Personalized Summarization

Tashvi Patel
Faculty Mentor: Dr. Sourish Dasgupta
Department of Information and Communication Technology
Adani University
Ahmedabad, Gujarat, India
Email: tashvi1510@gmail.com
Phone: +91 9879003964

ABSTRACT

Text summarization has emerged as an essential task for information access, but most existing approaches focus on producing generic summaries that overlook user-specific preferences. In practice, a summary that is useful for one reader may omit critical details for another, highlighting the importance of personalization. Traditional encoder–decoder models or prompting-based approaches fail to adequately capture evolving user interactions, such as click, skip, generate-summary (GenSumm), and summary-generation (SummGen), which strongly influence user intent.

In this work, we propose a behavior-modulated personalized summarization framework that explicitly integrates user actions into the summarization pipeline. Engagement is modeled not as a single step, but as a recurring cycle: even after a GenSumm request, the system proceeds to SummGen, and the outcome is fed back into the preference trajectory. This ensures that user interactions continuously shape future summaries.

We apply prefix tuning on a pre-trained encoder–decoder backbone, enabling efficient adaptation without retraining the full model while still learning to associate user behavior with document content. Experimental results on an interaction-annotated dataset demonstrate that our personalized approach produces summaries more contextually aligned with user intent compared to generic baselines. This study provides evidence that user-aware representation learning can significantly advance the quality and relevance of personalized summarization.

I. INTRODUCTION

The rapid growth of digital content, particularly in the news domain, has created an overwhelming demand for efficient summarization systems. Traditional summarization models, both extractive and abstractive, often generate outputs that are general in nature and fail to align with the individual preferences of users. This lack of personalization reduces the effectiveness of summaries in terms of user engagement and satisfaction.

To address this challenge, recent work has explored the integration of personalization into natural language processing (NLP) pipelines. Personalized summarization aims to generate summaries that are not only accurate and concise, but also tailored to the unique needs, behaviors, and interests of individual users. Such systems are expected to go beyond surface-level text compression by adapting to patterns in user interaction, such as their tendency to read, skip, or engage with specific types of summaries.

In this work, we present a personalized summarization framework that leverages prefix tuning in encoder–decoder architectures. Our approach incorporates user engagement signals—Click, Skip, GenSumm, and SummGen—into the summarization process, enabling the model to dynamically refine its outputs based on real-time feedback. Engagement is treated as an iterative cycle: even after a generated summary (GenSumm) is produced, subsequent engagement through user re-summarization (SummGen) continues to shape the personalization pipeline.

The contributions of this study are three-fold:

1) We propose a novel engagement-driven summarization pipeline that models user behavior explicitly.
2) We adapt prefix tuning methods for effective fine-tuning of large language models in a low-resource, user-specific setting.
3) We demonstrate the potential of incorporating interaction feedback to improve the personalization and quality of generated summaries.

By aligning summarization models with user engagement cycles, our approach provides a foundation for building adaptive, user-centric NLP systems that move closer to real-world applications.

II. PROPOSED WORK

In this work, we extend the T5 model with prefix-tuning to incorporate user engagement signals for personalized summarization. The main idea is to map user behavior into the encoder, so that the model not only learns from textual content but also from how users interact with the generated summaries. We consider four actions: Click, Skip, Gen_Summ, and Summ_Gen. Each action is represented mathematically to guide the model training.

A. Click

$$V_i^{CLK} = \left(1 + \alpha_i \odot \hat{h}_i\right) \cdot T_i \cdot \mathbf{RWC}(v_i) \tag{1}$$

Here, α_i represents a learnable attention or importance weight vector that highlights which parts of the user's memory

are most relevant when computing the click score for the current item. The term $\hat{h}_i$ denotes the user's memory state derived from past interactions, ensuring that historical behavior contributes to the prediction. The temporal score T_i modulates this influence based on recency, giving more importance to recent interactions. The function $\text{RWC}(v_i)$ captures the reliability of user i's interactions with the item v_i. The $(1+)$ term ensures that even if there is no past memory ($\hat{h}_i$ absent), the click has a full influence of 100%.

Thus, V_i^{CLK} predicts the click score for a given user-item pair, where a higher score indicates a stronger likelihood of the user clicking the current item. In effect, the formulation balances how much weight should be given to the user's prior clicks versus the present context.

B. Skip

$$v_i^{SKP} = \tanh(\lambda \cdot k_i + W_h \hat{h}_i) \cdot T_i \cdot \text{RWC}(v_i) \tag{2}$$

where the skip curvature k_i is defined as

$$k_i = \beta_1 \cdot L_i + \beta_2 \cdot \Delta_{pull}, \tag{3}$$

and the attraction–repulsion pull term is given by

$$\Delta_{pull} = a \odot \max(\langle h_i, v_{i+1} \rangle, \langle h_i, v_i \rangle) + b \odot (1 - \langle h_i, v_i \rangle). \tag{4}$$

a) Explanation.: The skip score v_i^{SKP} for item i is designed to capture the user's likelihood of abandoning or diverting from the current item. The first part of the formulation, $\lambda \cdot k_i + W_h \hat{h}_i$, combines two key components: the skip curvature k_i, which represents the user's natural tendency to lose interest, and the transformed memory state $\hat{h}_i$ passed through the weight matrix W_h. This combination is passed through the $\tanh(\cdot)$ nonlinearity, which produces a smooth gating effect that regulates the strength of skip behavior, ensuring that the response is not binary but continuous.

The skip curvature k_i itself is a linear combination of L_i, the inherent skip curvature mass based on the content's structure or layout, and Δ_{pull}, which introduces attraction and repulsion dynamics. The attraction term favors whichever of the current or next item is more aligned with the user's memory vector h_i, while the repulsion term penalizes the current item if it shows weak similarity to memory. The coefficients β_1 and β_2 determine how much weight is given to these two forces.

The Δ_{pull} formulation explicitly models these dynamics by using the similarity scores $\langle h_i, v_i \rangle$ and $\langle h_i, v_{i+1} \rangle$. The parameter a scales the attraction effect, while b scales the penalty for low similarity. The element-wise product $\odot$ ensures that these effects act dimensionally across the embedding space.

Finally, the modulation by T_i (user trust score) controls the temporal importance of memory in the skip action, while $\text{RWC}(v_i)$ provides a stable item embedding as a reliability anchor. Altogether, this ensures that the skip score is not only dependent on the mismatch of the current item but also sensitive to the pull of the next item, while being grounded in both user reliability and temporal context. A higher v_i^{SKP} indicates a greater likelihood that the user will skip the current item.

C. GenSumm: Pre-Interest vs. Post-Interest

$$v_i^{GenSum} = \sigma(\hat{h}_i) \cdot T_i \odot \text{RWC}(v_i^{headline}) \tag{5}$$

a) Explanation.: The generated summary interest score v_i^{GenSum} models the user's engagement based on their memory state after having consumed prior content. Here, $\hat{h}_i$ represents the updated user memory that incorporates signals from previously observed items. To avoid over-reliance on raw memory activations, the function $\sigma(\hat{h}_i)$ applies a sigmoid gating mechanism, softly filtering which components of the memory are most relevant for generating interest in the summary. This ensures that memory contributions are bounded between 0 and 1, enabling a controlled influence.

The user trust score T_i modulates this influence further, scaling the effect of memory according to how reliable the user's prior interactions are considered. Meanwhile, $\text{RWC}(v_i^{headline})$ provides a stable content representation through the Random Walk Conformer embedding of the item's headline, which captures structural and semantic properties of the summary text.

The element-wise multiplication $\odot$ between the trust-weighted memory signal and the headline embedding allows the model to align user memory with specific aspects of the summary. In this way, v_i^{GenSum} reflects a balance between pre-interest (driven by memory and trust) and post-interest (anchored in the item's headline embedding). A higher v_i^{GenSum} indicates that the headline resonates strongly with the user's memory-influenced preferences, making it more likely that the user engages with the summary.

D. SummGen: Summary Generation

$$v_i^{SummGen} = \big(\sigma(\hat{h}_i) \odot v_{i-1}\big) \cdot \text{RWC}(v_i^{summary}) \tag{6}$$

a) Explanation.: The summary generation score $v_i^{SummGen}$ captures how well a generated summary aligns with the user's memory and their immediately preceding interaction. Here, $\hat{h}_i$ is the current user memory vector that encodes information accumulated from past interactions. The function $\sigma(\hat{h}_i)$ applies a sigmoid gating to softly activate only the most relevant dimensions of memory, ensuring that noise or less relevant aspects are suppressed.

This gated memory signal is then combined element-wise ($\odot$) with v_{i-1}, the representation of the item viewed just before the current one. This design ensures that summary generation is influenced not only by long-term user memory but also by short-term context from the most recent interaction.

Finally, this composite representation is projected against $\text{RWC}(v_i^{summary})$, the Random Walk Conformer embedding of the actual summary text for item i. This embedding provides a stable, content-specific reference that encodes structural and semantic aspects of the summary.

Altogether, $v_i^{SummGen}$ evaluates how consistent the generated summary is with both the user's evolving memory and the context of their last interaction. A higher value of $v_i^{SummGen}$ indicates that the generated summary not only reflects user preferences but also flows naturally from the previously consumed item, reinforcing coherence in the user's content experience.

E. Drift: Measuring Semantic Shift

$$\Delta_{drift} = \text{RMSD}\big(v_i^{summary}, v_{i-1}^{headline}\big) \tag{7}$$

a) Explanation.: The drift score Δ_{drift} quantifies the semantic shift between consecutive items by comparing their embeddings. Specifically, $v_i^{summary}$ represents the Random Walk Conformer embedding of the actual summary of the current item, while $v_{i-1}^{headline}$ denotes the headline embedding of the previous item. The comparison between these two captures how much the user's focus transitions from the pre-interest stage (as signaled by the headline of the previous item) to the post-interest stage (as described by the summary of the current item).

The function RMSD$(\cdot)$, or Root Mean Square Deviation, serves as a robust distance metric that measures the degree of mismatch between the two embedding vectors. A smaller Δ_{drift} indicates that the summary of the new item is semantically close to the headline of the previous item, implying smoother topical continuity and lower cognitive effort for the user. Conversely, a larger Δ_{drift} reflects a significant semantic gap, signaling a strong shift in user attention or a possible disruption in narrative flow.

This formulation provides an interpretable measure of semantic drift, which is crucial for modeling how users transition between items in a sequential browsing or reading environment.

F. Temporal Trust

$$T_i = \gamma_i \cdot \frac{i}{i+\epsilon} \tag{8}$$

a) Explanation.: The temporal trust T_i models how user reliability evolves over the course of interactions. Here, i denotes the current interaction step, such as the number of items the user has viewed so far. The parameter γ_i is a learnable gain factor that scales the overall trust level for each user, allowing personalization of trust dynamics. The denominator introduces a small constant ϵ to avoid division by zero at the initial step, while also smoothing the early growth rate of trust.

This formulation ensures that trust grows monotonically with the number of interactions: as i increases, the fraction $\frac{i}{i+\epsilon}$ approaches 1, thereby allowing T_i to asymptotically converge to γ_i. In effect, the model assumes that the more a user interacts with items, the more reliable their behavioral signals become. At the same time, the scaling factor γ_i guarantees that trust remains user-specific, rather than universal.

Thus, T_i provides a temporal prior on user reliability, strengthening the influence of long-term behavioral patterns in downstream formulations while still accounting for uncertainty in early interactions.

G. Memory Updation

a) Positive Memory.:

$$h_i^{(+)} = \text{LSTM}(\alpha_i, v_{i-1}, h_{i-1}^{(+)}) \tag{9}$$

Here, α_i is a learned interaction influence vector that captures the strength of the current positive signal (e.g., a click). The embedding v_{i-1} represents the most recently interacted item, acting as contextual evidence of endorsed content. The state $h_{i-1}^{(+)}$ is the prior positive memory cell that stores accumulated user approval history. The LSTM function integrates these signals, retaining long-term relevance while adaptively updating short-term preferences.

—

b) Negative Memory.:

$$h_i^{(-)} = \text{LSTM}(R_i, v_{i-1}, h_{i-1}^{(-)}) \tag{10}$$

Here, R_i is a rejection or disengagement signal, encoding the user's lack of interest (e.g., a skip or low attention). The embedding v_{i-1} again anchors the context of the previously viewed item. The prior negative state $h_{i-1}^{(-)}$ tracks accumulated disinterest patterns. The LSTM function allows this representation to evolve across time, while forgetting outdated rejection signals.

—

c) Drift-Aware Memory.:

$$h_i^{(\Delta)} = \text{LSTM}(\Delta_{\text{drift}}, v_{i-1}, h_{i-1}^{(\Delta)}) \tag{11}$$

Here, Δ_{drift} quantifies the semantic shift between pre-interest and post-interest representations (e.g., headline vs. summary). The embedding v_{i-1} provides contextual continuity, while $h_{i-1}^{(\Delta)}$ accumulates drift-sensitive memory that highlights mismatches or topical divergence. The LSTM gating mechanism ensures that interest-shift signals are incorporated while maintaining stability of long-term patterns.

—

d) Final Aggregated Memory.:

$$\hat{h}_i = \text{softmax}\left(h_i^{(+)}, h_i^{(-)}, h_i^{(\Delta)}\right) \tag{12}$$

The final user memory $\hat{h}_i$ is obtained as a convex combination of three specialized pathways:

- $h_i^{(+)}$: Positively reinforced memory, capturing confirmed user interests.
- $h_i^{(-)}$: Negatively reinforced memory, encoding avoidance and disinterest.
- $h_i^{(\Delta)}$: Drift-aware memory, tracking expectation mismatches and interest shifts.

The softmax$(\cdot)$ assigns normalized weights to each branch, enabling dynamic re-weighting based on current context. This ensures that $\hat{h}_i$ reflects a balanced, adaptive user representation across positive, negative, and drift-driven influences.

H. Stitching Engagements

To obtain a unified representation of user engagement history, we iteratively stitch the embeddings from prior interactions.

$$\hat{E}_i = \sum \left(W_i^{(Q)} \cdot \hat{E}_{i-1} \right) \tag{13}$$

Here, $\hat{E}_i$ is the stitched engagement embedding at step i, and $W_i^{(Q)}$ determines how much importance is assigned to past embeddings. Depending on modeling assumptions, $W_i^{(Q)}$ can take one of the following forms:

—

a) Exponential Decay (Queue Style).:

$$W_i^{(Q)} = \exp\left(-\gamma(t-i)\right) \tag{14}$$

This style applies a temporal decay, where γ controls the rate of forgetting. Recent interactions are weighted more heavily than older ones, mimicking a queue where older items gradually lose influence.

—

b) Strictly Memoryless (Stack Style).:

$$W_i^{(Q)} = \mathbb{1}_i \tag{15}$$

In this case, only the most recent embedding is retained, while all prior signals are discarded. This mimics a stack-style update where the system relies exclusively on the latest interaction, treating history as memoryless.

—

c) Attention-Styled (Priority Queue).:

$$W_i^{(Q)} = \text{softmax}\left(\langle W_q \hat{E}_{1:i-1}, W_h \hat{h}_{1:i-1} \rangle \right) \tag{16}$$

Here, attention weights are computed using the dot-product similarity between past stitched embeddings $\hat{E}_{1:i-1}$ and historical memory states $\hat{h}_{1:i-1}$, parameterized by projection matrices W_q and W_h. The softmax$(\cdot)$ normalizes the relevance scores, allowing the model to selectively prioritize certain past interactions based on contextual similarity.

—

d) Interpretation.: The choice of $W_i^{(Q)}$ determines the balance between recency bias, memory compression, and adaptive prioritization:

- Exponential Decay: emphasizes recency while gradually forgetting older engagements.
- Strictly Memoryless: discards history, relying solely on the latest interaction.
- Attention-Styled: dynamically retrieves relevant past signals, adapting based on contextual needs.

This flexible weighting framework enables the model to stitch engagements into a coherent, context-aware sequence embedding $\hat{E}_i$.

III. Baseline Pipeline

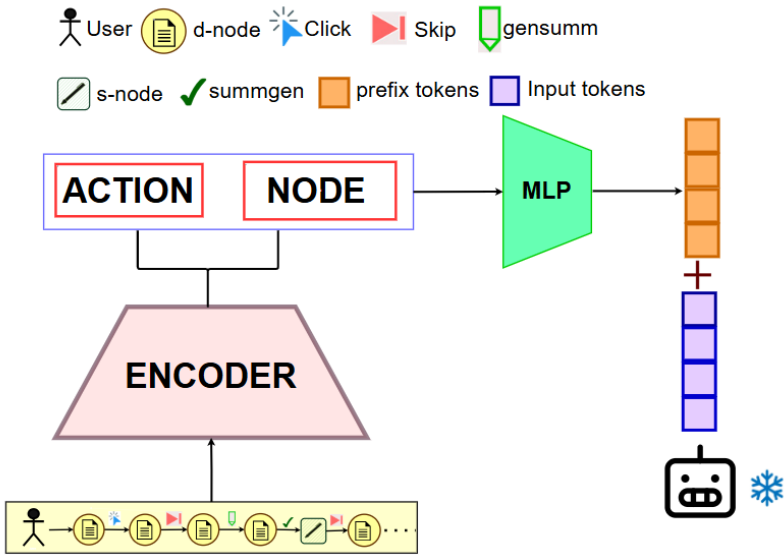

Fig. 1. Baseline pipeline: user encoder, engagement predictor, prefix adaptation, and frozen T5 decoder.

IV. Baseline Pipeline

Figure 1 illustrates the baseline pipeline that we implemented. This pipeline integrates user interaction encoding with a frozen T5 decoder, and serves as the foundation before injecting our custom formulations.

a) Input Interactions.: The pipeline begins with user interactions (e.g., click, skip, read). These are encoded as embeddings and processed sequentially.

b) User Encoder.: A recurrent encoder (GRU/LSTM) processes the sequence and produces hidden states h_i via backpropagation through time (BPTT). This captures the user's engagement history.

c) Engagement Prediction Head.: From each hidden state h_i, the model predicts:

- $\hat{a}_i$: the next action type (e.g., click/skip).
- $\hat{e}_i$: the engagement strength.

These predictions are supervised with classification/regression losses.

d) Prefix Adaptation.: The hidden state h_i is mapped via an MLP into prefix tokens. These tokens are continuous vectors that condition the downstream T5 model without modifying its parameters.

e) Frozen T5 Decoder.: The prefix tokens are concatenated with the text input and passed into the HuggingFace T5 decoder. The T5 model remains frozen; only the prefix embeddings and encoder/predictor are updated.

f) Summary Generation.: The decoder generates an abstractive summary $\hat{S}$, which is compared with the gold summary S^* using cross-entropy loss.

g) Training Objective.: The total objective combines:

- Engagement/action prediction losses.
- Summary generation loss from T5.

h) Summary.: Thus, the baseline pipeline encodes user interactions, predicts engagement, and conditions a frozen T5 decoder via prefix-tuning to generate summaries, while keeping the backbone language model fixed.

V. Conclusion

In this report, we have described the baseline pipeline for our personalized summarization framework. The baseline combines user interaction encoding, an engagement prediction head, and prefix-tuning with a frozen T5 decoder. This pipeline provides a stable foundation for integrating user signals into the summarization process.

We also outlined a set of proposed formulations, including drift modeling, temporal trust, and multi-path memory updates. At this stage, these formulations are presented as theoretical extensions and have not yet been injected into the baseline. Furthermore, it is important to emphasize that these formulations are still evolving; as our understanding deepens, modifications or refinements may be introduced. Thus, the equations presented here should be seen as an ongoing design process rather than finalized contributions.

Since our work so far is limited to establishing the baseline and proposing formulations, no experimental results are reported in this document. Future iterations will focus on implementing these formulations, updating the baseline accordingly, and evaluating their impact on personalization performance.