NLP with Transformers: Dialogue Summarization Project

Veysel Berk

Date: 27 October- 8 November 2025

Business Context & Stakeholders

Acme Communications is exploring AI-powered features to improve user experience and reduce information overload in group messaging. This project supports that goal by developing a prototype for automated dialogue summarization, helping users quickly catch up on missed conversations without scrolling through every message.
The business goals:

- Reducing information overload for users
- Improving engagement by making conversations more accessible
- Enhancing the platform with AI-powered features

Key stakeholders:

- The **product team**, focused on user experience and retention
- The **engineering team**, responsible for building and integrating the solution
- The **end users**, who want a simple, fast way to stay updated without reading every message

1. Business Problem and Proposed Solution

1.1 Problem Description

Group chats often become long and hard to follow, especially when many people are active. When users step away for a few hours, they may return to hundreds of new messages with no easy way to find what matters. Important updates can get buried under casual chatter, leading to missed information and disengagement.

Multiple studies in 2024–2025 show the scale of this issue: **55% of users report feeling overwhelmed by notifications** as the top reason for digital detox, and **71% uninstall or mute messaging apps due to excessive notifications**. In group chats, high message volume makes it harder to focus, increasing mental load and reducing productivity by almost 30%. These trends show a clear opportunity to build smarter tools that help users catch up quickly and stay engaged without the frustration of scrolling through long threads.

1.2 Impact Assessment

When users feel overwhelmed by long conversations, they often disengage, mute notifications, or even uninstall the app. This directly affects user retention, satisfaction, and time spent on the platform. Competitors are already investing in AI tools that make digital communication simpler and more personal. If Acme Communications can offer an effective dialogue summarization feature, it would not only improve user experience but also help the company stay competitive in a growing market.

1.3 Solution Vision

The goal is to create an **automated dialogue summarization feature** that gives users a quick, clear summary of what they missed. This would make conversations easier to follow, reduce mental fatigue, and encourage users to rejoin discussions.
For Acme Communications, this solution adds real value by making the platform smarter, more user-friendly, and competitive. It also opens the door to premium features like smart recaps or priority message highlights.

1.4 Success Criteria

A successful solution should generate summaries that are both accurate and concise. I will use **ROUGE scores** to evaluate the quality of generated summaries compared to human-written references. From a business perspective, success could also be measured by improved user engagement, such as faster catch-up times or increased retention in group chats. Technically, the system should deliver results efficiently without requiring heavy infrastructure, making it suitable for real-world deployment.
To define clear success criteria:

- Target ROUGE-1 ≥ 0.40 and ROUGE-L ≥ 0.35 on validation summaries
- Reduce average user catch-up time by at least 10%
- Achieve 5% improvement in engagement (measured by message reads or replies)
- Generate summaries in under 2 seconds per dialogue during testing

2. Problem Solving Process

2.1 Process Framework

Step 1: Input
Load the SAMSum dataset

Step 2: Data Exploration & Preparation
Inspect the dataset structure, and understand dialogue and summary formatting. Perform cleaning, tokenization, and create training-validation splits.

Step 3: Model Architecture Design & Selection
Evaluate multiple encoder–decoder model options using pretrained transformers, such as BERT + GPT-2, BART, and T5. Based on available resources and early results, select the best-performing architecture for fine-tuning on the SAMSum dataset.

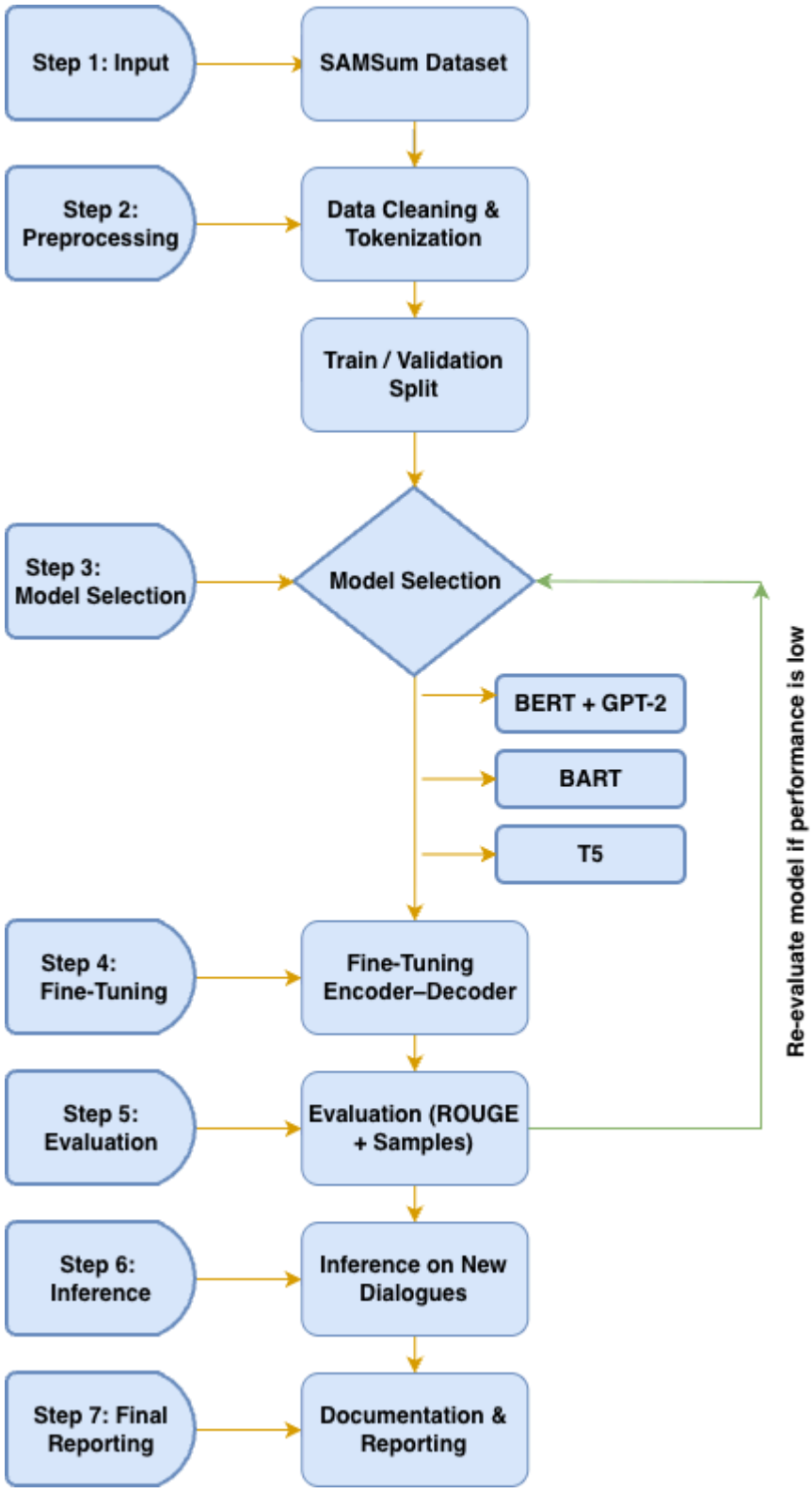
Step 4: Fine-Tuning Encoder–Decoder
Fine-tune and train the model on the dataset. Implement early stopping, checkpoint control, best model saving and other optimization strategies to improve convergence and prevent overfitting.

Step 5: Evaluation
Evaluate the model using ROUGE metrics and qualitative analysis. Compare generated summaries to human references to assess accuracy, fluency, and relevance.

Step 6: Inference & Output Sampling
Generate sample summaries on unseen conversations to test real-world behavior.Explore decoding strategies like beam search for better fluency.

Step 7: Reporting & Documentation
Document the process, results, and insights. Create visual examples, share limitations, and prepare materials for stakeholder review and future deployment planning.

2.2 Conceptual Representation



2.3 Methodology Justification

I chose a **transformer-based encoder–decoder model** because it performs well on abstractive summarization tasks.

- **BERT** works as an encoder to understand the full dialogue context.
- **GPT-2** serves as the decoder to generate fluent, natural-sounding summaries.
- Models like **BART** and **T5** offer similar architectures and can be used for comparison.

Using pre-trained models is efficient, they already understand language patterns, so fine-tuning them on dialogue data requires less time and compute power than training from scratch.

To measure quality, I'll use **ROUGE metrics** for quantitative evaluation and human review for qualitative feedback.

I also plan to experiment with **beam search** and **early stopping** to balance fluency, accuracy, and efficiency.

To summarize:

- **BERT as encoder, GPT-2 as decoder:** Capture dialogue context and generate fluent, natural-sounding summaries.
- **Pre-trained transformers:** improve performance with limited data and compute.
- **ROUGE and human evaluation:** balanced evaluation.
- **Beam search and early stopping:** support fluent generation and stable training.

2.4 Alignment with Requirements

This approach meets the project's business and technical requirements by building a full summarization pipeline, from data preparation to evaluation and reporting.
The use of pre-trained transformer models helps me build a working solution within realistic time and resource limits, while still targeting high-quality results.
The business need for reduced information overload is addressed directly through automated summaries that help users catch up quickly.
The solution is also flexible, opening the door to premium features in the future by balancing accuracy, efficiency, and clarity, this project delivers both technical value and a clear path toward product integration.

3. Timeline and Scope

3.1 Research & Preparation (Oct 27 – Oct 29)

The first phase focuses on understanding the SAMSum dataset, researching encoder–decoder architectures, and reviewing evaluation methods for summarization. I'll explore how different models handle dialogue-style input and finalize which ones to test. This stage also includes setting up the development environment and any required preprocessing steps.

3.2 Implementation Phases

Data Preprocessing and Exploration (Oct 30)

Explore the data and finalize preprocessing steps (cleaning, tokenization, train/validation split).

Model Architecture Implementation(Oct 31 – Nov 2)
Implement encoder–decoder setup using selected pre-trained models. Start with BERT + GPT-2 as the primary setup, with optional comparison to BART or T5-small.

Training and Optimization(Nov 3 – Nov 5)
Train and fine-tune the model using early stopping, best model saving etc. and decoding strategies (e.g., beam search). Monitor results and adjust parameters as needed.

Evaluation and Analysis(Nov 6)
Run ROUGE evaluation and review generated summaries for quality and readability.

Documentation and Presentation(Nov 7 – Nov 8)
Prepare the final report, sample outputs, and short presentation for submission.

3.3 Iteration Points

I've included time for iteration during the training and evaluation phases. If early results show low performance, I will revisit model choice, try different decoding strategies, or adjust preprocessing steps.
Feedback from peers or instructors will also be reviewed before final submission, with time allocated to refine outputs, clarify documentation, or improve model behavior based on insights.

3.4 Risk Management

The main risk is limited compute power, since fine-tuning transformer models can be resource-intensive. To manage this, I will use smaller model variants (like distilBART or T5-small) during testing, and optimize batch sizes and training time.
Another potential challenge is generating low-quality summaries. If this happens, I'll analyze outputs early and adjust preprocessing or training strategies. I've kept some flexibility in the timeline to handle any last-minute issues or changes.

3.5 Final Delivery Schedule

- **Project Critique Submission (MVP Discussion):Nov 5**
  Share the near-complete version of the project with sample outputs, ROUGE results, and reflection for peer feedback.
- **Final Implementation Complete: Nov 7**
  Finalize model training, evaluation, and notebook cleanup.
- **Presentation and Report Submission: Nov 8**
  Submit written project report (PDF) and 5-minute presentation video, along with the complete Jupyter notebook.