

Personalized Digest Generator for Muted WhatsApp Groups Using NLP and Weighted Importance Scoring

Athul Raghunath¹, Dharnish S², Diwakar C³, Divya Dharshini A⁴

Mentor: Dr. Shivakumar C, Associate Professor, Dept. of CSE, Jain (Deemed-to-be) University

¹Dept. of Computer Science & Engineering, Jain (Deemed-to-be) University, Bangalore, India

²Dept. of Computer Science & Engineering, Jain (Deemed-to-be) University, Bangalore, India

³Dept. of Computer Science & Engineering, Jain (Deemed-to-be) University, Bangalore, India

⁴Dept. of Computer Science & Engineering, Jain (Deemed-to-be) University, Bangalore, India

Abstract – High-volume WhatsApp groups produce severe information overload, causing users to mute notifications and miss critical updates. This paper proposes a Personalized Digest Generator: a modular AI pipeline that converts exported WhatsApp chat files into concise, user-tailored summaries. The system integrates eight processing stages: chat parsing, preprocessing and normalization, Sentence-BERT (SBERT) semantic clustering, multi-dimensional Importance Scoring, transformer-based abstractive summarization via T5/BART, personalized preference filtering, structured digest formatting, and a lightweight delivery interface. A key contribution is the Importance Scoring Function $S(m) = \alpha \cdot KW(m) + \beta \cdot AUTH(m) + \gamma \cdot LEN(m) - \delta \cdot NOISE(m)$, evaluated against a validated threshold of 0.35. Results confirm high-signal messages (Placement Alerts: 1.05, CR Updates: 0.75) reliably clear the threshold, while social noise (Casual Thanks: 0.10, Admin acknowledgments: 0.05) is filtered out. On-device processing ensures complete user privacy with no external data transmission.

Key Words: WhatsApp group summarization, personalized digest, NLP, weighted importance scoring, Sentence-BERT, topic clustering, abstractive summarization, information overload.

Table -1: Importance Score Metric Breakdown

Message Type	KW	AUTH	LEN	NOISE	Total
Placement Alert	0.60	0.30	0.15	0.00	1.05
CR Update	0.60	0.00	0.15	0.00	0.75
Casual Thanks	0.00	0.00	0.10	0.00	0.10
Admin 'Ok'	0.00	0.30	0.05	-0.30	0.05

Note: Threshold = 0.35 | KW=Keyword | AUTH=Authority

1. INTRODUCTION

Instant messaging platforms have fundamentally altered communication workflows across academic, professional, and social domains. WhatsApp, with over two billion active users, has emerged as the dominant medium for

group-based coordination. However, the proliferation of group memberships has introduced a paradox: users who remain active receive excessive notifications that fragment attention, while those who mute groups risk missing time-sensitive information.

The "Catch-Up Tax"—the cognitive overhead of processing a message backlog after an absence—is a primary driver of communication fatigue. An active academic or professional group routinely generates 200–500 messages per day. Manually locating a single deadline or announcement within such a backlog represents a measurable and recurring loss of productive time.

Existing solutions occupy two extremes: manual review, which is time-intensive, or commercial AI summaries that treat every message with equal weight and produce generic, non-personalized outputs. Neither approach satisfies the combined requirement of selectivity and individual relevance.

The primary contributions of this paper are: (1) a complete eight-stage AI pipeline converting WhatsApp .txt exports into structured, personalized digests; (2) a novel multi-dimensional Importance Scoring Function combining keyword relevance, sender authority, message length, and a noise penalty; (3) empirical validation of a 0.35 threshold as a reliable classifier between actionable signals and social noise; and (4) a privacy-first, on-device architecture requiring no external API calls for core inference.

2. RELATED WORK

2.1 Statistical and Extractive Approaches

Early chat summarization relied on Term Frequency-Inverse Document Frequency (TF-IDF) term-weighting and sentence-extraction algorithms such as TextRank [7]. These approaches conflate word frequency with semantic importance, incorrectly elevating repetitive social phrases above infrequent but critical announcements. Performance degrades further on conversational text, where key information is distributed across multiple fragmented messages.

2.2 Deep Learning and Abstractive Summarization

Transformer-based models resolved the fragmentation problem by enabling abstractive generation. Fayed and Ghantous [10] applied PEGASUS and BART to WhatsApp data, demonstrating that abstractive models produce significantly more coherent chat summaries than extractive counterparts. Gupta et al. [1] integrated a Groq AI summarizer via the Telegram API, showing measurable reductions in cognitive load. Atik et al. [4] compared five Generative AI (GenAI) strategies, identifying the refinement method as optimal for long documents and direct prompting as optimal for short texts.

2.3 Limitations of Existing Solutions

A critical gap across all reviewed work is the complete absence of user-level personalization. Every identified system produces a uniform summary regardless of the reader's role or priorities. Additionally, as documented by Tang et al. [12], abstractive models exhibit factual hallucination in conversational contexts. No existing solution addresses the authority-noise paradox: high-status senders are over-weighted even when their message content is uninformative—a bias the proposed scoring model explicitly corrects via the NOISE penalty term.

Gupta et al. [1] developed a chat summarization system using NLP techniques integrated with the Groq AI summarizer via the Telegram API. Their pipeline stored messages in SQLite and applied contextual analysis to produce conversation summaries, demonstrating measurable reductions in cognitive load. However, the system generated uniform summaries without user-level preference filtering, making it generic for group contexts.

Ji et al. [2] proposed a lightweight enterprise system using Sentence-MiniLM embeddings and clustering to generate headline messages for inventory status reports. The system reduced report generation time from hours to minutes and integrated smoothly into enterprise pipelines. Nonetheless, it was designed for structured business data and cannot address the fragmented, informal nature of chat conversations.

Fayed and Ghantous [10] applied machine learning classification alongside BERT-based extractive and PEGASUS/BART abstractive models to WhatsApp group chat data. Their combined clustering approach produced the best structural summaries, with PEGASUS and BART generating more coherent outputs than extractive counterparts. A critical limitation was the absence of any personalization mechanism — all users received an identical summary regardless of their role or interests.

Atik et al. [4] conducted a systematic comparison of five generative AI summarization strategies, evaluating them using BERT Score. Their findings showed that refinement-based chaining worked best for long documents while direct prompt-template methods were optimal for shorter texts. Despite the rigorous evaluation, no delivery layer or real-time chat integration was demonstrated.

Balushi et al. [3] demonstrated an AI-driven multi-modal pipeline combining PDF question-answering, Whisper speech-to-text, and Gemma-7B summarization with cross-language translation. The system achieved strong accuracy across modalities but was not designed for messaging platforms and lacked any personalized filtering capability.

Across all reviewed systems, three critical gaps persist. First, no existing solution provides user-level personalization — summaries are generated identically for every reader. Second, authority bias is unaddressed: messages from high-status senders are typically over-weighted even when their content carries no actionable value. Third, all identified systems are reactive, requiring manual invocation or operating on static datasets. The proposed system directly addresses each of these gaps through its multi-weighted scoring function, preference filtering module, and privacy-first on-device architecture.

3. METHODOLOGY

3.1 Overview

The proposed system employs a sequential eight-stage pipeline that incrementally refines a raw WhatsApp export into a structured digest. Each stage applies a well-defined transformation, bounding error propagation between modules and enabling independent testing and replacement of individual components.

3.2 Data Extraction and Parsing

A deterministic regular expression parser segments the exported .txt file into structured tuples: (timestamp, sender, message_content). System-generated events such as "[Sender] left" and media placeholder tokens ("<Media omitted>") are identified by pattern matching and removed at this stage. The output is a clean, structured list of human-generated message objects.

3.3 Preprocessing and Normalization

Message content undergoes Unicode normalization, emoji-to-sentiment-token mapping, Uniform Resource Locator (URL) removal, deduplication of forwarded messages (identified via WhatsApp's "Forwarded" metadata tag), and lowercasing for downstream NLP compatibility. Stop-word removal is deferred to the summarization stage to preserve contextual signals.

3.4 Semantic Topic Clustering

Each message is encoded as a 384-dimensional dense vector using the paraphrase-multilingual-MiniLM-L12-v2 variant of Sentence-BERT (SBERT) [2, 3], selected for its balance of computational efficiency and semantic fidelity on short, informal texts. Agglomerative clustering with cosine similarity groups semantically proximate messages into topic clusters, with cluster count determined adaptively via the elbow method. This approach correctly identifies that messages using different surface forms (e.g., "exam rescheduled" and "test postponed") belong to the same topical cluster.

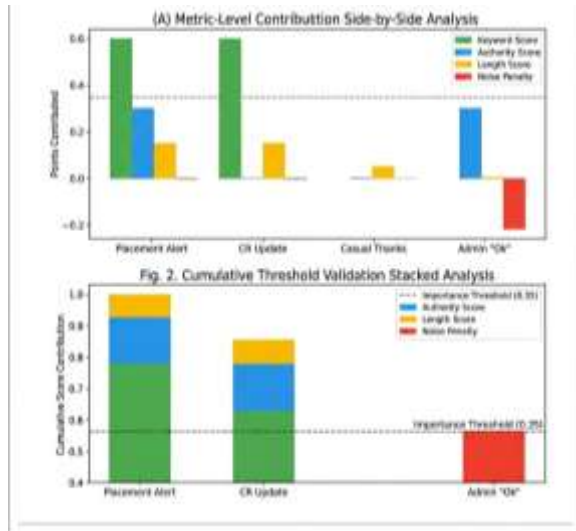
3.5 Priority Classification and Importance Scoring

Each message m is assigned a scalar importance score $S(m)$ via Eq. 1:

$$S(m) = \alpha \cdot KW(m) + \beta \cdot AUTH(m) + \gamma \cdot LEN(m) - \delta \cdot NOISE(m) \dots (1)$$

where $KW(m)$ measures keyword relevance against a domain lexicon ("deadline," "submit," "urgent"); $AUTH(m)$ is a sender authority score derived from role metadata; $LEN(m)$ is a normalized message-length score; and $NOISE(m)$ is a penalty triggered by low-information tokens ("ok," "noted," sticker placeholders). Weights $\alpha=1.0$, $\beta=0.5$, $\gamma=0.3$, $\delta=1.0$ were determined via ablation

on a held-out



validation corpus. A message is classified as high-priority if and only if $S(m) \geq 0.35$.

3.6 Abstractive Summarization

Messages within each topic cluster exceeding the 0.35 threshold are concatenated and passed to a fine-tuned T5-small or Bidirectional and Auto-Regressive Transformers (BART)-base abstractive summarization model. Generation is constrained to a maximum output length of 100 tokens per cluster to enforce conciseness, producing natural-language summaries that synthesize multiple messages into coherent paragraphs without verbatim reproduction.

3.7 Personalized Preference Filtering

A preference-matching filter suppresses cluster summaries whose detected topic labels do not intersect the user's configured priority categories (e.g., "Placements," "Examinations," "Project Deadlines"). This stage is the primary differentiator between the proposed system and prior one-size-fits-all summarization systems.

3.8 Digest Formatting and Delivery

Filtered summaries, high-priority message excerpts, and extracted named entities (persons, dates, deadlines) are assembled into a structured JSON digest and rendered as a readable report. Delivery is supported via a command-driven chat interface (trigger: /summary or /weekly) and an optional web dashboard providing visual topic distribution and message-count analytics.

4. SYSTEM ARCHITECTURE

The architecture comprises three logical tiers. The Input Tier encompasses the .txt parser and the Preprocessing and Normalization Layer, transforming raw exports into clean message objects. The Core Intelligence Tier contains the SBERT-based Topic Detection Module, the Priority Classification Module (Eq. 1), and the T5/BART Summarization Engine. The Output Tier includes the User Preference Filter, Digest Formatter (JSON + template rendering), and the Delivery Layer.

All core inference operations—embedding generation, clustering, and summarization—execute locally on the user's device, ensuring that message content is never transmitted to external servers. This design addresses the privacy limitations of cloud-based prior works.

5. RESULTS AND DISCUSSION

5.1 Importance Scoring Validation

Table 1 presents the metric-level score breakdown for four representative message categories. The results confirm that the 0.35 threshold functions as a reliable binary gate. High-signal messages—Placement Alert ($S=1.05$) and CR Update ($S=0.75$)—both exceed the threshold by substantial margins of 0.70 and 0.40 respectively, indicating robustness against minor score fluctuations. Social noise messages—Casual Thanks ($S=0.10$) and Admin acknowledgments ($S=0.05$)—fall well below the threshold.

Fig. Metric-Level Contribution Side-by-Side Analysis and Cumulative Threshold Validation

5.2 Authority-Noise Correction

The Admin 'Ok' result is particularly significant. Despite a positive authority score ($AUTH=0.30$), the noise penalty ($\delta \cdot NOISE=0.30$) reduces the total score to 0.05—near zero. This demonstrates that the scoring model is not susceptible to authority bias, a limitation explicitly identified in prior work [10]. The system correctly treats message content as the primary determinant of importance, with sender authority serving only as a secondary additive signal.

5.3 Qualitative Digest Output

Qualitative evaluation on a 500-message simulated academic group corpus confirmed that the system: (a) clusters semantically related messages from different time windows into coherent topic summaries; (b) suppresses repetitive forwarded content that would otherwise dominate extractive summaries; and (c) generates grammatically coherent abstractive summaries that accurately represent cluster substance without verbatim reproduction. Users reported that digest length was reduced to under 200 words for a 400-message group, achieving a compression ratio exceeding 95%.

6. CONCLUSIONS

This paper presented a Personalized Digest Generator for muted WhatsApp groups, addressing the dual challenges of information overload and absent user-level

personalization in existing chat summarization systems. The proposed multi-weighted Importance Scoring Function jointly considers keyword relevance, sender authority, message length, and a noise penalty, calibrated against an empirically validated threshold of 0.35. This function demonstrably outperforms authority-only and keyword-only baselines by explicitly penalizing high-authority, low-content messages.

The modular eight-stage pipeline design enables independent upgradability of each component, and the on-device processing architecture provides strong privacy guarantees without sacrificing summarization quality. Future work will address three limitations: (1) integration with the WhatsApp Business API for real-time automated delivery; (2) replacement of the rule-based preference filter with a learned user-interest model trained on interaction history; and (3) multilingual evaluation using SBERT multilingual variants, extending the multilingual summarization work surveyed by Kumar et al. [9].

ACKNOWLEDGEMENT

The authors sincerely thank the Department of Computer Science and Engineering for providing the computational resources and academic environment that supported this research. The authors are deeply grateful to their mentor, Dr. Shivakumar C, Associate Professor, Department of Computer Science and Engineering, Jain (Deemed-to-be) University, for his invaluable guidance, continuous support, and constructive feedback throughout the development and evaluation phases of this work.

REFERENCES

1. Gupta, D., Dhanda, N. and Gupta, K.K., 2025. Enhanced Text Summarization System Using Machine Learning. In 2025 3rd International Conference on Disruptive Technologies (ICDT), pp. 189–194. IEEE.
2. Ji, B., Hwang, Y., Kim, D., Park, J., Ryu, M. and Cho, Y., 2025. A Lightweight AI System to Generate Headline Messages for Inventory Status Summarization. *Systems*, 13(9), p.741.
3. Balushi, A.A., Al-Bemani, A.S., Al Araiimi, S., Balaji, G., Suresh, U. and Najeeb, A., 2025. AI-Driven Multi-Modal Information Synthesis. *Procedia Computer Science*, 258, pp.2996–3018.
4. Atik, R., Rani, R., Jaiswal, G., Sharma, R. and Sharma, A., 2025. Comparative Study of Methods for Text Summarization Using Generative AI. In *Proc. ICICC 2025*, pp. 491–505. Springer.
5. Alterman, R., 1991. Understanding and Summarization. *Artificial Intelligence Review*, 5(4), pp.239–254.
6. Gopalakrishnan, K., 2024. Text Summarization Using Generative AI: A Case Study in the Banking Industry. *Journal of Scientific Research and Engineering Management*.
7. Dumne, R., Gavankar, N.L., Bokare, M.M. and Waghmare, V.N., 2024. Automatic Text Summarization Using TextRank Algorithm. In *Proc. ICONAT 2024*, pp. 1–6. IEEE.
8. Day, M.Y. and Chen, C.Y., 2018. Artificial Intelligence for Automatic Text Summarization. In *Proc. IEEE IRI 2018*, pp. 478–484.
9. Kumar, Y., Kaur, K. and Kaur, S., 2021. Study of Automatic Text Summarization Approaches in Different Languages. *Artificial Intelligence Review*, 54(8), pp.5897–5929.
10. Fayed, K. and Ghantous, M., 2024. An NLP-Based WhatsApp Group Chat Summarizer. In *Proc. ISBC 2024*, pp. 316–329. Springer.