LLMs for AEC Code and Spec Review

Advik Mehta, Anant Bhide, Falak Sethi, Hanzhe Ye, Shreyank Hebbar

Department of Computing Science
University of Alberta

{advik, bhide, fsethi, hanzhe, shebbar}@ualberta.ca

Abstract

This paper presents an investigation into the potential of Large Language Models (LLMs) to accurately interpret and summarize technical documentation pertinent to the Architecture, Engineering, and Construction (AEC) sector. By fine-tuning foundational models with prompt-response pairs derived from technical documents, such as the ASME B31.3 Process Piping document and employing a Retrieval-Augmented Generation (RAG) architecture for sourced retrieval, this study aims to demonstrate the effectiveness of LLMs in navigating and extracting information from complex technical texts. The methodology encompasses the creation of a benchmark suite of questions alongside target answers to evaluate the model's accuracy. Experiment results, including data analysis, model performance, and statistical tests, are presented to underscore the capabilities and limitations of the proposed approach. This work seeks to lay the groundwork for developing domain-specific language models that can serve as sophisticated tools in the AEC field, enhancing accessibility to technical information and facilitating decision-making processes.

1 Introduction

1.1 Background

The Architecture, Engineering, and Construction (AEC) industry increasingly relies on comprehensive and intricate technical documentation to ensure its projects' precision, safety, and compliance. Navigating these documents can be a time-consuming and error-prone process due to their complexity and volume. Recent advances in Large Language Models (LLMs) have opened new avenues for automating the interpretation and summarization of such documents, promising significant improvements in efficiency and accuracy.

1.2 Motivation

The motivation for this study arises from the critical need to improve the accessibility and understandability of technical documentation in the AEC sector. Given the high stakes involved in architectural and engineering projects, errors or misinterpretations of technical standards and regulations can lead to costly and potentially hazardous outcomes. Traditional methods like document indexing or keyword search that we tested fail in such scenarios because most queries are complex and it is required to understand their semantic meaning. For instance, a user query might need context from multiple locations in source documents and traditional indexing fails to either retrieve all content of importance, or can not understand the semantic meaning behind certain terms used. LLMs, particularly those fine-tuned on domain-specific datasets, offer an unprecedented opportunity to minimize these risks by providing accurate, concise, and readily accessible interpretations of complex texts. Furthermore, this research is driven by the broader goal of exploring the feasibility of creating a domain-specific language model tailored to the AEC industry's unique requirements, thereby setting a precedent for similar applications in other specialized fields.

This paper builds on the hypothesis that LLMs can be effectively fine-tuned and utilized with an RAG architecture to interpret and summarize technical documents within the AEC domain. By detailing our methodology, presenting our experimental results, and discussing the implications of our findings, we aim to contribute valuable insights into the potential of language models to transform how technical documentation is engaged within the industry.

2 Related Work

This research builds upon a rich foundation of advancements in large language models (LLMs), retrieval-augmented generation (RAG), and fine-tuning methodologies to tailor LLMs for domain-specific applications. Our foundational model is the Mistral-7B OpenOrca, optimized for generative tasks, which serves as the base LLM for our initial experiments and adaptations (OpenOrca, 2023).

2.1 Large Language Models in Domain-Specific Applications

Recent developments in the field have increasingly focused on adapting large pre-trained language models to specific domains, enhancing their applicability to specialized fields. A pertinent example within the Architecture, Engineering, and Construction (AEC) sector is detailed in "GPT Models in Construction Industry: Opportunities, Limitations, and a Use Case Validation" (ScienceDirect, 2023), which explores the utility and challenges of implementing GPT models for construction project management. This study underscores the transformative potential of fine-tuned LLMs in improving domain-specific information retrieval and decision support, aligning closely with the objectives of our research.

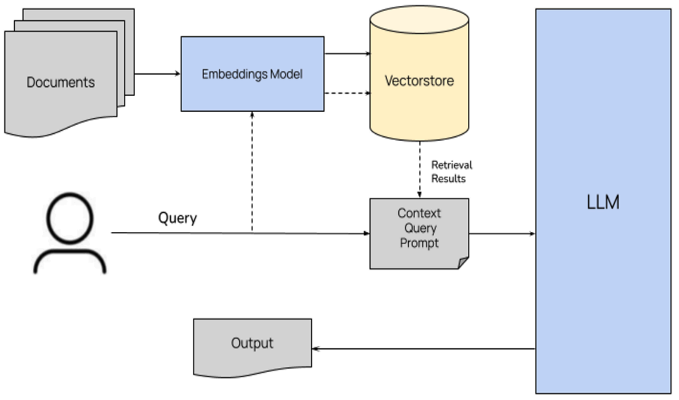


Figure 1: The RAG architecture. Source: https://llmstack.ai

2.2 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) represents a pivotal enhancement in the application of Large Language Models (LLMs) within technical and specialized fields such as the Architecture, Engineering, and Construction (AEC) industry. This section delves into the intricate mechanisms of RAG, specifically focusing on how documents are retrieved through similarity search (Lewis et al., 2021) in a vector database and subsequently integrated into the language model's generative process.

First, RAG includes transforming documents into high-dimensional vectors and storing in a vector database. When a query is received, the system converts it into a vector and performs a similarity search to retrieve the most relevant documents. This is typically achieved using the k-nearest neighbors (k-NN) algorithm. The retrieved document snippets are then used to construct an enriched prompt that combines the original query with contextual information from these documents. This enriched prompt is fed into the LLM, which generates a response that is not only contextually relevant but also informed by specific details from the retrieved texts.

2.3 Fine-Tuning Techniques

Fine-tuning is an essential technique for adapting Large Language Models (LLMs) to specialized domains (Hu et al., 2021). The fine-tuning process involves modifying a pre-trained model so it can better understand and respond to the specific data and requirements of a targeted field. A key method used in this process is LoRa (Low-Rank Adaptation), which allows for efficient and effective fine-tuning. LoRa introduces trainable low-rank matrices to the transformer layers of LLMs. This adaptation focuses primarily on the self-attention mechanisms, which are pivotal in processing and generating responses based on the input data. By adjusting these matrices, LoRa modifies the model's internal representations to align more closely with domain-specific content.

2.3.1 Fine-Tuning Process with LoRa

Dataset Preparation: The fine-tuning process starts with creating a dataset derived from key AEC documents, such as the ASME B31.3 process piping standards. This dataset is converted into prompt-response pairs that reflect real-world AEC scenarios.

Here is an example of the prompt-response pairs used for fine-tuning:

- ***Prompt:*** *What is the allowable stress for occasional loads of short duration for materials other than those with non ductile behavior in Elevated Temperature Fluid Service?*
- ***Response:*** *The allowable stress for occasional loads of short duration for materials other than those with non ductile behavior in Elevated Temperature Fluid Service is the strength reduction factor times 90% of the yield strength at temperature.*

Model Adjustment: Using LoRa, the LLM is fine-tuned on this dataset. The low-rank matrices are adjusted to refine the model's focus on terms and structures relevant to AEC tasks without overwhelming the model's general capabilities.

Training and Evaluation: The model undergoes a series of training epochs on this tailored dataset, with periodic evaluations to monitor improvements and ensure the model remains aligned with AEC-specific requirements.

2.4 Retrieval Augmented Fine Tuning (RAFT)

Building further on these methodologies, our research employs the RAFT framework, which synergizes retrieval-augmented techniques with fine-tuning adaptations (2024, RAFT). This hybrid approach not only retrieves relevant information but also integrates this knowledge seamlessly into the generation process, specifically tailored to the needs of the AEC industry. By leveraging and integrating these works, our research aims to push the boundaries of what LLMs can achieve in specialized fields. This cohesive approach offers a sophisticated tool designed to enhance productivity and decision-making within the AEC industry, drawing on the strengths of each component to develop a domain-specific language model capable of effectively interpreting and summarizing complex technical documentation.

3 Methodology

This study evaluates the performance of various configurations of the Mistral-7B OpenOrca model, tailored to enhance its ability to interpret and summarize technical documentation within the Architecture, Engineering, and Construction (AEC) industry. Our methodology involves a comparative analysis of four distinct model configurations:

- Base Mistral-7B OpenOrca model.
- Base Mistral-7B OpenOrca + Retrieval-Augmented Generation (RAG).
- Fine-tuned Base Mistral-7B OpenOrca on ASME B31.3 documentation.
- Fine-tuned Base Mistral-7B OpenOrca on ASME B31.3 documentation + RAG.

The foundational experiment deployed the Mistral-7B OpenOrca model in its pre-trained state to establish a baseline, assessing the enhancement effects of subsequent configurations.

3.1 Incorporation of Retrieval-Augmented Generation (RAG)

The second configuration introduced a RAG component to the base model, designed to dynamically retrieve relevant sections of external documents to inform the model's response generation, thereby enhancing accuracy and context relevance.

3.2 Fine-Tuning on Technical Documentation

The third configuration involved fine-tuning the base model on the ASME B31.3 process piping document. This adjustment refined the model's weights to better reflect the specific terminology, style, and content relevant to the AEC industry, improving domain-specific performance.

3.3 Integration of Fine-Tuning with RAG

Our most comprehensive configuration combined fine-tuning on ASME B31.3 documentation with the RAG component. This dual approach was designed to synergistically enhance domain-specific adaptation and contextually aware retrieval synergistically, aiming to maximize accuracy and relevance in response generation.

3.4 Dataset Preparation and Fine-Tuning

Automated Dataset Generation: Utilizing HelixAI, we automated the generation of over 1,000 high-quality prompt-response pairs from the ASME B31.3 document. These pairs simulate real-world queries and are intricately detailed, closely reflecting the document’s technical content.

Fine-Tuning Process: The dataset facilitated the fine-tuning of the Mistral-7B OpenOrca model using the LoRA technique, which introduces trainable low-rank matrices to adjust the self-attention layers of the transformer architecture, efficiently adapting the model while maintaining its general capabilities.

Training Duration: The fine-tuning was executed over a predetermined number of epochs to balance learning and overfitting.

Learning Parameters: Adjustments to learning rates and regularization strategies were optimized for the training process.

Evaluation During Training: Periodic evaluations using a validation set derived from the same dataset ensured the effectiveness of the fine-tuning.

3.5 Integration of RAG

Post-fine-tuning, the model was integrated with the RAG system, which dynamically retrieves information from the ASME B31.3 documentation in response to query inputs, thereby enriching the generation process with domain-adapted and contextually relevant information.

3.6 Experimental Setup

Each model configuration was evaluated using a standardized set of queries designed to span a broad range of topics and complexities within the ASME B31.3 document. Responses were assessed against a benchmark for accuracy, relevance, and conciseness.

3.7 Evaluation Metrics

To assess the effectiveness of each model configuration, we employed a comprehensive set of metrics:
Accuracy, Precision, Recall, and F1-Score: These standard metrics evaluate the correctness and relevance of the generated responses.
ROUGE Metrics: Including ROUGE-N for n-gram overlap, ROUGE-L for the Longest Common Subsequence, and ROUGE-S for skip-bigram co-occurrences, these metrics assess the textual overlap and structural similarity between the generated text and reference texts.
Cosine Similarity: This metric measures the semantic similarity between the generated text and reference texts, capturing the model's ability to paraphrase and retain the essence of technical content.
Composite f-score: A tailored metric that combines the above methods with variable weights, depending on whether the response requires fact-reporting or summarization, to provide a nuanced evaluation of text generation.
These diverse metrics ensure a robust evaluation, reflecting both the factual accuracy and the contextual relevance of the responses to the Architecture, Engineering, and Construction (AEC) sector.

4 Evaluation

The goal is to develop an evaluation metric, a similarity score between the model's answer and reference answer, that will help to compare all four approaches and decide which works best. We will use two evaluation schemes for calculating similarity scores: ROUGE and cosine similarity for semantics.

4.1 ROUGE

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations. It works by a string-to-string comparison of automatically produced summary or translation against a set of reference summaries provided by a domain expert. ROUGE[1] is associated with two aspects: recall and precision.

[1] Refer to Appendix A.1

$$\text{Recall} = \frac{\text{overlapping words}}{\text{words in reference answer}} \qquad \text{Precision} = \frac{\text{overlapping words}}{\text{words in model answer}}$$

4.2 Beta F-score: Combine precision and recall

$$F_\beta = \frac{(1+\beta^2) \cdot recall \cdot precision}{recall + \beta^2 \cdot precision} \qquad \beta = \frac{precision}{recall}$$

4.3 Cosine Similarity

Cos similarity is useful for semantic similarity between the answers. Convert the model and reference answers into vectors x and y. The dot product between the generated vectors gives cosine similarity.

$$\text{Cosine Similarity} = \cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

The smaller the angle between the vectors, the more similar the answers are.

4.4 Similarity F-score

We will design a balanced measure integrating ROUGE metrics and cosine similarity to evaluate our model approaches comprehensively. Define a function F that takes all necessary rouge F scores as well as cosine similarity scores and returns a weighted final F-score.

$$F = \begin{pmatrix} F_\beta(rouge1) & F_\beta(rouge2) & F_\beta(rouge3) & F_\beta(rougeL) & F_\beta(rougeS) & cossimilarity \end{pmatrix} \cdot \begin{pmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{pmatrix}^T$$

Weights will be adjusted according to the type of question. For fact questions, it is better to prefer rouge-1 and cos-similarity, w = [0.5, 0, 0, 0, 0, 0.5]. On the other hand, we assign the following weights for summary-based questions: w = [0.15, 0.15, 0.15, 0.15, 0.15, 0.25]

5 Experiments and Results

We performed statistical tests to show that the RAFT approach works best for this domain.

5.1 Experimental Design

Our experiments were designed to assess the effectiveness of the Mistral-7B OpenOrca model in various configurations: the base model, base model with Retrieval-Augmented Generation (RAG), fine-tuned on ASME B31.3 documentation, and fine-tuned with RAG integration. Each model was evaluated using a benchmark suite of technical queries from the AEC sector, reflecting realistic scenarios likely to be encountered in professional settings.

5.2 Data Collection

The dataset consisted of over 1,000 prompt-response pairs derived from the ASME B31.3 standards document. These pairs were carefully crafted to cover a wide range of technical questions and scenarios that test the model's domain-specific understanding and information retrieval capabilities.

5.4 Results

The results were quantitatively analyzed using a suite of metrics including accuracy, precision, recall, F1-score, ROUGE scores, and cosine similarity. Each model's performance was benchmarked against these metrics to determine its efficacy in producing accurate and relevant responses.

- **Base Model**: Showed competency in understanding general queries but lacked the precision in domain-specific terminology and concepts.
- **Base + RAG**: Improvement in contextual understanding was noted, with better handling of queries requiring detailed technical knowledge.
- **Fine-tuned Model**: Demonstrated a significant leap in accuracy and relevance, particularly for complex queries specific to AEC standards.
- **Fine-tuned + RAG**: This configuration yielded the highest scores across all metrics, confirming our hypothesis that the integration of fine-tuning with retrieval-augmented generation provides the most robust solution for domain-specific applications.

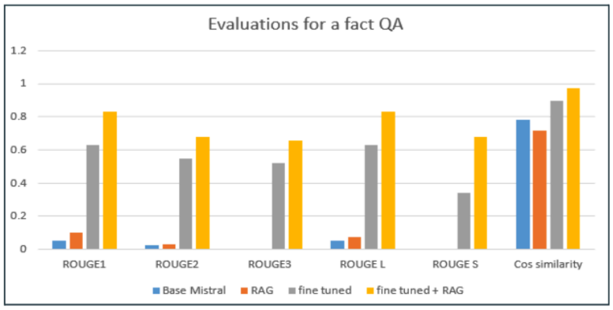


Figure 2.1: Model Evaluations for generating facts

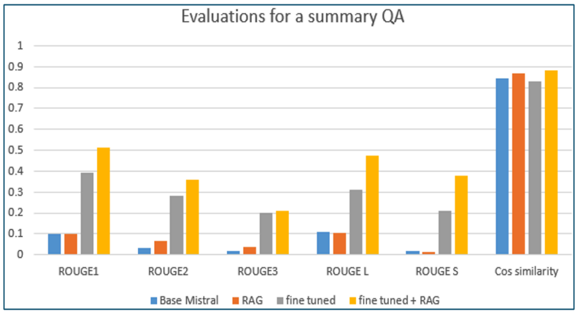
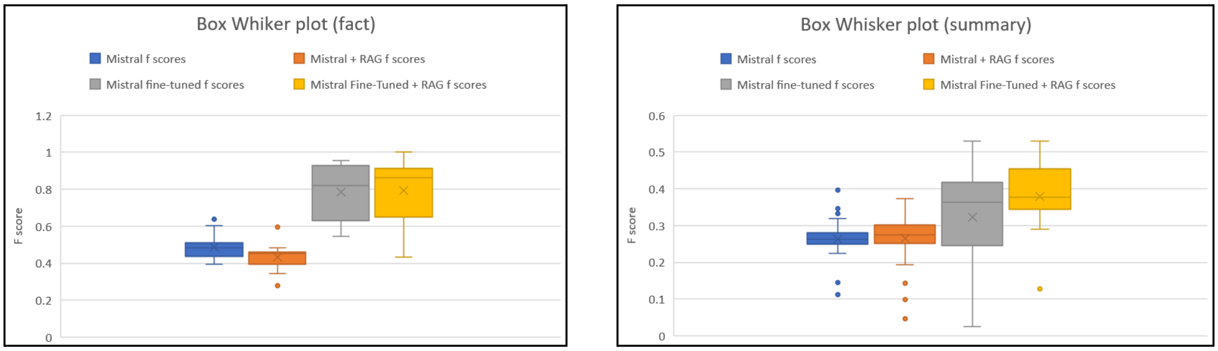


Figure 2.2: Model Evaluations for generating summaries

As shown in Figure 2.1, we observe that the model's performance is being assessed based on its ability to provide precise answers to fact-based questions. The 'fine-tuned + RAG' configuration appears to excel consistently across all metrics, particularly in ROUGE L and Cosine Similarity, indicating a high degree of overlap with reference answers and semantic closeness. However, in the ROUGE2 and ROUGE3 metrics, which emphasize the co-occurrence of bigrams and trigrams, the 'fine-tuned' configuration alone performs comparably well. It is noteworthy that the 'Base Mistral' configuration shows the lowest performance across all metrics, underlining the impact of fine-tuning and RAG on enhancing the model's precision and relevance in processing fact-based queries. Similarly, Figure 2.2 assesses the model's ability to summarize content in response to queries. Here, we again see the 'fine-tuned + RAG' configuration outperforming the others.

5.5 Statistical Analysis

ANOVA tests were conducted to statistically verify the differences in performance across the configurations. The results indicated a statistically significant improvement in the fine-tuned + RAG configuration over others.



Figures 3.1 and 3.2: Comparison of composite f-scores of all approaches

According to the above whisker plot, we can see that all f-scores across different models are roughly normally distributed. For each model, there are more than 30 data samples, which implies that this conclusion is statistically significant.

μ_{Mf}: population mean of Mistral f scores, μ_{M+Rf}: population mean of Mistral + RAG f scores, μ_{Mftf}: Mistral fine-tuned f scores, μ_{Mft+Rf}: Mistral fine-tuned + RAG f scores

Hypotheses: $H0$: $\mu_{Mf} = \mu_{M+Rf} = \mu_{Mftf} = \mu_{Mft+Rf}$; Ha: μ_{Mft+Rf} is greater than other means

After performing ANOVA[2] on our testing dataset, we calculated that F-value is around 10.67 and p-value is around 2.877 * 10^-6, which is way smaller than the typical alpha value 0.05. Hence, we can reject the null hypothesis and conclude, with 95% confidence level, that using the RAFT approach is better than using the other three approaches, for the AEC domain.

Conclusion

The advancement highlighted by the RAFT (Retrieval Augmented Fine Tuning) framework is underscored by its superior performance in generating accurate, concise, and semantically rich responses. The statistical significance of these results, validated through ANOVA testing and paired difference testing, attests to the effectiveness of integrating domain-specific fine-tuning with RAG to create models that are highly specialized yet maintain a broad understanding of natural language.

We envision that the continued development and refinement of such models will pave the way for new levels of productivity and decision-making within the AEC industry. It is our expectation that the methodologies and insights yielded by this study will contribute to future efforts in LLM research and applications, especially in sectors where the precision and accuracy of information retrieval and interpretation are of paramount importance. This study not only reinforces the transformative potential of LLMs in specialized fields but also sets a precedent for future research aimed at harnessing the power of LLMs to meet the unique challenges presented by complex, domain-specific datasets.

[2] Refer to Appendix B

Acknowledgements

The InnovAiTors team would like to acknowledge Brian Gue as our subject matter expert. We would also like to thank our advisory team, Russ Greiner and Weijie Sun, for all of their help throughout this project.

References

[1] A. Saka, R. Taiwo, B. A. Salami, S. Ajayi, K. Akande, and H. Kazemi. GPT models in construction industry: Opportunities, limitations, and a use case validation. Developments in the Built Environment. doi: https://doi.org/10.1016/j.dibe.2023.100052, 2023. URL: https://www.sciencedirect.com/science/article/pii/S2666165923001825.

[2] ASME. ASME B31.3 Process Piping Guide. [Date N/A]. URL https://engstandards.lanl.gov/esm/pressure_safety/Section%20REF-3-R0.pdf.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021. URL: https://arxiv.org/abs/2106.09685.

[4] Fan, T., Kang, Y., Ma, G., Chen, W., Wei, W., Fan, L., & Yang, Q. (2023). Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*.

[5] freeCodeCamp.org. (2017, October 24). An intro to Rouge, and how to use it to evaluate summaries. https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/

[6] J. E. Gonzalez, T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez. Raft: Adapting language model to domain specific rag. arXiv preprint, [Date N/A]. URL: https://arxiv.org/html/2403.10131v1#S7.

[7] Lin, C.-Y. (n.d.). Rouge: A package for automatic evaluation of summaries. https://www.researchgate.net/publication/224890821_ROUGE_A_Package_for_Automatic_Evaluation_of_summaries.

[8] Liu, M., Ene, T. D., Kirby, R., Cheng, C., Pinckney, N., Liang, R., ... & Ren, H. (2023). Chipnemo: Domain-adapted llms for chip design. *arXiv preprint arXiv:2311.00176*.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv preprint arXiv:2005.11401, 2021. URL: https://arxiv.org/abs/2005.11401.

[10] Qian, J., Jin, Z., Zhang, Q., Cai, G., & Liu, B. (2024). A Liver Cancer Question-Answering System Based on Next-Generation Intelligence and the Large Model Med-PaLM 2. *International Journal of Computer Science and Information Technology*, *2*(1), 28-35.

[11] Song, Lei & Zhang, Chuheng & Zhao, Li & Bian, Jiang. (2023). Pre-Trained Large Language Models for

Industrial Control.

[12] Wang, Z., Yang, F., Zhao, P., Wang, L., Zhang, J., Garg, M., ... & Zhang, D. (2023). Empower large language model to perform better on industrial domain-specific question answering. *arXiv preprint arXiv:2305.11541*.

[13] Yager, K. G. (2023). Domain-specific chatbots for science using embeddings. *Digital Discovery*, *2*(6), 1850-1861.

[14] Yang, R., Tan, T. F., Lu, W., Thirunavukarasu, A. J., Ting, D. S. W., & Liu, N. (2023). Large language models in health care: Development, applications, and challenges. *Health Care Science*, *2*(4), 255-263.

[15] Yao, J., Xu, W., Lian, J., Wang, X., Yi, X., & Xie, X. (n.d.). Knowledge plugins: Enhancing large language models for domain-specific recommendations. Papers With Code. https://paperswithcode.com/paper/knowledge-plugins-enhancing-large-language

[16] Zheng, O., Abdel-Aty, M., Wang, D., Wang, C., & Ding, S. (2023, July 28). TrafficSafetyGPT: Tuning a pre-trained large language model to a domain-specific expert in transportation safety. arXiv.org. https://arxiv.org/abs/2307.15311

Appendix

A. Additional Formulas and Explanations

A.1.1 ROUGE-N: N-gram Co-Occurrence Statistics

Rouge-N is the overlap proportion of n-grams, contiguous sequences of N words.

$$\text{ROUGE-N Recall} = \frac{\text{ngram matches}}{\text{ngrams in reference answer}} \quad \text{ROUGE-N Precision} = \frac{\text{ngram matches}}{\text{ngrams in model answer}}$$

A.1.2 ROUGE-L: Longest Common Subsequence

Rouge L is the overlap proportion of the longest matching sequence of words using LCS.

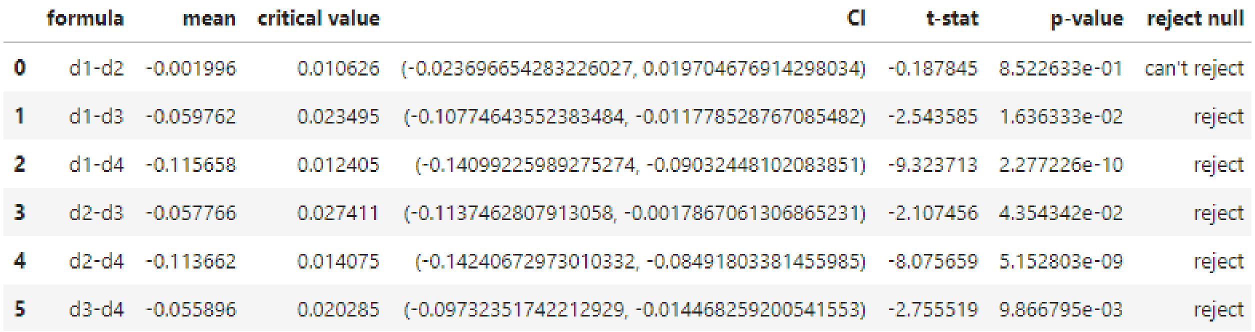
$$\text{ROUGE-L Recall} = \frac{\text{length of LCS}}{\text{words in reference}} \quad \text{ROUGE-L Precision} = \frac{\text{length of LCS}}{\text{words in model answer}}$$

A.1.3 ROUGE-S: Skip-Bigram Co-Occurrence Statistics

Rouge S is the overlap proportion of 2-grams, allowing for arbitrary gaps.

$$\text{ROUGE-S Recall} = \frac{\text{skip bigram matches}}{\text{skip bigrams in reference answer}} \quad \text{ROUGE-S Precision} = \frac{\text{skip bigram matches}}{\text{skip bigrams in model answer}}$$

B. Statistical Test data



	formula	mean	critical value	CI	t-stat	p-value	reject null
0	d1-d2	-0.001996	0.010626	(-0.023696654283226027, 0.019704676914298034)	-0.187845	8.522633e-01	can't reject
1	d1-d3	-0.059762	0.023495	(-0.10774643552383484, -0.011778528767085482)	-2.543585	1.636333e-02	reject
2	d1-d4	-0.115658	0.012405	(-0.14099225989275274, -0.09032448102083851)	-9.323713	2.277226e-10	reject
3	d2-d3	-0.057766	0.027411	(-0.1137462807913058, -0.0017867061306865231)	-2.107456	4.354342e-02	reject
4	d2-d4	-0.113662	0.014075	(-0.14240672973010332, -0.08491803381455985)	-8.075659	5.152803e-09	reject
5	d3-d4	-0.055896	0.020285	(-0.09732351742212929, -0.014468259200541553)	-2.755519	9.866795e-03	reject

* $d1 = \mu_{Mf}$, $d2 = \mu_{M+Rf}$, $d3 = \mu_{Mftf}$, $d4 = \mu_{Mft+Rf}$

Based on the above table, RAFT, represented as d4 in the table, appears to be the best model. The confidence intervals of mean differences involving d4 (which are d1-d4, d2-d4, and d3-d4) are all negative and notably larger in absolute value than the other differences. This suggests that the performance of RAFT is better than those for the other models.