بسم الله الرحمن الرحيم

Automatic Text Summarization Using Deep Learning Models

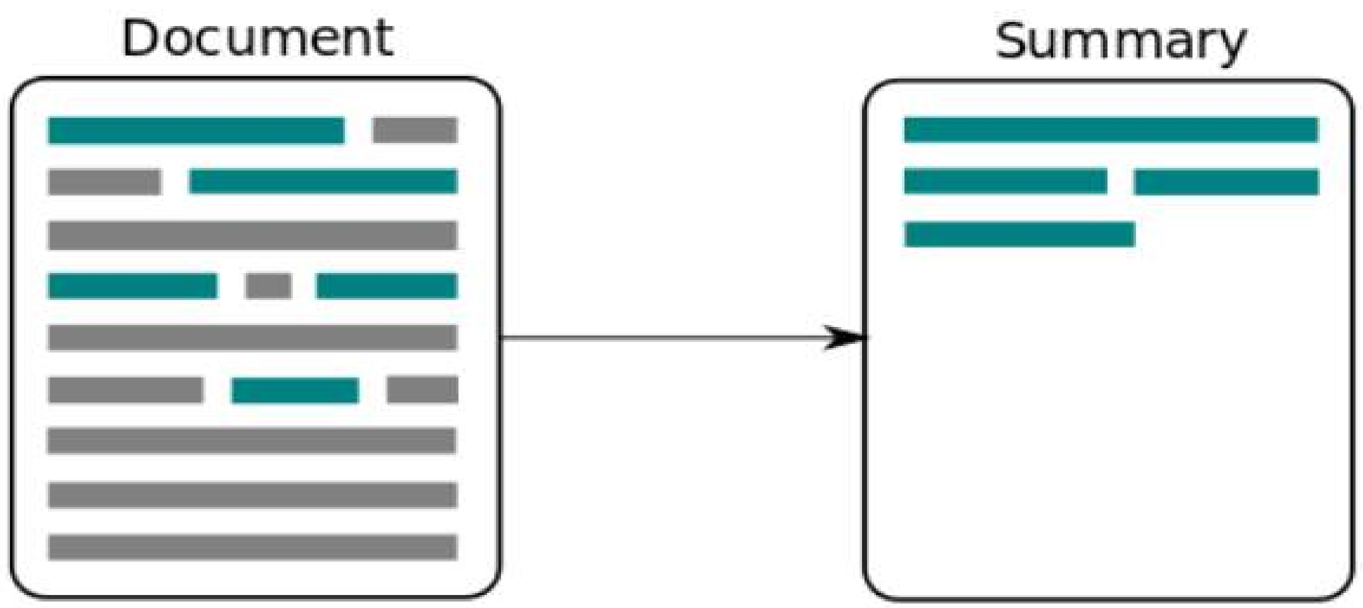
BY
Zohaib Ali Shah
Prof. Dr. Ibrar Ali Shah
Prof. Dr. Muhammad Usman
Bilal Khan

SEQUENCE

- Introduction
- Literature Review
- Research Methodology
- Comparative Analysis
- Results
- Implementation Setup
- Future Work
- References

INTRODUCTION

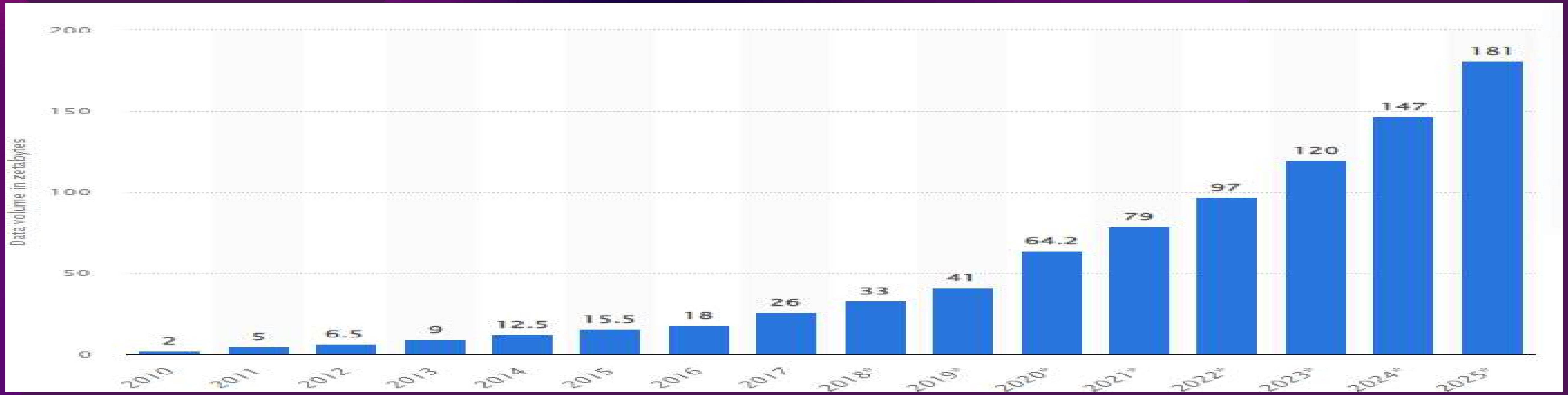
Automatic text summarization (ATS) is the process of shortening a text document by automatically creating a short, accurate, and fluent summary with the main points of the original document using software or tools.



Three main methodologies drive text summarization: Extractive, Abstractive and Hybrid.
In an extractive summary, the most important sentences are extracted and joined to get a brief summary. While abstractive method generates new sentences that capture the essence of the original text, often using techniques like natural language generation. It's more like a human writer paraphrasing the content. - just as humans do

INTRODUCTION

- Since humans have the capacity to understand the meaning of a text document and extract the most important information from the original source using their own words, we are generally quite good at making summaries of a text. However, manual creation of summaries is very time consuming, and therefore a need for automatic summary has arisen. Not only are the automatic summarization tools much faster, they are also less biased than humans.
- There is a need for a robust and scalable solution that leverages state-of-the-art natural language processing (NLP) models to automate text summarization tasks. This project processes the development of a text summarization system using advanced deep learning-based models, integrating these model into a user-friendly mobile and web-based application.
- Processing such vast amounts of data manually is virtually impossible, highlighting the need for ATS.



LITERATURE REVIEW

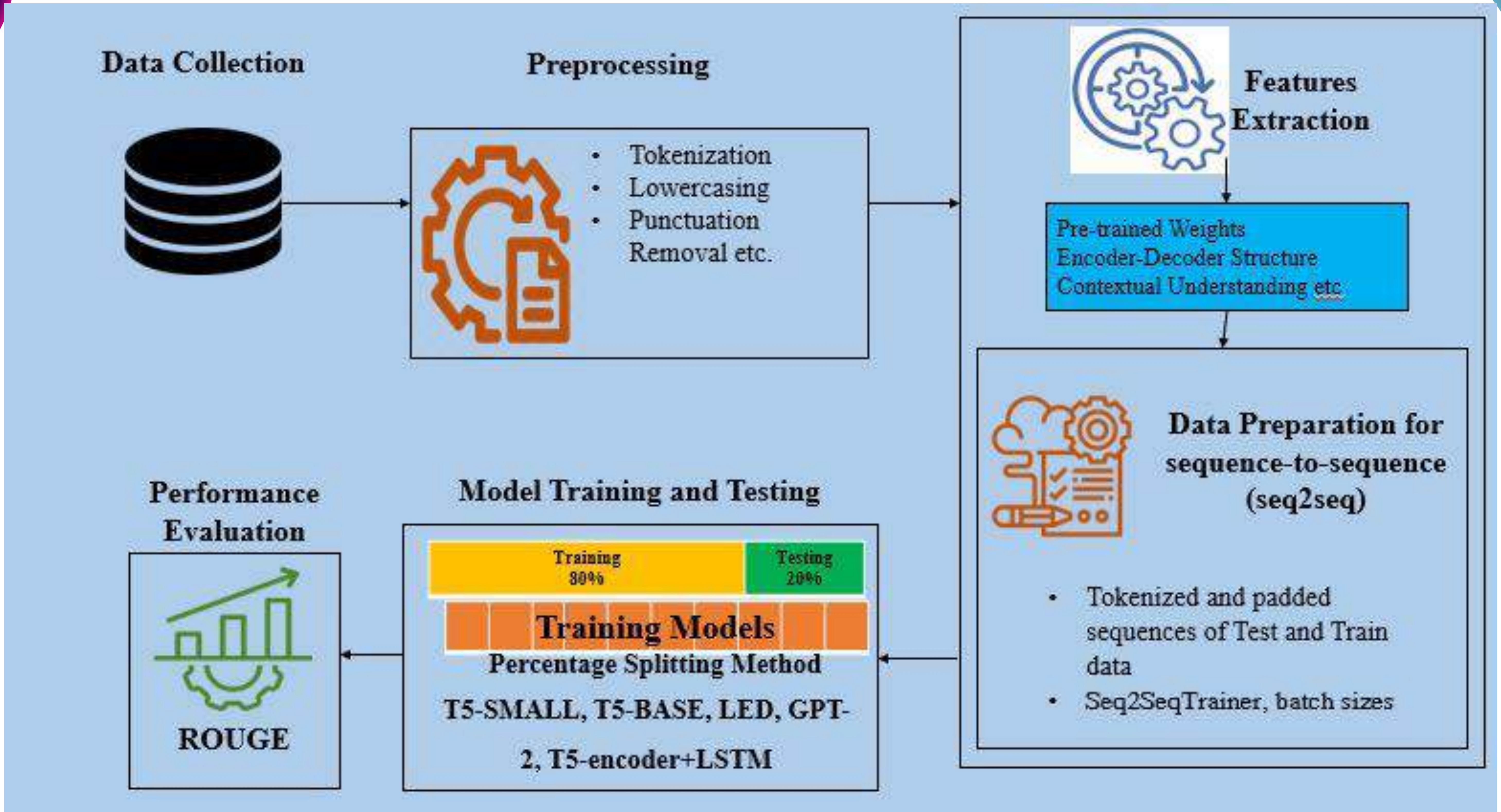
Text summarization using deep learning algorithms is a significant research focus, addressing the challenges of manual summarization, which is time-intensive and prone to inaccuracies. This literature review covers theoretical frameworks and key research contributions in automated text summarization systems.

- Neural Attention Model: Liu and Lapata developed a deep learning model for extractive summarization, significantly improving summarization accuracy by identifying key sentences.
- Multi-Task Learning: Gehrmann et al. introduced a framework that predicts sentence importance while generating summaries, enhancing extractive summarization via joint learning.
- Reinforcement Learning: Zhou et al. applied reinforcement learning to improve coherence and informativeness in summaries.
- Hierarchical Attention: Zhao proposed a model using attention mechanisms to capture word relationships in abstractive summarization.
- Reader's Knowledge Model: A study incorporated reader’s prior knowledge using hierarchical attention, adapting summaries based on contextual relevance.
- Multi-Task Learning with Reinforcement: Another study combined multi-task learning and reinforcement learning to create coherent and informative abstractive summaries.

LITERATURE REVIEW

Year	Technique/Model	Contribution	Problem Addressed
2022	Neural Attention Model	Introduced a deep learning-based model for extractive summarization, significantly improving summarization accuracy.	Identifying salient sentences from input documents.
2023	Multi-Task Learning Framework	Presented a framework leveraging deep learning for joint learning tasks, simultaneously predicting sentence importance and generating summaries.	Enhancing the effectiveness of extractive summarization.
2018	Deep Reinforcement Learning	Explored reinforcement learning for text summarization, improving the coherence and informativeness of generated summaries.	Improving coherence and informativeness in generated summaries.
2022	Hierarchical Attention Network	Investigated attention mechanisms for abstractive summarization, capturing relationships between words and phrases effectively.	Enhancing abstractive summarization by focusing on word relationships.
2020	Hierarchical Attention Model	Proposed incorporating reader's prior knowledge into summarization, using a hierarchical attention mechanism.	Incorporating prior knowledge into the summarization process.
2020	Multi-Task Learning Approach with Reinforcement Learning	Proposed a multi-task learning approach for abstractive summarization, focusing on generating summaries that are both informative and coherent.	Combining informativeness and coherence in abstractive summaries.

RESEARCH METHODOLOGY



DATASET

Data was gathered from various sources like forums, books, datasets, surveys, and questionnaires, creating a dataset of 2,741 entries with main texts and expert-verified summaries to ensure accuracy in reflecting psychological content. This data serves as the foundation for model training and testing

The collected dataset consists two columns: the main text and the short version of the text which is their summaries.

Category	Number of Instances	Description
Original Text	2,741	Comprehensive texts collected from various sources
Summarized Text	2,741	Summaries generated using different summarization tools

PREPROCESSING AND FEATURE EXTRACTION

- **Preprocessing**
  Preprocessing included tokenization, lowercasing, punctuation removal, lemmatization, and stemming to prepare the raw text for model training. Batching and input limits (1024 tokens) were applied to optimize memory and performance, making the dataset ready for the summarization model.
- **Feature Extraction**
  The LED model (Longformer-Encoder-Decoder) uses a transformer structure, pre-trained weights, and tokenization to capture linguistic features. It processes long-range dependencies and outputs structured, coherent summaries, leveraging deep learning's contextual analysis capabilities.

DATA PREPROCESSING

- **Data Preprocessing for Sequence-to-Sequence (Seq2Seq)**
  Seq2Seq preprocessing involved tokenizing input texts and summaries, padding sequences for uniformity, and defining training parameters like batch size, learning rate, and weight decay. This setup optimized training and inference for generating quality summaries.

MODEL TRAINING and TESTING

To ensure the model's robustness and performance, we divided the dataset into two parts:

- **Training Set (80%):** This is used to train all the models that are employed in this research. It assists the models to get familiar with the patterns and relations in the data in order for it to produce good summaries.
- **Testing Set (20%):** This portion is intended to be used for evaluation of the employed models. This makes it possible to test the effectiveness of the trained models when subjected to data that they have not encountered throughout the learning process

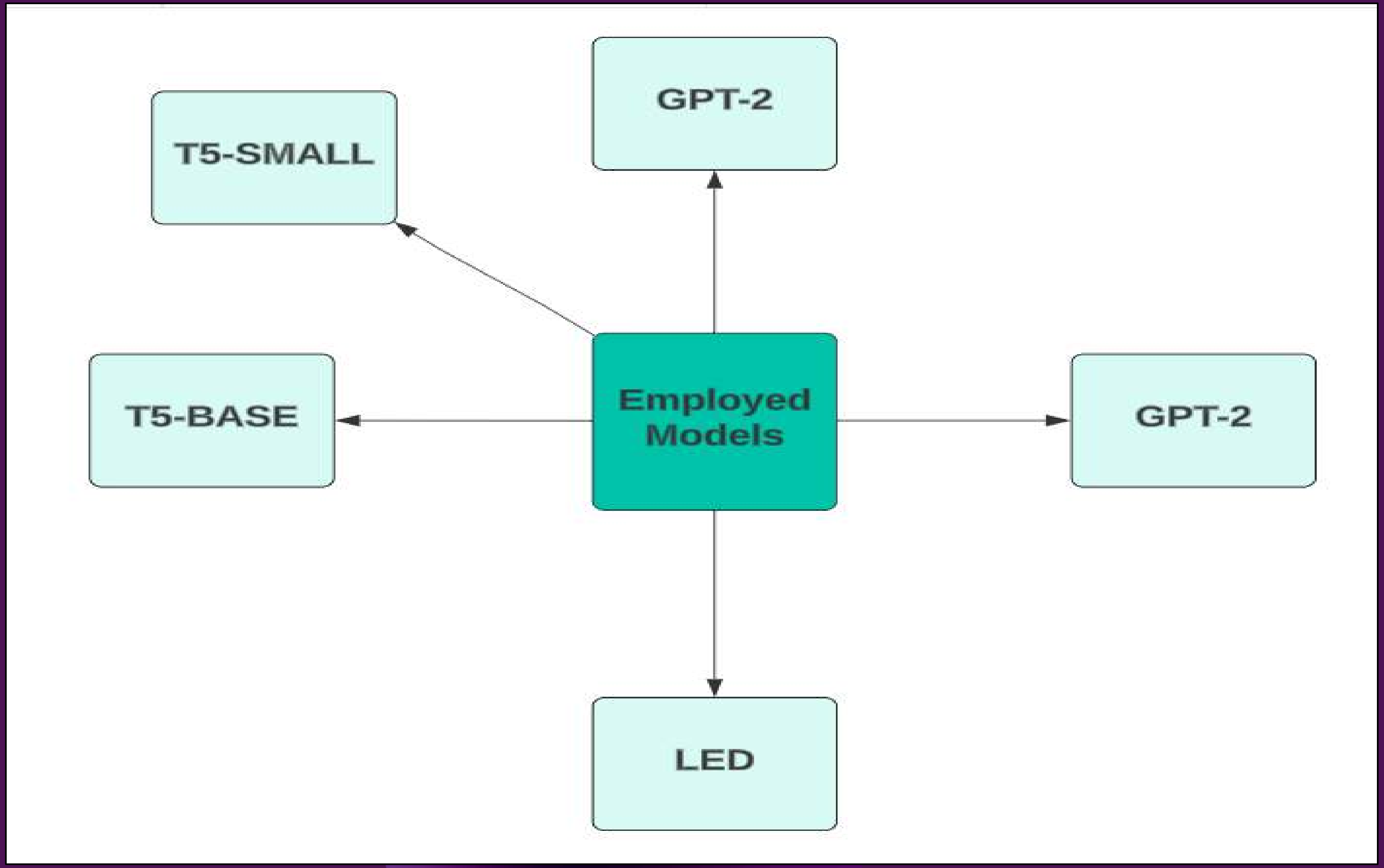
Dataset	Percentage	Instances
Training	80	2192
Testing	20	549

PERFORMANCE EVALUATION

- **ROUGE(Recall-Oriented Understudy for Gisting Evaluation)** is widely used for text summarization because it measures how well the generated summaries align with the actual summaries (focuses on exactness, not just closeness). It's preferred for text summarization because it evaluates both **precision** and **recall**, making it more comprehensive.
- **ROUGE-N** and ROUGE-L are the most common metrics, with ROUGE-N measuring the overlap rate of N-grams, and ROUGE-L focusing on the longest common subsequence (LCS) between the reference and generated summaries.
- **ROUGE-1**=∑unigramsCountmatch(unigram)/ ∑unigramsCount-reference(unigram)
- **ROUGE-2** = ∑bigramsCountmatch(bigram)/ ∑bigramsCount-reference(bigram)
- **ROUGE-L** = LCS (reference, generated) / Length of reference
- **ROUGE-Lsum** = LCS (reference summary, generated summary) / Length of reference summary). Where LCS (reference, generated) is the length of the longest common subsequence between the reference and generated summaries

COMPARATIVE ANALYSIS

In this comparative Analysis we basically train different kind of models on our own psychological dataset and find out the rouge scores of all the employed models.



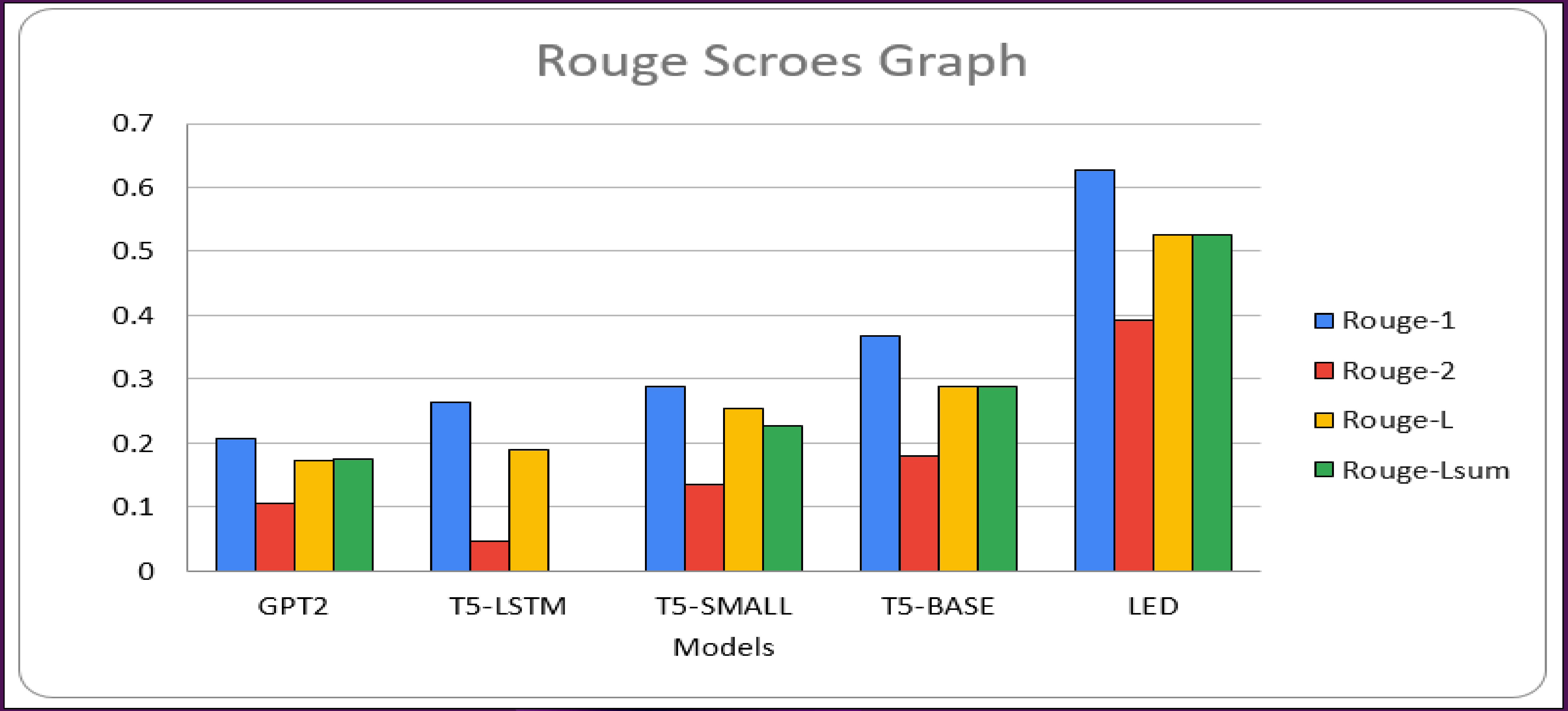
RESULT COMPARATIVE ANALYSIS

The comparison of all the employed models by focusing on the Evaluation performance scores is presented in tabular format.

ROUGH SCORE	MODELS NAME				
	T5-encoder-LSTM	T5-SMALL	T5-BASE	GPT2	LED
Rouge-1	0.2628	0.2881	0.366762	0.20607	0.6269
Rouge-2	0.0450	0.1339	0.178565	0.10605	0.3921
Rouge-L	0.1887	0.2532	0.2874	0.17116	0.5261
Rouge-Lsum	0.1885	0.2259	0.287713	0.17381	0.5246

RESULT COMPARATIVE ANALYSIS

The comparison of all the employed models by focusing on the Evaluation performance scores is presented in graphically.



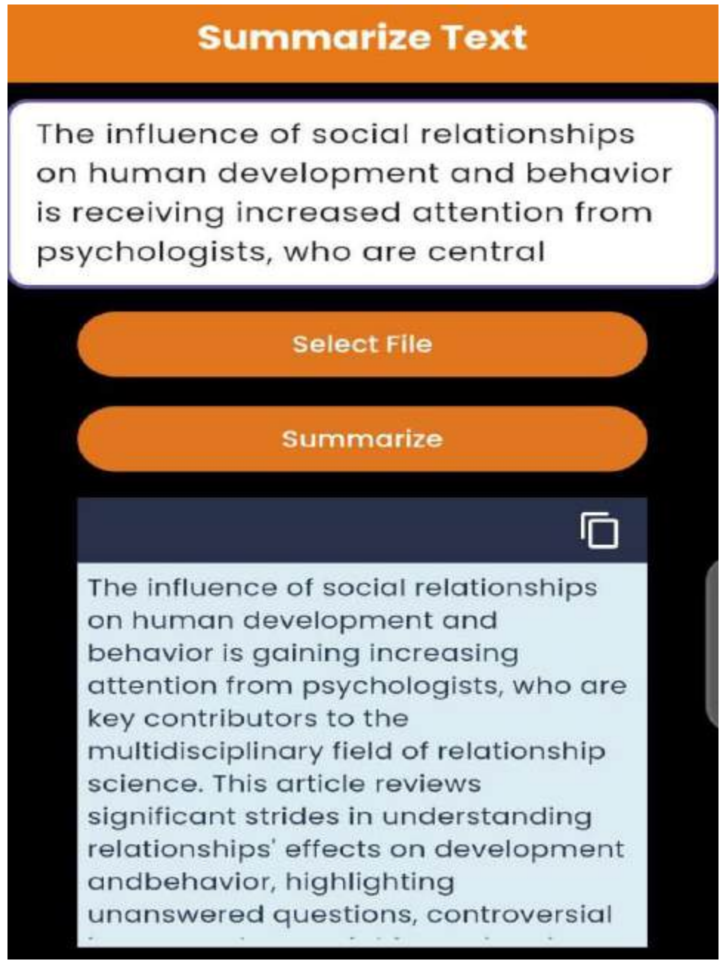
IMPLEMENTATION SETUP

LED outperformed the other models, with enhanced ROUGE scores. After this success, we hosted the model on the Hugging Face platform with the help of FastAPI which is a modern web framework for Python for creating APIs.

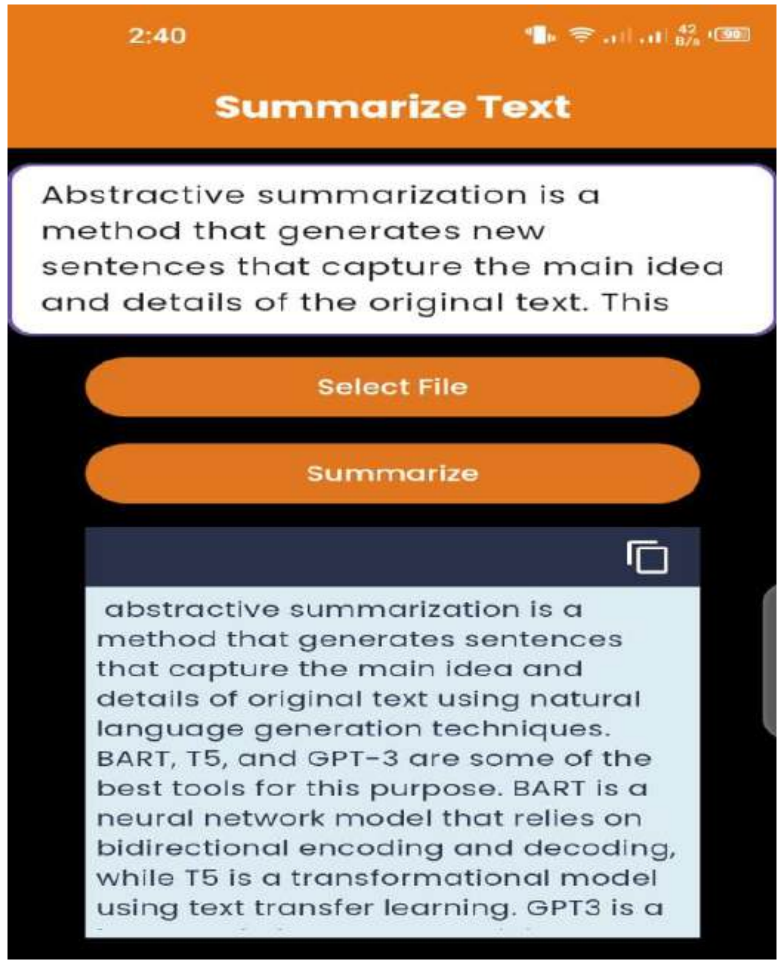
The developed mobile application using Flutter for cross-platform has a simple layout that enables easy inputting of text and provides summarized outputs. Key features include:

- **Graphical User Interface (GUI)**
- **Text Input and Summarization**
- **File Upload Feature**
- **FAQ**

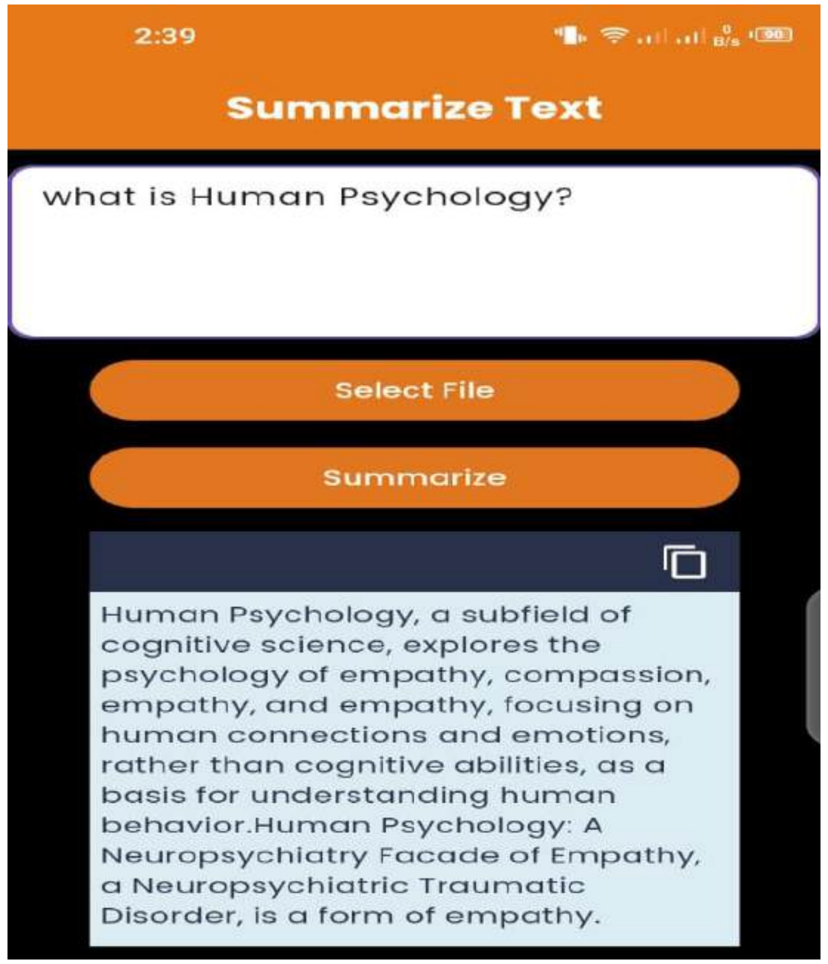
MOBILE APPLICATION



Text Input and Summarization



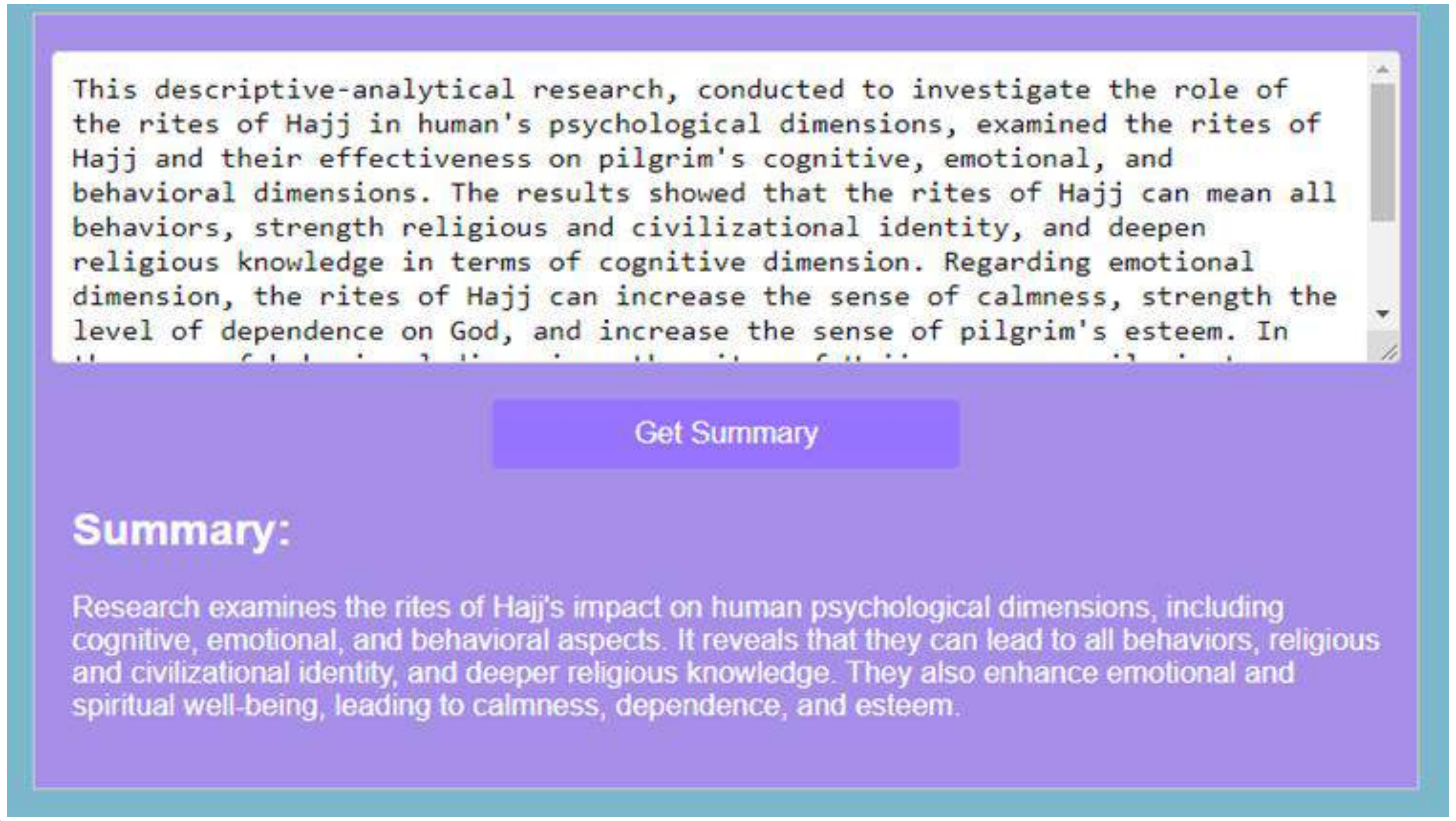
File Upload

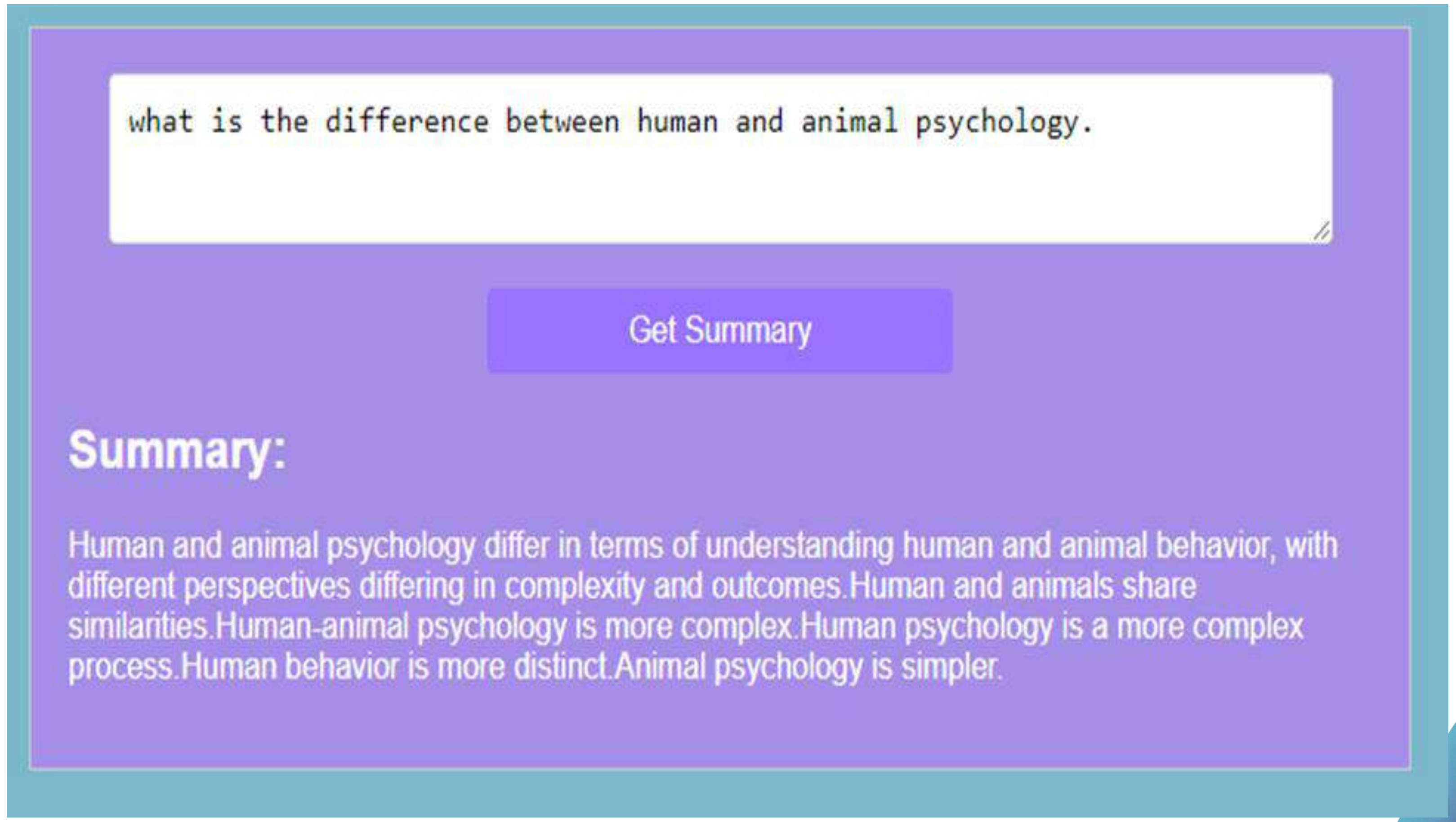


FAQ’s

WEB BASED APPLICATION

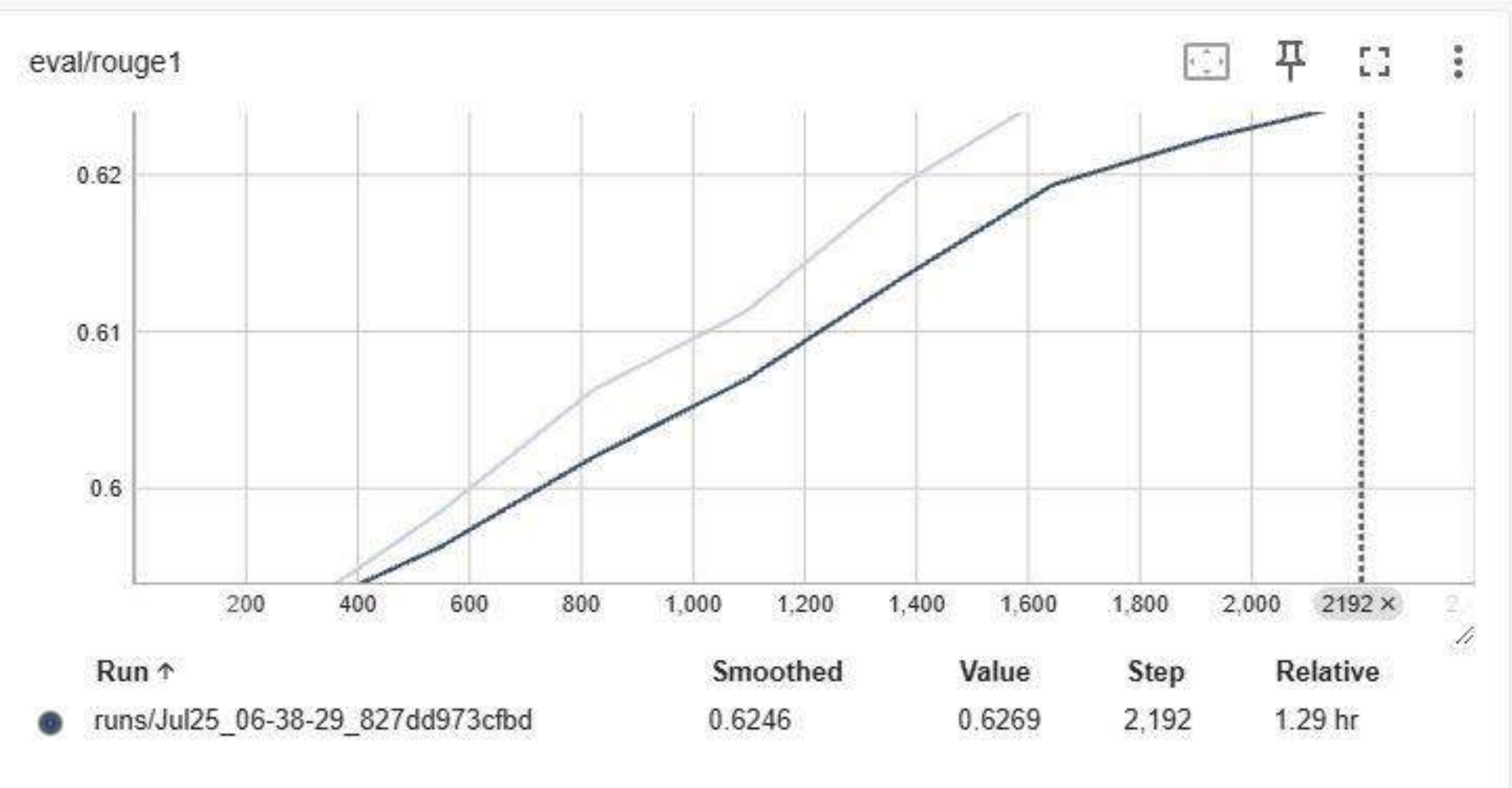
The text summarization system uses Python, Jupyter Notebook, and libraries like PyTorch, transformers by Hugging Face, BartTokenizer, and BartForConditionalGeneration. It uses an open-source deep learning framework, pandas, torch.data, and get_linear_schedule_with_warmup for training and summarizing the model. The model training and processing processes were accelerated using a GPU to handle computationally intensive tasks.





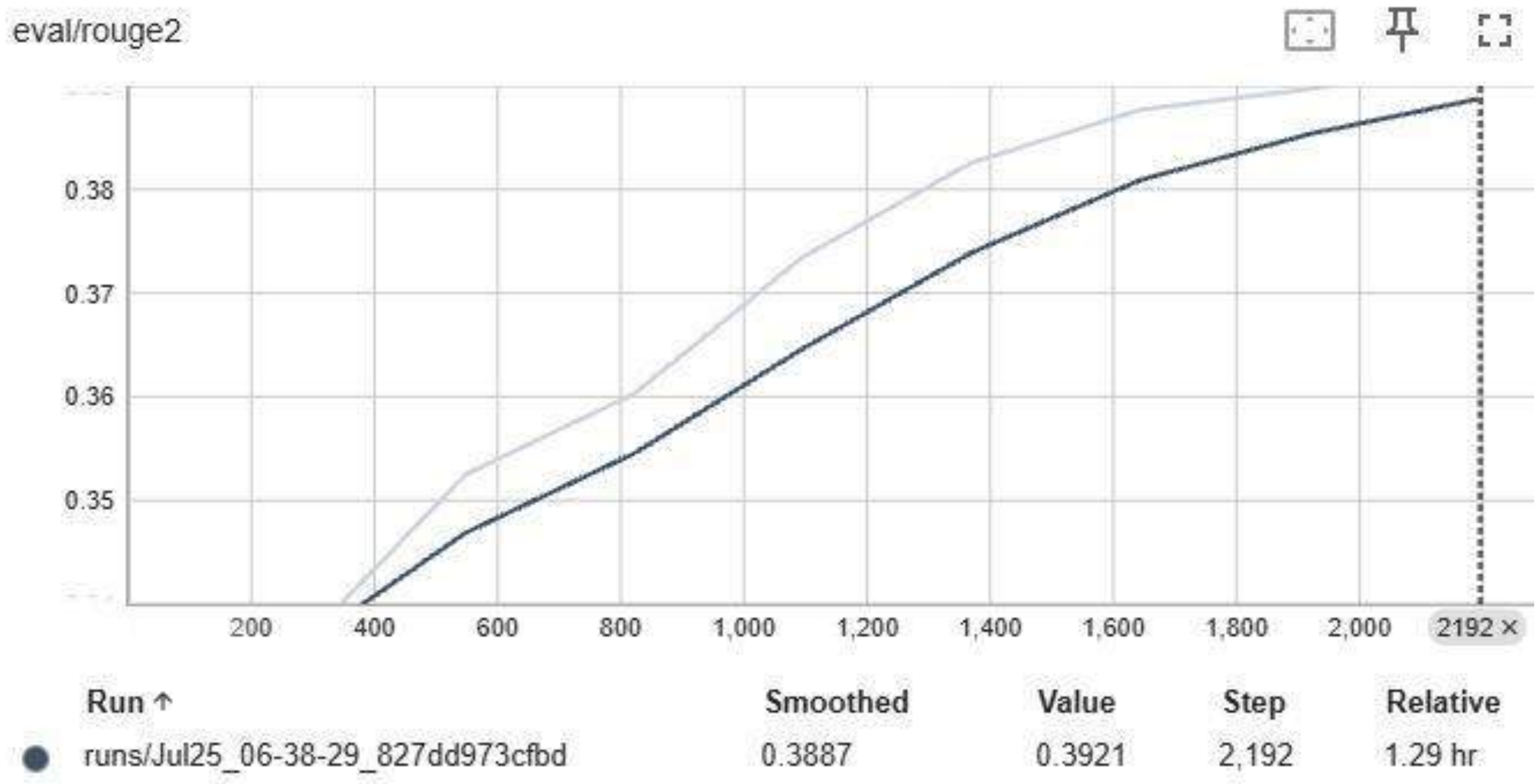
ROUGE 1 SCORES

- ROUGE-1 Scores



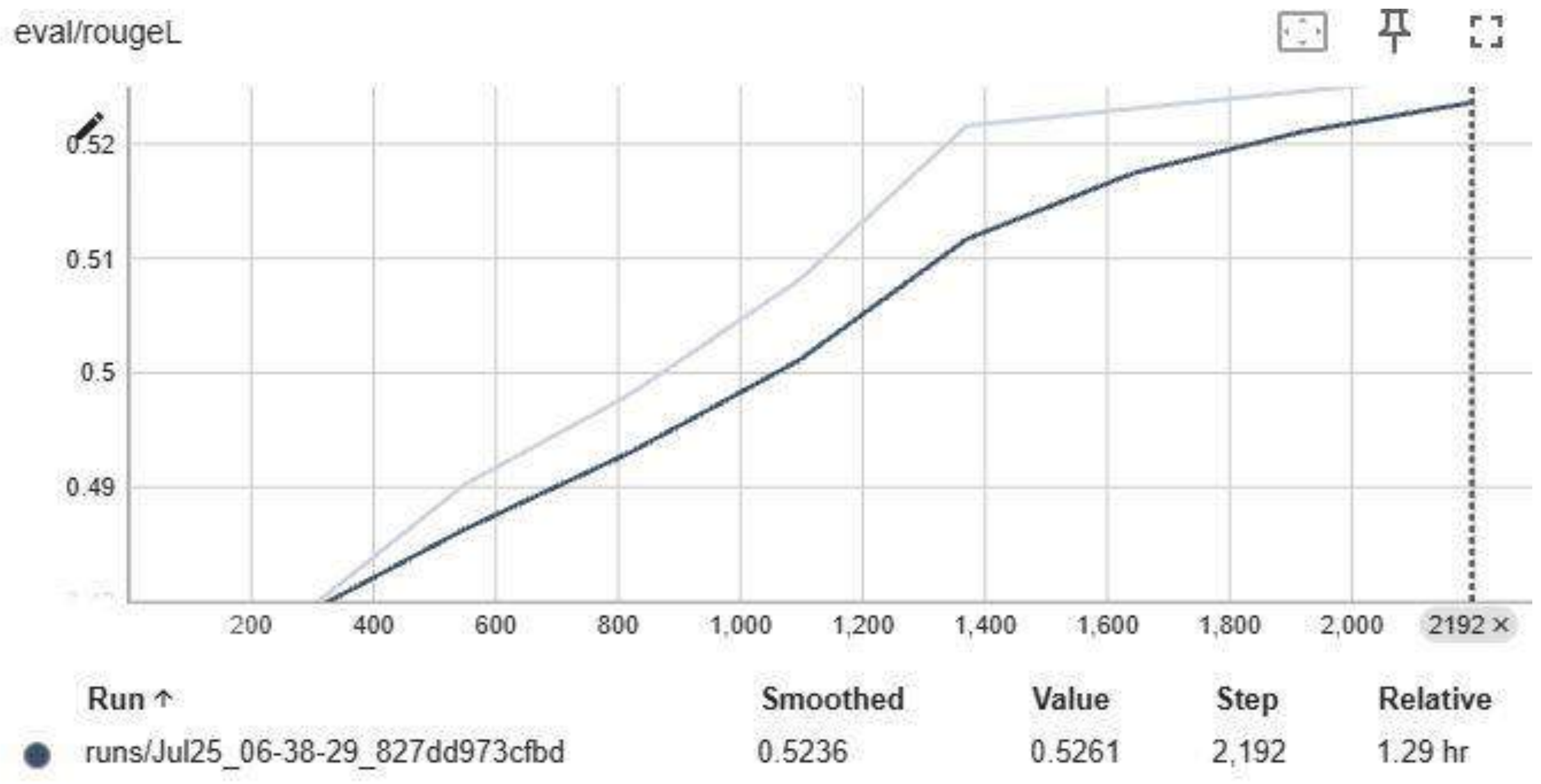
ROUGE 2 SCORES

ROUGE-2 Scores



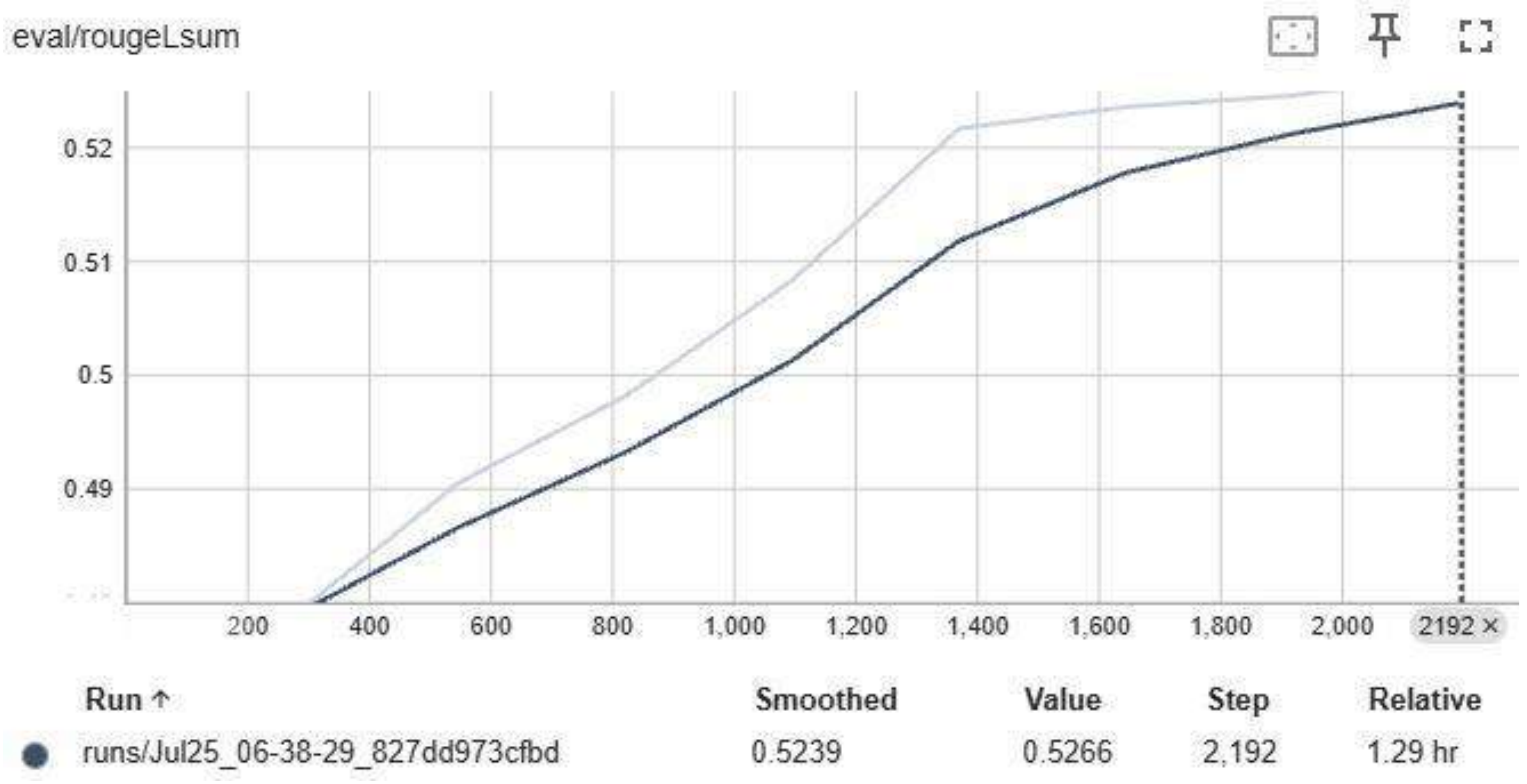
ROUGE-L SCORES

ROUGE-L Scores



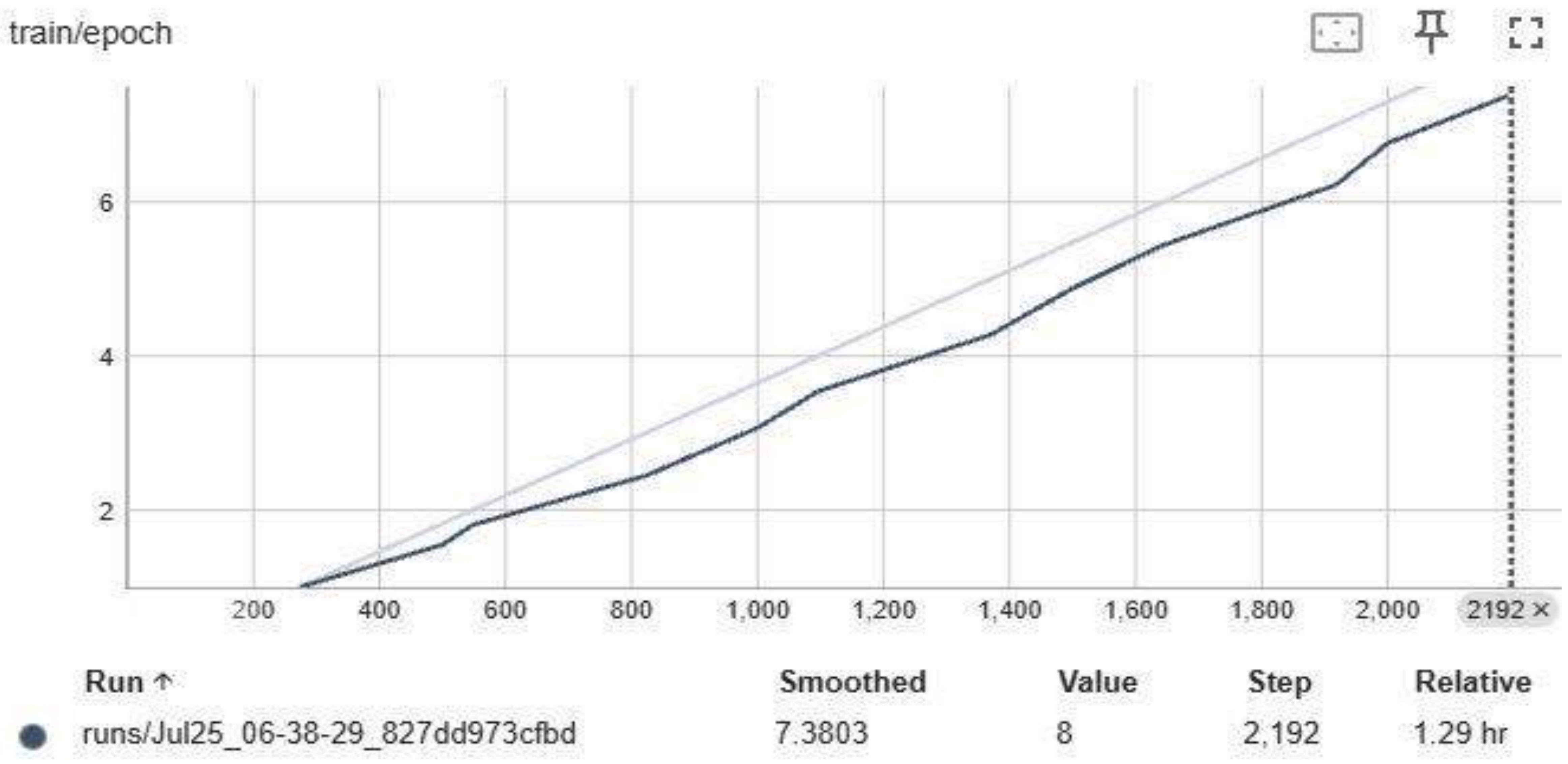
ROUGE LSum SCORES

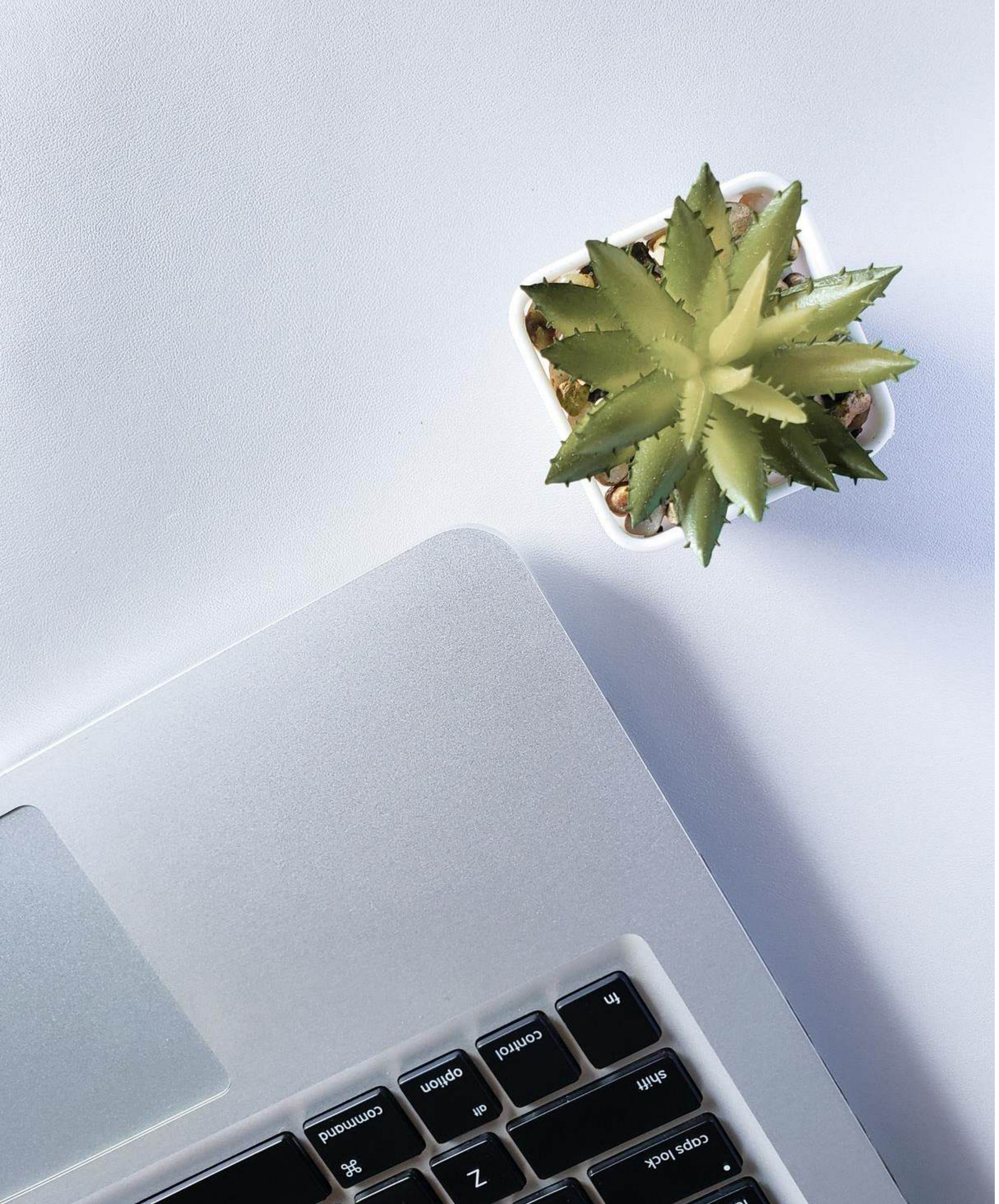
ROUGE LSum Scores



Number of Epochs

- As we see the increase in the number of EPOCHS basically increasing the rouge scores.





FUTURE WORK

- **Fine-tuning**: Further fine-tune the model on domain-specific datasets to enhance performance on specialized text types.
- **Parameter Optimization**: Explore different training configurations and hyperparameters to achieve higher ROUGE scores and more accurate summaries.
- **Dataset Expansion**: Increase the dataset size to allow the model to learn from a broader range of data, improving generalization.
- **Model Exploration**: Investigate additional models and fine-tune them to identify top performers based on evaluation metrics.

References

- N. Alami, M. Meknassi, N. En-nahnahi, Y. El Adlouni, and O. Ammor, “Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling,” Expert Syst. Appl., vol. 172, Jun. 2021, Art. no. 114652, doi: 10.1016/j.eswa.2021.114652.
- Karen Sparck Jones, " Automatic summarising: factors and directions " https://arxiv.org/abs/cmp-lg/9805011
- Karen Spa¨rck Jones, " Automatic summarising: The state of the art," https://www.sciencedirect.com/science/article/abs/pii/S0306457307000878.
- P. J. Liu and M. Lapata, "Attention-based neural generative models for text summarization," arXiv preprint arXiv:1611.09366, 2016.
- Lin, Chin-Yew. "ROUGE: A package for automatic evaluation of summaries." Text Summarization Branches Out (2004): 74-81 https://aclanthology.org/W04-1013.pdf.
- B. Khan, Z. A. Shah, M. Usman, I. Khan, and B. Niazi, "Exploring the Landscape of Automatic Text Summarization: A Comprehensive Survey," IEEE Access, vol. 11, pp. 109819-109840, 2023, doi: 10.1109/ACCESS.2023.3322188. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10272614&isnumber=10005208
- A. Vaswani et al., "Attention is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), 2017.


Thank You