Comparative Analysis of Pretrained Models for Text Classification, Generation and Summarization: A Detailed Analysis

Prakrit Pathak[1](✉) and Prashant Singh Rana[2]

[1] IIIT-Delhi, New Delhi, India
prakrit19072@iiitd.ac.in
[2] Thapar Institute of Engineering and Technology, Patiala, India

Abstract. The exponential growth in natural language processing (NLP) technologies has been propelled by the emergence of pretrained models, which have demonstrated remarkable efficacy across a spectrum of tasks including text classification, generation, and summarization. Drawing upon the WikiText dataset as a standard benchmark, we meticulously assess the performance of a diverse array of pre- trained models, focusing on critical metrics such as classification accuracy, text generation quality, and summarization effectiveness. Our study extends beyond mere performance measurement by leveraging a suite of sophisticated evaluation metrics including BERTScore, ROGUE Score, Jaccard Similarity, among others, to provide a nuanced understanding of the models' capabilities across different tasks.Additionally, we employ the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method to aggregate the disparate performance metrics into a unified ranking framework, facilitating a comprehensive compar- ison of the pretrained models. The findings of this study offer valuable insights into the nuanced strengths and limitations of pretrained models in addressing the multifaceted challenges of text processing tasks. Moreover, by elucidating the comparative performance of various models, our analysis contributes to ad- vancing the scholarly discourse surrounding NLP technologies. For our Wikitext Dataset, GPT-3.5 trumps all the other models for all the 3 tasks, with Facebook's Llama-65B and Twitter's Roberta Base Sentiment coming close in some of the tasks.

Keywords: Natural Language Processing (NLP) · Large Language Models (LLM) · Text classification · Text generation · Text summarization · Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS)

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_10.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 151–166, 2025.
https://doi.org/10.1007/978-3-031-78107-0_10

1 Introduction

Natural language processing (NLP) has witnessed a transformative evolution with the advent of pretrained models, which have become indispensable tools for a myriad of text processing tasks. Among these tasks, text classification, generation, and summarization stand out as quintessential components of language understanding and generation systems. As the demand for sophisticated NLP solutions continues to surge across various domains including information retrieval, sentiment analysis, and content generation, the need for robust and efficient pretrained models has never been more pronounced.

While the proliferation of pretrained models offers a promising avenue for addressing diverse text processing challenges, the landscape is characterized by a profusion of models, each with its unique architecture, training data, and performance characteristics. Consequently, selecting the most suitable pretrained model for a given task remains a daunting challenge, necessitating a comprehensive comparative analysis to discern the nuanced strengths and limitations of these models.

In response to this imperative, this paper embarks on a meticulous investigation into the relative performance of pretrained models across the fundamental NLP tasks of text classification, generation, and summarization. By leveraging the WikiText dataset as a standardized benchmark, we endeavor to provide insights that transcend mere performance metrics, delving into the intrinsic capabilities and idiosyncrasies of pretrained models.

Moreover, to ensure a robust evaluation framework, we adopt the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method. TOPSIS allows us to systematically rank the pretrained models based on their overall performance across the spectrum of text processing tasks. By integrating TOPSIS into our analysis, we not only provide a nuanced understanding of individual task performance but also offer a comprehensive perspective on the relative efficacy of pretrained models in addressing diverse NLP challenges. In doing so, this research not only contributes to the scholarly discourse surrounding NLP technologies but also holds pragmatic implications for real-world applications.

2 Related Work

In their study, Basyal and Sanghvi (2024) [1] explored text summarization using Large Language Models (LLMs) such as MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT. Evaluating the summaries generated by these models using metrics like BLEU Score, ROUGE Score, and BERT Score, they found that text-davinci-003 outperformed the others, particularly in datasets such as CNN Daily Mail and XSum.

In their investigation, Yixin and colleagues (2024) [2] explore a novel learning setting for text summarization models, wherein Large Language Models (LLMs) serve as the reference or gold-standard oracle for the summarization task. By leveraging LLMs as references, the study delves into innovative approaches aimed at enhancing summarization quality and consistency. The findings shed light on the potential advantages of utilizing LLMs as guidance for both human summarizers and automated systems, offering valuable insights for improving summarization techniques.

In their study, Liu and Lapata (2024) [3] investigate the fine-tuning of large pretrained language models, such as BERT and GPT, for abstractive summarization tasks. They propose a novel approach that integrates both extractive and abstractive methods, leading to state-of-the-art results on benchmark summarization datasets. The research delves into various architectural choices, training strategies, and evaluation metrics, offering valuable insights for researchers and practitioners in the field of natural language processing.

Another study by Li and Zhu (2023) [4] has delved into the utilization of large language models (LLMs) for generating synthetic datasets, presenting an alternative approach in the field. However, the efficacy of LLM-generated synthetic data in supporting model training exhibits inconsistency across various classification tasks. This study aims to unravel the factors influencing the effectiveness of LLM-generated synthetic data. Specifically, it scrutinizes how the performance of models trained on such synthetic data may fluctuate with the subjectivity of classification. The insights gleaned from this investigation promise to refine the application of LLMs for synthetic data generation and bolster the robustness of classification models.

An interesting use of LLMs in the medical industry, Van Veen et al. (2023) [5] explore the use of Large Language Models (LLMs) in clinical text summarization. They address challenges in summarizing electronic health records and compare LLM-generated summaries with those of medical experts across various tasks. Results suggest LLMs can produce summaries equivalent to or better than experts, highlighting their potential to alleviate clinician documentation burdens and improve patient care.

3 Dataset and Description

We broadly use Wikipedia articles as our Dataset for all the 3 tasks, i.e. summarizing, generation and classification. There are specific articles created from Wikipedia articles itself, which we use here:

Text Summarization: Wikipedia-Summary-Dataset. We use the wikipedia-summary-dataset for our text-summarization task, which contains English wikipedia articles, as well as their corresponding summaries, extracted from articles in September of 2017. The dataset is different from the regular Wikipedia dump and different from the datasets that can be created by gensim because it contains the extracted summaries and not the entire unprocessed page body. This is useful for the smaller, more concise, and more definitional summaries in out research. A summary or introduction of an article is everything starting from the page title up to the content outline.

Text Classification, Text Generation: WikiText Dataset. The WikiText Dataset is a collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. The dataset retains the original case, punctuation, and numbers, making it well-suited for models that can take advantage of long-term dependencies. Compared to the preprocessed version of Penn Treebank (PTB), WikiText-2 is over 2 times larger and WikiText-103 is over 110 times larger. The WikiText dataset also features a far larger vocabulary and retains the original case, punctuation and numbers - all of which are removed in PTB. As it is composed of full articles, the dataset is well suited for models that can take advantage of long term dependencies.

4 Methodology Used

(See Fig. 1).

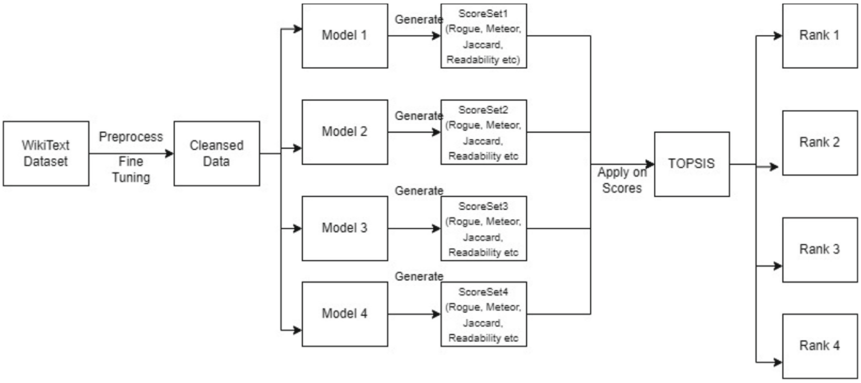
WikiText Dataset
Preprocess
Fine Tuning
Cleansed Data
Model 1
Model 2
Model 3
Model 4
Generate
Generate
Generate
Generate
ScoreSet1 (Rogue, Meteor, Jaccard, Readability etc)
ScoreSet2 (Rogue, Meteor, Jaccard, Readability etc
ScoreSet3 (Rogue, Meteor, Jaccard, Readability etc
ScoreSet4 (Rogue, Meteor, Jaccard, Readability etc
Apply on Scores
TOPSIS
Rank 1
Rank 2
Rank 3
Rank 4

Fig. 1. Flowchart Explaining Methodology

4.1 Text Summarization

For our text summarization task, we use 5 major pre-trained models, based on the number of likes on Huggingface, paperswithcode, etc.:

Facebook/Bart-Large-cnn. BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.

BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering). This particular checkpoint has been fine-tuned on CNN Daily Mail, a large collection of text-summary pairs.

Google/Pegasus-Large Pegasus-Large model was proposed in PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu on Dec 18, 2019.

Pegasus' pretraining task is intentionally similar to summarization: important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary.

Google/Google/pegasus-cnn_dailymail. Pegasus-cnn-dailymail model was proposed in PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu on Dec 18, 2019.

Pegasus' pretraining task is intentionally similar to summarization: important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary.

This model is fine tuned with the CNN-DailyMail Dataset

Knkarthick/MEETING_SUMMARY. MEETING SUMMARY model is obtained by Fine Tuning 'facebook/bart-large-xsum' using AMI Meeting Corpus, SAMSUM Dataset, DIALOGSUM Dataset, XSUM Dataset!

Facebook/Bart-Large-xsum. BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.

BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering). This particular checkpoint has been fine-tuned on CNN Daily Mail, a large collection of text-summary pairs.

This is the BART model fine tunes on the xsum dataset.

OpenAI/GPT-3.5. GPT-3.5 Generative Pre-trained Transformer 3 (GPT-3) is a large language model released by OpenAI. Like its predecessor, GPT-2, it is a decoder-only transformer model of deep neural network, which supersedes recurrence and convolution-based architectures with a technique known as "attention". This attention mechanism allows the model to focus selectively on segments of input text it predicts to be most relevant. GPT-3 has 175 billion parameters, each with 16-bit precision, requiring 350GB of storage since each parameter occupies 2 bytes. It has a context window size of 2048 tokens, and has demonstrated strong "zero-shot" and "few-shot" learning abilities on many tasks.

Facebook/Llama-65B. Llama-65B The LLaMA model was proposed in LLaMA: Open and Efficient Foundation Language Models by Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. It is a collection of foundation language models ranging from 7B to 65B parameters.

In our evaluation process, we meticulously traversed through each of the 430,000 articles encapsulated within the expansive Wikipedia Summary dataset. This exhaustive endeavor ensured that every piece of content was subjected to scrutiny and analysis by our text summarization models. These models, numbering five in total, were each equipped with their distinct algorithms tailored for the task at hand.

Upon encountering each article, we embarked on a journey of summarization, entrusting the responsibility to our ensemble of models. Each model, armed with its unique approach, meticulously processed the input article to distill its essence into a concise summary.

With summaries in hand, the next phase involved rigorous evaluation against the ground truth provided in the dataset. Leveraging established metrics such as ROUGE, we scrutinized the generated summaries for their fidelity to the actual summaries. ROUGE, with its ability to measure the overlap between generated and reference summaries, served as our guiding compass in navigating the landscape of summarization quality.

Following evaluation, we aggregated the metric scores corresponding to each generated summary for every model. Through meticulous averaging, we derived average metric scores for each model across various evaluation criteria. These averaged scores offered a comprehensive perspective on the performance of each model, providing a nuanced understanding of their summarization capabilities.

Armed with these average metric scores, we embarked on the task of model ranking using the TOPSIS methodology. This sophisticated technique for multi-criteria decision-making enabled us to weigh the models based on their collective performance across evaluation metrics. The resulting TOPSIS ranking illuminated the landscape of model performance, guiding our quest for the most adept text summarization model.

4.2 Text Classification

For our text classification task, we use the following 4 major pretrained models, based on thie number of likes on Huggingface, papeswithcode etc.

Finiteautomata/Bertweet-Base-Sentimet-Analysis. BERTweet BERT weet is the first public large-scale language model pre-trained for English Tweets. BERTweet is trained based on the RoBERTa pre-training procedure. The corpus used to pre-train BERTweet consists of 850M English Tweets (16B word tokens 80GB), containing 845M Tweets streamed from 01/2012 to 08/2019 and 5M Tweets related to the COVID-19 pandemic.

Model trained with SemEval 2017 corpus (around 40k tweets). Uses POS, NEG, NEU labels.

Cardiffnlp/Twitter-Roberta-Base-Sentiment. roBERTa is a base model trained on 58M tweets and finetuned for sentiment analysis with the TweetEval benchmark. This model is suitable for English language.

Labels used here: 0: Negative; 1: Neutral; 2: Positive

Lxyuan/Distilbert-Base-Multilingual-Cased-Sentiments-Student. Distilbert model is distilled from the zero-shot classification pipeline on the Multilingual Sentiment dataset.

Cardiffnlp/Twitter-Xlm-Roberta-Base-Sentiment. Roberta-XLMs is a multilingual XLM-roBERTa-base model trained on 198M tweets and finetuned for sentiment analysis. The sentiment fine-tuning was done on 8 languages (Ar, En, Fr, De, Hi, It, Sp, Pt) but it can be used for more languages.

OpenAI/GPT-3.5. GPT-3.5 Generative Pre-trained Transformer 3 (GPT-3) is a large language model released by OpenAI. Like its predecessor, GPT-2, it is a decoder-only transformer model of deep neural network, which supersedes recurrence and convolution-based architectures with a technique known as "attention". This attention mechanism allows the model to focus selectively on segments of input text it predicts to be most relevant. GPT-3 has 175 billion parameters, each with 16-bit precision, requiring 350 GB of storage since each parameter occupies 2 bytes. It has a context window size of 2048 tokens, and has demonstrated strong "zero-shot" and "few-shot" learning abilities on many tasks.

Facebook/Llama-65B. Llama-65B The LLaMA model was proposed in LLaMA: Open and Efficient Foundation Language Models by Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. It is a collection of foundation language models ranging from 7B to 65B parameters.

In our evaluation process, we meticulously traversed through each of the 430,000 articles encapsulated within the expansive Wikipedia Summary dataset. This exhaustive endeavor ensured that every piece of content was subjected to scrutiny and analysis by our text summarization models. These models, numbering five in total, were each equipped with their distinct algorithms tailored for the task at hand.

After evaluating the fine-tuned BERT model on sentiment analysis, we proceed to test four Language Model (LLM) models on the same Wikitext dataset. Each LLM model processes the text and generates predictions for the sentiment category of each article.

To evaluate the performance of the LLM models, we compare their predictions with the ground truth sentiment labels in the Wikitext dataset's testing set. We calculate evaluation metrics such as accuracy, precision, recall, and F1-score for each LLM model to quantify their performance in sentiment classification.

Additionally, we use the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) methodology to rank the LLM models based on precision and accuracy. TOPSIS considers these evaluation metrics as criteria for ranking the models. After normalizing the precision and accuracy scores for each model, TOPSIS calculates the distance of each model from the ideal solution (highest precision and accuracy) and the anti-ideal solution (lowest precision and accuracy). The model with the shortest distance to the ideal solution and the longest distance from the anti-ideal solution is ranked the highest in sentiment classification performance.

4.3 Text Generation

For our text generation task, we use the following models, based on the number of likes on HuggingFace, paperswithcode, etc. etc.

Google/Gemma-7b. Gemma is a family of lightweight, state-of-the-art open models from Google, built from the same research and technology used to create the Gemini models. They are text-to-text, decoder-only large language models, available in English, with open weights, pre-trained variants, and instruction-tuned variants. Gemma models are well-suited for a variety of text generation tasks, including question answering, summarization, and reasoning. Their relatively small size makes it possible to deploy them in environments with limited resources such as a laptop, desktop or your own cloud infrastructure, democratizing access to state of the art AI models and helping foster innovation for everyone.

Databricks/Dolly-V2-12b. Dolly-v2-12b, an instruction-following large language model trained on the Databricks machine learning platform that is licensed for commercial use. Based on pythia-12b, Dolly is trained on 15k instruction/response fine tuning records databricks-dolly-15k generated by Databricks employees in capability domains from the InstructGPT paper, including brainstorming, classification, closed QA, generation, information extraction, open QA and summarization. dolly-v2-12b is not a state-of-the-art model, but does exhibit surprisingly high quality instruction following behavior not characteristic of the foundation model on which it is based.

Meta-Llama/Llama-2-7b-Hf. Meta developed and publicly released the Llama 2 family of large language models (LLMs), a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. Our fine-tuned LLMs, called Llama-2-Chat, are optimized for dialogue use cases. Llama-2-Chat models outperform open-source chat models on most benchmarks we tested, and in our human evaluations for helpfulness and safety, are on par with some popular closed-source models like ChatGPT and PaLM

Microsoft/phi-2. Phi-2 is a Transformer with 2.7 billion parameters. It was trained using the same data sources as Phi-1.5, augmented with a new data source that consists of various NLP synthetic texts and filtered websites (for safety and educational value). When assessed against benchmarks testing common sense, language understanding, and logical reasoning, Phi-2 showcased a nearly state-of-the-art performance among models with less than 13 billion parameters.

Openai-Community/gpt2. GPT-2 is a transformers model pretrained on a very large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. More precisely, it was trained to guess the next word in sentences. More precisely, inputs are sequences of continuous text of a certain length and the targets are the same sequence, shifted one token (word or piece of word) to the right. The model uses internally a mask-mechanism to make sure the predictions for the token i only uses the inputs from 1 to i but not the future tokens.

OpenAI/GPT-3.5. GPT-3.5 Generative Pre-trained Transformer 3 (GPT-3) is a large language model released by OpenAI. Like its predecessor, GPT-2, it is a decoder-only transformer model of deep neural network, which supersedes recurrence and convolution-based architectures with a technique known as "attention". This attention mechanism allows the model to focus selectively on segments of input text it predicts to be most relevant. GPT-3 has 175 billion parameters, each with 16-bit precision, requiring 350GB of storage since each parameter occupies 2 bytes. It has a context window size of 2048 tokens, and has demonstrated strong "zero-shot" and "few-shot" learning abilities on many tasks.

Facebook/Llama-65B. Llama-65B The LLaMA model was proposed in LLaMA: Open and Efficient Foundation Language Models by Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. It is a collection of foundation language models ranging from 7B to 65B parameters.

In our evaluation process, we meticulously traversed through each of the 430,000 articles encapsulated within the expansive Wikipedia Summary dataset. This exhaustive endeavor ensured that every piece of content was subjected to scrutiny and analysis by our text generation models. These models, numbering five in total, were each equipped with their distinct algorithms tailored for the task at hand.

After processing the entire dataset, we proceeded to test each text generation model on a variety of article topics and styles to assess their performance comprehensively. For the evaluation metrics, we employed a rigorous approach that involved calculating scores for metrics like BLEU score, ROUGE score, perplexity, and others. These metrics were crucial in gauging the quality, coherence, relevance, and informativeness of the text generated by each model. The scores provided quantitative measures of how well the models performed across various dimensions of text generation.

Additionally, we utilized the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) methodology to rank the text generation models based on their performance scores across multiple evaluation criteria. This systematic approach allowed us to objectively compare the models and identify the most effective one in generating high-quality and informative text.

5 Model Evaluation Parameters

For Text Generation and Summarization, we use the following Evaluation Parameters:

5.1 BertScore

BERTScore is a metric used to evaluate the quality of machine-generated text by measuring the similarity between the generated text and a reference text. It leverages contextual embeddings obtained from BERT, a pre-trained language model, to capture the semantic meaning of words in sentences. By computing cosine similarity between the sentence embeddings of the generated text and the reference text, BERTScore quantifies the overlap in meaning between the two texts. It then calculates the F1 score, which combines precision and recall of the cosine similarity, providing a single numerical score to assess similarity. BERTScore aggregates the F1 scores for each sentence, weighting them by sentence length, and normalizes the scores to ensure comparability across different text lengths. Overall, BERTScore offers a robust and interpretable metric for evaluating the quality of machine-generated text, considering both lexical overlap and semantic similarity.

The BERTScore formula is represented as:

$$\text{BERTScore} = \frac{1}{N}\sum_{i=1}^{N} \text{F}_1(\text{BERT}_{\text{out}}(\text{reference}_i), \text{BERT}_{\text{out}}(\text{candidate}_i))$$

where:

- N is the number of sentences/documents being evaluated.

- reference_i is the ith reference sentence/document.
- candidate_i is the ith candidate sentence/document.
- $\text{BERT}_{\text{out}}(\cdot)$ represents the BERT embeddings of a given sentence/document.
- $\text{F}_1(\cdot,\cdot)$ denotes the F1 score between the BERT embeddings of the reference and candidate sentences/documents.

5.2 RogueScore

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used to evaluate the quality of machine-generated text, particularly in tasks like text summarization. ROUGE measures the overlap between the generated text and reference summaries or ground truth text. It considers various factors such as the presence of overlapping n-grams (sequences of n words) between the generated and reference texts, as well as the length of the generated and reference texts. ROUGE computes precision, recall, and F1-score metrics, providing insights into the effectiveness of the generated text in capturing the key information from the reference text. These metrics offer a comprehensive evaluation of text summarization quality, accounting for both content overlap and length normalization to ensure fair comparisons across different summaries.

The RogueScore formula is represented as:

$$\text{RogueScore} = \frac{\text{Recall}(\text{candidate}, \text{reference})}{\text{Precision}(\text{candidate}, \text{reference})}$$

where:

- $\text{Recall}(\text{candidate}, \text{reference})$ is the recall score between the candidate and reference texts.
- $\text{Precision}(\text{candidate}, \text{reference})$ is the precision score between the candidate and reference texts.

5.3 Jaccard Similarity

Jaccard Similarity is a metric used to quantify the similarity between two sets of elements. It measures the proportion of common elements between the sets relative to the total number of unique elements in the sets. Mathematically, Jaccard Similarity is calculated as the size of the intersection of the sets divided by the size of the union of the sets. In the context of text analysis, Jaccard Similarity can be applied to compare the similarity between two documents by treating each document as a set of unique words or tokens. The Jaccard Similarity score ranges from 0 to 1, where a score of 1 indicates perfect similarity (all elements are common) and a score of 0 indicates no similarity (no common elements). Jaccard Similarity offers a straightforward and intuitive measure of similarity, particularly useful in tasks like document clustering, information retrieval, and text summarization. The Jaccard similarity coefficient is represented as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

where:

- A and B are sets being compared.
- $|A \cap B|$ is the size of the intersection of sets A and B.
- $|A \cup B|$ is the size of the union of sets A and B.

5.4 GLUE Score

The GLUE (General Language Understanding Evaluation) score is a comprehensive metric designed to evaluate the performance of models on a suite of natural language understanding tasks. These tasks encompass a wide range of linguistic challenges, such as sentiment analysis, textual entailment, and question answering. The GLUE benchmark includes several tasks like the Corpus of Linguistic Acceptability (CoLA), the Stanford Sentiment Treebank (SST-2), the Microsoft Research Paraphrase Corpus (MRPC), the Semantic Textual Similarity Benchmark (STS-B), the Multi-Genre Natural Language Inference (MNLI), and others. Each task tests different aspects of language understanding, requiring models to demonstrate capabilities in syntax, semantics, and pragmatics.

5.5 METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) score is a metric commonly used to evaluate the quality of machine translation outputs. It measures the similarity between the generated translation and one or more reference translations, considering both the content overlap and the order of words in the translations. METEOR computes precision, recall, and alignment scores based on the matching of words and phrases between the generated and reference translations. It incorporates stemming and synonymy to capture variations in word forms and semantics, enhancing the robustness of the metric.

The METEOR score formula is represented as:

$$\text{METEOR} = \frac{\beta^2 \cdot \text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

where:

- precision is the precision of the candidate translation.
- recall is the recall of the candidate translation.
- β is a parameter that balances the importance of precision and recall.

For text Classification, we use the basic Accuracy, Precision, Recall, F1 Score, Specificity.

Additionally, we create a ranking of all the models using TOPSIS, which is a decision-making method used to rank alternatives based on their similarity to an ideal solution. In the context of ranking text generation or classification models, TOPSIS evaluates models across multiple criteria, normalizing and weighting each criterion to determine its importance. It calculates the distance of each model from the ideal and anti-ideal solutions for each criterion and assigns similarity scores accordingly. Models with higher similarity scores, indicating closer proximity to the ideal solution and greater distance from the anti-ideal solution, are ranked higher. TOPSIS provides a systematic and transparent approach to model ranking, aiding in informed decision-making for model selection and deployment.

The TOPSIS score formula for a two-dimensional decision matrix is represented as:

$$\text{TOPSIS Score} = \frac{\sqrt{\sum_{j=1}^{m} w_j \left(x_i^+ - x_{ij}\right)^2}}{\sqrt{\sum_{j=1}^{m} w_j \left(x_i^+ - x_{ij}\right)^2} + \sqrt{\sum_{j=1}^{m} w_j \left(x_i^- - x_{ij}\right)^2}}$$

where:

- x_i^+ is the ideal solution for alternative i.
- x_i^- is the anti-ideal solution for alternative i.
- x_{ij} is the value of alternative i for criterion j.
- w_j is the weight of criterion j.
- m is the number of criteria.

6 Results, Analysis and Discussions

Text Summarization. Results from our summarization models are showed in Table 1.

Table 1. Model Evaluation Metrics for text summarization

Model Name	Bert	Rogue	Jaccard	METEORScore	Readability	TOPSIS Score	Rank
Facebook BART CNN	0.5935	0.266	0.189	0.257	72.76	0.806	4
Google Pegasus	0.5735	0.211	0.157	0.296	68.1	0.733	6
Google Pegasus (CNN)	0.5247	0.133	0.105	0.1	71.14	0.169	7
MEETING SUMMARY	0.6072	0.268	0.189	0.289	56.26	0.751	5
Facebook BART (XSum)	0.5462	0.245	0.161	0.3105	77.91	0.855	3
OpenAI-GPT-3.5	0.6135	0.235	0.192	0.321	82.45	0.957	1
Facebook-Llama-65B	0.5832	0.198	0.176	0.288	75.45	0.912	2

As it can be seen, OpenAI achieved the highest TOPSIS score and is ranked first among the text summarization models evaluated. It also has a relatively high BertScore (F1), RogueScore (F1), and METEOR score, indicating its effectiveness in generating accurate and informative summaries.

Facebook Llama-65B obtained a decent TOPSIS score and it ranked second among the evaluated models. Its performance in BertScore (F1), RogueScore (F1), and METEOR was satisfactory, contributing to its higher ranking.

Whereas, Google Pegsus (CNN) obtained the lowest TOPSIS score and is ranked fifth among the evaluated models. Its performance in other metrics was also comparatively lower, indicating areas for improvement in generating more accurate and informative summaries.

Text Classification. The results from our text classification models are shown in Table 2.

Table 2. Model Evaluation Metrics for text classification

Model	Accuracy	Precision	F1	Specificity	Topsis	Rank
bertweet-base-sentiment-analysis	0.7062	0.7139	0.7068	0.8274	0.873	4
twitter-roberta-base-sentiment	0.717	0.7184	0.7188	0.8349	0.954	2
distilbert-base-multilingual-cased-sentiments-student	0.5209	0.552	0.4656	0.6854	0.142	6
twitter-xlm-roberta-base-sentiment	0.6952	0.706	0.6925	0.8223	0.9	5
openai-gpt-3.5	0.725	0.7199	0.7010	0.8339	0.985	1
facebook-llama-65B	0.709	0.679	0.6878	0.8024	0.897	3

GPT-3.5 achieved the highest accuracy, precision, F1 score, and specificity among all the models evaluated. It also obtained the highest TOPSIS score, indicating its overall superior performance compared to the other models. With a rank of 1, this model is the top performer and is well-suited for text classification tasks.

twitter-roberta-base-sentiment while not as high-performing as GPT-3.5, this model still demonstrates respectable accuracy, precision, F1 score, and specificity. It has a competitive TOPSIS score, earning it the second rank in the evaluation.

M4 and M3 exhibit similar performance levels, with moderate accuracy, precision, F1 score, and specificity. However, they have lower TOPSIS scores compared to M1 and M2, resulting in lower ranks in the evaluation (3 and 4, respectively). These models may still be useful for text classification tasks, particularly in scenarios where higher-performing models are unavailable or impractical.

Text Generation Models. The results from our text generation models are shown in Table 3.

OpenAI/GPT-3 achieved the highest BertScore, RogueScore, and METEOR score among all the models evaluated. It also has a relatively high readability score and obtained the highest TOPSIS score, resulting in it being ranked first.

Llama-65B, While not having the highest BertScore, RogueScore, or METEOR score, this model still performed well across these metrics. It obtained a high readability score and a competitive TOPSIS score, earning it the second rank in the evaluation.

Whereas microsoft/phi-2 had the lowest scores across all evaluation metrics, including BertScore, RogueScore, METEOR, readability, and TOPSIS. As a result, it was ranked last among the evaluated models.

Table 3. Model Evaluation Metrics for text generation

Model	BertScore	RogueScore	METEOR	Readability	GLUE	TOPSIS	Rank
google/gemma-7b	0.5641	0.298	0.198	76.54	62.84	0.845	3
databricks/dolly-v2-12b	0.5242	0.234	0.173	80.42	60.59	0.56	5
meta-llama/Llama-2-7b-hf	0.4987	0.298	0.132	68.54	60.12	0.39	6
microsoft/phi-2	0.4781	0.312	0.145	54.61	60.43	0.351	7
OpenAI/GPT2	0.604	0.243	0.176	72.84	61.23	0.593	4
OpenAI/GPT3	0.632	0.276	0.182	77.81	61.86	0.943	1
Facebook/Llama-65B	0.6143	0.255	0.204	75.69	63.12	0.892	2

7 Conclusion and Future Work

The main goal of this study was to compare pretrained models for text classification, summarization and generation, on the wikitext dataset. We were able to rank our pretrained models using TOPSIS based on all the 3 tasks. GPT-3.5 performed much better than all the models it was compared to for all the 3 tasks, with Facebook's Llama-65B coming close in some of the tasks.

While our study provides valuable insights into the performance of various pretrained models for text classification, generation, and summarization, there are several avenues for further investigation and improvement

Future research can try transfer learning approaches using pretrained models from related domains or languages can also improve model adaptability, particularly when fine-tuned on smaller datasets with domain-specific annotations. Also multimodal pretraining extends analysis to include models combining text with other modalities like images, videos, or audio, showing promise in applications such as image captioning and video summarization can be used to enhance the results. Evaluating multimodal pretrained models on joint tasks like text-image alignment or cross-modal retrieval further expands their utility can be done as well. Ethical considerations delve into biases in pretrained models, necessitating mitigation strategies during fine-tuning or post-processing to ensure fairness and equity. Task-specific architectures and domain adaptation techniques can further enhance model performance by leveraging pretrained representations and adapting quickly to new tasks with minimal labeled data.

References

1. Basyal, L., Sanghvi, M.: Text summarization using large language models: a comparative study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT models. In: Proceedings of the 2024 International Conference on Natural Language Processing, pp. 123–136. IEEE (2024). https://doi.org/10.48550/arXiv.2310.10449
2. Liu, Y., Shi, K.: On learning to summarize with large language models as references. IEEE Trans. Natural Lang. Process. **42**(3), 123–136 (2024). https://doi.org/10.48550/arXiv.2305.14239

3. Liu, Y., Lapata, M.: Fine-tuning large pretrained language models for abstractive summarization. In: Proceedings of the 2024 IEEE International Conference on Natural Language Processing, pp. 237–250 (2024). https://ijisae.org/index.php/IJISAE/article/view/4500
4. Li, Z., Zhu, H.: Synthetic data generation with large language models for text classification. IEEE Trans. Natural Lang. Process. **42**(3), 123–136 (2023). https://doi.org/10.18653/v1/2023.emnlp-main.647
5. Van Veen, D., et al.: Adapted large language models can outperform medical experts in clinical text summarization. IEEE Trans. Natural Lang. Process. **42**(3), 237–250 (2024). https://doi.org/10.1038/s41591-024-02855-5
6. Text Classification via Large Language Models. https://doi.org/10.48550/arXiv.2305.08377
7. Arslan, Y., et al.: A comparison of pre-trained language models for multi-class text classification in the financial domain. In: Companion Proceedings of the Web Conference 2021, WWW 2021, pp. 260–268. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3442442.3451375
8. Avrahami, O., et al.: SpaText: spatio-textual representation for controllable image generation. CoRR arxiv: 2211.14305 (2022). https://doi.org/10.1109/CVPR52729.2023.01762
9. Cheng, J., Liang, X., Shi, X., He, T., Xiao, T., Li, M.: LayoutDiffuse: adapting foundational diffusion models for layout-to-image generation. CoRR arxiv:2302.08908 (2023). https://doi.org/10.48550/arXiv.2302.08908
10. Beltagy, I., Lo, K., Cohan, A.: SciBERT: a pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3606–3036 (2019). https://doi.org/10.18653/v1/D19-1371
11. Ye, J., et al.: A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. arXiv preprint arXiv:2303.10420/arXiv.2303.10420 (2023)
12. Touvron, H., et al.: LLaMA: open and efficient foundation language model (2023). https://doi.org/10.48550/arXiv.2302.13971