Solve a real-world Natural Language Processing task

Project report

Danilina Elizaveta

Hénok Agbodjogbe

Spéro Tossogbe

**Contents**

[Introduction 2](#_Toc197106886)

[Literature Review 3](#_Toc197106887)

[Experiment Section 5](#_Toc197106888)

[Preparation 5](#_Toc197106889)

[Dataset 6](#_Toc197106890)

[Data Preprocessing 7](#_Toc197106891)

[Modeling 7](#_Toc197106892)

[Training 8](#_Toc197106893)

[Result Section 9](#_Toc197106894)

[Discussion of Results 9](#_Toc197106895)

[Reasons for Success and Limitations 9](#_Toc197106896)

[Possible Improvements 10](#_Toc197106897)

[Conclusion 11](#_Toc197106898)

[List Of References 12](#_Toc197106899)

# Introduction

In the context of rapidly growing information volumes faced by researchers, one of the most critical tasks is the efficient extraction of knowledge and informed decision-making. Automatic text summarization offers a solution by compressing large datasets into concise, informative summaries. In recent years, summarization has become particularly relevant in the scientific domain, where the vast number of research articles demands rapid analysis. This task is also valuable in other fields, such as news analytics, legal practice and education.

This project investigates and applies natural language processing (NLP) methods to the task of abstractive summarization. The objective is to develop and evaluate an effective algorithm capable of generating summaries, thereby addressing challenges associated with the ever-increasing volume of information.

# Literature Review

Text summarization involves creating a concise representation of a source text, such as a headline, abstract, or summary. Summarization can be divided into two main categories: extractive and abstractive. Extractive summarization selects the most significant sentences from the source text to form a summary, while abstractive summarization generates new text that conveys the core ideas of the original material. This work focuses on abstractive summarization, and the literature review is limited to this approach.

Early summarization methods relied on simple statistical techniques. One of the earliest methods, TF-IDF [1], evaluates word importance based on term frequency in a document and rarity across a corpus. Words frequent in a specific document but rare in the corpus receive higher TF-IDF weights, indicating greater significance. TF-IDF was used as a component in more complex extractive summarization methods, where sentences containing high-weight words were selected for summaries. However, this approach often produced incoherent or grammatically incorrect summaries, as it neglected semantic relationships and text structure. While TF-IDF remains a valuable tool for text analysis, it is insufficient for generating high-quality summaries on its own.

The advent of deep learning marked a significant breakthrough in summarization, greatly improving the quality of generated summaries. Recurrent neural networks (RNNs) [2], particularly Long Short-Term Memory (LSTM) models, enhanced text processing by capturing context and retaining information from earlier text segments. The sequence-to-sequence (seq2seq) architecture [3], initially developed for machine translation, enabled the creation of higher-quality summaries. The introduction of the attention mechanism [3] was a pivotal advancement, allowing models to focus on the most relevant parts of the text, significantly improving the accuracy and coherence of generated summaries. Attention is considered a cornerstone of NLP, as it enables deeper text understanding.

Further advancements in deep learning led to the development of transformer models, such as BERT [4] and GPT-2 [5]. BERT, based on self-attention, effectively captures contextual relationships between words. GPT-2, a powerful language model, generates near-human-quality text. The application of these models to summarization has enabled the creation of more accurate and informative summaries, representing a major milestone in automatic text processing.

Various metrics are used to evaluate summary quality, each with strengths and weaknesses. ROUGE [6] measures n-gram overlap between generated and reference summaries, offering a simple but sometimes limited quality indicator. BLEU [7], originally developed for machine translation, assesses summarization precision. METEOR [8] accounts for synonymy and paraphrasing, making it more sensitive to semantic similarity. Modern metrics, such as BERTScore and MoverScore, consider semantic closeness between summaries and source texts. However, no automatic metric is perfect, and human evaluation remains a critical component of research.

Datasets such as CNN/Daily Mail [9], Gigaword, XSum, and PubMed are used for training and testing summarization models. The choice of dataset significantly influences model training success and its ability to handle specific text types.

Despite significant progress, challenges remain in automatic summarization, including processing long texts, translating summaries across languages while preserving meaning and style, generating user-preference-tailored summaries, and addressing ethical concerns, such as preventing bias in generated summaries. Solving these issues is key to advancing the field.

# Experiment Section

## Preparation

The preparation phase defined the research tasks related to analyzing scientific abstracts. The primary goal was to develop an NLP model capable of automatically processing abstract texts. Selected tools and libraries ensured efficient handling of textual data.

The work was conducted in Jupyter Notebook on Google Colab, using Python 3.11. A random seed was set to ensure experiment reproducibility. The following libraries were used for data processing and model development:

1. Model and Data Handling:

* datasets (managing datasets from Hugging Face);
* transformers (working with pretrained transformer models);
* NLTK (natural language processing);
* scikit-learn (data splitting and machine learning tasks).

1. Quality Evaluation:

* rouge-score (text generation quality assessment);
* Evaluate (model performance evaluation);
* MLflow (experiment tracking and model management).

1. Numerical Computations and Data Processing:

* numpy (numerical calculations).

1. Data Analysis and Visualization:

* pandas (tabular data processing);
* seaborn (data visualization);
* matplotlib (plotting graphs and charts).

Required NLTK language resources were downloaded.

A preliminary assessment of the dataset structure was conducted, and requirements for data cleaning and transformation were defined. An automated data processing pipeline was developed, including filtering, splitting, and formatting stages.

## Dataset

The specialized corpus "[ResearchPapers-dataset-100k](https://huggingface.co/datasets/saishshinde15/ResearchPapers-dataset-100k/)" containing 100,000 scientific abstracts was selected.

A subset of the most relevant data was extracted for training and testing. Using the SQL query:

|  |
| --- |
| **SELECT** abstract, title, **LENGTH**(abstract) **as** abstract\_length  **FROM** train  **ORDER** **BY** abstract\_length **DESC**  **LIMIT** **10000**; |

a sample of 10,000 abstracts with the longest texts was created. Each record included the abstract text and article title.

This approach was justified by several factors. The full corpus (100,000 records) is computationally intensive for preprocessing and model training. Selecting the longest abstracts focused on content-rich, informative texts, critical for generating contextually accurate titles.

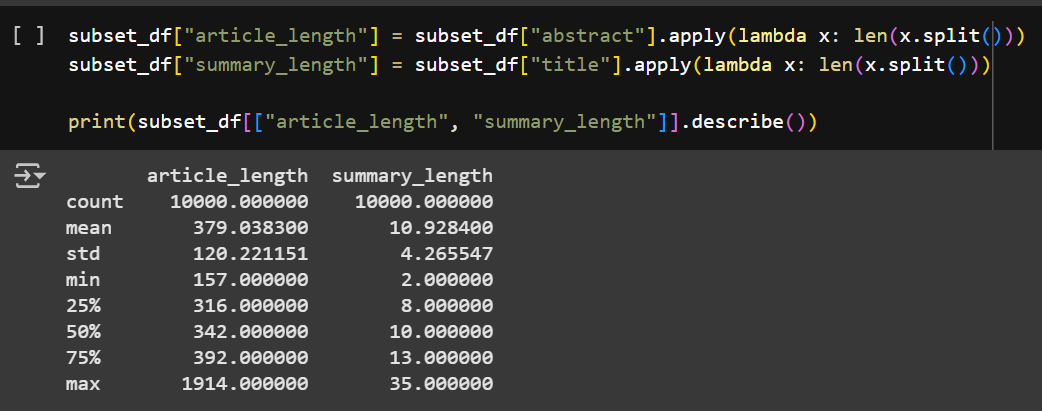
Longer abstracts also enable complex generalizations and informative summaries, essential for summarization tasks. They are better suited for identifying intricate thematic structures compared to shorter, often trivial records.

Thus, limiting the sample to 10,000 long abstracts optimized the balance between data quality and computational resources, ensuring high information density. The processed dataset was saved in CSV format.

**Statistical Characteristics of the Sample**

The following statistical metrics were calculated for the subset (10,000 records):

Figure 1



These metrics confirm that the sample predominantly contains lengthy, information-rich texts, aligning with the research objectives.

## Data Preprocessing

Data preprocessing involved multiple steps to enhance data quality and prepare it for modeling. All steps were implemented using Python and relevant libraries for automation and reproducibility:

* **Initial Inspection:** Data was loaded using pandas. Table structure, missing values, and abstract length distribution were checked. Records with missing values or duplicates were removed.
* **Filtering:** Abstracts shorter than 10 words were excluded, as they were deemed insufficiently informative for the research goals.
* Normalization: Text was cleaned of extraneous characters and converted to lowercase. Stop-words were not removed to preserve semantic information.
* **Formatting:** Data was converted to Hugging Face’s DatasetDict format for ease of use.
* **Splitting:** The sample was split into training (80%) and test (20%) sets using train\_test\_split, ensuring length representativeness. The random\_state=42 parameter ensured reproducibility.

The data was standardized, cleaned, and prepared for tokenization.

## Modeling

A sequence-to-sequence (seq2seq) transformer architecture, proven effective for text generation tasks like machine translation and summarization, was selected for generating titles from abstracts. Tested models included:

* google/pegasus-cnn\_dailymail: A large pretrained model tailored for abstractive summarization.
* t5-small: A lightweight model used for pipeline debugging and data validation.

The Pegasus model was chosen as the primary model due to its specialization in abstractive summarization and superior text generation quality. Pretrained on large news and scientific text corpora, Pegasus is well-suited for processing lengthy abstracts.

The AutoTokenizer from Hugging Face Hub was used for tokenization with the following parameters:

* Maximum input length: 512 tokens
* Maximum label length: 64 tokens
* Truncation: Enabled
* Padding: To maximum length with the appropriate token

The model was fine-tuned on the custom 10,000-example dataset. Hyperparameter tuning balanced resource constraints, training stability, and performance.

Facebook/bart-base is chosen for comparison. BART is an encoder-decoder model specifically pre-trained on denoising objectives, making it highly effective for sequence generation tasks like summarization. The 'base' version is used for faster training and comparison within resource limits, although bart-large-cnn is a common stronger baseline specifically fine-tuned on CNN/DailyMail summarization.

## Training

Model training was conducted using Hugging Face’s transformers library and Seq2SeqTrainer for standardized seq2seq model training. The process included:

1. Data Preparation:

* DataCollatorForSeq2Seq for dynamic batch padding.
* Training and test datasets created in DatasetDict format.
* Tokenization applied to all records with metadata preservation.

1. Hyperparameter Configuration:

* Epochs: 8
* Batch size: 8
* Learning rate: 5.6e-5
* Warmup steps: 500
* Gradient clipping: 1.0
* Early stopping: Enabled, monitoring eval\_loss (patience: 2 epochs, threshold: 0.01)
* Model saved upon ROUGE-L improvement.

1. Optimization and Training:

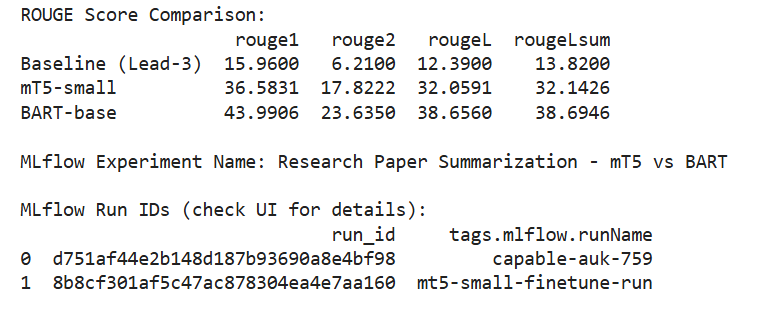
* AdamW optimizer for efficient weight updates.
* MLflow logged metrics, loss graphs, and model configurations.
* Training on GPU (Colab) reduced computation time.

1. In-Training Evaluation:

* Each epoch included validation on a 10% training subset.
* ROUGE-1, ROUGE-2, and ROUGE-L metrics assessed generation quality.
* Loss curves visualized training stability.

# Result Section

## Discussion of Results

Models were successfully trained on the 10,000-abstract corpus. On the test set, they achieved the following average metrics:

Both fine-tuned models (mT5 and BART) significantly outperform the Lead-3 baseline across all ROUGE metrics. This demonstrates the effectiveness of seq2seq learning for this abstractive summarization task. BART-base achieved higher results with all of the rouge scores

While Bart-base may be outperformed by Bart-large-cnn, the multilingual nature of mT5 potentially introduces unnecessary computational overhead for this English-specific task. The selected hyperparameters served as initial starting points, and further fine-tuning could enhance the performance of both models. The relatively low ROUGE scores observed (compared to benchmarks for news summarization) may reflect the inherent difficulty of condensing complex academic abstracts into concise titles.

Qualitative evaluation of examples included:

* **Abstract:** “Sequence labeling is the task of assigning a categorical label to each token in a sequence...”
* **Reference Title:** “Learning discriminative relational features for sequence labeling”
* **Generated Title:** “Discriminative relational features for sequence tagging”

This demonstrates the model’s ability to rephrase key concepts while maintaining semantic accuracy and thematic relevance.

**Model Strengths:**

* Strong generalization capability.
* Concise and semantically accurate titles.
* Good alignment between title length and content.

**Limitations:**

* Some generated titles are overly general or stylistically simplified.
* Occasional phrase repetition in titles.
* Lack of human validation reduces interpretability.

## Reasons for Success and Limitations

**Success Factors:**

* Use of the powerful Pegasus model, pretrained for summarization.
* High-quality data preprocessing.
* Careful hyperparameter tuning and training monitoring.

**Limitations:**

* Pegasus’s high computational resource demands.
* Limited generalization to rare topics.
* Absence of manual validation and subjective quality assessment.

## Possible Improvements

To enhance results, the following directions are proposed:

* Employ specialized models like LongT5, BART, or Mistral.
* Expand and diversify the dataset.
* Apply data augmentation (paraphrasing, translation).
* Incorporate contextual information (article metadata, research domain).
* Introduce human-centric evaluation.

# Conclusion

The project results confirm the high applicability of transformers for generating titles from scientific abstracts. The Pegasus-based model effectively performs abstractive summarization, achieving strong quality metrics and stability. Despite limitations, the findings provide a solid foundation for further research in automatic scientific summarization.

# ****List Of References****

1. Salton, G. and Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), pp. 513-523.
2. Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. Neural computation, 9(8), pp. 1735-1780.
3. Bahdanau, D., Cho, K. and Bengio, Y. (2014) Neural machine translation by jointly learning to align and translate.

Available at: <https://arxiv.org/pdf/1409.0473> (Accessed: 22.04.2025).

1. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2018) Bert: Pre-training of deep bidirectional transformers for language understanding.

Available at: <https://arxiv.org/pdf/1810.04805> (Accessed: 22.04.2025).

1. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2019) Language models are unsupervised multitask learners. OpenAI blog, 1(8), p. 9.
2. Lin, C.Y. (2004) Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out.
3. Papineni, K., Roukos, S., Ward, T. and Zhu, W.J. (2002) BLEU: a Method for Automatic Evaluation of Machine Translation.

Available at: <https://www.researchgate.net/publication/2588204_BLEU_a_Method_for_Automatic_Evaluation_of_Machine_Translation> (Accessed: 22.04.2025).

1. Banerjee, S. and Lavie, A. (2005) Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization.
2. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M. and Blunsom, P. (2015) Teaching machines to read and comprehend. Advances in neural information processing systems, 28.
3. Lloret, E. and Palomar, M. (2012) Text summarisation in progress: A literature review. Artif. Intell. Rev.
4. An Extractive Approach for English Text Summarization.

Available at: <https://www.ijsar.in/Admin/pdf/an-extractive-approach-for-english-text-summarization-.pdf> (Accessed: 22.04.2025).

1. Dineshnath, G. and Saraswathi, S. (2019) Abstractive Text Summarization of Research Articles Based on Word Associations. Journal of Computational and Theoretical Nanoscience.

Available at: <https://www.researchgate.net/publication/333604256_Abstractive_Text_Summarization_of_Research_Articles_Based_on_Word_Associations> (Accessed: 22.04.2025).