Dialogue Summarization on ACME

By Alejandro Silva

## Executive Summary

It was built a reliable chat-summarization baseline that turns multi-speaker conversations into concise, consistent notes to reduce analyst review time and improve CRM data quality. Using an industry-standard dataset and a proven seq2seq model, we optimized inference (not retraining) to improve readability at low cost and are ready to pilot. Outputs are accurate on core topics but longer than ideal and occasionally drift on intention; We will launch the pilot with settings that produce shorter summaries and simple checks to keep facts straight. Success will be judged by faster case handling, fewer analyst edits, and stable response times. If we hit those goals, the project will scale up to evaluate stronger models and meaning-based metrics to further improve quality.

## Problem Statement & Approach

The business problem is turning messy, multi-person chat threads into quick, accurate summaries that speed support triage, keep CRM notes consistent, and capture meeting outcomes.

The approach is a reliable summarization engine using a proven model, tuned on a representative chat dataset, and improved results by optimizing generation settings with no extra training cost. Quality is measured with an industry standard score and hands-on reviews of sample cases, and we analyzed length and coverage to ensure summaries stay concise and useful. Success metrics: quality, brevity, faithfulness, usability and latency

## Timeline

Timeline from pitch report adjusted to real time

|  |  |  |
| --- | --- | --- |
| Phase | Estimated Time | Actual Time |
| Research & Preparation | 1-2 days | 0.5 day / 4 hours |
| Implementation – Phase 1 | 1 day | 0.5 day / 4 hours |
| Implementation – Phase 2 | 1-1.5 days | 1 day / 8 hours |
| Implementation – Phase 3 | 1 day | 1 day / 8 hours |
| Evaluation & Iteration | 1 day | <1 day / 6 hours |
| Documentation & Delivery | 1-1.5 days | <1 day / 6 hours |
| Risk Management | 0.5 day | 0.5 day / 4 hours |

## Dataset Analysis

We used the SAMSum chat dataset with human-written summaries (train/validation/test). One test record had an empty dialogue and was removed; validation and test are clean, and seeds/versions are documented for reproducibility. The training split includes 14,731 conversations with 28,935 unique dialogue words; chats are short—about 94 words per dialogue and ~8 words per turn (median 10 turns). Human summaries are concise (median 18 words), yielding an average compression ratio ≈ 0.28–0.30. On a 200-sample validation slice, dialogues average 91.5 words (median 70.5, median 9 turns), human summaries average 18.9 words (median 17), while our model’s outputs average 43.3 words (tight 30–54 range)—about 3× longer than references and ~81% of the source length, underscoring the need for brevity controls in deployment.

A graph of a number of words

AI-generated content may be incorrect.A graph of a distribution of words

AI-generated content may be incorrect.

## Encoder / Decoder Selection

A graph of different colored bars

AI-generated content may be incorrect.It was compared two backbones: BERT+GPT-2 (decoder only) vs BERT2BERT (encoder–decoder). Both backbones used the same preprocessing and a comparable decoding baseline during the head-to-head. BERT2BERT clearly outperformed across all validation ROGUE metrics providing higher accuracy with less engineering and reducing delivery risk (see chart, ROGUEx100).

BERT2BERT was chosen as the backbone because encoder–decoder models are purpose-built for summarization: they compress and copy source content more reliably, reach useful quality with less custom engineering, and train more predictably. Net result: higher accuracy, faster time-to-value, and lower delivery risk for the project.

### Phase A - Training Optimization – Learning the right behavior

Encoder–decoder BERT2BERT model was fine-tuned on SAMSum to learn how to turn multi-speaker chats into summaries. We ran a small, cost-controlled sweep over core training knobs—learning rate, weight decay, label smoothing, warmup—and selected the best checkpoint using validation ROUGE-Lsum. Outcome: ROUGE-Lsum ≈ 0.2475.

Phase A creates the *capability*—a stable, reproducible model that captures the key content of a conversation.

### Phase B — Decoding Optimization - Polish for production

With the trained weights fixed, generation settings were optimized to shape the final summary (beam size, length penalty, no-repeat n-gram, and max new tokens). This is inference-only (no retraining cost), so it was run on the full validation set to choose robust defaults.

The numerical gain over Phase A (+0.0008 ~0.3%) is modest because decoding can only rearrange what the model already knows, and ROUGE under-rewards improvements in brevity and factuality.

Phase B sets deployable defaults that balance quality and cost/latency, producing cleaner, more consistent summaries without additional training spend.

## Performance evaluation and results

After selecting the best model via Phases A and B (using ROUGE-Lsum), we conducted a qualitative review on three representative validation samples from the validation set. The aim is to assess real-world summary quality beyond metrics—faithfulness to the source dialogue, brevity/clarity, role and numeric accuracy, and tone handling (e.g., humor). For each case, we compare the model’s output to both the original dialogue and the human reference (flagging any noisy references). This review validates our decoding choices and surfaces concrete failure modes to address before deployment.

### Quantitative

Phase A best ROUGE-Lsum: ≈ 0.2475 with learning rate=5e-5, weight decay= 0.01, label smoothing=0.1 and warmup ratio=0.03. Phase B best ROUGE-Lsum: ≈ 0.2483 with  
num\_beams=6, length\_penalty=1.0, no\_repeat\_ngram\_size=3, and max\_new\_tokens=64 with avg gen\_len ≈ 44.7 tokens. Relative gain: ~0.0008 absolute (~0.3%) — small but consistent.

The gain is small because decoding can only rearrange probabilities; it can’t change what the model knows. Noisy references (some human summaries contain inventions) and verbosity also limit ROUGE movement. ROUGE rewards n-gram overlap, not tone/faithfulness.

### Qualitative

It was selected three case studies

* idx=654 (humor). Model echoes “20% decrease” but sometimes mutates it (“20% faster”) and misses the joke; Phase B reduces speculation vs. Phase A but numeric fidelity remains a risk. The *reference summary itself* is misleading (treats a joke as fact), so we don’t over-weight this sample for metric judgment.
* idx=114 (noisy reference). Human reference invents details (beer with kids, cookies, dad). Phase B is shorter than Phase A and stays closer to the cleaning theme but still adds some unsupported family claims.
* idx=25 (roles). Human reference is strong. Phase B improves concision vs. Phase A but still confuses who is launching the restaurant; repetition reduced, role attribution remains the main error.

Business framing: For support/chat summarization, outputs should be concise and faithful. Current model is usable as a baseline but needs stricter brevity and fact controls to avoid misreporting. Following the best models in both phases and the final generated summary histogram.

A graph of blue bars

AI-generated content may be incorrect.A screen shot of a chart

AI-generated content may be incorrect.

The histogram shows model outputs cluster tightly around 40–48 words (mean ≈ 43), indicating near fixed-length behavior. Dialogues average 91.5 words with 8.17 words per turn. Compression ratios highlight the gap: human summaries are 0.280× the dialogue length, while model summaries are 0.809×—about 3.06× longer than humans. Implication: the system is verbose relative to SAMSum’s style; adopt brevity/length-adaptive decoding (e.g., max\_new\_tokens 32–40, length\_penalty≈0.6) to better match human concision without retraining.

## Conclusion

It was done a dependable chat-summarization model that converts multi-speaker threads into quick, consistent notes, cutting analyst reading time without adding training cost. Using typical reading speed (~200 wpm), an average dialogue (~91.5 words) takes ~27.5s to read; our model’s summary (~43.3 words) takes ~13.0s—saving ~14.5s per chat (~53% less time), or ~24 minutes per 100 chats. With planned brevity controls (adaptive max\_new\_tokens, mild length penalty), summaries can approach human length (~18.9 words, ~5.7s), pushing potential savings toward ~79%.

It wasdelivered an end-to-end baseline that is reproducible and explainable. Phase B decoding on full validation produced the best configuration (ROUGE-Lsum ≈ 0.2483), with qualitative improvements (less redundancy, slightly better grounding) but lingering issues in faithfulness and verbosity. Following an example of the generated summary

## Challenges & Solutions

* Phase A could not be run on the full corpus, so we trained and tuned hyperparameters on 800 training and 200 validation dialogues to select the best checkpoint.
* Verbosity & length control. Generated summaries are 3× human length in average.  
  *Mitigation:* Decoding search + proposed short/length-adaptive caps.
* Faithfulness (numbers/entities/roles). Occasional numeric slips and role mix-ups.  
  *Mitigation:* no\_repeat\_ngram\_size=3; propose a lighter repetition penalty (1.05–1.15) and optional numeric/entity post-checks or constrained decoding.
* Noisy references (SAMSum). Some human summaries don’t reflect the dialogue (limits ROUGE; confuses training). *Mitigation:* Qualitative review; note reference issues explicitly in analysis; treat ROUGE cautiously.
* Despite tighter compute budgets, we ran Phase B on the full validation set (inference-only, no retraining) to avoid subset bias—trading modest extra runtime for robust, representative decoding defaults ready for production.
* To optimize resources, the project was engineered to run seamlessly on either GPU or CPU, automatically detecting the available hardware and switching paths accordingly. This required device-agnostic code, ensuring consistent results and predictable costs across environments.

## Potential Improvements & Limitations (Future Work)

**Model-Level Improvements:** Fine-tune on SAMSum with other encoder–decoder architectures, which might perform better on abstractive summarization than BERT2BERT. Use a cased variant during preprocessing to improve style and readability of names and proper nouns. Apply domain-adaptive pretraining (DAPT) to better capture conversational tone and humor.

**Data Enhancements:** Human references in SAMSum often contain inaccuracies. Building a curated subset with corrected summaries would provide more reliable supervision and evaluation. Test another language than SAMSum (DialogSum, MediaSum), it could increase robustness and reduce hallucinations.

**Advanced Decoding & Post-Processing:** Add lexical constraints to enforce copying of numbers, percentages, or named entities directly from the dialogue to prevent numeric hallucinations. Lightweight regex or entity-matching rules to validate generated facts against the source dialogue (e.g., ensure “20% decrease” isn’t changed to “20% faster”).

**Evaluation & Metrics:** Complement ROUGE with semantic metrics (BERTScore, BLEURT, QAFactEval) to better capture factual correctness and conversational intent. Ask annotators (interns) to rate factuality, conciseness, and tone preservation, which ROUGE cannot measure.

**System-Level Extensions:** Allow end-users to set a style preference (concise fact vs. conversational highlight) to adapt decoding dynamically. Train the model to both summarize and classify dialogue tone (informative vs. humorous), helping the system decide whether to preserve humor or condense facts.

**Actionable in decoding** based on other metrics: Tighten decoding for brevity, make length adaptive to each dialogue, reduce filler, track faithfulness at human-like compression.