USTH
VIETNAM FRANCE UNIVERSITY

Group Project Report: Text Summarization

Group ID: 21

Student Name	Student ID
Nguyen Tuan Dung	22BI13107
Tran Duc Duong	22BI13117
Vu Hung Anh	**22BI13045**
Truong Dai An	22BI13008
Tran Trong Nghia	22BI13332
Nguyen Thanh Nam	22BI13325

Contents

1 Introduction

This project aims to develop a practical use of a Natural Language Processing (NLP) model that performs Text Summarization to generate concise and meaningful summaries from lengthy texts. The tool enhances productivity by reducing reading time and enabling quick comprehension of large amounts of information.

2 Project Objectives

The primary objectives of this project are:

- Implement a user-friendly web interface for text input and summary output.
- Fine-tune and deploy an NLP model for high-quality summarization.
- Develop a backend that processes text input and returns real-time summaries.
- Ensure system flexibility to accommodate different models and datasets.

3 Target Audience

The application targets:

- **Students & Researchers**: Quickly extract key points from academic materials.
- **Professionals**: Save time by summarizing lengthy reports and emails.
- **Journalists & Writers**: Efficiently distill essential information from news and articles.
- **General Public**: Easily digest long-form content such as articles and blogs.

4 Methodology

The project is implemented using modern web development frameworks and NLP models for seamless summarization, covering both frontend and backend development along with model fine-tuning.

4.1 Backend Development

- Built using Flask, a lightweight Python web framework.
- Provides an API that accepts text input, processes it with the summarization model, and returns a JSON summary.
- Uses the `flask_cors` library for cross-origin requests.
- Runs on port 5000 and is started using `python app.py`.

4.2 Frontend Development

- Developed with Node.js (for dependency management) and React for the user interface.
- Offers a simple UI with a textarea for text input, a ”Summarize” button, and a ”Clear” button.
- Uses the fetch API to communicate with the backend in real-time.
- Utilizes `useState` from React to manage state.

4.3 Dataset

- Utilizes the SAMSum dataset from Hugging Face, consisting of conversational dialogues and summaries.
- Ideal for training NLP models on summarization tasks, especially with conversational data.

4.4 NLP Model

- Employs the PEGASUS model (pegasus-samsum) as the primary summarization model.

4.5 Fine-Tuning Process

- **Load Dataset** – Import and preprocess the SAMSum dataset.
- **Tokenization** – Tokenize text using the Hugging Face AutoTokenizer.
- **Trainer Definition** – Set model parameters and training configurations.
- **Training** – Fine-tune the PEGASUS model on the dataset.
- **Model Saving** – Save the trained model for inference.
- **Evaluation** – Evaluate using metrics such as ROUGE scores.

5 Achieved Results

The project successfully meets its objectives by delivering a functional text summarization web application:

5.1 Functional Outcomes

- **Backend API**: Processes text and returns summaries using a fine-tuned PEGASUS model.
- **Frontend UI**: Provides an intuitive interface for text input and instant summary display.

- **Flexible Model Usage**: Supports multiple summarization models for diverse tasks.

5.2 Performance and Accuracy

- Produces high-quality, coherent summaries that capture essential information.
- Fine-tuning enhances performance, ensuring relevance and fluency.
- Evaluation with ROUGE scores confirms efficient retention of key details.

5.3 Deployment Readiness

- The backend API can be hosted on a cloud server or containerized via Docker.
- The frontend can be deployed on platforms like Netlify, Vercel, or Firebase Hosting.
- Scalability options include database integration for storing user-generated summaries.

6 Future Enhancements

- **Multilingual Support:** Incorporate models like mT5 for summarizing non-English texts.
- **External Integration:** Develop plugins for Google Docs, MS Word, and browser extensions.
- **Model Customization:** Allow users to choose models and adjust summary length dynamically.
- **Mobile Development:** Create a mobile-friendly version for broader accessibility.

7 Conclusion

The Text Summarization Web Application efficiently extracts key insights using a Flask backend and a React frontend with the PEGASUS model. It offers a user-friendly experience, high-quality summaries, and flexibility for future enhancements such as multilingual support and mobile integration.

8 References

- Hugging Face Transformers
  https://huggingface.co/transformers/
- SAMSum Dataset
  https://huggingface.co/datasets/samsum
- PEGASUS Paper
  https://arxiv.org/abs/1912.08777