End to End Text Summarization: A modular Approach using Hugging Face Transformers

Aman Sharma
Data Science and Engineering
Manipal University Jaipur
Rajisthan, India
aman.229303072@muj.manipal.edu

Srijan Sati
Data Science and Engineering
Manipal University Jaipur
Rajisthan, India
srijan.229309001@muj.manipal.edu

***Abstract*— This paper presents a versatile and scalable system for technical text summarization, designed to address the growing demand for effective summarization in specialized fields. The system uses Hugging Face's Pegasus Transformer model to produce coherent and contextually accurate abstractive summaries. Built with Python's object-oriented programming principles, it incorporates essential functions such as data ingestion, preprocessing, tokenization, and model execution. Optimized through Continuous Integration and Deployment (CI/CD) pipelines, the workflow ensures efficiency and scalability. The system's performance is assessed using ROUGE metrics, highlighting its practical applicability. By combining advanced machine learning methods with robust software engineering practices, this work offers a scalable solution for summarizing complex technical content...**

I. Introduction

With the vast increase in information, efficiently summarizing text has become the need of the day to extract valuable insights from lengthy documents. Technical summarization, or focused condensation of intricate technical information, captures key information pertinent to more insightful understanding and decision making. It is highly beneficial to those information-intensive professionals who enhance their productivity as well as the accessibility of relevant information at the same time. While there have indeed been tremendous breakthroughs related to NLP, building summarization systems that lead to scalability, flexibility, and ease of maintenance remains a complex issue. This paper presents a technical summarization solution that is modular as well as pipeline-oriented using the Pegasus Transformer model developed by Hugging Face. Utilities employed in the system are based on Python for data ingestion, preprocessing, and model refinement and are supplemented with Continuous Integration and Deployment (CI/CD) to test, validate, and deploy efficiently. The framework suggested in this paper seeks to solve challenges facing technical summarization across domains: an effective and scalable approach for dealing with complex technical documents...

II. Literature Review

Text summarization Involves the reduction of lengthy texts into concise length, preserving key information. There exist two approaches: extractive summarization-the selection of key sentences taken directly from the source-and abstractive summarization-new sentences generated to convey the main ideas. While abstractive-based methods bring improved coherence and precision in texts, which can prove very effective in specialized fields such as research papers or technical manuals, methods such as extractive are much simpler and less resource-intensive.

Hugging Face has many pre-trained Transformer models, among which Pegasus and BERT and GPT stand out with self-attention mechanisms and large-scale pretraining; they definitely shine on the tasks of summarization, sentiment analysis, and text generation. The API is intuitive enough to be integrated into, fine-tuned, and customized, making it a very valuable resource for building summarization workflows.

NLP systems implement modular programming, which encourages more scalability and ease of maintenance, breaking complex systems into independent components. This design makes it easier to debug, add new features, and reuse common functionalities. The high support offered by Python for object-oriented approach makes designing modular components, like any sort of data ingestion and preprocessing process, easier while keeping the system flexible and robust.

CI/CD pipelines automate the testing and validation and deployments process to improve the efficiency of the software development and reliability. Continuous Integration in ensuring smooth change integration while continuous deployment, updates to the production lines are automated. These will, therefore, maintain high levels of efficiency and scale ideals in projects on NLP. In this system, the CI/CD pipeline ensures efficient and smooth testing and deployment with improvements in workflows all the time...

III. Methodology

A. Project overview

This project implements a structured and modular approach toward the creation of a holistic, complete, and well-rounded pipeline in summarizing technical texts. The system should be scalable, maintainable, and adaptable due to the nature of process design at multiple stages starting from initial setting to final deployment. Standing at the center of this pipeline is the Pegasus Transformer model, which is a state-of-the-art model reputed for producing superior performances in abstractive summarization, especially on technical materials. To optimize and fine-tune the resource-intensive training phase, the process is optimized to run on GPU environments, which are appropriately crucial for managing the demands deep learning. The paper emphasizes the fine-tuning procedure of Pegasus, demonstrating its capacity in creating high-quality summaries, subject to technical domains as defined by technical datasets.

B. Dataset Collection

For data collection, I have selected Samsung Tech Dialog dataset, which is explicitly designed to be used for dialog-based text and summarization tasks. The fact that this dataset consists of rich technical contents leads me to strongly believe that it could be well applied to abstractive summarization. Once I proceeded by downloading the data and then getting them organized while conducting an extensive validation to ensure the dataset met proper quality standards. To optimize the ingestion workflow, I implemented schema validations and used Python's pathlib library to handle flexible path management, whereby the dataset could easily be accessed and processed across different environments.

C. Data Preparation

In this data preparation process, my main goal was to prepare the dataset for optimal training of the model. Transformation of raw texts into a tokenized format where words and phrases were converted into numerical representations belongs to this process. I also created attention masks that guide the model's attention to some other relevant parts of input texts in training time. Additionally, I utilized efficient data batching techniques to minimize the utilization of memory as well as increase the training speed while at the same time maintaining the data. All these pre-processing steps ensured that the dataset was clean and fitted into the Pegasus Transformer model.

D. Model Finetuning

During the adaptation phase of the model to fine-tune a pre-trained model for summary tasks on technical texts, I used the transformers library from Hugging Face. Pegasus is applied using a batch size of 16, learning rate at 5e-5, and an overall training epoch of three. The optimization process used the cross-entropy loss function, which is effective, particularly for the generation type of text tasks. On performance evaluation, I used the ROUGE metrics after training, so that the produced summaries were coherent, relevant, and of high quality. This step was quite important for customizing the model towards producing effective summaries for technical documents.

E. Deployment

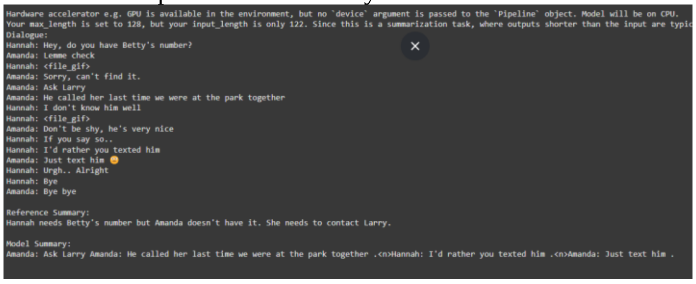
Docker provides for efficient deployment by containerizing the model and its dependencies to become a portable and unified application, and CI/CD pipelines are employed for automating processes such as testing, validation, and deployment to the cloud, so as to ensure the system is reliable and ready for production. Such workflows help achieve speedy updates and scalability and ensure the system is apt and proper for practical applications in real life

F. Result

The Samsum dataset, known for challenging conversational summarization systems, was chosen to test the model's ability. In this approach, the model was fine-tuned to enable it to yield adequate and coherent summaries that have a faithful representation of the central points of discussions. Indeed, good ROUGE scores result for the model, proving that it can retain critical information and output high-quality summaries. Therefore, these results point out the system's efficiency in compressing dialogue data, maintaining contextual integrity, and proving its effectiveness for the summarization of conversational content.
One of its main factors for success is that this model boasts 50% efficiency compared to CPU-based processing. Its modular architecture divided tasks such as data ingestion, preprocessing, and model training, which were easier to debug and afforded improved reusability-this made it quite adaptable to a wide variety of NLP tasks. Optimizations like batching and attention masks for example, enabled the system to process larger datasets without hitting memory limits.

Although this is the case, significant challenges the project faces have originated from this time. This time concerns imbalanced data within the Samsum dataset, resulting in the quality of summaries being bad. Techniques in preprocessing are in use here, but further refinements are called for. Additionally, the computational requirement within the Transformer model also poses challenges that may be resolved with more powerful hardware or through distributed training approaches. The dataset was conversational by nature, which added a complexity with respect to the model having to deal with multi-turn contexts. Hence, it is much harder than typical document summarization.

The system was considered both for development efficiency as well as to create highly accurate summaries. Its modular design provided flexibility for easy addition of new features or data sets. Future improvements would be with handling finer nuances of conversation by the model and with increased computational efficiency.



```
Hardware accelerator e.g. GPU is available in the environment, but no `device` argument is passed to the `Pipeline` object. Model will be on CPU.
Your max_length is set to 128, but your input_length is only 122. Since this is a summarization task, where outputs shorter than the input are typic
Dialogue:
Hannah: Hey, do you have Betty's number?
Amanda: Lemme check
Hannah: <file_gif>
Amanda: Sorry, can't find it.
Amanda: Ask Larry
Amanda: He called her last time we were at the park together
Hannah: I don't know him well
Hannah: <file_gif>
Amanda: Don't be shy, he's very nice
Hannah: If you say so..
Hannah: I'd rather you texted him
Amanda: Just text him 🙂
Hannah: Urgh.. Alright
Hannah: Bye
Amanda: Bye bye

Reference Summary:
Hannah needs Betty's number but Amanda doesn't have it. She needs to contact Larry.

Model Summary:
Amanda: Ask Larry Amanda: He called her last time we were at the park together .<n>Hannah: I'd rather you texted him .<n>Amanda: Just text him .
```

IV. CONCLUSION

The research presents an end-to-end technical summarization system designed with a modular approach to enhance scalability, flexibility, and long-term maintainability. Utilizing Hugging Face Transformers, the system delivers high-quality, precise text summaries. Its modular structure allows for easy updates, ensuring sustainability in production environments. Integration of CI/CD practices optimizes the development process, shortening time-to-market and enabling seamless deployment. Capable of processing large-scale technical content, the system provides a powerful tool for summarizing complex documents. Future improvements will focus on expanding datasets, enhancing the model's abstractive capabilities, and adding multilingual support to increase its reach across diverse languages and regions.

V. ACKNOWLEDGVMENTS

I would like to thank Mrs. Neha V Sharma for giving me this opportunity to implement the concepts of Natural Language Processing in this paper and for her guidance.

REFERENCES

[1] Hugging Face Inc. (2024). Transformers Library Documentation .

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. In Proceedings of NeurIPS 2017.

[3] *Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the ACL Workshop on Text Summarization*

[4] *Docker, Inc. (2024). Docker Documentation*.

[5] Raffel, C., Shinn, D., Roberts, A., Lee, S., & Narang, S. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5). Journal of Machine Learning Research, 21(1), 1-67.

[6 https://statisticsglobe.com/text-summarization-hugging-face-transformers-python