019, pp. 187–196.

[15] T. Tang, X. Tang, and T. Yuan, "Fine-tuning BERT for multi-label sentiment analysis in unbalanced code-switching text," *IEEE Access*, vol. 8, pp. 193248–193256, 2020, doi: 10.1109/ACCESS.2020.3030468.

[16] R. Cai, B. Qin, Y. Chen, L. Zhang, R. Yang, S. Chen, and W. Wang, "Sentiment analysis about investors and consumers in energy market based on BERT-BiLSTM," *IEEE Access*, vol. 8, pp. 171408–171415, 2020, doi: 10.1109/ACCESS.2020.3024750.

[17] H. S. Alatawi, A. M. Alhothali, and K. M. Moria, "Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT," *IEEE Access*, vol. 9, pp. 106363–106374, 2021, doi: 10.1109/ACCESS.2021.3100435.

[18] G. Zhang, J. Wu, M. Tan, Z. Yang, Q. Cheng, and H. Han, "Learning to predict U.S. policy change using New York times corpus with pre-trained language model," *Multimedia Tools Appl.*, vol. 79, pp. 34227–34240, Dec. 2020, doi: 10.10072/s11042-020-08946-y.2020.

[19] Z. Jahanbakhsh-Nagadeh, M.-R. Feizi-Derakhshi, and A. Sharifi, "A semi-supervised model for Persian rumor verification based on content information," *Multimedia Tools Appl.*, vol. 80, nos. 28–29, pp. 35267–35295, Nov. 2021, doi: 10.1007/s11042-020-10077-3.
[20] A. Gupta and R. Bhatia, "Ensemble approach for web page classification," *Multimedia Tools Appl.*, vol. 80, pp. 25219–25240, Jul. 2021, doi: 10.100/s11042-021-10891-3.2021.
[21] P. Ganesh, Y. Chen, X. Lou, M. A. Khan, Y. Yang, H. Sajjad, P. Nakov, D. Chen, and M. Winslett, "Compressing large-scale transformer-based models: A case study on BERT," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 1061–1080, Sep. 2021, doi: 10.1162/tacl_a_00413.
[22] X. Pan, A. Jiang, and H. Wang, "Edge-cloud computing application, architecture, and challenges in ubiquitous power Internet of Things demand response," *J. Renew. Sustain. Energy*, vol. 12, no. 6, Nov. 2020, Art. no. 062702, doi: 10.1063/5.0014059.
[23] North Carolina State University. (Apr. 8, 2019). *New Technique Cuts AI Training Time by More Than 60 Percent*. ScienceDaily. Accessed: Jul. 9, 2022. [Online]. Available: https://www.sciencedaily.com/releases/2019/04/190408114322.htm
[24] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.
[25] I. Annamoradnejad and G. Zoghi, "Computational humor using BERT sentence embedding in parallel neural networks," 2020, *arXiv:2004.12765*.
[26] R. Anggrainingsih, G. M. Hassan, and A. Datta, "BERT based classification system for detecting rumours on Twitter," 2021, *arXiv:2109.02975*.
[27] M. Roman, A. Shahid, S. Khan, A. Koubaa, and L. Yu, "Citation intent classification using word embedding," *IEEE Access*, vol. 9, pp. 9982–9995, 2021, doi: 10.1109/ACCESS.2021.3050547.
[28] R. Man and K. Lin, "Sentiment analysis algorithm based on BERT and convolutional neural network," in *Proc. IEEE Asia–Pacific Conf. Image Process., Electron. Comput. (IPEC)*, Apr. 2021, pp. 769–772, doi: 10.1109/IPEC51340.2021.9421110.
[29] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a bert-based deep learning approach," *Multimedia Tools Appl.*, vol. 80, no. 8, pp. 11765–11788, 2021, doi: 10.1007/s11042-020-10183-2.2021.
[30] B. Yang, D. Li, and N. Yang, "Intelligent judicial research based on BERT sentence embedding and multi-level attention CNNs," in *Proc. 2nd Int. Conf. Inf. Sci. Electron. Technol. (ISET)*, 2019, pp. 226–232.
[31] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 833–844.
[32] S. Kulshreshtha, J. Luis Redondo-García, and C.-Y. Chang, "Cross-lingual alignment methods for multilingual BERT: A comparative study," 2020, *arXiv:2009.14304*.
[33] T. Schuster, O. Ram, R. Barzilay, and A. Globerson, "Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing," in *Proc. NAACL-HLT*, vol. 1, 2019, pp. 1599–1613.
[34] S. Cao, N. Kitaev, and D. Klein, "Multilingual alignment of contextual word representations," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*. Addis Ababa, Ethiopia, 2020. [Online]. Available: https://openreview.net/pdf?id=r1xCMyBtPS
[35] Y. Wu, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.
[36] J. Miao and L. Niu, "A survey on feature selection," *Proc. Comput. Sci.*, vol. 91, pp. 919–926, Jan. 2016, doi: 10.1016/j.procs.2016.07.111.2016.
[37] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.
[38] R. Socher, A. Perelygin, and J. Wu, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2013, pp. 1631–1642.

YUHENG KIT was born in 1997. He is currently pursuing the Ph.D. degree with Universiti Teknologi Malaysia. His research interests include natural language processing, face recognition, and deep learning.

MUSA MOHD MOKJI received the M.Eng. and Ph.D. degrees specializing in image processing from Universiti Teknologi Malaysia, in 2001 and 2008, respectively. He is currently an Associate Professor with the Faculty of Engineering, Universiti Teknologi Malaysia, where he is also the Head of the Digital Signal and Image Processing Research Group. His research interests include signal and image processing, pattern recognition, data mining, applying these models to agriculture, surveillance systems, natural language processing, and medical. He has published more than 50 publications in these areas. He gives lectures on signal processing and image processing to undergraduate and postgraduate students at Universiti Teknologi Malaysia.