*Journal of Quantum Computing*
DOI: 10.32604/jqc.2022.026658
***Article***

Tech Science Press

# Online News Sentiment Classification Using DistilBERT

**Samuel Kofi Akpatsa[1,*], Hang Lei[1], Xiaoyu Li[1], Victor-Hillary Kofi Setornyo Obeng[1], Ezekiel Mensah Martey[1], Prince Clement Addo[2] and Duncan Dodzi Fiawoo[3]**

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China
[2]Faculty of Applied Sciences and Mathematical Education, Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development, Kumasi, Ghana
[3]Jasikan College of Education, University of Cape Coast, Jasikan, Ghana
*Corresponding Author: Samuel Kofi Akpatsa. Email: 201814090019@std.uestc.edu.cn
Received: 15 December 2021; Accepted: 28 April 2022

**Abstract:** The ability of pre-trained BERT model to achieve outstanding performances on many Natural Language Processing (NLP) tasks has attracted the attention of researchers in recent times. However, the huge computational and memory requirements have hampered its widespread deployment on devices with limited resources. The concept of knowledge distillation has shown to produce smaller and faster distilled models with less trainable parameters and intended for resource-constrained environments. The distilled models can be fine-tuned with great performance on a wider range of tasks, such as sentiment classification. This paper evaluates the performance of DistilBERT model and other pre-canned text classifiers on a Covid-19 online news binary classification dataset. The analysis shows that despite having fewer trainable parameters than the BERT-based model, the DistilBERT model achieved an accuracy of 0.94 on the validation set after only two training epochs. The paper also highlights the usefulness of the ktrain library in facilitating the building, training, and application of state-of-the-art Machine Learning and Deep Learning models.

**Keywords:** Natural language processing; DistilBERT; text classification; sentiment analysis

## 1 Introduction

Online news networks have become reliable platforms that provide useful information in educating and informing the public with the latest updates and current happenings around the globe. Due to the massive amount of textual data generated daily from these networks, the use of Machine Learning and NLP techniques to analyze and classify such content allows researchers and the scientific community to understand whether the public perception of developing events has a positive, negative, or neutral sentiment. With the advancements of Deep Learning models, the research interest of the NLP community has shifted towards the distributed representation of words (word embeddings)

which extract rich semantic features from text sequences effectively improve the performance of learning algorithms [1–4]. The use of word embeddings has given better performance compared to the traditional bag of words model.

Although Machine Learning and Deep Learning models are still relevant in NLP [5–7], the paradigm shift towards the concept of transfer learning has seen the introduction of several pre-trained models, allowing researchers to achieve cutting-edge performance across multiple benchmarks. One such approach that has been more successful and supported by a wave of research is the transformer-based BERT model developed by Google [8]. Since its introduction, BERT model has become the model of choice for many researchers and practitioners in NLP, replacing the RNNs [9]. Due to the learned knowledge from the BERT model being pre-trained using a large unlabeled corpus, fine-tuning them on downstream tasks have become the most successful technique in NLP [10]. The pre-trained BERT models have shown to document impressive results and even outperform human beings on many tasks.

However, despite their impressive performance, BERT models require massive computational resources and large amounts of data for training. As a result, many studies in the recent past have explored the concept of knowledge distillation to reduce the large-scale BERT model into a compact model without losing its capabilities. Knowledge distillation has proven to be a useful pruning technique that can yield significant speedup and memory size reduction while preserving much capacity of the original model [11,12]. Moreover, the complexity of the transformer-based models remains one of the major challenges that impede widespread adoption by the research community [13]. However, the ktrian library developed by [14] is designed to make state-of-the-art Machine Learning and Deep Learning models more accessible and easier to apply by practitioners. ktrain provides several pre-trained models for fine-tuning on NLP tasks, such as sentiment analysis, document classification, named entity recognition, part of speech tagging, etc.

One of the most popular pre-canned text classifiers under the ktrain library is DistilBERT model. It is an efficient and scaled-down distilled version of BERT model purported to be smaller, lighter, cheaper, and faster, yet retain 97% of BERT's capabilities [15]. The fine-tuning performance of the DistilBERT model on binary classification tasks suggests that it can generalize better than other pre-trained language representation models on a wide range of downstream NLP tasks [16].

This paper fine-tuned five different pre-canned text classifiers from the ktrain library on a binary classification dataset. These classifiers include Logistic Regression (logreg), Naïve Bayes infuses with Support Vector Machines (nbsvm), Bidirectional Gated Recurrent Unit (bigru), BERT-base model, and DistilBERT models. The main objective is to evaluate the performance of DistilBERT model in predicting the sentiment category for a given Covid-19 related online news. Our analysis also contributes to a better understanding of the usefulness of the ktrain library in developing and training Machine Learning and Deep Learning models for NLP tasks. The rest of the paper is structured as follows. Section two describe the dataset and the experimental setup for this study. Section three presents an overview of the models used for the classification task. Section four reports the results from the experimental evaluation of the models, while the last section draws conclusions and suggestions for future work.

## 2 Experimental Setup

### *2.1 Dataset*

This study used data collected from the websites of three most popular online news providers (*10news.com, cnn.com, and foxla.com*). A scraper was used to download more than 10000 articles from

May 2020 to September 2020 using 'covid-19' and 'coronavirus' as the keywords. Thus, the collected data is highly related to covid-19 as the text was about articles on covid-19 at the websites of the online news providers. The downloaded articles varied significantly in the number of sentences and word counts. To normalize the text size, we reconstruct the news articles to create paragraphs with approximately ten sentences to utilize as many data points as possible. A script that automatically applies NLTK's sentence tokenizer was run to extract 41839 text entities to be labeled into different sentiment categories. Data labeling was handled manually with human annotation. The annotators assigned 1 (positive class) for texts with positive sentiment and 0 (negative class) for texts with negative sentiment. Some texts were removed from the dataset due to the difficulty of interpretation. The final dataset consisted of 33324 texts, where 20810 were assigned to the positive class, and 12489 were assigned to the negative class. The developed dataset was published on Mendeley data repository (https://data.mendeley.com/datasets/r6nn5s37tp/2).

We analyzed the length of the text documents in terms of the number of words and characters and found that it consists of fairly long samples. The average number of words for each text is 210, with the longest document containing 1104 words and 8429 characters. Figs. 1 and 2 represent the distribution of the words and characters counts of the dataset. The words and characters are both right-skewed with a minimal number of outliers creating a right tail.

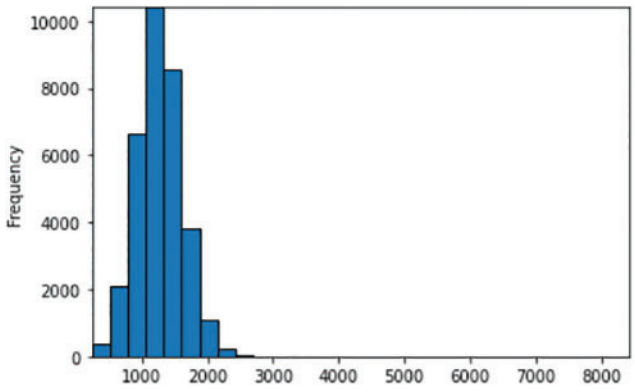


**Figure 1:** Frequency distribution of word counts

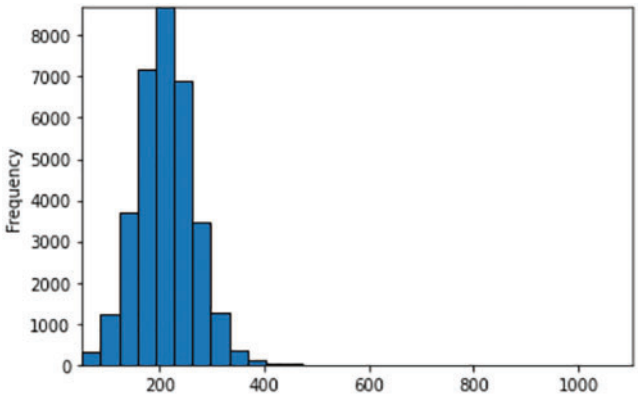


**Figure 2:** Frequency distribution of character counts

### *2.2 Data Preprocessing*

Even though texts from online news articles come with much fewer irregularities and noise, we perform some fundamental cleaning and preprocessing to improve the quality of the dataset. Texts from the dataset were filtered using a python script to remove background noises such as white spaces, punctuations, hashtags, urls, special characters, hyperlinks, and stop words, while all letters were converted to lower case. The data-loading function within the ktrain package allowed us to load the preprocessed text data into a suitable format for the various text classifiers. Finally, different experiments were applied to analyze which classifier performed best on the dataset.

### *2.3 Fine-Tuning*

The selected classifiers were fine-tuned on the dataset using the ktrain library written in the Python programming language. The dataset was randomly split into 70% training set and 30% test set using stratified sampling. We use the get_learner function to load and wrap the model and the data into a learner object used to train the network. We employ 1cycle policy learning rate schedule with a maximum learning rate of 2e−5, and a Stochastic Gradient Descent with Restart (SGDR) policy learning rate schedule to automatically adjust the learning rates during training [17,18]. To protect model training from over-fitting, we trained the logreg and nbsvm classifiers for 32 epochs over a batch size of 16. The deep learning bigru model was trained for 16 epochs with a batch size of 8. Due to the relatively large size of the transformer-based bert and distilbert models, training was carried out for only two epochs with a batch size of 6. Optimization of the hyper-parameters in the training of each distinct model was done using Adam Optimizer.

Experimentations were carried out in Google Colab on Graphics Processing Unit (GPU) with the following configurations: GPU 0: Tesla T4, with 12 GB of RAM and 68 GB of disk space. The Google Colab Notebook is an integrated development environment (IDE) that runs in the cloud and highly integrated with Google Drive. As a result, the GPU offers speedups that significantly accelerate the training procedure.

## 3 Methodology

### *3.1 Models*

ktrain is an open-source Python package that acts as a lightweight wrapper for TensorFlow and many other libraries, making it easy to build, train, and deploy Machine Learning models on standard desktop. This study selected five pre-canned text classifiers (logreg, nbsvm, bigru, bert, and distilbert) from the ktrain library and fine-tuned them on the dataset. These classifiers have been categorized into Machine Learning, Deep Learning, and Transformer-based models for text classification.

#### *3.1.1 Machine Learning Classifiers*

The first set of pre-canned classifiers under the ktrain library are the logreg and nbsvm Machine Learning algorithms. These classifiers were included in the experiment to serve as a comprehensive baseline. The logreg classifier uses a trainable embedding layer, while the nbsvm classifier uses a linear Support Vector Machines (SVM) model infuses with a Naive Bayes (NB) log-count ratios as features [19]. These classifiers are much simpler than the deep learning and transformer-based models but have proven to be fast and efficient on a wide range of text classification tasks [19–21].

### *3.1.2 Deep Learning Classifier*

The Deep Learning approach used in our experiment is the bidirectional variant of Gated Recurrent Unit (BiGRU) model for the classification task. BiGRU is a more advanced variant of RNN and has proven to be less complex than BiLSTM [22]. BiGRU works as a better window-based feature extractor to extract textual information from both directions simultaneously, improving the context understanding of words. The input to the bigru model is a word vector obtained from a pre-trained fasttext word vector. fasttext differs from Word2vec in that it considers character n-grams as the smallest unit rather than a word. Thus, the vector of a word is the sum of its character n-grams vectors. This makes it appropriate for addressing morphological richness and has proven effective for text classification tasks [23].

### *3.1.3 Transformer-based Model*

The transformer-based BERT model is a general-purpose language model pre-trained on a very large corpus of unlabeled text, including the entire English Wikipedia (2500 M words), and Book Corpus (800 M words). The many parameters allow them to dig deep and learn how language works. However, training the pre-trained BERT model takes a longer inference time, making it difficult to deploy on edge devices such as mobile phones. Recent literature suggests that due to the redundancy in pre-trained language models such as BERT, it is essential to reduce the computational overhead in such models [24]. Different model compression techniques have been proposed in literature to speed up inference of deep models and reduce model size while retaining their performance [25]. The most commonly used technique is knowledge distillation in a teacher-student framework [26]. Knowledge distillation aims at training a small student model to replicate the behavior of a large teacher model. The memory usage and the time overhead are both decreased when using a small distilled model for inference. Successful implementation of this concept has seen the proposal of a method to pre-train a smaller general-purpose language representation model called DistilBERT [15]. The smaller DistilBERT model has demonstrated to produce good performances similar to the larger BERT model when fine-tuned on a wide range of downstream tasks.

The general architecture of the DistilBERT model is similar to that of the BERT model [8], except that it has 40% fewer trainable parameters and is intended for environments with limited computing resources [15] (Tab. 1). Other variants of the BERT model such as RoBERTa, TinyBERT, and ALBERT, have very recently produced state-of-the-art performances in many NLP tasks [10,27,28]. The success of BERT-like models stems from their ability to capture bi-directional contextual information from text sequences during the training phase. In addition, the compactness of the DistilBERT makes them scalable and capable of producing state-of-the-art performance on most NLPT tasks in real-time while preserving 97% of BERT's capacity.

**Table 1:** Differences between BERT-base and DistilBERT-base models

| Trainable parameters | BERT-base | DistilBERT-base |
|---|---|---|
| No. of layers (transformer blocks) | 12 | 6 |
| No. of hidden units | 768 | 768 |
| No. of self-attention heads | 12 | 12 |
| Total trainable parameters | 110 M | 66 M |

### *3.2 Metrics of Evaluation*

The primary metrics for comparing the models were accuracy, precision, recall, and f1-score. The metrics were calculated in terms of the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The accuracy is calculated as the number of all the correct predictions divided by the total number of instances (Eq. (1)). The precision is calculated as predicted instances which were correct divided by the size which was the predicted size of the instance (Eq. (2)). The recall, also known as sensitivity, is measured as the total correctly predicted instances divided by the actual number of instances (Eq. (3)).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

### *3.3 Dealing with Data Imbalance*

Our dataset consists of uneven class distribution. This could lead to a bias towards the majority class at the expense of the minority class as the classifiers attempt to achieve high accuracy scores. To solve this problem, we need to find evaluation metrics that give the classes in the target class equal weight, and at the same time, find a way to balance the data. We consider the f1-score with a Macro average to be an important evaluation measure (Eq. (4)). It calculates the f-score for each target class and outputs their unweighted mean, allowing each class to have the same weight as the other classes regardless of the number of instances.

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

We then apply the manual undersampling technique to balance out the data by arbitrarily reducing the number of observations from the majority class. The resulting balanced dataset has an equal number of positive and negative classes.

## 4 Result and Discussion

This study used Python programming language to set up the models. The entire study was done using the k-train library on Google Colab. Several experiments were carried out to evaluate the effectiveness of the DistilBERT model and other pre-canned text classifiers on a binary text classification task. The performance of the classifiers implemented for the sentiment classification of the Covid-19 online news dataset is presented below.

### *4.1 Result Using the Original Imbalanced Dataset*

An initial experiment was performed on the original dataset with class imbalanced using the pre-canned ktrain classifiers. The results of the selected classifiers in terms of accuracy, precision, recall, and f1-score are shown in Tab. 2. The practical trade-off between the complexity of the models and their performance becomes noticeable from the results. For example, the f1-score gap between logreg and nbsvm classifiers is only 1% but reaches 5% between logreg and bigru models, 10% between logreg and distilbert models, and 12% between logreg and bert models. The result indicates that the machine learning classifiers show poor performance compared to the deep learning and the Transformer-based

models. The primary reason is that the machine learning models fail to learn long-term relationships within the long text documents and could not capture the context across the text due to memory issues.

**Table 2:** Performance of classifiers on the original imbalanced dataset

| Method | Classifier | Accuracy | Class | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Machine learning | logreg | 0.86 | 0 | 0.88 | 0.74 | 0.80 |
| | | | 1 | 0.86 | 0.94 | 0.90 |
| | | | Macro avg | 0.87 | 0.84 | 0.84 |
| | nbsvm | 0.86 | 0 | 0.89 | 0.75 | 0.80 |
| | | | 1 | 0.85 | 0.94 | 0.90 |
| | | | Macro avg | 0.87 | 0.85 | 0.85 |
| Deep learning | bigru | 0.90 | 0 | 0.89 | 0.83 | 0.86 |
| | | | 1 | 0.90 | 0.94 | 0.92 |
| | | | Macro avg | 0.90 | 0.88 | 0.89 |
| Transformer | distilbert | 0.94 | 0 | 0.93 | 0.95 | 0.94 |
| | | | 1 | 0.97 | 0.96 | 0.95 |
| | | | Macro avg | 0.94 | 0.94 | 0.93 |
| | bert | 0.96 | 0 | 0.95 | 0.94 | 0.94 |
| | | | 1 | 0.97 | 0.95 | 0.96 |
| | | | Macro avg | 0.96 | 0.95 | 0.95 |

### *4.2 Result Using the Balanced Dataset*

The next set of experiments focuses on evaluating the pre-canned ktrain classifiers on the balanced dataset, and the result is presented in Tab. 3. We observe that all the classifiers follow the same performance pattern on the datasets. Nonetheless, there was a marginal increase in performance for all classifiers on the balanced dataset. For example, the distilbert classifier appreciated slightly from 0.94 accuracy, 0.94 precision and 0.93 f1-score values on the imbalanced dataset, to 0.95 accuracy, 0.96 precision, and 0.95 f1-score values on the balanced dataset. Similarly, the deep learning bigru classifier increase from 0.90 accuracy, 0.90 precision, and 0.89 f1-score, to 0.91 accuracy, 0.91 precision, and 0.91 f1-score values. The performance difference is due to the class balance, which reduces the probability of models over-fitting on the majority class. The result obtained from this study further strengthens the importance of a balanced dataset to increase the overall performance of classification models.

**Table 3:** Performance of classifiers on the balanced dataset

| Method | Classifier | Accuracy | Class | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Machine learning | logreg | 0.87 | 0 | 0.87 | 0.88 | 0.85 |
| | | | 1 | 0.86 | 0.87 | 0.87 |
| | | | Macro avg | 0.86 | 0.87 | 0.86 |
| | nbsvm | 0.86 | 0 | 0.86 | 0.87 | 0.85 |
| | | | 1 | 0.85 | 0.86 | 0.86 |
| | | | Macro avg | 0.86 | 0.86 | 0.86 |
| Deep learning | bigru | 0.91 | 0 | 0.90 | 0.91 | 0.91 |
| | | | 1 | 0.91 | 0.90 | 0.91 |
| | | | Macro avg | 0.91 | 0.90 | 0.91 |
| Transformer | distilbert | 0.95 | 0 | 0.96 | 0.95 | 0.94 |
| | | | 1 | 0.95 | 0.96 | 0.96 |
| | | | Macro avg | 0.96 | 0.95 | 0.95 |
| | bert | 0.97 | 0 | 0.96 | 0.97 | 0.97 |
| | | | 1 | 0.98 | 0.98 | 0.96 |
| | | | Macro avg | 0.97 | 0.97 | 0.96 |

Previous studies have indicated that BERT-based models have a deeper bi-directional context-awareness in text sequences [29,30]. The results from this study corroborate the strength of transformer-based models over other classifiers (Tab. 4). In line with literature, we conclude that the transformer-based distilbert model is more robust than conventional machine learning and deep learning models on a binary text classification task. In addition, the deep learning bigru classifier, which was pre-trained on fasttext word vectors, outperforms the baseline machine learning (logreg, nbsvm) classifiers on the validation set. These findings are consistent with previous studies demonstrating that deep learning and transformer-based models typically outperform conventional machine learning models [6,30,31]. However, the improved classification performance comes at a high cost in computing time, particularly during training, as the many hyperparameters of the transformer-based models need to be fine-tuned to optimize performance

**Table 4:** A comparison of performance on the balanced and imbalanced datasets

| Classifier | Imbalanced dataset | | Balanced dataset | |
|---|---|---|---|---|
| | Accuracy | Fl-score | Accuracy | F1-score |
| logreg | 0.86 | 0.84 | 0.87 | 0.86 |
| nbsvm | 0.86 | 0.85 | 0.86 | 0.86 |

(Continued)

**Table 4:** Continued

| Classifier | Imbalanced dataset | | Balanced dataset | |
|---|---|---|---|---|
| | Accuracy | Fl-score | Accuracy | F1-score |
| bigru | 0.90 | 0.89 | 0.91 | 0.91 |
| distilbert | 0.94 | 0.93 | 0.95 | 0.95 |
| bert | 0.96 | 0.95 | 0.97 | 0.96 |

## 5 Conclusion

The study fine-tuned five different pre-canned text classifiers from the ktrain library on an online news binary classification dataset. The experiments confirmed the superiority of the transformer-based (BERT, and DistilBERT) models over other Machine Learning and Deep Learning models on a downstream NLP task. We show that a class imbalance in the dataset can affect model performance and point out that under-sampling can be a useful technique in dealing with imbalanced data. Further, we noticed that the implementation of text classifiers in the ktrain package is far less complicated and easier to apply. Alongside the class balance, we assume that the quality of the annotation could also impact model performance. For future work, we would like to investigate how the polarity levels of news in our dataset influence the performance of the various pre-canned classifiers.

**Acknowledgement:** This study was supported by the National Key R\&D Program of China, Grant No. 2018YFA0306703.

**Data Availability:** The dataset is available at Mendeley Repository: https://data.mendeley.com/datasets/r6nn5s37tp/2.

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

**Conflicts of Interest:** The authors declare that the research was conducted without any commercial or financial relationships that could be interpreted as a potential conflict of interest.

## References

[1] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," arXiv Preprint, arXiv1301.3781, 2013.

[2] H. P. Wu, Y. Liu and J. Wang, "Review of text classification methods on deep learning," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1309–1321, 2020.

[3] P. Cen, K. X. Zhang and D. Zheng, "Sentiment analysis using deep learning approach," *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 17–27, 2020.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Advances in Neural Information Processing Systems*, Lake Tahoe in Spain. pp. 3111–3119, 2013.

[5] S. Shah, H. Ge, S. A. Haider, M. Irshad and T. Younas, "A quantum spatial graph convolutional network for text classification," *Computer Systems Science and Engineering*, vol. 36, no. 2, pp. 369–382, 2021.

[6] S. K. Akpatsa, X. Li and H. Lei, "A survey and future perspectives of hybrid deep learning models for text classification," in *Proc. ICAIS*, Dublin, Ireland, pp. 358–369, 2021.

[7] J. Jumadinova, O. Bonham-Carter, H. Zheng, M. Camara and D. Shi, "A novel framework for biomedical text mining," *Journal on Big Data*, vol. 2, no. 4, pp. 145–155, 2020.

[8] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv Preprint, arXiv1810.04805, 2018.
[9] T. Wolf, L. Debut, V. Sanh, J. Chaumond and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proc. the 2020 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, pp. 38–45, 2020.
[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, California, USA, pp. 5998–6008, 2017.
[11] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova *et al.,* "Distilling task-specific knowledge from bert into simple neural networks," arXiv Preprint, arXiv1903.12136, 2019.
[12] M. Wasserblat, O. Pereg and P. Izsak, "Exploring the boundaries of low-resource bert distillation," in *Proc. SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, Online, pp. 35–40, 2020.
[13] E. Strubell, A. Ganesh and A. McCallum, "Energy and policy considerations for deep learning in NLP," arXiv Preprint, arXiv1906.02243, 2019.
[14] A. S. Maiya, "Ktrain: A low-code library for augmented machine learning," arXiv Preprint, arXiv2004.10703, 2020.
[15] V. Sanh, L. Debut, J. Chaumond and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv Preprint, arXiv1910.01108, 2019.
[16] B. Büyüköz, A. Hürriyetoğlu and A. Özgür, "Analyzing ELMo and DistilBERT on socio-political news classification," in *Proc. the Workshop on Automated Extraction of Socio-Political Events from News 2020*, Marseille, France, pp. 9–18, 2020.
[17] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," arXiv Preprint, arXiv1803.09820, 2018.
[18] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," arXiv Preprint, arXiv1608.03983, 2016.
[19] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, pp. 90–94, 2012.
[20] I. L. Laily, I. Budi, A. B. Santoso and P. K. Putra, "Mining Indonesia tourism's reviews to evaluate the services through multilabel classification and LDA," in *Proc. 2020 Int. Conf. on Electrical Engineering and Informatics*, Yogyakarta, Indonesia, pp. 1–7, 2020.
[21] G. Mutanov, V. Karyukin and Z. Mamykova, "Multi-class sentiment analysis of social media data with machine learning algorithms," *Computers, Materials & Continua*, vol. 69, no. 1, pp. 913–930, 2021.
[22] L. Zhang, Y. Zhou, X. Duan and R. Chen, "A hierarchical multi-input and output bi-GRU model for sentiment analysis on customer reviews," in *Proc. IOP Conf. Series: Materials Science and Engineering*, Xi'an in China, vol. 322, no. 6, 2018.
[23] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, "Bag of tricks for efficient text classification," arXiv Preprint, arXiv1607.01759, 2016.
[24] O. Kovaleva, A. Romanov, A. Rogers and A. Rumshisky, "Revealing the dark secrets of BERT," arXiv Preprint, arXiv1908.08593, 2019.
[25] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *Fiber*, vol. 56, no. 4, pp. 3–7, 2015.
[26] G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," *Computer Science*, vol. 14, no. 7, pp. 38–39, 2015.
[27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," arXiv Preprint, arXiv1909.11942, 2019.
[28] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen *et al.,* "Tinybert: Distilling bert for natural language understanding," arXiv Preprint, arXiv1909.10351, 2019.
[29] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu *et al.,* "Pre-trained models: Past, present and future," *AI Open*, vol. 147, no. 1, pp. 1834–1841, 2021.

[30] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification," arXiv Preprint, arXiv2005.13012, 2020.
[31] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu *et al.,* "Deep learning–Based text classification: A comprehensive review," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–40, 2021.