Citation: V. Prema, V. Elavazhahan. Empowering sentiment analysis in resource-constrained environments: Leveraging lightweight pre-trained models for optimal performance. *Journal of Harbin Institute of Technology (New Series)*, 2025, 32(1): 76-84. DOI: 10.11916/j.issn.1005-9113.2023103

Empowering Sentiment Analysis in Resource-Constrained Environments: Leveraging Lightweight Pre-trained Models for Optimal Performance

V. Prema[1*] *and V. Elavazhahan*[2]

(*1. Department of Computer and Information Science, Annamalai University, Chidambaram 608002, India;*
2. Department of Computer Science, Government Arts and Science College, Vadalur 607303, India)

Abstract: Sentiment analysis, a cornerstone of natural language processing, has witnessed remarkable advancements driven by deep learning models which demonstrated impressive accuracy in discerning sentiment from text across various domains. However, the deployment of such models in resource-constrained environments presents a unique set of challenges that require innovative solutions. Resource-constrained environments encompass scenarios where computing resources, memory, and energy availability are restricted. To empower sentiment analysis in resource-constrained environments, we address the crucial need by leveraging lightweight pre-trained models. These models, derived from popular architectures such as DistilBERT, MobileBERT, ALBERT, TinyBERT, ELECTRA, and SqueezeBERT, offer a promising solution to the resource limitations imposed by these environments. By distilling the knowledge from larger models into smaller ones and employing various optimization techniques, these lightweight models aim to strike a balance between performance and resource efficiency. This paper endeavors to explore the performance of multiple lightweight pre-trained models in sentiment analysis tasks specific to such environments and provide insights into their viability for practical deployment.

Keywords: sentiment analysis; light weight models; resource-constrained environment; pre-trained models

CLC number: TP183 **Document code**: A **Article ID**: 1005-9113(2025)01-0076-09

0 Introduction

Sentiment analysis, the art of deciphering emotions and opinions from textual content, has experienced remarkable progress owing to the appearance of deep learning and pre-trained language models. This advancement has significantly improved sentiment classification across diverse domains, from social media sentiment tracking to market trend forecasting. Yet, as the influence of sentiment analysis expands to resource-constrained environments, a distinct challenge emerges: How to achieve accurate sentiment analysis as well as accommodating limitations in computing power, memory, and energy availability.

Resource-constrained environments, typified by mobile devices, edge computing nodes, and Internet of Things devices, have become pervasive in modern computing landscapes. These contexts, however, demand that sentiment analysis methods should differ from traditional paradigms. Conventional large-scale pre-trained models, while proficient in sentiment classification, often demand excessive computational resources and memory, rendering them impractical for deployment in such environments. The pre-trained models face issues resulted from their large model size and significant latency, preventing us from directly implementing them on mobile devices with limited resources[1]. Managing these extensive models at the edge and/or within restricted computational training or inference limitations continues to be a challenge[2]. Lower hardware costs makes it possible to deploy deep learning models in resource-constrained environments or in applications where cost is a significant factor[3].

This paper addresses the compelling need to meet this gap by exploring the application of lightweight pre-trained models in resource-constrained sentiment

Received 2023-09-13. Published online 2024-03-06.

* Corresponding author. V. Prema, Research Scholar. Email: premaveluchamy78@gmail.com.

analysis. Models such as DistilBERT (smaller and more efficient version of the BERT model, which are bidirectional encoder representations from transformers, developed by Google), MobileBERT (Lightweight version of BERT), ALBERT (a lite BERT, another variant of the BERT), TinyBERT (light weight, compressed and smaller version of BERT), ELECTRA (efficiently learning an encoder that classifies token replacements accurately), and SqueezeBERT (variant of BERT) have arisen as potential solutions, offering a synthesis of performance and efficiency. These models utilize techniques such as knowledge distillation, parameter reduction, and optimization algorithms render them both compact and capable.

1 Related Work

1.1 Pre-trained Lightweight Models

Sun et al. [1] proposed MobileBERT, which is a task-agnostic compact variant of BERT. DistilBERT, introduced by Sanh et al.[2], is a condensed version of the BERT model. It offers a more compact, efficient, and cost-effective alternative while retaining 97% of its language comprehension capabilities. It is trained by using a triple loss function that combines language modeling, distillation, and cosine-distance losses. Ren et al. [3] proposed building a lightweight and efficient model based on Convolutional Neural Networks (CNNs) and argued that CNN-based models have the potential to strike a suitable balance between reducing computational costs without degrading performance. ALBERT integrates two methods for reducing parameters, which effectively addressed the main challenges in scaling pre-trained models[4]. TinyBERT delivers competitive performance by substantially decreasing the model size and inference time, offering an efficient solution for deploying BERT-based NLP models on edge devices[5]. ELECTRA works effectively even with relatively small amounts of computational resources, enabling the deployment of deep learning models in resource-constrained environments or applications where cost is a significant factor[6]. Nguyen et al.[7] released a new model BERTweet. Jin et al.[8] used pre-trained BERT in sentiment analysis to check robustness of BERT. Hu et al.[9] incorporated knowledge into prompt verbalizer for text classification. Hierarchy-aware prompt tuning achieves state-of-the-art performances on 3 popular hierarchy-aware prompt tuning datasets and is adept at handling the imbalance and low resource situation[10].

1.2 Variants of Pre-Trained Lightweight Models

Yao et al.[11] introduced a compact text classification model with certified robustness. It features a reduced parameter count suitable for utilization on mobile devices with limited computational resources. To address the issue of reducing the number of parameters in models, researchers like Lopez et al.[12] have explored Tensorized Neural Networks (TNNs). The goal of TNNs is to replicate the functionality of conventional networks while using fewer parameters, which can thereby reduce computational costs associated with training operations. Mewada et al.[13] suggested the development of a novel approach called Synthetic Attention in Bidirectional Encoder Representations from Transformers (SA-BERT) coupled with an Extreme Gradient Boosting (XGBoost) classifier for the purpose of sentiment polarity classification in a review dataset. Tanvir et al.[14] harnessed the potential of Generative Adversarial Network-Bidirectional Encoder Representations from Transformers (GAN-BERT), a modified variant of BERT tailored for low-resource settings in Bangla Natural Language Processing (BNLP). Ding et al.[15] introduced a new term 'delta-tuning', where 'delta' a mathematical notation often used to denote changes, is borrowed to refer to the portion of parameters that are 'changed' during training. Kim et al.[16] proposed I-BERT, a novel quantization scheme for transformer-based models. I-BERT (Integer BERT) quantizes the entire inference with integer-only arithmetic. It relies on lightweight integer-only approximation methods for nonlinear operations, such as Gaussian Error Linear Unit activation function (GELU), Softmax, and Layer Normalization. I-BERT performs an end-to-end integer-only BERT inference without any floating-point calculations.

2 Light Weight Models for Resource Constrained Environment

2.1 Importance of Lightweight Models for Efficient Sentiment Analysis

Sentiment analysis for resource-constrained environments is underscored by the pervasive demand for sentiment classification in contexts where computing resources, memory, and energy are

constrained. In a landscape increasingly defined by mobile devices, edge computing platforms and Internet of Things (IoT) devices, the limitations imposed by limited memory and computing speed have become paramount. Traditional sentiment analysis models, while proficient in sentiment classification, are often unsuited for such environments due to their memory-intensive nature and high computational demands.

This study not only addresses the gap but also extends its significance to individual researchers and small companies equipped with devices featuring a minimum of 8 GB RAM. In these scenarios where access to extensive computational resources may be restricted. Therefore, the adaptation of lightweight pre-trained models such as DistilBERT, MobileBERT, ALBERT, TinyBERT, Electra, and SqueezeBERT can be a transformative solution. By enabling accurate sentiment prediction within the constraints of such devices, this research serves as a crucial enabler for sentiment analysis applications even in resource-limited settings.

As above, individual researchers and small businesses can leverage these efficient models to integrate sentiment analysis into their projects without being hindered by memory and computing speed limitations, thus promoting a broader accessibility to sentiment analysis capabilities. As a consequence, those endeavors not only enhance the efficiency of sentiment analysis tasks but also democratizes access to this technology, empowering a wider range of stakeholders to benefit from sentiment analysis in their endeavors. Table 1 shows the contributors of light weight models whose contribution can be utilized for the aforementioned purposes.

Table 1 Contributors of the light weight models

Model	Researcher	Year
MobileBERT	Sun et al.[1]	2019
DistilBERT	Sanh et al.[2]	2019
ALBERT	Lan et al.[4]	2019
Tiny BERT	Jiao et al.[5]	2019
ELECTRA	Clark et al.[6]	2020
IBERT	Kim et al.[16]	2021
T5 small	Raffel et al.[17]	2020
SqueezeBERT	Iandola et al.[18]	2020

2.2 Strategies for Sentiment Analysis in Resource-Constrained Environments

Conducting sentiment analysis in resource-constrained environments necessitates the employment of specialized strategies to ensure accurate predictions while optimizing memory and computational resources.

2.2.1 Model selection

In order to strike a balance between model size and performance, the selection of a compact sentiment analysis model is crucial. Models like BERT, DistilBERT, ALBERT, TinyBERT, and MobileBERT offer various levels of optimization tailored to different resource constraints.

2.2.2 Model quantization

To efficiently utilize memory, model quantization techniques can be applied. By reducing the precision of model weights and activations, significant memory savings can be achieved without substantial loss in accuracy. Many deep learning frameworks provide quantization tools that can facilitate this process.

2.2.3 Model pruning

Model pruning, involving the removal of unnecessary connections or neurons, is an effective means to reduce model size while maintaining performance. This technique is particularly advantageous in resource-constrained scenarios where memory efficiency is paramount.

2.2.4 Knowledge distillation

Employing knowledge distillation involves training a smaller "student" model to replicate the behavior of a larger and more accurate "teacher" model. This approach leverages the knowledge captured by the teacher model to achieve competitive performance with reduced computational demands.

2.2.5 Feature extraction

Consider utilizing feature extraction techniques, which involve extracting relevant features from input text before employing a simpler classifier for sentiment classification. This approach reduces the complexity of the model without compromising on accuracy.

2.2.6 Optimized frameworks

Opt for lightweight deep learning frameworks or libraries optimized for sentiment analysis tasks in resource-constrained environments. These frameworks offer efficient implementations and reduced computational overhead, which makes them well-suited for deployment on devices with limited resources.

By incorporating these strategies, sentiment analysis can be effectively conducted even in settings with restricted resources. The judicious combination of model selection, quantization, pruning, knowledge

distillation, feature extraction, and optimized frameworks empowers sentiment analysis to be both accurate and efficient, thereby expanding its applicability across a spectrum of resource-constrained environments.

3 Experimental Setup

3.1 Dataset

3.1.1 Dataset description

Dataset used for fine tuning and predicting is customized dataset, which is a collection of reviews sourced from social media platforms, especifically from Twitter and YouTube via using Application Programming Interfaces (APIs). The dataset comprises two main columns: "Review" and "Labels". These reviews are centered around two distinct categories of content, which are restaurant reviews (Fine tuning) and webinar reviews (Predicting Sentiments), representing different contexts in which sentiment analysis can be applied.

The "Review" column consists of textual content extracted from Twitter and YouTube. Each entry in this column corresponds to a user-generated review, expressing opinions, thoughts, and sentiments about a specific restaurant or webinar. These reviews are essentially unstructured text, reflecting the users' experiences and impressions of the respective subjects. The "Labels" column serves as the ground truth for sentiment analysis. It contains binary labels indicating the sentiment polarity of each review which is labelled manually. The sentiment labels are categorized into "Positive" and "Negative". This binary classification reflects the sentiment tone conveyed by the review. The labeling is balanced, which means that there is an equal representation of both positive and negative sentiment labels.

The dataset encompasses a range of sentiments expressed by users, covering both positive and negative viewpoints. The inclusion of restaurant and webinar reviews offers diversity in the types of content and sentiment expressions. By collecting data from both Twitter and YouTube, the dataset captures sentiments from different user bases and communication styles inherent to these platforms. Table 2 shows the detailed description of the dataset used for the study.

3.1.2 Pre-processing

To ensure data quality and analysis consistency, the dataset has undergone a pre-processing. Pre-processing involves text cleaning and normalization to remove irrelevant characters, symbols, or inconsistencies that could potentially interfere with sentiment analysis algorithms. The pre-processed text is prepared for the oncoming analysis, such as tokenization and feature extraction. For tokenization, the preprocessed text is segmented into individual tokens or words, breaking down the input into meaningful units. This step is crucial for further analysis as it allows the sentiment analysis model to understand the context of the text.

Table 2 Dataset description

Purpose	Nature of dataset	Positive labels	Negative labels
Fine tuning	Restaurant reviews (custom)	2850	2196
Prediction of sentiments	Webinar reviews (custom)	567	512

3.2 Environment and Execution

3.2.1 Platform selection and instance configuration

The experiments were conducted on virtual machines (VMs) hosted on Azure and Google Colab platforms, utilizing CPU instances with 28 GB of RAM. The programming environment employed for the experiments was Python.

Azure VM: The Azure platform provided a virtual machine with CPU resources, well-suited for running resource-efficient tasks. The 28 GB of RAM is facilitated to handle larger datasets and models.

Google Colab: Leveraging the free version of Google Colab, CPU instances with 28 GB of RAM were utilized to execute the experiments. This platform provided cloud-based computing resources without a need for local hardware.

3.2.2 GitHub repository and code availability

The complete set of experiment implementations, including data pre-processing, model fine-tuning, prediction, and performance evaluation, are available within a dedicated GitHub repository. This repository serves as a comprehensive resource for accessing the codebase and reproducing the experiments conducted on both Azure VM and Google Colab platforms[19].

3.2.3 Resource limitations and adaptation

Given that the experiments were carried out on free-tier instances, considerations were made to accommodate resource constraints. This involved careful dataset selection, model choice, and hyper

parameter tuning to ensure that the experiments were feasible within the provided computing resources.

4 Methodology

4.1 Selection of Model

Upon a meticulous review of transformer-based models for text classification, a deliberate selection process led to the inclusion of DistilBERT, MobileBERT, SqueezeBERT, iBERT, and ELECTRA as the chosen lightweight transformer architectures for sentiment analysis. The rationale behind this selection rested upon their inherently optimized attributes catering to resource-constrained contexts. By incorporating pre-trained versions of these models, we harnessed their ingrained linguistic knowledge during subsequent fine-tuning processes. Adjustments were made to the final layers of each model architecture, ensuring their alignment with the binary sentiment classification task. Each model underwent meticulous fine-tuning leveraging binary cross-entropy loss functions and appropriately tuned hyper parameters. Consistent with the pre-processing pipeline used for restaurant reviews, identical pre-processing steps were applied to the webinar reviews. In pursuit of sentiment prediction, the fine-tuned models were deployed, yielding predicted sentiment labels for the webinar reviews. Table 3 shows the methodology used in the light weight models utilized for this study.

Rigorous assessment ensued, encompassing conventional sentiment analysis metrics, encompassing accuracy, precision, recall, and F1-Score, applied to the forecasted sentiment labels of the webinar content. Through a comprehensive performance evaluation, we discerned the nuanced variances in the models' predictive prowess, illuminating which among these lightweight models optimally adapts to the distinctive challenges posed by resource-constrained scenarios. In Table 4, light weight models are analyzed in various aspects for resource specific environments.

Table 3 Methodology of the light weight models

Ref.	Model	Paper title	Methodology
[1]	MobileBERT	MobileBERT: a compact task-agnostic BERT compression framework	Task-agnostic BERT compression using advanced techniques
[2]	DistilBERT	DistilBERT, a distilled version of BERT	Knowledge distillation to create a smaller version of BERT.
[4]	ALBERT	ALBERT: a lite BERT for self-supervised learning of language representations	Factorized embedding parameterization for more efficiency
[5]	TinyBERT	TinyBERT: distilling BERT for natural language understanding	Knowledge distillation of transformer based models
[6]	ELECTRA	ELECTRA: pre-training text encoders as discriminators rather than generators	Introduction of a new generator-discriminator pre-training
[16]	IBERT	i-BERT: integer-only BERT quantization	Integer-only quantization of BERT to minimize resource use
[17]	T5 small	Exploring the limits of transfer learning with a unified text-to-text transformer	Formulation of tasks as text-to-text problems for Learning
[18]	SqueezeBERT	SqueezeBERT: what can computer vision teach NLP about efficient neural networks?	Neural architecture design with efficiency insight

4.2 Performance Metrics

The sentiment analysis experiment was rigorously assessed using key performance metrics. Accuracy quantified the models' precision in predicting sentiments, while precision gauged the models' ability to accurately predict positive sentiments among positive predictions. Recall illuminated the models' aptitude in capturing true positive predictions among actual positive instances, and F1-Score provided a balanced assessment by harmonizing precision and recall. These metrics collectively offered a nuanced technical evaluation of the models' sentiment prediction capabilities, shedding light on their accuracy, precision, recall, and balance between minimizing false positives and false negatives in sentiment classification for webinar reviews.

5 Fine-Tuning Pre-Trained Models

Fine-tuning pre-trained models is a critical technique in transfer learning, where a neural network model trained on one task is adapted to perform another task. This process capitalizes on the knowledge encoded within the pre-trained model and tunes it to excel in the target task, often with a smaller dataset. When employing fine-tuning for sentiment analysis, the process involves several key steps and considerations.

Table 4 Analysis of pre-trained light weight models for resource specific environment

Model	Model size (MB)	Training time	Task-specific performance	Pre-processing complexity	Resource availability	Compatibility	Customization & fine-tuning	Robustness	Interpretability	Sustainability
DistilBERT	Varies	Moderate	Varies	Moderate	Widely available	High	Moderate	Moderate	Moderate	Strong
MobileBERT	Varies	Moderate	Varies	Moderate	Moderate	High	Moderate	Moderate	Moderate	Moderate
TinyBERT (compressed)	Varies	Varies	Varies	Moderate	Moderate	High	High	Varies	Moderate	Varies
ALBERT (small)	Varies	Moderate	Varies	Low	Moderate	High	High	Moderate	Moderate	Strong
Electra (small)	Varies	High	Varies	Low	Moderate	High	High	Moderate	Low	Moderate
VADER sentiment	Minimal	N/A	Varies	Minimal	Widely available	High	Minimal	Low	High	Strong

Notes: "Varies" indicates that the value can vary widely depending on the specific model variant, dataset, and other factors. "N/A" indicates that the factor is not applicable. "Moderate", "High", "Low", "Widely Available", and "Strong" are subjective qualitative indicators.

5.1 Transfer Learning Principle

Fine-tuning is grounded in the principle of transfer learning. A pre-trained model, usually a deep neural network like BERT or DistilBERT, has already learned rich features from a vast dataset on a related task. This pre-trained model serves as a feature extractor, capturing general linguistic patterns and semantics.

5.2 Task-Specific Adaptation

During fine-tuning, the model's architecture is preserved, but its final layers are adjusted to align with the new sentiment analysis task. These final layers are typically responsible for making predictions based on learned features. In our experiment, the task is to predict the sentiment labels for the reviews in the given dataset.

5.3 Loss Function and Optimization

The fine-tuning process involves updating the model's weights based on a task-specific dataset. The loss function used in training guides the adjustment of these weights. For sentiment analysis, the binary cross-entropy loss function is commonly employed:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{n}[y_i\log(p(y_i)) + (1-y_i)\log(1-p(y_i))]$$

where θ represents the loss function, often denoted by L, which is a function of the model parameters θ. In the context of binary cross-entropy, it measures the discrepancy between the predicted probabilities and the actual target labels. N: The total number of samples or instances in the dataset. y_i: The true label (ground truth) for the i-th sample. It is a binary value that can be either 0 or 1. $p(y_i)$: The predicted probability that the i-th sample belongs to the positive class (1) according to the model. In binary classification, this value typically ranges between 0 and 1.

5.4 Regularization

To prevent overfitting, regularization techniques like dropout and weight decay are applied. These techniques control the complexity of the model and enhance its ability to generalize to new data.

5.5 Learning Rate and Hyper Parameters

The learning rate dictates the step size in updating model weights. Careful selection of the learning rate and other hyper parameters is crucial to achieve stable convergence during fine-tuning.

5.6 Transfer and Fine-Tuning Phases

The process often involves two main phases: a transfer phase where the model is pre-trained on a large general text corpus, and a fine-tuning phase where the model's last layers are updated using a

smaller task-specific dataset.

6 Comparison of Fine Tuned Models

6.1 Model Size

The comparison of the five lightweight models based on their size, including both model parameters and tokenizer size, reveals notable differences among them.

DistilBERT: With 66955010 parameters, DistilBERT demonstrates a model size of 255.413 megabytes. The tokenizer associated with DistilBERT processes 30522 tokens, contributing to a tokenizer size of 3553.74 megabytes.

Albert: Albert possesses 11685122 parameters, resulting in a model size of 44.58 megabytes. Its tokenizer handles 30000 tokens, resulting in a tokenizer size of 3492.97 megabytes.

MobileBERT: Mobile Bert's model size is 24581888 parameters, occupying 93.77 megabytes. Its tokenizer encompasses 30522 tokens, leading to a tokenizer size of 3553.74 megabytes.

i-BERT: The i-BERT model encompasses 51094272 parameters, resulting in a substantial model size of 475.48 megabytes. Its tokenizer manages 50265 tokens, accounting for a tokenizer size of 9638.1 megabytes.

SqueezeBERT: Similar to i-BERT, SqueezeBERT also encompasses 51094272 parameters, contributing to a model size of 194.91 megabytes. Its tokenizer handles 30528 tokens, resulting in a tokenizer size of 3555.14 megabytes.

The lightweight models exhibit variations in both model size and tokenizer size.

While DistilBERT, Albert, and MobileBERT maintain relatively smaller model sizes, i-BERT and SqueezeBERT have larger model sizes. Notably, tokenizer sizes closely align with model sizes for each respective model. These size considerations are essential when deploying models in resource-constrained environments, as they directly impact memory usage and storage requirements. Table 5 clearly indicates the model and tokenizer sizes of models.

6.2 Performance Metrics

In the evaluation of the fine-tuned models using custom dataset, distinct performance metrics offer an insightful comparison. DistilBERT showcases strong performance with an accuracy of 88%, while maintaining a balanced precision and recall at 86% and 92%, respectively, leading to an F1-Score of 89%. Albert exhibits notable precision at 89%, but its overall accuracy of 74% is accompanied by a lower recall of 57%, yielding an F1-Score of 70%. Notably, MobileBERT demonstrates exceptional recall at 99%, although its overall accuracy and precision are comparatively lower at 52%, resulting in an F1-Score of 69%. SqueezeBERT attains a relatively balanced combination of precision (62%) and recall (56%), resulting in an F1-Score of 59%. IBERT, while achieving a respectable precision of 74%, faces challenges in recall at 53%, yielding an F1-Score of 62%. In summary, DistilBERT emerges as a standout performer in terms of balanced accuracy, precision, recall, and F1-Score, offering promising sentiment analysis capabilities for the custom dataset. The details are given in Table 6.

Table 5 Comparison of light weight models

Model name	Model size		Tokenizer size	
	Parameters	Megabytes	Tokens	Megabytes
Distilbert	66955010	255.413	30522	3553.74
Albert	11685122	44.580	30000	3492.97
MobileBERT	24581888	93.770	30522	3553.74
i-BERT	51094272	475.480	50265	9638.10
SqueezeBERT	51094272	194.910	30528	3555.14

Table 6 Performance metrics of fine-tuned models for custom dataset

Fine-tuned model name	Base model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score(%)
ResEff1	Distilbert	88	86	92	89
ResEff2	Albert	74	89	57	70
ResEff3	MobileBERT	52	52	99	69
ResEff4	SqueezeBERT	61	62	56	59
ResEff5	IBERT	67	74	53	62

7 Results and Discussions

7.1 Interpretation of Results and Performance Trends

Based on the results obtained from the comprehensive evaluation of the lightweight models, a discernible pattern emerges, shedding light on their efficacy and characteristics in the context of sentiment analysis. Notably, DistilBERT emerges as a prominent contender, exhibiting an accuracy of 88% alongside a harmonious balance between precision (86%) and recall (92%), resulting in a commendable F1-Score of 89%. This underscores its aptitude in accurately identifying sentiments and its capacity to effectively capture both positive and negative instances.

Conversely, Albert, while displaying a noteworthy precision of 89%, is accompanied by an accuracy of 74% and a comparatively lower recall of 57%, contributing to an F1-Score of 70%. While precision remains high, its performance in correctly classifying positive instances is somewhat hindered. On another tangent, MobileBERT distinguishes itself through an exceptional recall of 99%, yet its overall accuracy (52%) and precision (52%) present certain limitations, culminating in an F1-Score of 69%. The challenge here lies in achieving a more balanced distribution of predictive prowess across both classes.

SqueezeBERT, with precision and recall at 62% and 56% respectively, strikes a balance that yields an F1 score of 59%. Its performance indicates the capacity to provide a reasonable trade-off between identifying positive instances and maintaining precision in predictions. Meanwhile, i-BERT demonstrates a commendable precision of 74%, but grapples with a recall of 53%, ultimately resulting in an F1-Score of 62%. This points to the need for enhancement in correctly identifying actual positive instances.

In summation, the findings collectively illuminate DistilBERT as a standout candidate, offering a well-rounded blend of accuracy, precision, recall, and F1-Score. Albert showcases remarkable precision, MobileBERT excels in capturing positive instances, SqueezeBERT strikes a balance, and i-BERT seeks refinement in capturing all positive instances. These insights serve as a valuable guide for selecting an appropriate model based on specific performance requirements and resource constraints, effectively advancing sentiment analysis endeavors in varied contexts.

7.2 Conclusions

The results provided make sense and offer valuable insights into the performance and characteristics of the different lightweight models as well as their associated sizes. The comparison of performance metrics, such as accuracy, precision, recall, and F1-Score, gives a clear understanding of how each model performs in sentiment analysis tasks. Likewise, the comparison of model sizes and tokenizer sizes provides information about the resource requirements and potential efficiency of deploying these models in various environments.

The results reflect the trade-offs between model size and performance. For instance, models like DistilBERT and MobileBERT achieve relatively good accuracy with smaller model sizes, making them suitable choices for applications with limited resources. On the other hand, models like i-BERT may have larger model sizes but offer competitive performance. These findings align with the concept of balancing model complexity with performance expectations and resource constraints

8 Future Direction and Enhancements

Looking ahead, the possibilities for sentiment analysis are vast and exciting. We can expand our approach to different areas, understanding sentiments in various types of content. This means making our models work well not just in one field, but in many different ones. We can also make our models smaller while keeping their accuracy, using methods like pruning and quantization. By combining different techniques and using smarter strategies, we can build models that are good at understanding sentiments and do not need a lot of computer power.

9 Acknowledgements

I extend my gratitude to Mr. Prabhakaran, Cloud Architect, for his invaluable contribution in providing the essential cloud space within Azure VM. This pivotal provision enabled the successful training of the models, a cornerstone in our pursuit of advancing sentiment analysis. His support and facilitation have played a crucial role in making our endeavors feasible and productive.

References

[1] Sun Z, Yu H, Song X, et al. Mobilebert: Task-agnostic compression of bert by progressive knowledge transfer. arXiv:2004.02984.2020.DOI: 10.48550/arXiv.2004.02984.

[2] Sanh V, Debut L, Chaumond J, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. 2019 . DOI: 10.48550/arXiv.1910.01108.

[3] Ren F, Feng L, Xiao D, et al. DNet: A lightweight and efficient model for aspect based sentiment analysis. Expert Systems with Applications, 2020, 151: 113393. DOI: 10.1016/j.eswa.2020.113393

[4] Lan Z, Chen M, Goodman S, et al. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942. 2019. DOI: 10.48550/arXiv.1909.11942.

[5] Jiao X, Yin Y, Shang L, et al. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351. 2019. DOI: 10.48550/arXiv.1909.10351.

[6] Clark K, Luong M T, Le Q V, et al. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003. 10555. 2020. DOI: 10.48550/arXiv.2003.10555.

[7] Nguyen D Q, Vu T, Nguyen A T. BERTweet: A pre-trained language model for English Tweets. arXiv preprint arXiv: 2005. 10200. 2020. DOI: 10.48550/arXiv.2005.10200.

[8] Jin D, Jin Z, Zhou J T, et al. Is bert really robust? A strong baseline for natural language attack on text classification and entailment. Proceedings of the AAAI Conference on Artificial Intelligence. Wishington DC: AAAI Press, 2020, 34(5): 8018-8025. DOI: 10.1609/aaai.v34ia5.6311.

[9] Hu S, Ding N, Wang H , et al. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. arXiv preprintarXiv:2108.02035. 2021. DOI: 10.18653/v1/2022.acl-long.158.

[10] Wang Z, Wang P, Liu T, et al. HPT: hierarchy-aware prompt tuning for hierarchical text classification. arXiv preprint arXiv:2204.13413. 2022. DOI: 10.48550/arXiv.2204.13413.

[11] Yao Q, Kumar S T, Brocanelli M , et al. Tiny rnn model with certified robustness for text classification. 2022 International Joint Conference on Neural Networks. Piscataway: IEEE, 2022: 1-8. DOI: 10.1109/IJCNN55064.2022.9892117.

[12] Lopez G, Nguyen A, Kaul J. Reducing computational costs in sentiment analysis: Tensorized recurrent networks vs. recurrent networks. arXiv preprint arXiv:2306.09705. 2023.

[13] Mewada A, Dewang R K. SA-ASBA: A hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting. The Journal of Supercomputing, 2023, 79(5): 5516-5551. DOI: 10.1007/s11227-022-04881-x.

[14] Tanvir R, Shawon M T, Mehedi M H, et al. A GAN-BERT based approach for bengali text classification with a few labeled examples. International Symposium on Distributed Computing and Artificial Intelligence. Berlin: Springer, 2022, 583: 20-30. DOI: 10.1007/978-3-631-20859-1_3.

[15] Ding N, Qin Y, Yang G, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, 2023, 5(3): 220-235. DOI: 10.1038/s42256-023-00626-4.

[16] Kim S, Gholami A, Yao Z, et al. I-bert: Integer-only bert quantization. International Conference on Machine Learning, 2021, 139: 5506-5518. DOI: 10.48550/arXiv.2101.01321.

[17] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551. DOI: 10.48550/arXiv.1910.10683.

[18] Iandola F N, Shaw A E, Krishna R, et al. SqueezeBERT: What can computer vision teach NLP about efficient neural networks? arXiv preprint arXiv: 2006.11316. 2020. DOI: 10.48550/arXiv.2006.11316.

[19] GitHub. Sentiment-Analysis-using-Pretrained-Deep-learning-models. https://github.com/Prema-Veluchamy/Sentiment-Analysis-using-Pretrained-Deep-learning-models.