Project Pitch Report: Automated Conversation Summarization

Problem Statement:
Acme Communications faces a key challenge: information overload in group chats. Lengthy conversations often become difficult to follow, and important details are lost in the noise, especially for users who step away temporarily. This problem reduces user satisfaction and overall platform engagement.

Approach:
Develop a dialogue summarization prototype using state of the art NLP models to condense long conversations into concise, accurate summaries that retain essential context, decisions, and action items. The prototype emphasizes both technical feasibility and measurable business value.

Indicative Metrics from Public Datasets:
Conversation Length: SAMSum shows ~20 dialogue turns on average with many exceeding 40 turns.
Speaker Diversity: Dialogues commonly involve 2 to 5 speakers complicating information attribution.
Information Density: DialogSum indicates key decisions are often mid dialogue requiring full thread scanning to catch up.
Summary Compression: Human summaries in both datasets reduce text length by ~80 to 90 percent while preserving core meaning.

Proposed 5 Step Process

Step 1: Data Exploration and Preparation
Ingest SAMSum and DialogSum, profile turns and speakers, remove noise, add speaker or role tags, create train val test splits, and prepare long dialog chunking windows.

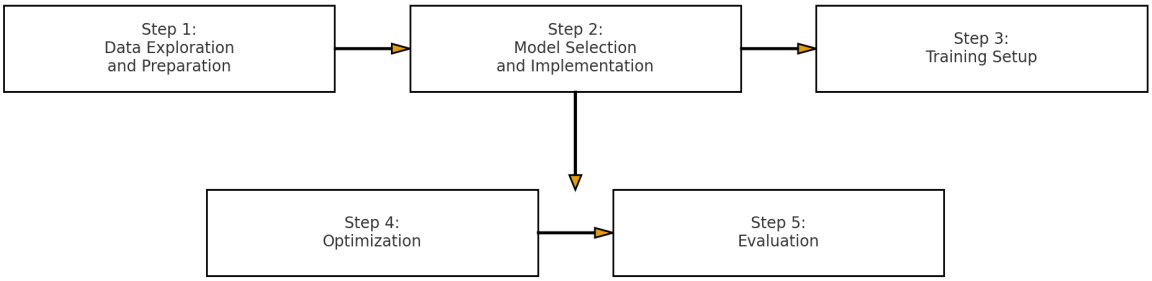
Step 2: Model Selection and Implementation
Start with BART or T5 encoder decoder for supervised abstractive summarization. Implement ChatGPT auto regressive prompting as a baseline and for qualitative comparisons.

Step 3: Training Setup
Tokenization, max length strategy with sliding windows, mixed precision, gradient clipping, label smoothing for example 0.1, and learning rate schedule with warmup and cosine or linear decay.

Step 4: Optimization
Hyperparameter search such as learning rate, batch size, max target length, early stopping on ROUGE L, checkpointing best validation model, and selective layer unfreezing if needed.

Step 5: Evaluation
Automatic metrics such as ROUGE 1, ROUGE 2, ROUGE L, and BERTScore plus human evaluation rubric for coverage, faithfulness, readability, and actionability. Error taxonomy for hallucination versus omission.

Solution Flowchart Data and Processing Pipeline



Rationale for Key Choices

BERT Based Encoder Decoder Models and ChatGPT Auto Regressive Modeling:
Encoder decoder architectures such as BART and T5 are highly effective for abstractive summarization. They combine powerful text encoding with controlled generation that can compress long inputs into fluent and faithful summaries. Auto regressive models such as ChatGPT serve as strong baselines for few shot summarization and allow direct comparison for readability and coherence.

Fine Tuning Pre Trained Models vs Training from Scratch:
Fine tuning pre trained models allows leveraging billions of tokens of prior knowledge, drastically reducing training time and compute while improving generalization. Training from scratch would require massive proprietary corpora and high costs without guaranteed performance benefits.

Evaluation Metrics ROUGE and Human Evaluation:
ROUGE 1, ROUGE 2, and ROUGE L provide reliable measures of lexical overlap with human summaries, while BERTScore captures semantic similarity. Human evaluation is critical for assessing qualities like faithfulness, coverage, and readability that automated metrics cannot fully capture.

Optimization Techniques:
Key techniques include learning rate warmup with cosine or linear decay, AdamW optimizer with weight decay, gradient clipping to prevent instability, and label smoothing to improve generalization. Data specific strategies such as turn aware chunking and speaker role embeddings further optimize model performance for dialogue summarization.

Business Alignment and Deliverables

Fulfills Deliverables: This report provides the problem statement and approach, a condensed 2 week timeline, dataset analysis, model architecture, rationale for choices, evaluation plan, and a visual pipeline flowchart.
Addresses Core Needs: Summaries reduce cognitive load, surface key decisions, and make catching up fast, directly tackling information overload.
Balanced Performance and Practicality: Latency and resource targets guide an efficient, scalable prototype with safety and monitoring built in.
Meaningful Outputs: Summaries framed to highlight action items, decisions, and owners make the feature useful for users and differentiating for the platform.

Condensed Two Week Timeline

Phase	Duration	Activities
Phase 1: Data Preparation	Day 1 to 3	Explore and clean data. Add role tags. Create splits.
Phase 2: Model Development	Day 4 to 7	Set up and fine tune BART or T5. Establish ChatGPT baseline.
Phase 3: Evaluation	Day 8 to 10	ROUGE and BERTScore. Human evaluation rubric.
Phase 4: Prototyping	Day 11 to 14	API and UI prototype. Latency and usability tests.