Enhancing Recruitment Drive Query Summarization: Benchmarking Models

DR. GEETANJALI KALE[1], PROF. PRANALI RAJENDRA NAVGHARE[2], ATHARVA SADANANLITAKE[3], APURVA AJIT KULKARNI[4], KISHANLAL CHHELARAM CHOUDHARY[5], ADITYA DARADE[6]

[1, 2, 3, 4, 5, 6]*Computer Engineering, Pune Institute of Computer Technology*

Abstract- ***Campus recruitment is an important life phase from a student perspective. Not only students, but the institute conducting the process also aspires to be the top institute providing recruitment to various talented minds across the globe. Training and placement cell from various institutes manages this process with the help of various authoritative persons. For every organization recruiting through campus placement drive, the process starts with registration and ends with the successful hiring of the candidates. During this process, students face various difficulties and complications at every stage of recruitment drive. Every authoritative person receives multiple queries to address the difficulties faced by the students. Leveraging text summarization to summarize the queries and resolve the difficulties in a short span of time proves to be useful for both the parties. This paper encompasses the use of various Large Language Models for summarizing the queries and benchmarking them against the standard metrics.***

Indexed Terms- ***- Text Summarization, Large Language Models, Queries, BERT.***

I. INTRODUCTION

In recent years, Large Language Models (LLMs) have shown remarkable advancements in natural language understanding tasks, including text summarization. Summarization plays a crucial role in condensing lengthy and information-dense text into concise and meaningful representations, particularly in student-facing systems like Training and Placement (TNP) portals, where effective communication is vital. This paper presents a comparative study of several state-of-the-art LLMs—ChatGPT 4o, Claude 3.7 Sonnet, Deepseek R1, Gemini 2.0 Flash, Llama 4, and Mistral 7B—by evaluating their summarization performance on a custom-created dataset of real-world student queries across ten distinct categories. We employ BERTScore as our primary evaluation metric to measure the semantic similarity between model-generated summaries and human-written references, enabling us to assess the strengths and limitations of each model in a practical academic support context.

II. LITERATURE SURVEY

Beginning in the early 2000s, the research on text summarization has been rapidly growing until today, both using extractive and abstractive approaches. Text Summarization research papers explore a variety of methodologies for summarization based on statistical, machine learning, and computational approaches. The advent of large language models (LLMs) has greatly improved the situation.

Mark Dredze and co-authors [6] explain the greater use of email has arisen due to heightened connectedness among people, and as the volumes of email grow, managing all these emails by hand will be increasingly difficult. To do this, they consider an unsupervised-learning-based method to summarize emails into keywords automatically with latent topic models. The proposed brief summary does a good job of encompassing what is discussed in the emails. The paper focuses on applying Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) as the topic model. The authors develop and evaluate four variants of the model - LDA-doc, LSA-doc, LDA-word, and LSA-word, and they compare performance to a TF-IDF baseline. The authors conduct their study using the publicly available Enron email dataset (25,000 emails across 150 unique users).

A summarization model was introduced by Mohamed Abdel Fattah and collaborators [7], which could be trained to achieve the highest-value sentences based on multiple characteristics, for example, position, occurrence of a keyword, length and centrality. The authors used Feature Extraction, Genetic Algorithm (GA) and Mathematical Regression (MR) to rank the sentences, and these features were used to assess the importance of the sentence. The GA process updated the characteristics using natural selection, while the MR described a perceived value to the importance of a feature in relation to the value of a human summary. The authors trained the summarization model using 50 religious texts that were manually summarized, and tested it against 100 religious articles written in English. All of the items came from the Internet Archive.

Rahim Khan and co-authors [12] have implemented clustering methods for extractive text summarization, specifically K-means clustering and TF-IDF. The goal was to lessen the burden of large texts, as well as extract important sentences for automatic summarization. Simple techniques were included with their study, along with preprocessing (including tokenization and stemming). The Elbow method and Silhouette method were utilized to find proper K values, grouping according to sentence importance based on TF-IDF scores, and they demonstrated relative performance against other summary techniques with respect to precision metrics. On news articles, their methods produced higher accuracy in summaries, and their study produced summaries that were readable.

Jingqing Zhang and others [19] propose PEGASUS, a model to examine the pretraining of an encoder-decoder model for abstractive summarization. There has been significant work in extractive summarization, but not quite as much work in the abstractive variety. The new PEGASUS proposes two new pre-training objectives called "Gap Sentence Generation" (GSG) and "Masked-Language Modeling" (MLM) that pretrain a transformer-based encoder-decoder. In pretraining, a vector of the original input sentences is masked and is then used to generate a summary of the input in the summary generation. This system has been assessed using twelve datasets across a range of domains and shows its efficacy according to human evaluations of the system.

Mihir Sanghvi. and co-authors [2] researched and evaluated the summarization performance of three large language models (LLMs) including text-davinci-003, mpt-7b-instruct, and falcon-7b-instruct, evaluating each model based on 25 samples. This research further states that the two datasets used in the study were CNN/Daily Mail 3.0.0 and XSum, in addition to the study conducting evaluations on three total metrics: BLEU, ROUGE, and BERTScore. Overall, the study discusses their findings on size and overall architecture of the model and impressions on summarization quality. The study recommends the OpenAI model as the best overall model for use in summarization tasks.

III. METHODOLOGY

The following section describes the process used to assess the performance and compare a range of state-of-the-art Large Language Models (LLMs) for text summarization on a custom dataset sourced from real-world student questions. The capability of each model to summarise and combine content is evaluated using the BERTScore metric.

1. Dataset Description

We have developed our own dataset made from 10 different types of student queries from the Training and Placement (TNP) system portal. The datasets represent real-life issues and challenges encountered by students during their placements on the campus. The datasets contain the following:

Dataset 1 - Account access - Login issues, account clearance, general system issues.

Dataset 2 - Profile changes - Issues with editing academic and personal information and uploading resumes.

Dataset 3 - Document submissions - Uploading issues, acceptable file types or follow up protocols, and approval timelines.

Dataset 4 - Applications status - Perceived issues, changes in the status of applications, clarifying record checks and communication breakdowns.

Dataset 5 - Technical issues with tests and interviews - Crashes, issues with timers, and consideration of

evaluations of transferable skills in tests or interviews.
Dataset 6 - Placement process queries - Queries related to policies around, and concerns about eligibility, timelines, and processes of offer.
Dataset 7 - Company specific queries - In relation to programming languages, relocations or reimbursement for relocations and compensation.
Dataset 8 - Queries about internships - In relation to the processes for internships, PPOs, and complexities of internships that would require schools to ask questions.
Dataset 9 - Concern following placement - In relation to post-acceptance offer experiences, onboarding delays, and team mismatches.
Dataset 10 - General TNP related queries - General queries for interview etiquette, placement statistics, and processes for international students.
Data points in each dataset are made up of a student question and a human produced reference summary. These were used for comparison of input/output of the model.

2. Model Selection
We examined the following list of LLMs, all of which allow for abstractive summary. Here is a brief description of each:

ChatGPT-4o (OpenAI): A multimodal, faster and more efficient version of GPT-4 that works in its own unique way to summarise with great coherence for longer context.
Claude 3.7 Sonnet: A model that demonstrates a strong model performance noting long context window and less risk related to summary inaccuracy and unsafe behaviours.
DeepSeek R1: An innovative open-source model optimized for reasoning and summarization that is able to produce competitive performance with little Christian-style fine-tuning.
Gemini 2.0 Flash): This lightweight Gemini model is optimized for speed and efficiency with attention to quality and reduced inference time.
LLaMA 4: This is Meta's latest LLM with a deeper understanding of natural language and suitable for both short- and long-form summarization.
Mistral 7B: A lightweight, open-weight model trained on a mixture of internet data that produces high-quality outputs and is efficient.
Each model was used originally in a zero-shot or minimally prompted condition mimicking real-world use, which has to some extent lessened the reliance on significant fine-tuning.

3. Summarization & Evaluation Strategy
Each of the 10 custom datasets was supplied as input to each chosen LLM to generate summaries for our experiment. This allowed for all models to be evaluated based on the same set of actual student queries, across different issue categories. The generated summaries were subsequently compared through the use of BERTScore, a measurement of similarity based on contextualized word embeddings.

4. Experimental Setup
The following experiments to test were conducted on GPU systems, while using libraries for Hugging Face Transformers, PyTorch, and OpenAI/Anthropic APIs. In the interest of fairness:

- All models received the same input.
- Prompt templates (where applicable) were standardized.
- Token limit, and decoding parameters, were tuned but were made uniform across the runs.

IV. RESULTS

We created a dataset that contained queries about training and placement, then applied multiple Large Language Models (LLMs), and examined their performance of the summarization, which also included overall criterion scores for summarization. The dataset was individually queried, and we enrolled the scores and scored their performance. The dataset are as follows:

- Sample 1 - Account Access Queries
- Sample 2 - Profile Update Queries
- Sample 3 - Document Submission Queries
- Sample 4 - Application Status Inquiries
- Sample 5 - Test and Interview Technical Queries
- Sample 6 - Placement Process Questions

The comparative bar graphs illustrating the performance scores of different LLM models on each dataset are presented below.

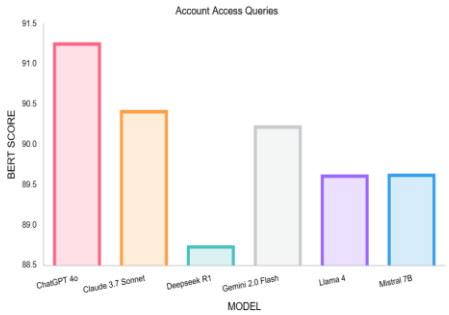

Figure 1. Comparative performance of various models

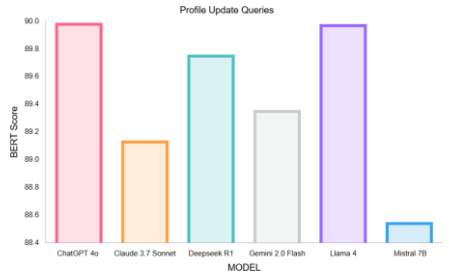

Figure 2. Comparative performance of various models

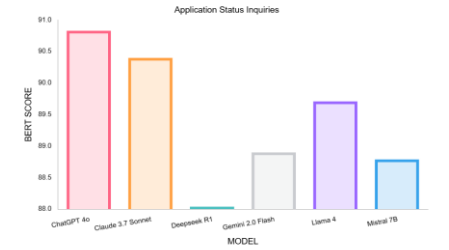

Figure 3. Comparative performance of various models

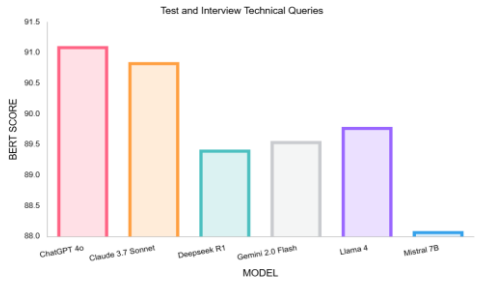

Figure 4. Comparative performance of various models

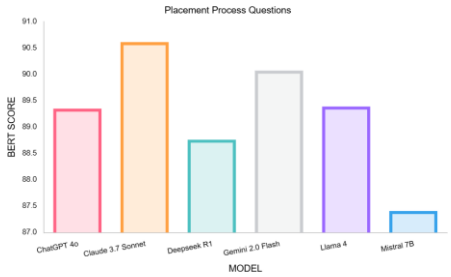

Figure 5. Comparative performance of various models

	ChatGPT 4o	Claude 3.7 Sonnet	Deepseek R1
Sample 1	91.27	90.43	88.75
Sample 2	89.99	89.14	89.76
Sample 3	89.58	90.69	89.25
Sample 4	90.83	90.4	88.04
Sample 5	91.11	90.85	89.42
Sample 6	89.35	90.61	88.76
Sample 7	89.9	89.25	88.15
Sample 8	89.51	89.65	89.41
Sample 9	90.31	90.64	89.42
Sample 10	91.61	91.47	89.49

Table 1. Comparative performance of various models

	Gemini 2.0 Flash	*Llama 4*	*Mistral 7B*
Sample 1	90.24	89.63	89.64
Sample 2	89.36	89.98	88.55
Sample 3	88.63	89.72	89.95
Sample 4	88.9	89.71	88.79
Sample 5	89.56	89.79	88.09

Sample 6	90.07	89.39	87.41
Sample 7	89.07	88.76	88.15
Sample 8	89.14	88.56	89.92
Sample 9	89.28	89.36	88.01
Sample 10	90.2	88.09	87.46-

Table 2. Comparative performance of various models

BERTScore is a metric utilized for evaluating the quality of generated text such as summarization, by measuring semantic similarity instead of surface level identical words. Traditional metrics like ROUGE or BLEU often rely on whether the text surface level (1 gram or n-gram) is a match, while BERTScore utilizes contextual embeddings from transformer-based models (i.e., BERT and RoBERTa models). In particular, BERTScore is a metric that quantifies similarity by contextualizing both the generated summary (candidate) and the reference summary to produce contextualized token embeddings. Next, cosine similarity calculates the similarity between each token of the generated summary in the candidate and all tokens in the reference summary in relationship to one another, and ranks the best matches according to their similarity. Finally, BERTScore calculates precision, recall, and F1 to derive an overall facet of how closely a candidate captured the meaning of the reference. BERTScore is useful for evaluations of summaries that employed paraphrasing or substitution of different synonymous words, while maintaining the same key idea and meaning. When conducting an evaluation of multiple LLMs to develop summative, BERTScore offers a more specific, meaning-aware level of evaluation than traditional lexical measures, although it still relies on sequential token matches, therefore becoming more involved than a traditional evaluation of bigrams or trigrams. It is a commonly selected metric for component analysis in recent NLP studies.

Average BERT scores by Model

Formula:

Model	Average BERT Score
ChatGPT 4o	90.61
Claude 3.7 Sonnet	90.23
Deepseek R1	88.94
Gemini 2.0 Flash	89.45
Llama 4	89.30
Mistral 7B	88.21

Table 3. Average BERT score of models

ChatGPT 4o is the best-performing model overall, and has the best average BERT F1-score (90.61). It consistently outperformed the others across a number of datasets.

CONCLUSION

In this study, we evaluated several large language models in detail: ChatGPT 4o, Claude 3.7 Sonnet, Gemini 2.0 Flash, and Llama 4, Deepseek R1 and Mistral 7B, by text summarization performance on 10 datasets. Each model summarized the same queries and we evaluated each model's summarization against human reference summaries using BERTScore, a useful method of semantic similarity. The results suggest that while the performance of individual models was similar for summarization, ChatGPT 40 provides the most appropriate tradeoff between precision and semantic accuracy across different types of context as of this time. This assessment provides useful information to researchers and developers who want to select unimodels or multimodal language models for tasks that require quality abstractive summarization

REFERENCES

[1] Alsaedi, N., Burnap, P., & Rana, O. (2016). Automatic summarization of real world events using twitter. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 10, No. 1, pp. 511-514).

[2] Basyal, Lochan, and Mihir Sanghvi. "Text summarization using large language models: a

comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models." arXiv preprint arXiv:2310.10449 (2023).

[3] Bogireddy, Srinivasa Rao, and Nagaraju Dasari. "Comparative analysis of ChatGPT-4 and LLaMA: Performance evaluation on text summarization, data analysis, and question answering." 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE, 2024.

[4] Hernández-Castañeda, Á., García-Hernández, R. A., Ledeneva, Y., & Millán-Hernández, C. E. (2020). Extractive automatic text summarization based on lexical-semantic keywords. IEEE Access, 8, 49896-49907.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. 2019 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol., Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Assoc. Comput. Linguistics, Jun. 2019, pp. 4171–4186.

[6] Dredze, M., Wallach, H. M., Puller, D., & Pereira, F. (2008, January). Generating summary keywords for emails using topics. In Proceedings of the 13th international conference on Intelligent user interfaces (pp. 199-206).

[7] Fattah, M. A., & Ren, F. (2008). Automatic text summarization. World Academy of Science, Engineering and Technology, 37(2), 192.

[8] Feng, X., Feng, X., & Qin, B. (2021). A survey on dialogue summarization: Recent advances and new frontiers. arXiv preprint arXiv:2107.03175.

[9] Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. arXiv preprint arXiv:1906.04165.

[11] Jin, H., Zhang, Y., Meng, D., Wang, J., & Tan, J. (2024). A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. arXiv preprint arXiv:2403.02901.

[12] Khan, R., Qian, Y., & Naeem, S. (2019). Extractive based text summarization using k-means and tf-idf. International Journal of Information Engineering and Electronic Business, 12(3), 33.

[12] Langston, O., & Ashford, B. (2024). Automated summarization of multiple document abstracts and contents using large language models. Authorea Preprints.

[13] Mao, Yingjie, et al. "Automated smart contract summarization via llms." arXiv preprint arXiv:2402.04863 (2024).

[14] Majić, Antonela. Comparative Analysis of Abstractive Text Summarizers. Diss. University of Zagreb. Faculty of Humanities and Social Sciences. Department of Linguistics, 2025.

[15] W. Luo, F. Liu, Z. Liu, and D. Litman, "Automatic summarization of student course feedback," in Proc. 2016 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol., K. Knight, A.Nenkova, and O. Rambow, Eds. San Diego, California: Assoc. Comput. Linguistics, Jun. 2016, pp. 80–85.

[16] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in EMNLP, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Assoc. Comput. Linguistics, Oct. 2014, pp. 1532–1543.

[17] Yang, Xianjun, et al. "Exploring the limits of chatgpt for query or aspect-based text summarization." arXiv preprint arXiv:2302.08081 (2023).

[18] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," Inf. Process. & Manage., vol. 41, no. 1, pp. 75–95, 2005, an Asian Digital Libraries Perspective.

[19] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (1912). Pegasus: pre-training with extracted gap-sentences for abstractive summarization (2019). arXiv preprint ArXiv:1912.08777.

[20] Zhang, M., Li, X., Yue, S., & Yang, L. (2020). An empirical study of TextRank for keyword extraction. IEEE access, 8, 178849-178858.

[21] Zhang, H., Yu, P. S., & Zhang, J. (2024). A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models. arXiv preprint arXiv:2406.11289.

[22] Zhang, Y., Ni, A., Yu, T., Zhang, R., Zhu, C., Deb, B., ... & Radev, D. (2021). An exploratory study on long dialogue summarization: What works and what's next. arXiv preprint arXiv:2109.04609.