Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

GPT vs. Open-Source LLMs: A Comprehensive Performance and Capability Assessment

Shaad Abid Qureshi
Department of Information Technology
SVKM's Narsee Monjee College of Commerce and Economics
Mumbai, India

Aditya Ojha
Department of Information Technology
SVKM's Narsee Monjee College of Commerce and Economics
Mumbai, India

Varun Sunil Guwal
Department of Information Technology
SVKM's Narsee Monjee College of Commerce and Economics
Mumbai, India

Taher Kezar Jhalodwala
Department of Information Technology
SVKMâ€™s Narsee Monjee College of Commerce and Economics Mumbai, India

Hardik Malhotra
Department of Information Technology
SVKM's Narsee Monjee College of Commerce and Economics
Mumbai, India

Hardi Thakkar
Department of Information Technology
SVKM's Narsee Monjee College of Commerce and Economics,
Mumbai, India

Reeba Khan
Department of Information Technology
SVKMâ€™s Narsee Monjee College of Commerce and Economics Mumbai, India

Anupama Jawale
Department of Information Technology
SVKMâ€™s Narsee Monjee College of Commerce and Economics Mumbai, India

***Abstract*— The increasing demand for use of large language models (LLMs), primarily for text generation and Question-Answer jobs, has created an urgent need to evaluate their performance suiting varied roles. In any Natural Language Processing (NLP) advancement, selecting the appropriate model is yet a cumbersome job. While there seem to be proprietary LLMs available that cater to this, however, there is a lack of detailed comparisons that could guide the best choice. This study examines the performance of three prominent open-source language models—GPT-2 Small, T5 Small, and DistilBERT—in the text completion task. The goal is to ascertain which of the three alternatives is most appropriate for this task. The Wikitext-2 dataset was employed to enhance the models, ensuring uniform training and testing conditions. Metrics such as accuracy, precision, recall, F1-score, BLEU, ROUGE, and perplexity were utilized to assess performance within a comprehensive evaluation framework. An extensive assessment of the model's efficacy and quality was achieved by analyzing memory usage, processing duration, and output variability. A standardized hardware setup was employed for the studies to ensure equity and repeatability. This study aims to elucidate the trade-offs between the quality of text generation and computational efficiency in the selection of the optimal open-source model for text completion tasks.**

Keywords - Computational Efficiency, Evaluation Metrics, Language Models, Text Completion, Wikitext-2 Dataset

I. INTRODUCTION

One of the main reasons for the quick progress of natural language processing has been the creation of large language models (LLMs), which have revolutionized text production jobs including question answering, translation, and summarization. From rule-based systems and statistical models like n-grams to text generation, neural networks and recurrent models like LSTMs have historically progressed. The introduction of transformers caused a paradigm shift by enabling the creation of more cohesive and contextually aware text. These models include, for instance, BERT and GPT. As LLMs became increasingly complicated, the challenge moved from capacity limitations to selecting the optimal model based on performance, computational efficiency, and application-specific requirements. This study compares three popular LLMs on text completion tasks: DistilBERT, T5 Small, and GPT-2 Small, using the Wikitext-2 dataset. The study illustrates the trade-offs between text quality and computational efficiency by assessing these models on metrics like BLEU, ROUGE, and memory utilization. This provides information on the best LLM option for different NLP applications.

II. LITERATURE REVIEW

A. T5-Small:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

T5- Small is a smaller version of Text-To- Text Transfer Transformer(T5) by Google. T5 is unique in its approach as it deals with strings regardless of the task in hand. In simpler terms both input and output are in the form of text strings [1]. It has a wide range of applications ranging from Text Summarization to Machine Translation, Question Answering, Classification, and Fill in the Blanks style tasks [2]. It is suitable to work in a constrained environment due to its compact parameters just 60M compared to 220M and 770M of T5-base and T5-large respectively. T5 models have been employed in various applications such as code generators, robotics, text summarization tools, translation systems and chatbots [1]. T5 uses common crawl web extracted text. T5 removes any lines that didn't end in a terminal punctuation mark. It also removes lines with the word JavaScript and any pages that had a curly bracket (since it often appears in code) [1]. The dataset is cleaned by removing any duplicated content, using a 3-chunk format with sliding windows. For example, if a paragraph is repeated, then it is removed in this process. This results in about 750 GB of clean data.

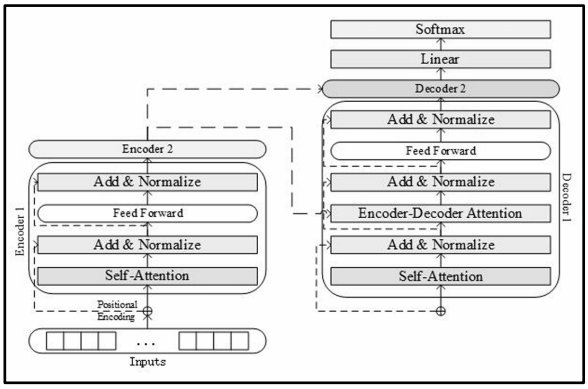

Fig 1. T5-Small Architecture [1]

Figure 1 shows the architecture of the T5 model that uses encoder decoder.

The main equation at the heart of T5-Small model is:

$$\text{CrossAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

B. DistilBERT:

In 2018, Google developed the powerful natural language processing method referred to as BERT (Bidirectional Encoder Representations from Transformers) [6]. It is a deep learning model that understands contextual connections between words in a text by undergoing pre-training on a large dataset of text utilizing self-supervised learning [7]. BERT utilizes the transformer architecture and is designed to process sequential data, such as text. It can identify long-distance connections between words in a text due to its multiple layers of self-attention mechanisms. In predicting outcomes, the bidirectional BERT model can consider context from both the left and the right. This helps BERT understand the significance and context of words in a sentence [7]. DistilBERT is 40% smaller, 60% faster, and a lighter version of the BERT with a 97% increase in performance.

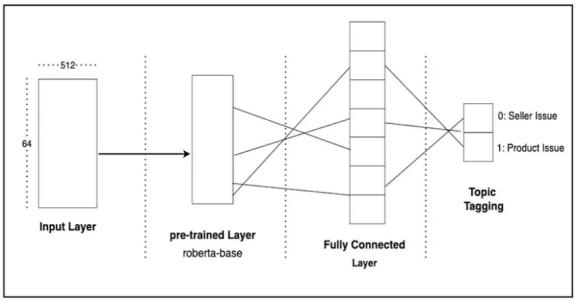

Fig 2. DistilBERT Architecture [7]

Fig. 3 shows the architecture of the DistillBERT with the different layers in it.

In the above figure BERT model is pre-trained on a large corpus of Amazon Negative Product Review data and can be fine-tuned for various NLP tasks. There are multiple layers in the BERT Model:

- *Input Layer:* IDs and masks are input into the BERT model. These inputs are tokenizer-obtained encoded representations of the input text [7].
- *Pre-trained Layer:* A neural network that has already been extensively trained on text data is referred to as the pre-trained layer. The pretrained layer of the Roberta model processes input text and generates a series of hidden states for every input token. The deep architecture of the RoBERTa model includes feed-forward neural networks and several self-attention layers [7]
- *Fully Connected Layer:* This is a linear layer that translates the pre-trained layer's output to the dimensionality that is needed [7].

The forward pass creates a concealed state sequence by passing the input ids and mask through the pre-trained layer. A vector of size two that represents the probabilities of the two classes in the topic tagging job is the final output of the model, which is obtained by passing the output of the dropout layer through the linear output layer [6].

The main equation used in DistilBERT is:

$$\text{Loss} = \alpha \cdot \text{KL-Divergence}(S, T) + (1 - \alpha) \cdot \text{CrossEntropy}(S, y)$$

C. GPT-2-Small:

GPT-2 is a transformer model that has been pre-trained in a self-supervised manner on a sizable corpus of English data [3]. It can use a large amount of publicly available data since it was pre-trained on the raw texts only, without any human labeling [3]. It then used an automated procedure to create inputs and labels from those texts. More specifically, it was trained to estimate phrases' next word. The targets are the same sequence, but with one token (word or word fragment) moved to the right [4]. The inputs are continuous text sequences of a specific length. To ensure that the predictions for the token I only use the inputs from 1 to i and not the future tokens, the model employs a mask-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

mechanism [3]. In this manner, the model acquires an internal representation of the English language, from which elements beneficial for subsequent tasks can be extracted [3]. However, the model excels at producing texts in response to a prompt, which is what it was pre-trained for.

The main formula used to compute the output of this model is:

$$\text{MaskedAttention}(Q,K,V) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V$$

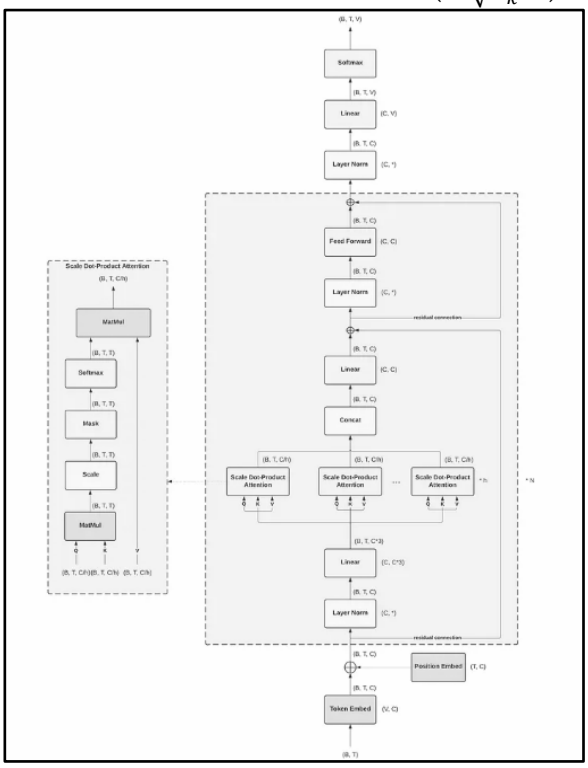

Fig 3. GPT-2-Small Architecture [8]

Figure 2 shows the architecture of GPT-2-Small and all its different components.

The encoder consists of an identical stack of N = 6 levels. Within each layer there are two sub-layers. The first is a simple, position-wise, fully connected feed-forward network, and the second is a multi-head self-attention mechanism. The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder adds a third sub-layer that performs multi-head attention over the output of the encoder stack.

III. METHODOLOGY

The methodology outlined is for the experiment to compare the performance of the model in the task of text completion. The observations of this experiment will be used to determine the effectiveness of the model to generate text accurately, coherently and efficiently.

A. Dataset

The dataset used in the experiment to train and test the models is 'wikitext-2-raw-v1'. The WikiText language modeling dataset is a collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. The dataset is available under the Creative Commons Attribution-ShareAlike License [9]. This is an open dataset loaded from HuggingFaces. The dataset was split into a training set of 1000 samples and a test set of 10 samples, both randomly shuffled in each iteration of the test. Both the training and test data were formatted for the task of text completion.

B. Models

The models described above (GPT-2-Small, T5-Small, DistilBERT) were all loaded from HuggingFaces and fine-tuned using the same training dataset. They were also evaluated on the same test dataset for a fair comparison.

C. Metrics

The models were evaluated by the following metrics:

1) *Loss:* For measuring errors in prediction. (Lower is better).

2) *Accuracy:* For measuring token-level prediction accuracy. (Higher is better)

3) *Precision, Recall, and F1-Score:* For evaluating the generated tokens' alignment with ground truth labels. (Higher is better)

4) *ROUGE:* Capturing the longest common subsequences in predictions. (Higher is better) This is calculated using the formula:

$$\text{ROUGE} = \frac{\text{Overlap of n-grams}}{\text{Total n-grams in reference}} \quad \text{where:}$$

Overlap of n-grams is the count of n-grams common to the hypothesis and reference,
Total n-grams in reference is the count of n-grams in the reference text.

5) *BLEU:* Evaluating n-gram overlaps between generated and reference texts. (Higher is better) This is calculated using the formula:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log p_n\right) \quad \text{where:}$$

$$\text{BP} = \exp\left(\min\left(0, 1 - \frac{r}{c}\right)\right)$$

p_n is the precision of n-grams (overlap between candidate and reference),
N is the maximum n-gram length considered,
c is the length of the candidate text, and
r is the length of the reference text.

6) *CHrF:* Measuring character-level similarity in shorter texts. (Higher is better) This is calculated using the formula:

$$\text{ChrF} = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \quad \text{where:}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

$$\text{Precision} = \frac{\text{Overlap of character n-grams}}{\text{Total character n-grams in candidate}}$$

$$\text{Recall} = \frac{\text{Overlap of character n-grams}}{\text{Total character n-grams in reference}}$$

β is the weight to balance precision and recall (typically set to 2 or 1).

7) *Perplexity:* Evaluating how well a model can predict the next token given the preceding context. It is mathematically calculated as the inverse of the (geometric) average probability assigned to each word in the test set by the model [10]. (Lower is better). This is calculated using the formula:

$$\text{Perplexity} = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log_2 P(w_i)} \quad \text{where:}$$

N is the total number of words in the sequence,
$P(w_i)$ is the predicted probability of the i-th word.

8) *Diversity:* Calculating the proportion of unique n-grams in the output to assess creativity and language variety. A diversified system contains more information and can better fit for various environments [11]. (Higher is better) This is calculated using the formula:

$$\text{Diversity} = \frac{\text{Unique n-grams}}{\text{Total n-grams}} \quad \text{where:}$$

Unique n-grams is the count of distinct n-grams in the output,
Total n-grams is the total number of n-grams in the output.

9) *Memory Usage:* Tracking memory usage of the CPU/GPU when training or querying the model.

10) *Timings:* Tracking time taken when training or querying the model.

D. Setup

All the tests were conducted on a system with the Intel Core i9-13900H CPU with 32 GB RAM and NVIDIA RTX 4080 Mobile GPU with 12 GB VRAM. The Python libraries Torch and PSUtil were used to track resource and memory consumption. Each model was trained for 1 epoch, for comparable resource usage. The optimizer used was AdamW with a learning rate of 5×10^{-5}. Each prediction was generated with a set of fixed hyperparameters, to ensure standardized inputs across all models. To ensure that the results can be reproduced, a fixed seed was used for all randomizations.

E. Procedure

Each model was tested over a series of 10 independent test runs. The results from all these runs were aggregated and analyzed to determine trends. The analysis focused on judging overall performance (average scores across all the test runs), finding trade-offs (which models trade efficiency for quality) and for determining how models deal with nuances in context when dealing with text completion tasks.

IV. RESULT ANALYSIS AND DISCUSSION

The data for the three models reveals the following observations. The graphical results have been scaled using Min-Max and Z-Score scaling to be displayed. The means of the data have been tabulated in Table 1.

A. Loss and Accuracy

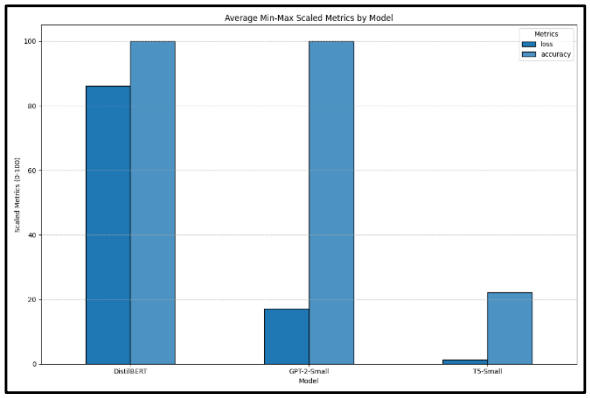

Fig 4. Loss and Accuracy Comparison

Fig. 4 compares the average min-max scaled Loss and Accuracy of the three models using a Bar Graph.

For DistilBERT, the loss is very high, and accuracy is 100%. This suggests overfitting or a mismatch between the loss and accuracy. For GPT-2-Small, the loss is moderate, but accuracy is still high. This suggests that the model performs well and can make predictions reliably. For T5-Small, the loss is low, and the accuracy is much lower than the others. This means that the model might not be able to generalize, or it is not optimized for this task. To conclude, DistilBERT and GPT-2 Small have perfect accuracy but have different loss values, with GPT-2 Small having the most balanced loss and accuracy relationship. T5 Small, has the lowest loss and shows lower accuracy, which can indicate an imbalance in how the model interprets loss and accuracy.

B. F1 Score, Precision and Recall

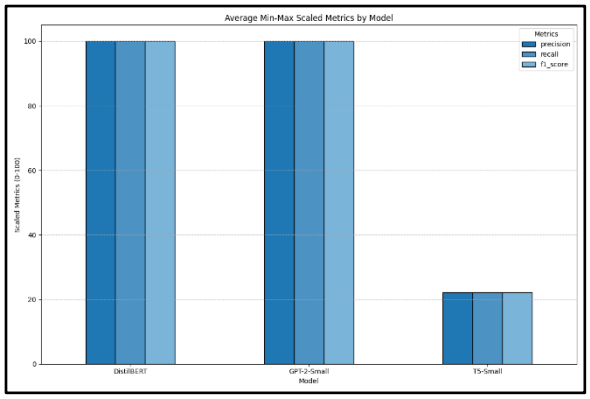

Fig 5. Precision, Recall, F1-Score Comparison

Fig. 5 compares the average min-max scaled Precision, Recall and F1-Score of the three models using a bar graph.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

DistilBERT and GPT-2-Small have perfect precision, recall, and F1 score, which means that it is very effective at identifying true positives and avoiding false positives/negatives. This can mean that the model is overfitting or being tested on a dataset with minimal complexity/noise.

C. *ROUGE, BLEU and CHrF*

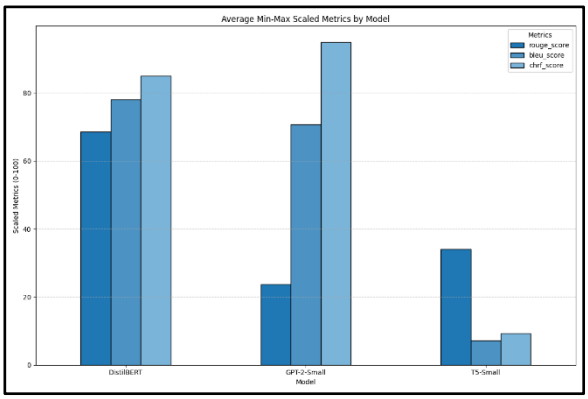

Fig 6. ROUGE, BLEU and CHrF Comparison

Fig. 6 compares the average min-max scaled ROUGE, BLUE and CHrE of the three models using a Bar Graph.

For DistilBERT, the scores are consistently high across all metrics, showing strong performance in capturing both lexical and semantic similarity. GPT-2-Small has moderate ROUGE and BLEU, but the highest CHrF score, which means that it excels in character-level similarity while being weaker in higher-level semantic or syntactic similarity. T5-Small has poor performance compared to the other models, especially BLEU and CHrF. This means that it struggles a lot in lexical and semantic similarity, which means that it's not well-suited for this specific task or dataset.

D. *Perplexity and Diversity*

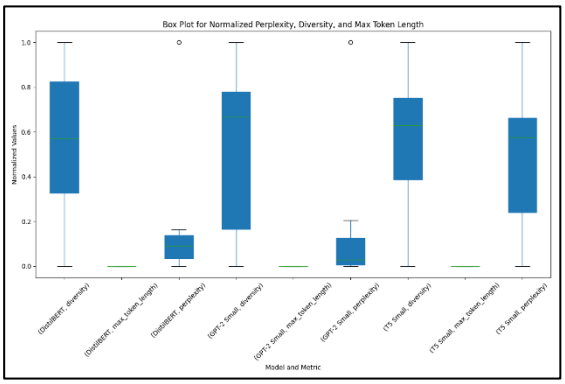

Fig 7. Perplexity and Diversity Comparison

Fig. 7 compares the average min-max normalized Perplexity and Diversity of the three models using a box plot.

DistilBERT exhibits extremely high perplexity, suggesting significant variability in its predictive confidence. Its high average diversity score indicates that the generated outputs are varied. The maximum token length aligns with its typical usage constraints for efficient processing. This model shows strong diversity for its moderate token capacity but suffers from inflated perplexity. GPT-2 Small demonstrates lower average perplexity, meaning it is more stable in its predictive confidence. However, its diversity score is lower, meaning its outputs are less varied. Its maximum token length provides a broader context window for generating text. This model is suitable for longer text generation tasks, with reduced diversity. T5-Small has exceptionally low perplexity, meaning it has high predictive confidence and consistency. Its diversity score is the highest, meaning it has highly varied outputs. Its maximum token length limits its ability to handle longer contexts. This model excels in diversity and confidence, but it's constrained in extensive input sequence tasks.

E. *Time and Memory Consumption*

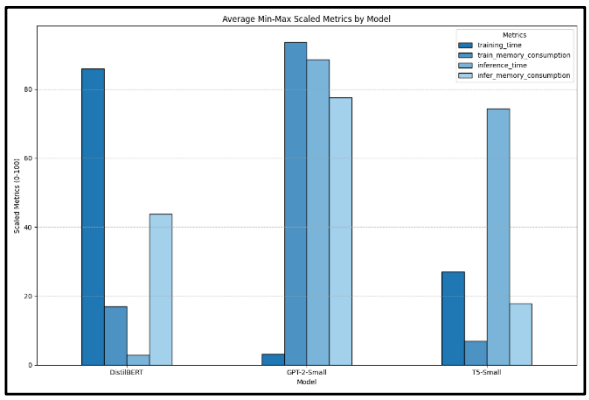

Fig 8. Time and Memory Consumption Comparison

Fig. 8 compares the average min-max scaled Time and Memory Consumption of the three models while running using a Bar Graph.

DistilBERT has moderate training time and memory usage. It has very fast inference speed and low memory requirements, which makes it ideal for environments requiring fast predictions and are constrained in resources. GPT-2-Small trains quicker than the other models, but demands much more memory for both training and inference. Its inference time is the highest, making it less suitable for real-time applications. T5-Small has the lowest memory consumption for training as well as inference. The inference speed of this model is slower than DistilBERT, but it is faster than GPT-2-Small, making it a suitable choice for environments with moderate constraints. While each model has its own merits and demerits, we can infer the following:

1) GPT-2-Small balances good loss, perplexity, CHrF, BLEU and ROUGE, and with decent efficiency. This makes it the best for classification related tasks. It is also robust for text metrics like CHrF.
2) T5-Small is lightweight and efficient, but this is at the cost of its performance in accuracy, CHRF, BLEU and ROUGE.
3) DistilBERT is reliable in many scenarios, but falls short in our experimental setup, where text completion

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

was being conducted. With the correct pre-processing and fine-tuning, it can also be as reliable as the other 2 models

TABLE I. SUMMARY OF MEANS OVER 10 RUNS

Parameter	DistilBERT	GPT-2-Small	T5-Small
Accuracy	1	1	0.3
Precision	1	1	0.3
Recall	1	1	0.3
F1-Score	1	1	0.3
Loss	14.3790	2.8634	0.2115
ROUGE	0.6905	0.3944	0.4631
BLEU	0.6076	0.5525	0.0827
CHrF	83.4279	90.1366	32.5990
Perplexity	3346370.7375	51265.7784	1.17937
Diversity	0.8990	0.6803	0.9505
Max. Token Length	512	1024	512
Training Time	74.6719	47.3869	55.2607
Inference Time	0.1637	0.5708	0.5028
Training Memory Consumption	834	1560	780
Inference Memory Consumption	9.0421	9.9981	8.0322
Precision	1	1	0.3

V. CONCLUSION

The aim of this research was to highlight the strengths and weaknesses of DistilBERT, GPT-2-Small, and T5-Small, emphasizing their merits and demerits in different applications. GPT-2-Small emerges as a versatile model, having good performance in many of the metrics, making it ideal for text generation and classification tasks where relevance to the context is important. DistilBERT excels in inference speed and diversity but falls short in probability-based tasks and longer context handling. With proper fine-tuning, it can be effective in certain tasks like text summarization. T5-Small appears to be lightweight and efficient and shows promise in probability-intensive and deterministic tasks. However, it has lackluster performance in generative metrics and token handling. Ultimately, the choice of model should depend on task requirements, making sure to balance performance trade-offs, computational demands, and memory constraints.

REFERENCES

[1] Q. Chen, "T5: a detailed explanation. Given the current landscape of transfer… | by Qiurui Chen | Analytics Vidhya," 8 June 2020. [Online]. Available: https://medium.com/analytics-vidhya/t5-a-detailed-explanation-a0ac9bc53e51. [Accessed 4 December 2024].

[2] M.-W. C. K. L. a. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"," in Conference of the North American Chapter of the Association for Computational Linguistics, 2019.

[3] "openai," OpenAI, 5 November 2019. [Online]. Available: https://openai.com/index/gpt-2-1-5b-release/. [Accessed 5 January 2025].

[4] O. L. A. V. Michael Hanna, "arxiv," 02 November 2023. [Online]. Available: https://arxiv.org/pdf/2305.00586. [Accessed 05 January 2025].

[5] "arxiv," 22 July 2020. [Online]. Available: https://arxiv.org/pdf/2005.14165. [Accessed 05 January 2025].

[6] L. D. J. C. T. W. Victor SANH, "arxiv," 01 March 2020. [Online]. Available: https://arxiv.org/pdf/1910.01108. [Accessed 05 January 2025].

[7] P. Kumari, "BERT and DistilBERT Models for NLP | by Priyanka Kumari," 22 June 2023. [Online]. Available: https://medium.com/@kumari01priyanka/bert-and-distilbert-model-for-nlp-7352eb16915e. [Accessed 29 December 2024].

[8] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, "arxiv," 02 August 2023. [Online]. Available: https://arxiv.org/pdf/1706.03762. [Accessed 05 January 2025].

[9] Salesforce, "Wikitext Dataset," Hugging Face. [Online]. Available: https://huggingface.co/datasets/Salesforce/wikitext. [Accessed: Jan. 5, 2025].

[10] S. Chen, D. Beeferman, and R. Rosenfeld, "Evaluation Metrics for Language Models," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, 1998.

[11] Z. Gong, P. Zhong, and W. Hu, "Diversity in Machine Learning," IEEE Access, vol. 7, pp. xxxx-xxxx, May 2019, doi: 10.1109/ACCESS.2017.DOI.