Enhancing Research Paper Summarization Through Advanced Language Techniques: Integrating Abstractive Methods, Fine-tuning Large Language Models, and Retrieval-Augmented Generation

Indrajeet Roy
Department of Electrical and Computer Engineering, Northeastern University
roy.i@northeastern.edu

Abstract

The aim of this paper is to develop a novel system for summarizing academic research papers, leveraging the latest advancements in natural language processing. Facing challenges due to limited datasets, traditional abstractive text summarization methods using encoder-decoder neural networks with attention mechanisms did not yield satisfactory results. In response, this study employs two advanced techniques: Retrieval-Augmented Generation (RAG) integrated with neo4j to construct knowledge graphs and efficiently answer questions about the research papers, and the fine-tuning of the Mistral 7B Large Language Model, a pre-trained generative text model with 7 billion parameters. This integrated approach not only addresses the inherent issues of data scarcity but also aims to transform how researchers interact with academic literature by providing concise, insightful, and context-rich summaries. The project confronts critical challenges such as data quality and availability, the complexity of abstractive summarization techniques, and the computational demands of fine-tuning large language models, all with the goal of enhancing the research paper reading experience.

Introduction

This research project is motivated by the need for reliable and accessible summarization of academic research papers. Existing applications, while useful, often suffer from a lack of precision. More advanced solutions such as chat GPTs, though more accurate, face constraints like token count limitations and are costly to implement. Critically, traditional Seq2Seq models show significant limitations when applied to small datasets—a common challenge in academic environments. This study explores how advanced techniques such as Large Language Model (LLM) fine-tuning and Retrieval-Augmented Generation (RAG) can leverage even limited data to deliver superior performance. Unlike traditional approaches, these methods are designed to optimize effectiveness and efficiency in data-constrained scenarios, potentially transforming the landscape of automated summarization.

To achieve fine-tuning for research paper summarization, various Large Language Models (LLMs), including LLaMA 2, were evaluated. Ultimately, Mistral 7B was selected for its superior performance. Despite the computational challenges associated with this model, techniques such as Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA) were employed to reduce these complexities effectively. Additionally, the project utilized Retrieval-Augmented Generation (RAG) to access external knowledge bases robustly and efficiently. This approach is crucial for enhancing the accuracy and depth of the generated summaries, ensuring that the summaries not only capture essential content accurately but also reflect a deep understanding of the underlying research topics.

Methods

1 Fine-tuning Large Language Models

Large Language Models (LLMs) have transformed the field of natural language processing, providing unparalleled proficiency in understanding and generating text that closely mimics human language. Fine-tuning these models for specialized tasks, such as summarization, requires precise adjustments to reflect the nuances and specific demands of the target domain. In this project, the focus was on enhancing an LLM's ability to summarize academic research papers. This involved fine-tuning the model using a carefully curated dataset of 100 research papers, which helped adapt the pre-trained model to the distinct language, style, and substantive requirements of academic journals.

To achieve this, Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA) were utilized. PEFT focuses on modifying a small subset of the model's parameters, making the fine-tuning process more efficient and less resource-intensive. This

approach is particularly beneficial when working with large models and limited computational resources. On the other hand, LoRA involves inserting trainable low-rank matrices into the pre-existing weight matrices of the LLM. By combining these two techniques, the project aimed to optimize the balance between efficiency and performance in the fine-tuning process.

1.1 PEFT

Parameter-Efficient Fine-Tuning (PEFT), as introduced by Hugging Face, enables the efficient adaptation of pre-trained language models to specific tasks by adjusting only a limited number of additional parameters. This approach significantly reduces computational and storage costs while maintaining performance levels comparable to those achieved through full fine-tuning.

1.2 LoRA

Low-Rank Adaptation (LoRA) enhances fine-tuning efficiency by updating model weights through low-rank decomposition, minimizing changes to the overall parameters and reducing computational load. This innovative approach preserves the original weight matrix, offering benefits such as memory reduction and potentially helping to mitigate catastrophic forgetting. As a result, LoRA has become a popular choice for efficient model fine-tuning.

1.3 Fine-tuning

The fine-tuning of the chosen Large Language Model was conducted utilizing the NVIDIA A100-SXM4-80GB. The initial phase involved evaluating multiple LLMs to determine the most suitable candidate for the specific task. After thorough testing and consideration, experiments were conducted with Google FLAN T5, Falcon AI, LLaMA 2 7B, and ultimately, Mistral 7B was selected for its optimal balance of performance and resource usage.

The next step involved configuring the model for fine-tuning, which included the setup of LoRA adapters, essential for the efficient training of the model. To guide the fine-tuning process, the dataset, consisting of research papers and their summaries, was processed. Prompts were engineered to direct the model to summarize the provided research papers. This was achieved by creating a new column in the data frame, formulated as follows:

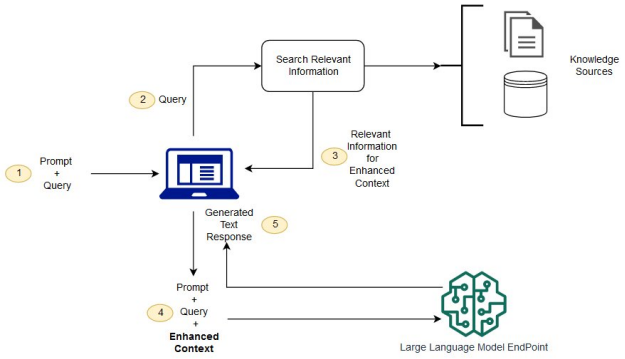
```
start_prompt = '###Human:\n Summarize the
    ↪ following research paper.\n\n'
end_prompt = '###Assistant:\n\nSummary: '
prompts = start_prompt + text + end_prompt
    ↪  + summary
```

In this formulation, text represents the content of the research paper, and summary is its corresponding summary. These prompts were designed to clearly communicate the task to the LLM, ensuring it understood the objective of summarizing the research papers effectively.

The final stage involved the tokenization of the data and the initiation of the training process. Key training parameters were established as follows: a train batch size of 2, a total of 10 epochs, and a learning rate of 2e-4. Upon the completion of the training, a final loss of 3.98 was achieved. While this value indicates room for improvement, the constraints of time and computational resources necessitated limiting the training to 10 epochs.

2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a fascinating technique that combines the best of retrieval-based and generative approaches to provide domain-specific applications without requiring fine-tuning. This makes RAG particularly interesting for applications where updating a model frequently with new data or domain-specific knowledge is necessary but computationally expensive.



Retrieval-Augmented Generation marks a pivotal innovation in augmenting Large Language Models with external, dynamically updated knowledge bases, diverging from the conventional fine-tuning approach where knowledge is statically integrated into the model. RAG introduces an external knowledge base similar to a traditional database, facilitating scalability, reliability, and the integration of up-to-date information. The core of RAG's methodology is the use of source knowledge inputted into the LLM to retrieve relevant data from an external source. This additional information is then incorporated into the model's prompt, enabling the LLM to generate responses that are not only more accurate but also reflect current knowledge.

While "naive RAG" offers a straightforward and efficient implementation, more complex variations exist that can handle nuanced queries with greater precision. Nonetheless, these advanced methods bring trade-offs in terms of response speed and operational costs. The interaction with the LLM required at each reasoning step can be both time-consuming and costly.

A significant benefit of RAG is its ability to reduce hallucinations, a common limitation in modern LLMs

where the model generates plausible but incorrect or unfounded information. By grounding responses in retrieved documents, RAG models can offer more accurate and verifiable outputs.

For the vector database essential in the retrieval step, FAISS (Facebook AI Similarity Search) is utilized due to its speed and ease of integration. FAISS can be stored locally, offering an advantage over cloud-based databases by scaling applications to a broader audience without incurring the latency and cost associated with online data retrieval. This local integration of FAISS with RAG not only enhances performance but also aligns with the need for efficient, scalable solutions in deploying advanced NLP applications.

The architecture implemented was a combination of traditional databases with knowledge graphs[7][9] for a Retrieval Augmented Generation system. These knowledge graphs are built from the research paper dataset and offer a more organized way to access and use data. This assists the model to pull in rich and related information from these graphs.

The LangChain framework was selected for integrating advanced features with the Large Language Model in the Retrieval-Augmented Generation system due to its robust capabilities in chaining language model calls with external data retrieval. This architecture is essential for RAG systems that require dynamic information fetching during the generation process. LangChain modular structure allows for tailored customization and scalability, addressing the project's need for accurate academic paper summarizations that incorporate diverse data sources.

3 Traditional Method - Seq2Seq Model

Exploring methods for summarizing research papers under limited data scenarios, a bidirectional LSTM (Long Short-Term Memory) Seq2Seq model augmented with an attention mechanism was implemented. This model employs an abstractive summarization approach, where it generates entirely new text based on the understanding of the original content, unlike extractive methods that simply select parts of the source text. Representing a traditional approach, this Seq2Seq model contrasts with the newer advancements such as Retrieval-Augmented Generation (RAG) and the fine-tuning of Large Language Models (LLMs). Unlike these contemporary methods, which may integrate external data or utilize extensive pre-trained networks for enhanced contextual understanding and adaptability, the LSTM-based Seq2Seq model relies on its intrinsic sequence processing capabilities. It effectively processes and understands text sequences by emphasizing segments crucial for generating accurate summaries, excelling in environments with constrained datasets.

Bidirectional LSTMs enhance context comprehension by processing text in both forward and backward directions, providing a comprehensive understanding of text sequences. This dual-direction processing is crucial for summarization tasks, which rely heavily on contextual accuracy to determine the relevance of text segments.

The integration of an attention mechanism optimizes model performance by selectively emphasizing the most informative parts of the text. By assigning varying weights to different sections, it focuses on segments that are crucial for encapsulating the core message of the content. This method not only improves the coherence of the summaries but also ensures their relevance to the main themes of the research papers.

This approach proves especially effective in data-scarce environments, where it utilizes limited information to produce concise and relevant summaries. Serving as a robust complement to newer techniques like Retrieval-Augmented Generation (RAG) and the fine-tuning Large Language Models (LLMs), this method enhances the available methodoliges for academic literature summarization.

3.1 Model Architecture

The Seq2Seq model employs Long Short-Term Memory (LSTM) networks to process and generate text sequences effectively. These networks, a specialized type of Recurrent Neural Network (RNN), incorporate gating mechanisms—input, output, and forget gates—that regulate information flow, enhancing the model's ability to manage long-range dependencies and contextual nuances.

Encoder: The encoder utilizes a LSTM layer, optimized with tanh and sigmoid activation functions for controlling state and gate dynamics, respectively. A dropout mechanism mitigates overfitting. Integration of a Keras Bidirectional wrapper allows the encoder to process text in both forward and reverse directions, merging outputs to form a comprehensive context vector. The setup is completed by a Keras Input layer that defines the fixed-size, tokenized input sequences, preparing a detailed encoder vector for the decoder.

Decoder: The encoder utilizes a LSTM layer, focusing on sequential text generation, utilizing the context vector from the encoder and outputs from previous time steps to generate contextually relevant words. A Keras Dense layer with a softmax activation function transforms LSTM outputs into a probability distribution over potential output words, facilitating coherent sequence generation.

Attention: The attention mechanism in the sequence-to-sequence model significantly enhances the decoder's ability to focus on pertinent segments of the input text during summary generation. Implemented using a Attention layer, this mechanism calculates attention scores that allow the model to dynamically allocate focus across different parts of the input sequence. For each word generated by the decoder, the attention mechanism computes a dynamic context vector. This is achieved by assigning weights to various segments

of the encoder's output, indicating the relevance of each input segment to the word currently being generated. These weights are determined based on the interaction between the current state of the decoder and the encoder's outputs. This process ensures that generated summaries are concise and contextually aligned with the input text's core content.

3.2 Model Training

During preprocessing, text is cleaned by removing non-essential characters and stopwords to streamline input for effective tokenization and summarization. The model employs the Adam optimizer for its adaptive learning rate capabilities, which efficiently handles sparse gradients. Categorical cross-entropy is utilized as the loss function to accurately measure the discrepancy between the predicted and actual word distributions in the summaries.

Key hyperparameters, including learning rate, batch size, and the number of training epochs, are optimized to enhance learning efficiency and model performance. The learning rate is set high initially to promote rapid convergence and is gradually reduced to allow for finer adjustments during later stages of training. Batch size and the number of epochs are adjusted to balance memory usage and performance, minimizing the risk of overfitting.

To further combat overfitting, dropout techniques are applied within LSTM layers, randomly omitting a fraction of neurons (dropout rates between 0.2 and 0.5) to diminish dependency on specific neuron patterns. Early stopping is implemented via callback function that monitors validation loss, halting training when no improvements in loss are observed, thereby preventing the model from learning noise or irrelevant data fluctuations. This strategy ensures that the generated summaries are both concise and contextually relevant to the original text, leading to a robust model capable of producing high-quality text summaries.

Results Evaluation

In the course of the experiments, the performance of the fine-tuned model was assessed by comparing the summaries it generated with the original summaries of the research papers. This evaluation process is crucial for understanding how well the model has learned to condense the content of research papers into concise, informative summaries.

1 Fine-tuning Large Language Models

Original Summary:

```
This paper proposes a novel technique to detect
anomalies in Electrocardiogram signals and classify
cardiac conditions from 45 patients in the MIT-BIH
Arrhythmia database. The proposed approach utilizes
two methods, one based on conventional Machine
learning algorithm (SVM) and the other based on a
deep learning method (CNN-based architecture
ALEXNET). The deep learning technique
improved precision and can be used in clinical
settings, with an average classification
accuracy of 87.2%.
```

Predicted Summary:

```
1. The paper proposes a comparison of machine
learning-based conventional classifier SVM and
deep CNN architecture AlexNet for classifying
arrhythmia dataset.
2. The raw data is preprocessed, feature
extracted, and cleaned data is divided into
train and test sets.
3. The SVM model with linear kernel and AlexNet
model with optimizer Adam and loss function
categorical cross entropy are trained.
4. The future work includes using different
deep learning architectures and
hyperparameters for ecg signal classification.
```

The provided summaries demonstrate the fine-tuned model's ability to capture key elements of the research paper, although with varying levels of detail and specificity. The original summary succinctly presents the core findings and methodology of the study, focusing on the novel technique for anomaly detection in ECG signals and the comparison between SVM and a deep learning approach using the ALEXNET architecture. It highlights the practical application and effectiveness of the deep learning method, evidenced by the reported average classification accuracy.

The predicted summary by the fine-tuned model provides a more granular overview of the study's process, from data preprocessing to model training, and mentions future work considerations. It reflects the model's capacity to extract and reiterate detailed procedural aspects of the research, though it could be more concise and focused on the study's primary contributions and outcomes.

This comparison illustrates the strengths and areas for improvement in the model's summarization capabilities. The model effectively identifies and communicates essential points, but the inclusion of superfluous details suggests that further refinements could enhance its ability to generate more focused and impactful summaries. This feedback loop is critical for iterative model improvements, aiming for summaries that are not only accurate but also aligned with the conciseness and clarity expected in academic summarizations.

2 Retrieval-Augmented Generation results

The capabilities of the Retrieval-Augmented Generation system are demonstrated through its responses to specific inquiries about a research paper. These examples illustrate the system's proficiency in understanding complex topics and its precision in retrieving accurate information.

Questions and Answers:

```
1. What is the research on?
The research focuses on abstractive summarization using
a model that incorporates bidirectional RNNs.

2. Which model did the author propose?
The author proposed a feature-rich encoder and a
switching generator pointer model.
```

Handling Hallucinations:

```
1. How did Trump contribute to the paper?
Trump was not mentioned in this paper.
```

The RAG system effectively identifies the primary focus of research papers, as demonstrated when pinpointing the key area of study in abstractive summarization, particularly highlighting the utilization of bidirectional RNNs. This accuracy underscores the system's adeptness at capturing the central theme of the research, navigating through the technical intricacies without distraction from complex details interspersed within the document.

When inquired about the model proposed by the author, the RAG system succinctly articulates it as a "feature-rich encoder and a switching generator pointer model." This response encapsulates the core attributes of the proposed model and illustrates the system's proficiency in condensing extensive information into a succinct summary. Such capability is essential for effectively communicating sophisticated research findings, facilitating a better understanding and dissemination of scientific advancements.

Hallucination in language models refers to the generation of responses that are ungrounded or factually incorrect, a common issue where the system might introduce irrelevant or fabricated information.The question about Trump's contribution to the paper serves as a test for this phenomenon. The system's precise response, "Trump was not mentioned in this paper," demonstrates its capability to avoid hallucinations and maintain factual accuracy, thereby underscoring its reliability in handling queries that require strict adherence to source material.

3 Seq2Seq Model

The evaluation of the Seq2Seq model, employing LSTM networks, has highlighted several insights into its capabilities and limitations in handling the task of text summarization. The following points highlight the key findings:

Keyword Identification Proficiency
The Seq2Seq model excelled at identifying and extracting key terms from the research papers, demonstrating its strong capacity for learning from sequential data and recognizing pertinent terms within the textual context. This indicates a robust ability to parse and interpret domain-specific vocabulary.

Repetitive and Fragmented Output
The outputs from the Seq2Seq model often exhibited issues with repetition and a lack of cohesive contextual structure, which reflects difficulties in the sequential generation process. These challenges are indicative of problems in maintaining contextual understanding over longer text spans, where the inherent limitations of LSTM-based Seq2Seq models in capturing and utilizing long-range dependencies become apparent.Retrieval-Augmented Generation (RAG) and fine-tuned Large Language Models (LLMs) demonstrate superior performance in handling extensive texts.

Retrieval-Augmented Generation (RAG) significantly enhances output diversity by dynamically leveraging external knowledge bases during the generation process. This allows RAG to introduce varied and contextually relevant information not present in the training data, effectively reducing repetitive patterns and enriching the content pool for summary generation. Simultaneously, Large Language Models (LLMs) fine-tuned for specific tasks like summarization derive immense benefits from extensive pre-training on diverse corpora. This broad exposure equips the LLMs with a deeper understanding of language and context, enabling them to produce coherent and contextually appropriate text. The fine-tuning process enhances their capability to synthesize and integrate information across extended text spans, thereby maintaining a consistent context and addressing the issues of repetitive and fragmented outputs found in traditional Seq2Seq model outputs.

Generalization Across Text Types
The Seq2Seq model's challenge in generalizing to new text types, not well represented in the training set, was significant. This aspect was particularly critical in the domain of research paper summarization, where the unique structure and language pose additional challenges. Meanwhile, the broad and varied pretraining of LLMs grants them enhanced generalization capabilities, making them more adept at adapting to varied textual formats.

Conclusion

This research project demonstrates the advantages of fine-tuning Large Language Models (LLMs) over traditional encoder-decoder methods in summarizing research papers, especially in scenarios with limited datasets. By employing advanced techniques such as Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA), the study successfully produced effective summaries from a modest dataset of only 100 research papers. Additionally, the effectiveness of Retrieval-Augmented Generation (RAG) was underscored, showing its capacity to deliver detailed and insightful responses without the need for extensive fine-tuning, thus proving its suitability for NLP tasks under data constraints.

This holistic strategy combines the concise summarization capabilities of fine-tuned LLMs with the dynamic, context-sensitive responsiveness of RAG. The integration of PEFT and LoRA methods allows for efficient model optimization with minimal computational overhead, leading to significant performance enhancements in specialized tasks like summarization, even without extensive datasets. Moreover, RAG's retrieval-based mechanism offers a powerful means to enhance the model's knowledge base on-the-fly, ensuring responses are both accurate and up-to-date—crucial in fast-evolving fields.

References

1. Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).
2. Hu, Zhiqiang, et al. "LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models." arXiv preprint arXiv:2304.01933 (2023).
3. Wang, Jianguo, et al. "Milvus: A purpose-built vector data management system." Proceedings of the 2021 International Conference on Management of Data. 2021.
4. Jiang, Albert Q., et al. "Mistral 7B." arXiv preprint arXiv:2310.06825 (2023).

5. Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in Neural Information Processing Systems 33 (2020): 9459-9474.

6. Topsakal, Oguzhan, and Tahir Cetin Akinci. "Creating large language model applications utilizing langchain: A primer on developing llm apps fast." Proceedings of the International Conference on Applied Engineering and Natural Sciences, Konya, Turkey. 2023.

7. Yao, Liang, et al. "Exploring large language models for knowledge graph completion." arXiv preprint arXiv:2308.13916 (2023).

8. LangChain Documentation. Available online: `https://python.langchain.com/docs/get_started/introduction` [Accessed on: date].

9. Heidloff, Niklas. "Efficient Fine-Tuning with LoRA." Available online: `https://heidloff.net/article/efficient-fine-tuning-lora/`.

10. Hugging Face. "Mistral 7B Model." Available online: `https://huggingface.co/mistralai/Mistral-7B-v0.1`.

11. Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

12. Poonja, Hasnain Ali, et al. "Evaluation of ECG based Recognition of Cardiac Abnormalities using Machine Learning and Deep Learning." 2021 International Conference on Robotics and Automation in Industry (ICRAI). IEEE, 2021.