International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimizing Text Summarization and Content Tagging: A Performance Comparison of General Purpose Large Language Models and Specialized Architectures

Bosco Chanam[1], Ashay Kumar Singh[2], Chris Dcosta[3], Arghadeep Das[4], Shwetambari Chiwhane[5]

[1,2,3,4]Final Year Student, Dept. of CSE, Symbiosis Institute of Technology, Pune, India
[5]Prof. Dr., Dept. of CSE, Symbiosis Institute of Technology, Pune, India

Abstract
In NLP, Text summarization and content tagging are essential problems that are dedicated to improving information accessibility and organization. Summarization reduces the quantity of information that has to be stored or transmitted, and content tagging enables information to be stored in categories.This project focuses on enhancing two critical tasks in natural language processing: Text summarization and content tagging are the two most typical applications of text comprehension. In the first task, the text summarization is accomplished with the help of a general-purpose large language model (LLM). It is then compared with other similar models for the purpose of summarization to check out for any enhancements in accuracy, coherence and relevance. This is in an effort to understand the efficiency of fine-tuning a general LLM compared to the application of task-specific models to fine-tune and improve text summarization for various usages. In the second task, content tagging, the BERT model is used on a classification data set where it is working on the specific task of labeling the given content with the appropriate tags. Then, the performance of the proposed BERT is compared with other classification models that were also presented in the research group and discussed earlier. The purpose here is to examine how effectively models can work in terms of being accurate, fast and smart in identifying content in line with context and the semantic analysis of the provided data. It is expected that the mentioned project will provide the overall description of both tasks together with determination of the models that are most suitable for the summarization and content tagging. This comparison aims to provide useful findings on the discrepancy between general and specialized models for accomplishing good quality text processing in real-world and practical context.

Keywords: Text Summarization, Content Tagging, NLP, LLM, BERT, Fine-Tuning, Text Classification, Semantic Analysis, Model Comparison

1. Introduction

This has been particularly challenging in the era of information explosion, whereby text analytics of large data volumes is a daunting task. The major tool that is currently on par with this challenge is Natural

Language Processing (NLP) that enhances the data handling and also information extraction. This research focuses on two pivotal tasks within NLP: text summarization and content tagging are two of such approaches. The aim is thus to propose and assess an automatic approach for these tasks based on several models and approaches.

1.1 Text Summarization

Text summarization is a reduction process of converting a large document into a smaller and summarized one in such a way that contains all the important information of the large document in simple and meaningful form. This task is important for different fields, such as News, Research, Content organization, etc., where the users need to get basic points without having to deal with full form documents.

Text summarization can be categorized into two primary types - Extractive Summarization and Abstractive Summarization. In extractive summarization, the method involves identifying the specific phrases, sentences or passages in the text that deserves to be quoted. The aim is to produce a summary which would be a reflection of the key aspects of the original text. Methods of extractive summarization are statistical methods, for instance frequency based methods and the machine learning methods, including supervised learning.

Whereas abstractive summarization synthesizes new sentences that bring out a summary of the entire text in a shorter form. This particular approach incorporates natural language features in the process and is more detailed than the previous one. It is also important to note that most abstractive summarization models are more complex and time consuming than extractive ones because, unlike extractive ones, they must not only identify potential portions of the text but also paraphrase the content meaningfully.

In this study, we evaluate a generic large language model (LLM) on text summarization tasks. The current deep learning models like GPT-Three and T5 have shown remarkable potentiality in mimicking human writing and they are used here to summarize. To do this, we test the general-purpose model against state-of-the-art summarization models such as BART and Pegasus. Both BART and Pegasus models have been especially developed for summarization tasks and according to the results of the experiment the models can generate high quality summaries.

To state, the purpose of the comparison envisaged is to assess the capacity of a generic LLM to create summaries, focusing on the summary quality, coherence and relevance of the outcomes in relation to specific models designed to read and summarize texts. To evaluate these models we use ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores among others. ROUGE scores give the percentage overlap between summary generated and reference summaries giving an idea of the accuracy and coverage of summarization.

1.2 Content Tagging

Content Tagging, also known as text classification entails assigning some tags or labels that have been predefined based on the content in the text. This task is crucial to store and sort the data, to avoid its chaos and to be able to find what is needed in the blink of an eye. From the above discussions, it can be noted that the content tagging has its use in the following domains: News classification, topic modeling and Information Retrieval systems.

The process of content tagging typically involves the following steps:The process of content tagging typically involves feature extraction, model training and prediction and evaluation. One will involve pre-processing of the data where the text data is transformed to make it suitable for passing to the ML models. Tokenization, Word2Vec and GloVe are some of the methods used in feature representation of text features.

The training data contains labeled text features hence the machine learning algorithms are used to learn text features and tags. Some of the techniques found include supervised learning techniques where the model learns from examples that are already labeled. After being trained, the concern model estimates second order tags in new, unseen textual data. Measurement of the model is done by using parameters like accuracy, precision, recall and F1 score in order to determine the efficiency of the model in the classification of text.

As the volume of additional data increases and becomes more complex and less structured, efficient tagging is important for performance as well as the user experience of content marketing. Widely scattered unstructured data across disparate tools and systems make content control and content discovery a challenge. This challenge is solved by automating classification and content tagging which categorizes information in a much better manner and also makes it easier to locate. This enables marketers to work on vast amounts of data conveniently, increase findability and personalize the interactions.

In this research, there is the use of BERT (Bidirectional Encoder Representations from Transformers) for content tagging. The method that has been employed in this work known as BERT is a transformer-based model that performs particularly well where context and dependencies are to be determined. It has been proved to be efficient in many NLP applications such as text classification. Compared to many other most recent models, BERT is a bidirectional model, which means that besides depending on the word it is currently analyzing, it takes into consideration the word that precedes and or the word that comes after it thus having a better shot at capturing context.

Here, we benchmark BERT with some of the go-ahead models, for example, RoBERTa (Robustly optimized BERT approach), DistilBERT, as well as XLNet. RoBERTa is the improved form of BERT with better training and performance all over. DistilBERT is the smaller and faster version of BERT but it's less accurate than BERT but closer to 70 percent. There is the so-called generalized autoregressive pretraining as an improved version of BERT as the XLNet model.

The purpose of the following comparison is to investigate which model yields the best accuracy and time complexity as well as computational requirement of news tagging for a given classification dataset.

2. Literature Review

Reference No.	Method Used	Dataset used	Performance / Outcome	Findings/ Limitations/ Future Scopes
[1]	PEGASUS. Transformer based pre training with Gap sentences Generation	C4, HugeNews, Multi News	Achieved novel results on multiple summarizatio n datasets.	High Computational Requirement Lack of FP16 support Less Flexible for other NLP requirements
[2]	BART. Denoising autoencoder for pre training sequence to sequence models	CNN/DailyMail, XSum	Outperformed existing summarizatio n models in many	Requires substantial computational resources for both training and inference. Performance greatly

			benchmarks.	relies on quality of input data + fine tuning.
[3]	Longformer. Transformer with a much longer attention span	ArXiv, PubMed	Improved performance on long document summarization	Hard to finetune Optimized but still requires a lot of computational resources
[4]	BERT. Use word and label semantics for multi label classification.	ARXIV Academic Paper Dataset, Reuters Corpus Volume I	Improved label specific information, demonstrating superiority on datasets and addressing imbalanced datasets	Limited context understanding and generation
[5]	RoBERTa. Robustly optimized BERT approach	GLUE, SQuAD	Enhanced performance over BERT on multiple benchmarks	High computation cost for pre training and finetuning
[6]	ALBERT. Lightweight version of BERT with some parameter reduction techniques for some improved performance metrics	GLUE, SQuAD	Achieved comparable performance to BERT with fewer parameters	Requires optimal and careful tuning for performance
[7]	T5 (Text To Text Transfer Transformer). Unified text to text framework	C4	Set new state of the art results in multiple NLP tasks, including summarization	Heavy Computational requirements Heavily depends on extensive pre training and fine tuning based on large datasets
[8]	DistilBERT.	GLUE, SQuAD	Enhanced	Requires extra fine

	Classification. Smaller, Faster, Cheaper version of BERT		runtime performance over BERT	tuning Lower accuracy than full size BERT models
[9]	PRIMERA. Supervised Learning. Hierarchical Transformer with multi task learning.	Multi News, WikiSum	Enhanced summarization for multi document datasets	Performance is very dataset dpendant Requires a lot of fine tuning for specific tasks
[10]	GPT 4. Large scale multimodal transformer model.	Not disclosed	Demonstrated strong few shot multi modal performance across various generation tasks	Extremely large model size Prone to hallucination.

[1] Pegsus is a Transformer-base model pre-trained with a new self-supervision task for text summarization in which the model hides important sentences and has to reconstruct them at the output while outputting the entire sequence at once. PEGASUS is also effective at summarization of 12 tasks, such as news and scientific texts, and proves to be excellent in outcomping previous techniques with the limited dataset as well as, to be within close proximity to human performance. On the other hand, this project compares a general large language model with other specific models such as PEGASUS and BART in the summarization task or between BERT and other sophisticated models in the content tagging task with the view of determining the optimal approaches for these NLP tasks.

[2] This paper provides BART which is a denoising autoencoder for pretraining sequence-to-sequence models and it is trained by making text noisy and then predicting them. BART is based on BERT's bidirectional encoder and GPT's left-to-right decoder, and they different experiments that involve noising include; shuffling of the sentences and in-filling. In the assessed text generation, comprehension and summarization tasks, BART presents helpful results with notable increases in the ROUGE scores.

[3] This paper aimed at designing the Longformer, which is a Transformer model that has been developed to help in the proper management of long sequences as compared to the other models that rely on attention that is proportional to the sequence length. Longformer's attention integrates the local and global to successfully pass over hundreds, or even thousands of tokens found in documents. It surpasses the performance of RoBERTa across character-level language modeling, and is particularly indicated to enhance performance with respect to multiple long-document tasks such as WikiHop and TriviaQA. Furthermore, the paper proposes Longformer-Encoder-Decoder (LED), a variant designed for generative sequence to sequence tasks and shows the improvements on arXiv summarization tasks.

[4] For multi-label classification BERT uses word and label semantics, performs well on ARXIV and Reuters Corpus Volume I datasets, enhances label wise information and works well to handle imbalanced label problems. But the weaknesses are in the identification of contextual semantics and in generation of text.

[5] This paper replicates BERT's pretraining in-depth in order to understand the impact that hyperparameters and size of training data have on it. The research also shows that BERT was trained considerably less than some other models, and with better hyperparameters, can outperform following models. The best performing model from this study offered equally competitive performance against the current state-of-the-art on GLUE, RACE, and SQuAD benchmarks. Hence, the results underline basic design decisions that were not given sufficient consideration in prior studies and question recent advances in the field. The paper also includes released models and code for future reference and extension of study.

[6] ALBERT (A Lite BERT) is a regularization of BERT that lowers the numbers of parameters that makes it ideal for devices that have little memory and high speeds. Unlike BERT, however, ALBERT makes use of efficient parameters by keeping the core Transformer network in tact. Instead, RoBERTa discovered by Facebook AI Research amplifies the BERT structure with enhanced training procedure and mors demands in data for higher performance in diverse NLP tasks. This work extends prior work by fine-tuning BERT, ALBERT, and RoBERTa in Indonesian language data sets for fake news categorization with the intention of evaluating their performance in this particular context.

[7] In this paper, we discuss transfer learning in NLP with a text-to-text unified framework that recasts numerous text-based tasks into the essentially unified text-to-text form. It compares and makes a comprehensive analysis to a range of pre-training objectives, architectures and the transfer techniques on multiple language understanding tasks. From their study and a new dataset named Colossal Clean Crawled Corpus, the authors set records for summarization, question answering and text classification benchmarks. The paper also shares their obtained dataset, pre-trained models and code for additional study of transfer learning in the NLP domain.

[8] DistilBERT holds 40% less parameters than BERT and is faster and less costly to deploy at runtime without compromising much on the quality. While it seems to handle tasks such as GLUE and SQuAD, one has to fine-tune the model with more detail as compared to other full-sized models. While it provides improved efficiency, it is, in most cases, less accurate than the full BERT models.

[9] PRIMERA is a pre-trained model for multi-document summarization which reduces the complexity to tailor the overarching structure as well as complex optimization. It uses a new pre-training task to adequately relate and summarize information from different documents in a Work. Built on the powerful encoder-decoder transformers, PRIMERA makes the manipulation of concatenated documents easy. Experimental results on six widely-used multi-document summarization benchmarks from different domains coupled with extensive ablations show that PRIMERA achieves state-of-the-art performance in zero-shot, few-shot, and fully supervised manners. The code file and the pre-trained models are provided below for the further use and analysis.

[10] This paper gives an account of the latest developments in Multimodal Large Language Models (MLLMs) such as GPT-4V that incorporates a robust language model to solve the tasks that involve interaction with semantic representations of different modes. It shows the strength of MLLMs like; ability to create stories from images and other complex reasoning other than OCR. This work presents a survey of the MLLM architectures, training methodologies, and assessment techniques. It also talks about how to extend MLLMs for different modalities and languages, and how to address issues of Multimodal

Hallucination and reasoning which includes, the Multimodal In-Context Learning (M-ICL) and LLM-Aided Visual Reasoning (LAVR). The last section of the paper presents the current issues and research potentialities in the study.

3. Problem statement

Develop and compare a system that automates text processing by completing two functions; summarizing text using a general language model and comparing its effectiveness, with specialized summary models; and categorizing content using BERT in comparison to other advanced models available today for benchmark purposes.The goal is to pinpoint the models for each task based on their accuracy ,speed and applicability, to real world content processing needs. This examination aims to identify the trade-offs between specialized models when it comes to achieving the outcomes for summarizing and categorizing content.

4. Objectives / Major Contributions

Evaluate Text Summarization Models: Evaluate how well the generic LLM does in producing summarization and how the results are like the setting up of specialized summarization varieties, BART, and Pegasus. The assessment criteria will be classification of summaries, their coherence as well as relevance of the summary to the topic in question.

Compare Content Tagging Models: Analyze how well BERT works for content tagging and compare it to other state-of-art classification models including; RoBERTa, DistilBERT, and XLNet. This paper will seek to compare the different models with a view of determining which of the models will be best suited for tagging news content.

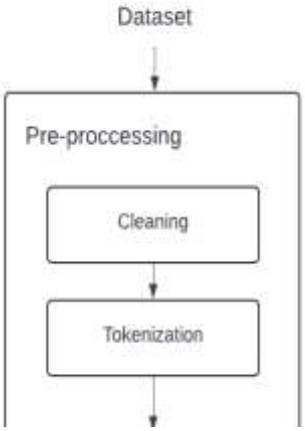
This is in a view to helping the research determine the suitability of these models for the two NLP tasks; summarization and tagging. Thus, the goal of the work is to compare general-purpose and specialized models to outline recommendations for practical text processing in the context of news aggregator, content management system, and/or information retrieval application.

5. Proposed Methodology/ Framework/ Model

The goal of this research is to. The technique consists of several critical elements that must be completed in order to ensure a solid and systematic approach to the problem.

5.1 Summarization:

A. Proposed System Architecture



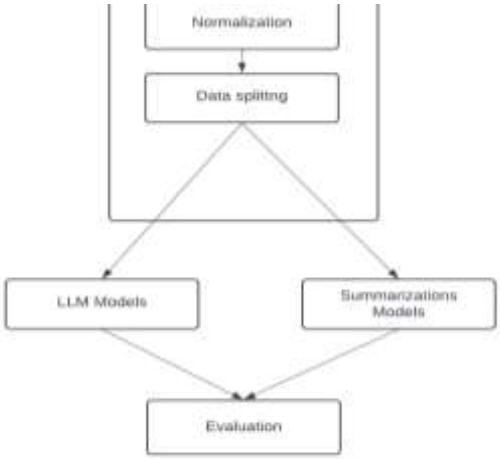


Figure 5.1: Process Diagram for Summarization Evaluation

B. Dataset

Our model was evaluated using three widely recognized datasets: XLSum dataset [11], XSum [2], and CNN/ Daily Mail [2] The description for these datasets that were applied for training and evaluation:

- XLSum: It's a summarization dataset of Indian-language news articles with multilingual support.
- XSum: Has highly extreme summarization data, where one sentence acts as a summary for each document.
- CNN/Daily Mail: probably the most famous dataset for news summarization where one actually uses the articles with their summaries.

These datasets provide varied samples of texts, which helped increase generalization capability across different types of content for the various models, sometimes coupled together to improve range.

C. Pre-processing

Pre-processing involved several steps in text data preparation:

- Text Cleaning: Removing HTML tags, special characters, and an excessive number of white spaces.
- Tokenization: This process divides the text into smaller units that are more friendly to processing.
- Normalization: The act of transforming a text into standard form-for instance, converting all characters into lower cases.
- Handling Long Documents: For long documents, such models are split into pieces so that it can fit within the limits of the model used.

D. Data Summarization Models

We have tested many models with different advantages:

1. Transformer-based BART and BERT-based Models: Their models bart-large-cnn-samsung, bart-fine tuned-text-summarization, etc, rely on high-quality summaries coming through the attention mechanism.
2. Abstractive summarization models: These PEGASUS models like the pegasus_summarizer model are abstractive summarizers. It is set to give concise coherent summaries.
3. Multilingual Models: A model like mT5_multilingual_XLSum can take care of multiple languages for the same reason it is suitable for different datasets.
4. Long Document Models: The long former model update-summarization-bart-large-long former focuses on large documents with the proper application of attention mechanisms that efficiently scale

with the document length.

5. LLMs: The architecture for a text summarization system using various LLMs and techniques used like Random Splitting and stratified sampling. We experimented with several state-of-the-art language models to compare for summarization tasks. The models chosen include T5, LLaMA, and Mixtral. T5, in particular the variant t5-small, is an all-purpose text-to-text model that has proven pretty effective in handling summarization by converting it into a generation problem of text. The LLaMA model, the one we tried with known as NousResearch/Hermes-3-Llama-3.1-8B, has shown impressive results on producing summaries of quality due to significant training on high volumes of data. Mixtral, with variant mixtral/mixtral-7b, is optimized towards being used in multi-task learning and cross-lingual applications, making the model eligible to work on a wide number of language-dependent summarization tasks. We apply summarization with these models through several steps: First, tokenization transforms the input text into the appropriate structure for running it through a model, and handling special tokens. The generation model takes the tokenized input and produces summaries from them using beam search, temperature sampling, and even length penalties to ensure quality and length in the summaries. Decoding finally transforms output tokens from the model into readable text by removing special tokens, and adding formatting, for coherent and accurate summaries. The comprehensive method allows for a robust evaluation of how well each model can actually generate effective summaries.

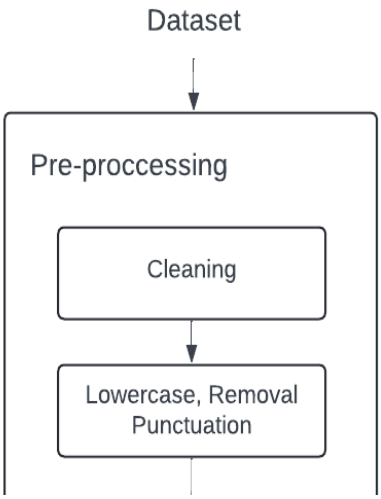
E. Evaluation Metrics

We evaluated LLM models using the Eleuther AI Language Model Evaluation Harness, which compares generative language models across various tasks. Our evaluation includes IFEval where this model is being tested on whether it would strictly follow explicit formatting instructions or not, that is how well it will follow directives to include certain keywords or use certain formats. BBH evaluates models on 23 challenging tasks extracted from the BigBench dataset which consist of complex arithmetic, algorithmic reasoning, language understanding, and world knowledge; their metrics are strongly related to human preferences. GPQA provides questions prepared by PhD-level experts in biology, physics, and chemistry to be challenging to a non-expert but accessible to an expert, with rigid validation to get them both difficult and correct.

5.2 Content Tagging

A. Proposed System Architecture

The fig shows the architecture of the proposed system for - content tagging



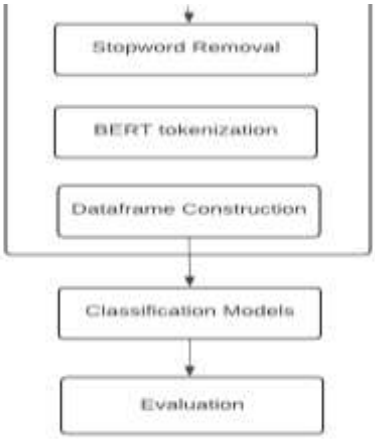


Figure 5.2: Process Diagram for Content Tagging Evaluation

B. Dataset

The dataset [12] contains 1 million news articles from 2000 sources over 4 large categories and is used widely for text classification techniques. It is vastly used in text processing because of balanced representations and real-world relevance. Each class contains 30,000 training samples and 1900 testing ones. The classes consist of World, Sports, Business, Science/Technology.

C. Pre-processing

Several data preprocessing steps were carried out to prepare the dataset for analysis to make the text cleaner and standardized.

The documents are read from their respective files, as every file contains text coming from a single newsgroup. And then the preprocessing pipeline follows:

- Tokenization: The Description text is broken up into individual words, or tokens.
- Lowercase and Removing Punctuation: Convert tokens to lowercase and remove non-alphanumeric tokens.
- Stopword Removal: Removing common stop words like "the," "and," and "is" from it to reduce the noise of the data.
- Stemming: Usage of Porter Stemmer which reduces words to their root form.
- BERT Tokenization: The BERT tokenizer cleans and stems the text, tokenizing and encoding the phrases into numerical representations for input and also
- Dataframe Construction: We create a DataFrame that holds the title of documents
  along with its corresponding preprocessed content and class labels. We then convert our dataset to TensorFlow tensors for both training and evaluation.

D. Data visualization

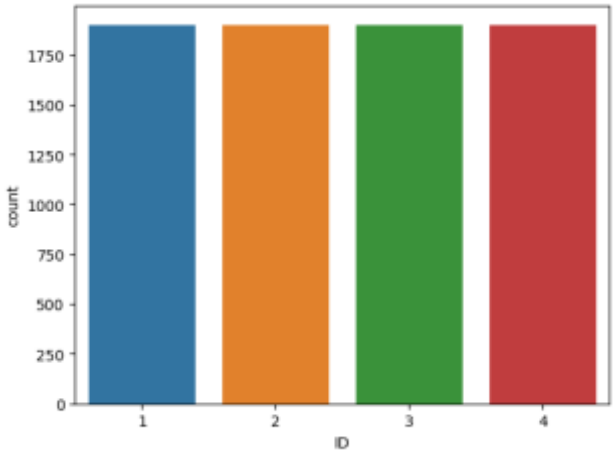


Figure 5.3: Countplot of ID column in Test Data

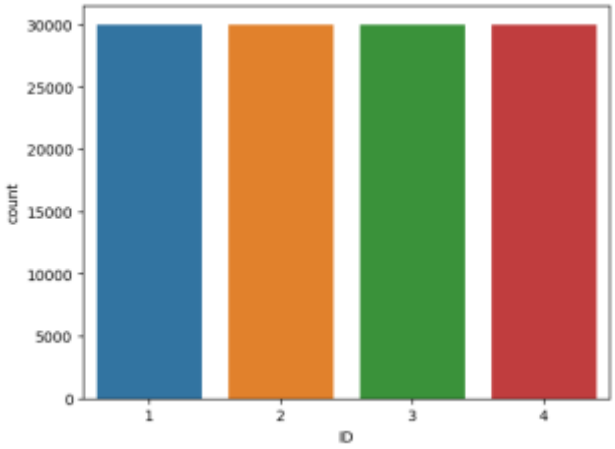


Figure 5.4: Countplot of ID column in Train Data

E. Data Splitting

The dataset is broken down into training and validation subsets as shown below:

- Shuffling: Done to assure the training and validation sets are representative of the whole dataset.
- Training and Validation Split: An 80-20 split is used , This helps to evaluate performance on unseen data.

F. Classification

We tried several classifiers for document classification: Multinomial Naive Bayes, Decision Trees, Gaussian Naive Bayes, SGD (Stochastic Gradient Descent), LightGBM, and Random Forests. But a normal BERT model outperformed the rest.

The BERT model was used to extract contextual embeddings. Input layers include input_ids and attention_mask provided with BERT embeddings from the pooled output. The output passes through a dense layer followed by ReLU activation, which captures the deeper features. Then it goes for a dropout layer at a 20% rate to prevent overfitting. Its output will pass through a dense layer with softmax activation to classify a document into one out of 10 categories. The model uses the Adam optimizer with a learning rate decay schedule, and categorical cross-entropy is used as the loss function.

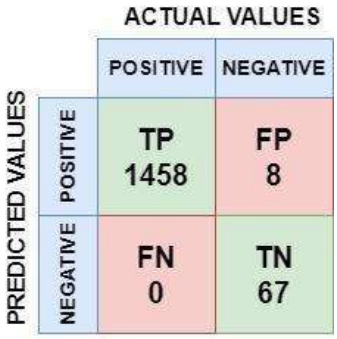


Figure 5.5: Countplot of ID column in Train Data

Results and Discussions

The metrics obtained through our detailed testing pipeline and evaluation provide a comprehensive assessment of the model's performance in classifying instances.

In this case, the BERT model demonstrates high accuracy, precision, recall and F1-Score, indicating its effectiveness in distinguishing between positive and negative cases.

	Accuracy	Precision	Recall
Bert	0.9315	0.9393	0.9263
Decision Tree	0.8173	0.8164	0.8270
Gaussian NB[13]	0.4615	0.4844	0.4687
SGD	0.6923	0.6990	0.7008
LGBM	0.8461	0.8457	0.8501
Random Forest	0.8461	0.8481	0.8553
Multinomial NB	0.4375	0.5096	0.4425

Table 5.1: Classification models comparison

	ROUGE-1	ROUGE-Lsum	ROUGE-L
Bart-large-cnn-samsum (xsum) [2]	41.3174	38.4149	32.1337
Pegasus_summarizer (cnndaily mail) [1]	36.604	32.902	23.884
mT5_multilingual_XLSum [7]	36.500	28.996	28.988

distilbart-xsum-12-6 (xsum) [8]	44.2553	36.2696	36.2639

Table 5.2: Summarization specific comparative analysis

	IFEval	BBH	GPQA
mistralai/Mixtral-8x22B-Instruct-v0.1	71.84	44.11	16.44
NousResearch/Hermes-3-Llama-3.1-8B	61.7	30.72	6.38
flan-t5-small	15.24	6.36	1.45

Table 5.3: LLMs models performance

The paper compares general-purpose large language models like T5 and LLaMA with specialized architectures like BART, PEGASUS, and BERT for tasks including text summarization and content tagging. While previous work, such as PEGASUS and BART, focuses more on the optimization of abstractive summarization, and models such as RoBERTa improve BERT for classification, this project underlines the efficiency of general LLMs for both summarization and tagging alike. It is an effort to add value based on a better understanding of the trade-off between general models and task-specific models regarding accuracy, coherence, speed, and the amount of computation needed, which can lead to identifying the most useful models for the challenges facing practical real-world NLP tasks.

Conclusion

As seen in the blossoming field of NLP, every model contributes to different tasks that include summarization, text generation, translation, as well as classification in the current world. Works like PEGASUS use transformer based pre-training with gap sentence generation, BART employs denoising autoencoder pretraining and Longformer proposes the extended attention span techniques for the enhancement of the summarization breakthrough. In contrast, generation models like GPT 4, CTRL and BlenderBot 3; employ large scale transformer and control codes for the regulation of the generation of text in the task oriented strategy. mBART and Marian NMT used for translation models which provide multilingual analysis and fast neural machine translation; BERT, RoBERTa,ELECTRA improve the performance in terms of strengthened pre-training method and selected masked language Models.

References

1. Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, November). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. International conference on machine learning (pp. 11328-11339). PMLR.
2. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

3. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140), 1-67.
4. Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.
5. Xiao, W., Beltagy, I., Carenini, G., & Cohan, A. (2021). PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. arXiv preprint arXiv:2110.08499.
6. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.
7. Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., ... & Joulin, A. (2021). Beyond english-centric multilingual machine translation. Journal of Machine Learning Research, 22(107), 1-48.
8. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
10. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.
11. Hasan, Tahmid, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar.(2021) XL-sum: Large-scale multilingual abstractive summarization for 44 languages.arXiv preprint arXiv:2106.13822.
12. Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems 28 (NIPS 2015).
13. Kumar, S., Gulati, A., Jain, R., Nagrath, P., Sharma, N. (2021). Categorizing Text Documents Using Naïve Bayes, SVM and Logistic Regression. In: Sharma, N., Chakrabarti, A., Balas, V.E., Martinovic, J. (eds) Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing, vol 1175. Springer, Singapore. https://doi.org/10.1007/978-981-15-5619-7_14