Project 3 Pitch – Dialogue Summarization for Acme Communications

1. Problem Description

Acme Communications users increasingly experience information overload in fast-moving group chats. Long message threads, frequent topic shifts, and dense interactions make it difficult for users to stay informed.

Key friction points include:

- Users needing several minutes to catch up after short absences
- Difficulty extracting the "main point" of discussions
- Missed action items and misunderstandings during decision-making

This negatively affects engagement, satisfaction, and retention. Acme must reduce cognitive load to maintain competitiveness in the messaging market.

2. Impact Assessment

Information overload reduces:

- User satisfaction and perceived usability
- Engagement (less reading, fewer responses)
- Retention (users avoid busy chats)

Competitive platforms increasingly offer AI-assisted message understanding. Without improvement, Acme risks losing market relevance. Summarization directly addresses these issues and supports enterprise and productivity-oriented use cases.

3. Solution Vision

We propose an AI-powered Dialogue Summarization feature using a transformer encoder–decoder architecture.

The system will automatically generate short, coherent summaries of multi-turn conversations.

User-facing value:

- Rapid catch-up in busy chats
- Clear understanding of decisions and next steps
- Reduced stress and improved daily usability

Business value:

- Increased platform stickiness
- Higher productivity perception
- Competitive differentiation in the market

4. Success Criteria

Measurable success indicators:

- ROUGE-1, ROUGE-2, and ROUGE-L scores competitive with baseline models
- Summaries preferred by users in A/B quality comparisons
- Latency compatible with near-real-time summarization
- High coherence and minimal hallucination rates

Product success indicators:

- Increased message reads
- Higher engagement after absences
- Reduced user reports of confusion or message fatigue

5. Problem-Solving Process

Step 1 – Data Loading & Exploration

- Import and inspect the SAMSum dataset
- Understand conversational style, length, summarization patterns

Step 2 – Preprocessing & Tokenization

- Clean text, standardize formatting
- Apply BERT tokenizer for encoder inputs

- Apply decoder tokenizer (e.g., GPT-2 or T5 style)

Step 3 – Model Architecture

- Encoder: BERT to extract contextual dialogue representations
- Decoder: Transformer-based generator to produce summaries
- Combined seq2seq training

Step 4 – Training & Optimization

- Fine-tune model on SAMSum
- Use attention masking, teacher forcing, checkpoints

Step 5 – Evaluation

- Compute ROUGE metrics
- Qualitative inspection of generated summaries
- Error analysis

Step 6 – Iteration

- Improve decoding (beam search, nucleus sampling)
- Adjust hyperparameters
- Additional cleaning / truncation strategies

Step 7 – Deployment Considerations

- Batch inference
- Latency and memory management
- Scalable deployment for group chats

6. Conceptual Representation (Flowchart)

Dialogue Input

│

▼

Preprocessing & Tokenization

│

▼

BERT Encoder → Latent Representation → Transformer Decoder

│

▼

Generated Summary

7. Methodology Justification

Why BERT as encoder:

• Strong contextual understanding of conversations

• Robust for long, multi-speaker inputs

Why Transformer decoder:

• Well-established for natural language generation

• Flexible decoding strategies

Why ROUGE evaluation:

• Industry standard for summarization

• Enables objective comparison to human reference summaries

This approach is computationally feasible and aligns with Acme's feature goals.

8. Alignment With Requirements

This solution meets all Project 3 requirements:

- Uses transformer encoder–decoder architecture
- Includes data preprocessing, modeling, training, evaluation
- Provides conceptual visuals and process explanation
- Presents clear, measurable success criteria
- Aligns business needs with technical strategy
- Produces a viable, production-oriented summarization feature

9. Timeline, Scope, and Final Delivery

Research & Preparation (Nov 29 – Dec 1)

- Explore dataset, finalize preprocessing, study model strategies

Implementation (Dec 1 – Dec 3)

- Build encoder-decoder pipeline
- Train model
- Perform initial ROUGE evaluation

Iteration (Dec 3 – Dec 4)

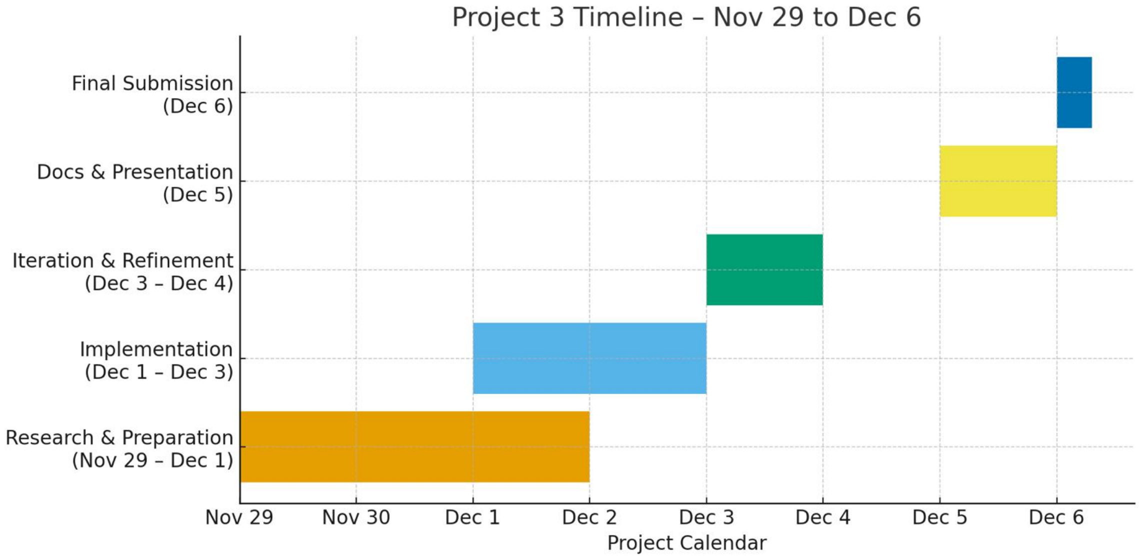
- Improve decoding strategies
- Conduct error analysis
- Apply feedback from MVP Discussion

Risk Management

- Compute limits → gradient accumulation, reduced batch sizes
- Unstable training → frequent checkpointing
- Low ROUGE → adjust cleaning/tokenization strategies

Final Delivery

- Project critique submission: Dec 3
- Final implementation completed: Dec 4
- Documentation & presentation prepared: Dec 5
- Full final submission: Dec 6


Project 3 Timeline – Nov 29 to Dec 6
Final Submission (Dec 6)
Docs & Presentation (Dec 5)
Iteration & Refinement (Dec 3 – Dec 4)
Implementation (Dec 1 – Dec 3)
Research & Preparation (Nov 29 – Dec 1)
Nov 29
Nov 30
Dec 1
Dec 2
Dec 3
Dec 4
Dec 5
Dec 6
Project Calendar