IMPerSumm: Information-Modulated User Preference Modeling for Personalized Text Summarization

Anonymous Author(s)

Abstract

Document summarization helps readers focus on the content-of-interest – a *highly subjective* and time-variant quantity, making *personalized summarization* essential. Prior state-of-the-art (SOTA) approaches either flatten evolving preferences into static personas, rely on LLM prompting that fails under long-horizon histories, or encode user interactions without explicit action types (*click*, *skip*, *summarize*, *click summary*), leaving preference dynamics under-modeled. To this end, we propose IMPerSumm, an information-modulated user preference encoder for personalized text summarization. IMPerSumm represents user histories as trajectories in a *user-interaction* graph. It dynamically modulates the mutual information between past interactions and new action-induced signals over a trajectory, and models short-term reactivity, long-term drift, and rare-event reinforcement using memory kernels. We empirically evaluate IMPerSumm on three benchmark datasets – PENS, OpenAI-Reddit, and PersonalSum using PerSEval, a personalization metric with strong human correlation. IMPerSumm outperforms SOTA personalized summarizers (PENS, GTP, Signature-Phrase) by 0.43 ↑ (on avg.), and strong LLM baselines (DeepSeek-R1-14B, LLaMA-2-13B, Mistral-7B, Zephyr-7B) by 0.37 ↑. Strikingly, we also observe that although the IMPerSumm encoder is trained on the personalized summarization task on the PENS dataset, it outperforms SOTA news recommendation systems on the standard *de-facto* MIND benchmark dataset (accuracy gain w.r.t best: 2.5/3.5/6.6 ↑ w.r.t MRR/nDCG@5/10). This shows that the IMPerSumm encoder develops cross-task capability of news recommendation.

ACM Reference Format:

Anonymous Author(s). 2026. IMPerSumm: Information-Modulated User Preference Modeling for Personalized Text Summarization. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

With information overload, personalized summarization is vital for tailoring updates to a reader's specific interests, especially for multi-aspect documents catering to diverse topics [Dasgupta et al. 2024]. Most existing methods rely on *static* persona traits [Dou et al. 2021; He et al. 2022; Li et al. 2023]. However, preference datasets representing user reading behavior, like MS/CAS PENS [Ao et al. 2021], show that user interests shift over time at fine-grained subtopic levels. This poses a challenge even for state-of-the-art (SOTA) LLMs, which underperform when detailed user-interaction histories are injected in prompts for in-context personalized summarization [Liu et al. 2024; Patel et al. 2024].

Recent models aim to *explicitly* represent the evolving user interaction (i.e., *click*) trajectories. However, they fail to model either the temporal ordering of the user-interaction sequence or the nuances of the other user-interactions (e.g. *skip*, *summarize*, etc.) that affect the information content flow over the trajectory [Ao et al. 2021; Lian et al. 2025; Okura et al. 2017; Song et al. 2023; Wu et al. 2019a]. This is empirically shown from the benchmarking by Dasgupta et al. [2024] w.r.t. PerSEval, the only personalized summarization metric with strong human-judgment correlation.

To address this gap, we propose IMPerSumm, an encoder-decoder-based personalized summarizer that represents user reading history as an *information-content* (IC) *flow* modulated by user actions (*click* vs. *skip*). At each timestep, the IMPerSumm encoder aligns the current document's IC with the accumulated history, estimates the document novelty (thereby, new interest) using a Kernel Density Estimation (KDE)-based distributional surprise model, and modulates the flow accordingly. A multi-layer perceptron (MLP)-based header predicts the next behavior embedding that entangles the query document and the corresponding latent summary. The decoder then disentangles the latent summary embedding and generates a contextualized query embedding. This is then fed into a pre-trained frozen *vanilla* summary-decoder for the final personalized summary.

We pose three core research questions to investigate IMPerSumm's effectiveness as an end-to-end personalized summarizer:

- **RQ-1:** To what extent does explicit action-specific user encoding affect personalized summarization quality (w.r.t PSE) compared to SOTA RNN and Transformer-based encoders?
- **RQ-2:** Beyond encoder-based modeling, can stronger personalization emerge either from history-prompted mid-size LLMs or from oracle-style summarizers with access to gold references?
- **RQ-3:** How well does IMPerSumm's encoder trained for personalized summarization on news datasets (PENS and PersonalSum) generalize to the related task of personalized recommendation, compared to SOTA news recommendation models?

To address RQ-1, we use PENS [Ao et al. 2021], Norwegian PersonalSum [Zhang et al. 2024], and OpenAI Reddit [Völske et al. 2017] datasets. We benchmark the PENS summarization framework (augmented with NAML [Wu et al. 2019a], EBNR [Okura et al. 2017], NRMS [Wu et al. 2019b] user encoder), GTP [Song et al. 2023] (using the TrRMIo encoder) and Signature-Phrase [Cai et al. 2023], and observe that IMPerSumm yields **0.49/0.48/0.5**↑ w.r.t PSE-JSD/SU4/METEOR. For RQ-2, we evaluate Zephyr-7B [Tunstall et al. 2023], LLaMa2-13B [Touvron et al. 2023], Mistral-7B [Jiang et al. 2023], and DeepSeek-R1-14B [DeepSeek-AI et al. 2025] as strong baselines. We observe that IMPerSumm outperforms the best baseline (DeepSeek-R1) by **0.27/0.41/0.42**↑. We also follow Patel

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

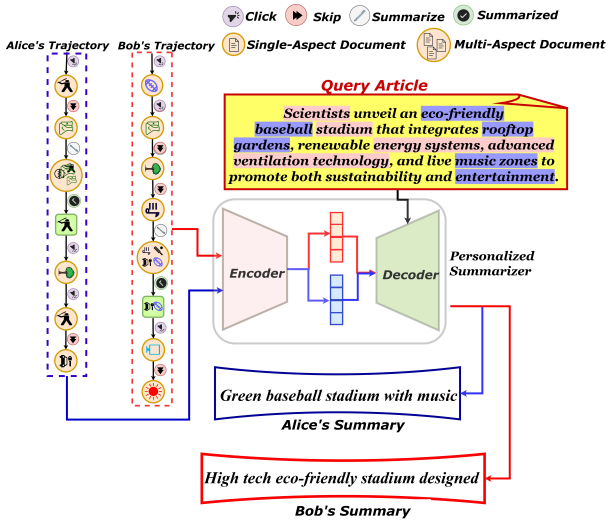

Figure 1: Personalized Summarization Pipeline: A behavior-aware encoder–decoder encodes each user's trajectory as the prefence condition to generate personalized summaries of the same query article.

et al. [2024] to evaluate IMPerSumm against oracle BigBird-Pegasus [Zaheer et al. 2020] and SimCLS [Liu and Liu 2021] (having access to gold references) – IMPerSumm outperforms by **0.32**↑. For RQ-3, we evaluate the cross-task capability of the PENS-trained IMPerSumm-encoder by comparing against 30 strong news recommendation baselines on the standard, widely adopted MIND-large news recommendation dataset. We find that IMPerSumm's encoder surpasses the best baseline FastFormer-PLM-NR [Wu et al. 2021] by **3.5/6.6**↑ in terms of nDCG@5&@10 and **2.5**↑ w.r.t. MRR.

2 Related Work

Personalized Summarization Evaluation: Personalized summarization aims to align generated summaries with a user's subjective expectations, as signalled by the evolving reading behavior (clicks, skips, summarize interactions). Traditional accuracy metrics fail to capture this personalization, as shown by Vansh et al. [2023], who proposed the EGISES metric. PerSEval [Dasgupta et al. 2024] refines EGISES by penalizing accuracy drops. We adopt PerSEval to evaluate our proposed IMPerSumm summarizer.

Training/Evaluation Datasets: Effective personalization demands datasets with (i) temporally-ordered user interactions, (ii) user-specific gold summaries for shared documents, and (iii) diverse, shifting topics. Unlike CNN/DM or MultiNews [Fabbri et al. 2019; Hermann et al. 2015], PENS [Ao et al. 2021] and translated PersonalSum [Zhang et al. 2024] satisfy these. We use both for our experiments, and also OpenAI-Reddit [Völske et al. 2017] to synthetically derive temporal orders for personalized summarization. PENS logs clicks/skips and summaries per user, averaging 13.6 topics and 52.83 sub-topics, with topic change rates of 0.77/0.93 per step. Its fine-grained behavioral diversity makes it a standard benchmark [Lian et al. 2025; Song et al. 2023; Yang et al. 2023].

Personalized Summarizers: Most personalized summarizers assume a *static* user persona guided summarization (e.g., GSUM, CTRLSum, TMWIN, Tri-Agent) [Dou et al. 2021; He et al. 2022; Kirstein et al. 2024; Xiao et al. 2024]. On the other hand, models such as PENS augment external *dynamic* user-trajectory encoders such as NRMS, NAML, and EBNR [Ao et al. 2021; Okura et al. 2017; Wu et al. 2019a,b]. Others, such as GTP [Song et al. 2023], generate control but omit temporal distinctions. SCAPE integrates content and stylistic features [Lian et al. 2025]. However, none capture session-aware, action-specific dynamics. Although prompting LLMs with preference history, termed *In-Context-Personalization Learning (ICPL)* [Patel et al. 2024], outperforms these models, it fails in practice. This is because most LLMs cannot handle long, complex prompts and suffer the "*lost-in-the-middle*" effect – i.e., they ignore important trajectory segments that are in the middle [Gao et al. 2024; Liu et al. 2024; Qiu et al. 2024]. Moreover, richer preference cues can degrade personalization, illustrating the iCOPERNICUS "*Paradox of Less is More*" [Patel et al. 2024]. To capture action-specific dynamics, IMPerSumm leverages MI-based relevance [Darrin et al. 2024; van den Oord et al. 2019; Zhang et al. 2021].

Personalized Summarization and Recommendation. While personalized summarization and recommendation have long evolved in parallel, recent work highlights their convergence. The PENS framework shows that user embeddings from recommendation models can guide abstractive summarization, while SumRecom [Ghodratnama and Zakershahrak 2024] casts summarization itself as a recommendation task via user feedback. Conversely, summarization enhances recommendation by generating explanatory profiles, as in PGHIS [Liu et al. 2026]. Multi-task approaches such as P5 [Zhang et al. 2022] further demonstrate that user preference signals transfer across retrieval, ranking, and generation. Collectively, these works point to a shared foundation in modeling user histories, with different outputs (ranked items vs. generated summaries). By enabling summarization-to-recommendation transfer without retraining, IMPerSumm advances this line toward unified user models bridging generative and predictive personalization.

3 Personalized Summarization: Formulation

In this direction, we introduce the User-Interaction-Graph (UIG), a data model for dynamic behavior trajectories.

3.1 Preference Data Modeling

We represent user histories as a **User–Interaction Graph** (UIG), a directed acyclic graph $G = \langle N, E \rangle$, where the node set N consists of three disjoint types: (i) **u-nodes** $u^{(t_0)}$ denoting a user at initial timestep t_0, (ii) **d-nodes** $d^{(t_p)}$ representing documents interacted at timestep t_p, and (iii) **s-nodes** $s_j^{(t_q)}$ representing user-specific summaries requested or generated at time t_q for a document viewed at t_{q-1}. The edge set E encodes user actions: $a_d^{(t_p)} \in \{click, skip, summarize\}$ on documents, and $a_s^{(t_q)}$ as the follow-up *summGen* action connecting a document $d^{(t_q-1)}$. A user **trajectory** τ_u is then a time-ordered sequence of such interactions, beginning at $u^{(t_0)}$. Each trajectory can be decomposed into **behavior triples** $b_{u_j}^{t_i} : \langle hd^{(t_{i-1})}, a^{(t_i)}, tl^{(t_i)} \rangle$, denoting the action-triggered transition connecting the *hd*-node $hd^{(t_{i-1})}$ and *tl*-node $tl^{(t_i)}$. Such triples

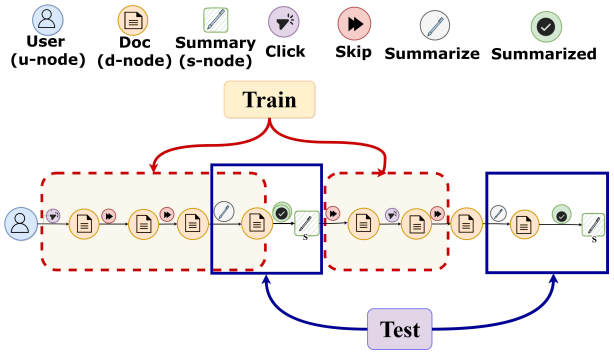

Figure 2: User–Interaction Graph construction linking u-, d-, and s-nodes through action edges (click, skip, summarize) over training and test partitions; see Appendix C for details.

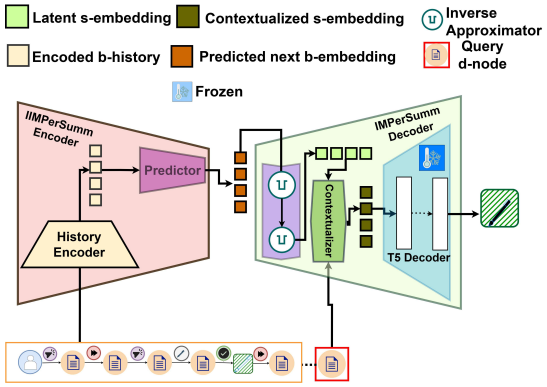

Figure 3: IMPerSumm End-to-end pipeline diagram: User trajectory is encoded by History Encoder; Predictor estimates the next behavior node; Query document, contextualized by extracted latent summary, is fed into (frozen) T5-decoder for personalized summary.

may represent a *document-to-document*, *document-to-summary*, and *summary-to-document* transitions. The UIG $\mathcal{T}$ is a pool of trajectories, used as $\mathcal{T}_{\text{train}}$ for training and $\mathcal{T}_{\text{test}}$ for evaluation. Hence, a UIG is a dynamic temporal knowledge graph (TKG) of user behavior (UIG construction: Figure 2; Appendix. C).

While UIG captures rich temporal detail, directly modeling its raw structure quickly becomes computationally expensive and noisy over long horizons. This is particularly challenging in personalized summarization, where fine-grained shifts in preference must be retained without overwhelming the model. Recent work in sequential recommendation suggests that *hierarchical abstractions* improve long-horizon accuracy by condensing low-level interactions into higher-order behavioral units [Cho and Hyun 2023; Ou et al. 2025; Pan and Wang 2021; Zhang et al. 2020; Zhu et al. 2023]. Motivated by this, we introduce a bi-level UIG: each behavior triple $b_{u_j}^{(t_i)}$ is a **b-node** in the higher-level **b-tier** $\tau_b^{u_j}$ constructed on top of the *u-tier* τ^{u_j}, linked by ***nextBehavior*** edges. Task-1 (next b-node prediction) operates on the b-tier. We construct u-tier and b-tier trajectories from PENS [Ao et al. 2021], English PersonalSum [Zhang et al. 2024], and OpenAI (Reddit) [Völske et al. 2017] as $\mathcal{T}^{\text{P}}_{\text{train}}$, $\mathcal{T}^{\text{P}}_{\text{test}}$, $\mathcal{T}^{\text{PS-EN}}_{\text{train}}$, $\mathcal{T}^{\text{PS-EN}}_{\text{test}}$, and $\mathcal{T}^{\text{OAI}}_{\text{train}}$, $\mathcal{T}^{\text{OAI}}_{\text{test}}$.

Problem Formulation. We formulate the personalized summarization task into three stages: **Task 1** - predict the next behavior triple $b_{(q,u_j)}$ (i.e., embedding) from τ^{u_j} embedding and d_q embedding as user's next interest, **Task 2** - decode $b_{(q,u_j)}$ to the extract the latent representation of $s_{(q,u_j)}$, and **Task 3** - generate personalized summary using a frozen pre-trained summarizer decoder by feeding in $s_{(q,u_j)}$ embedding contextualized by d_q embedding.

4 Approach

4.1 Preference as Information Flow Modulation

A central challenge in modeling evolving user preferences is balancing **continuity with the past** against **adaptation to novelty**. When a user clicks, skips, or requests a summary, the model must decide: *does this reinforce existing interests, or mark a shift in focus?* We frame this as a problem of **information flow modulation**, where each new interaction is judged by how much information it shares with accumulated history. Mutual Information (MI) provides a principled measure of such dependency: high MI implies continuity with prior interests, while low MI suggests novelty or drift. Since preference signals reside in high-dimensional embedding spaces, MI is estimated using **kernel density estimation (KDE)** [Silverman 1986; Zhelezniak et al. 2020], which smooths distributions over interaction embeddings to compute alignment between current and historical representations. Preference evolution, however, unfolds across **multiple timescales**. Short-term kernels capture immediate reactivity, long-term kernels encode gradual drift, and rare-event kernels highlight infrequent but decisive shifts. Moreover, not all actions contribute equally: skips convey weaker or negative signals, clicks reinforce engagement, while summary requests mark deeper intent. In this way, a user trajectory τ^{u_j} is modeled as a sequence of **regressive information updates**, where timescale-specific kernels offer complementary views of history and action-specific gates modulate their strength. This yields a dynamic, information-theoretic account of preference evolution sensitive to both *when* an action occurs and *how* it is expressed. Section 4.2 instantiates these principles within the IMPerSumm architecture.

4.2 The IMPerSumm Model

Building on preference-as-information-flow modulation, we propose IMPerSumm as an encoder–decoder instantiation. IMPerSumm addresses three tasks defined in Section 3.1 sequentially using an encoder for task 1, a decoder contextualizer for task 2, and a frozen pre-trained T5-decoder for task 3. The end-to-end pipeline is shown in Figure 3 (Notations: Table 7 in Appendix).

4.2.1 Task 1: Next b-node Prediction. We now detail how IMPerSumm encoder models $\tau_b^{u_j}$ via a regressive information modulation cell (Section 4.1), producing a high-dimensional embedding for $\tau_b^{u_j}$.

Initialization of u-tier. To enable Task-1 on the b-tier, we initialize the user's u-tier trajectory τ^{u_j} by embedding each document and summary node using the E5 model [Wang et al. 2024]. For each behavior triple $b_{u_j}^{(t_i)}$, the head and tail embeddings are $\mathbf{e}_{hd}^{(t_{i-1})}$ and

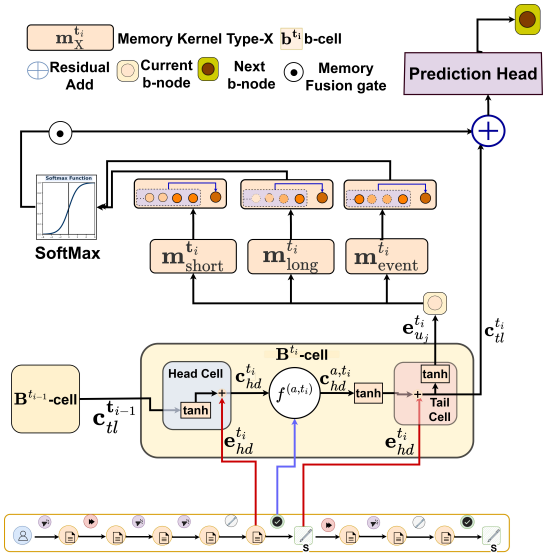

Figure 4: IMPerSumm Encoder: A chain of b-cells (Head & Tail cells), each representing a b-node of the underlying UIG (see Section 3.1), form b-tier; Information Content Flow is modulated in the b-cell based on user action to generate b-node embedding; Memory Functions (w/ kernels) further modulate b-node embedding; Prediction head predicts next b-node.

$\mathbf{e}_{tl}^{(t_i)}$, respectively. The initial u-node embedding $\mathbf{e}_{u_j}^{(t_0)}$ derives from the title embedding of $d^{(t_0)}$, addressing cold starts. This seeds the user's starting information state, anchored in the first document's topical semantics before any behavior-driven modulation.

b-tier Architecture. IMPerSumm's *b*-tier encoder models the evolving $\tau_b^{u_j}$ using a recurrent architecture of **b-cells**, each of which produces a *b*-node embedding ($\mathbf{e}_{b_{u_j}}^{(t_i)}$) at timestep t_i. Each *b*-cell integrates user interactions through three sequential components, (i) the **head-cell**, (ii) the **action-control**, and (iii) the **tail-cell**. The t_i-th head-cell fuses the incoming **behavior history** (i.e., the previous b-cell's **tail-cell content** $\mathbf{c}_{tl}^{(t_{i-1})}$) and the *hd*-node embedding $\mathbf{e}_{hd}^{(t_{i-1})}$ to generate the **head-cell content** $\mathbf{c}_{hd}^{(t_i)}$ as follows (W_h, W_{hd} are learnable): $\mathbf{c}_{hd}^{(t_i)} = \tanh\left(W_h \cdot \mathbf{c}_{tl}^{(t_{i-1})} + W_{hd} \cdot \mathbf{e}_{hd}^{(t_i)}\right)$. The action-control ($a$), representing one of the four possible transition actions, then adjusts the $c_{hd}^{(t_i)}$, and modulates what information is to be retained, enhanced, or discarded using an information-alignment function $f_\sigma^{(a,t_i)}$ as: $\mathbf{c}_{hd}^{(a,t_i)} = f_\sigma^{(a,t_i)}(\mathbf{c}_{hd}^{(t_i)}, \mathbf{e}_{tl}^{(t_i)})$. The tail-cell then combines the action-modulated $\mathbf{c}_{hd}^{(a,t_i)}$ with the tail-node embedding $\mathbf{e}_{tl}^{(t_i)}$ to produce the **tail-cell content** $\mathbf{c}_{tl}^{(t_i)}$ and corresponding **b-node embedding** $\mathbf{e}_{b_{u_j}}^{(t_i)}$ as: $\mathbf{c}_{tl}^{(t_i)} = \tanh(W_a \cdot (\mathbf{c}_{hd}^{(a,t_i)} + \mathbf{e}_{tl}^{(t_i)}))$; $\mathbf{e}_{b_{u_j}}^{(t_i)} = \tanh(W_b \cdot \mathbf{c}_{tl}^{(t_i)} + \mathbf{b}_b)$.

Base Encoder. In its simplest form, the **information-alignment function** $f_\sigma^{(a,t_i)}$ is a cosine similarity, such that $\mathbf{c}_{hd}^{(a,t_i)} = \lambda_c \cdot f_{\cos}(\mathbf{c}_{hd}^{(t_i)}, \mathbf{e}_{tl}^{(t_i)})$, where λ_c is a learnable influence factor to capture the influence of every alignment signal. However, embedding similarity (such as cosine) lacks the ability to specifically capture how much information is carried over *in contrast to new information*. Moreover, it does not capture the nuances of action-specific updates explicitly for *clicks, skips, or summarize*.

MI (Mutual Information) Encoder. To capture nuanced, action-specific information flow (or modulation) across each *b*-cell, we model the information-alignment function $f_\sigma^{(a,t_i)}$ as **KDE-based pair-wise mutual information modulation** (see Section 4.1) as: $f_{\mathrm{I}}^{(a,t_i)} = I(tl^{(t_i)}; hd^{(t_{i-1})}) = I_K(\mathbf{e}_{tl}^{(t_i)}; \mathbf{c}_{hd}^{(t_i)})$, where $I(\cdot;\cdot)$ denotes mutual information. High I indicates continuity with past preferences, while low I indicates novelty or drift. In practice, we estimate I using a **kernelized estimator**: $I_K(\mathbf{e}_{tl}^{(t_i)}; \mathbf{c}_{hd}^{(t_i)})$. The marginals and joints are obtained via **Gaussian kernel density estimation (KDE)** [Zhelezniak et al. 2020]. Given cumulative history embeddings $\{\mathbf{c}_{hd}^{(t_j)}\}_{j=1}^{i-1}$, the marginal density is

$$\hat{p}(\mathbf{e}_{tl}^{(t_i)}) = \frac{1}{(i-1)\,\lambda} \sum_{j=1}^{i-1} K\left(\frac{\mathbf{e}_{tl}^{(t_i)} - \mathbf{c}_{hd}^{(t_j)}}{\lambda}\right),$$

where $K(\cdot)$ is the Gaussian kernel and λ the bandwidth. Here $(i-1)$ is the number of past interactions. Smaller λ emphasizes fine-grained alignment with recent history, larger λ smooths over longer dependencies[1]. The modulated head-cell content $\mathbf{c}_{hd}^{(clk,t_i)}$ of the *click* action is generated using the learnable *click-gate* $g^{(clk,t_i)}$ that depends on the (normalized) time spent (called *dwell time*; δ_{dwell}) by the user on any d-node. In other words, if a user spends less time on a node, then the information content of the b-cell should decrease. $\mathbf{c}_{hd}^{(clk,t_i)}$ is computed as: $\mathbf{c}_{hd}^{(clk,t_i)} = g^{(clk,t_i)} \odot I_K(\mathbf{e}_{tl}^{(t_i)}; \mathbf{c}_{hd}^{(t_i)})$, where: $g^{(clk,t_i)} = \frac{\delta_{\text{dwell}}^{t_i}}{\max_{j=1:(N-U)} \delta_j} \cdot \tanh(W_g^{clk} \cdot \mathbf{e}_{tl}^{(t_i)})$

$\mathbf{c}_{hd}^{(skp,t_i)}$ for the *skip* action is generated using the learnable *skip-gate* $g^{(skp,t_i)}$. Since *skip* denotes user's disinterest in the content of the tail-node to certain degree, $g^{(skp,t_i)}$ determines *how much of* $\mathbf{e}_{tl}^{(t_i)}$ *is to be suppressed* (i.e., overlapping information with the incoming history should not be erased), and *how much of the inflowing history is to be retained* (i.e., redundant information in the history needs to be erased). $\mathbf{c}_{hd}^{(skp,t_i)}$ is computed as:

$$\mathbf{c}_{hd}^{(skp,t_i)} = g^{(skp,t_i)} \odot \mathbf{e}_{tl}^{(t_i)} + (1 - g^{(skp,t_i)}) \odot I_K(\mathbf{e}_{tl}^{(t_i)}; \mathbf{c}_{hd}^{(t_i)}), \tag{1}$$

where $g^{(skp,t_i)} = \tanh(W_g^{skp} \cdot \mathbf{e}_{tl}^{(t_i)})$. The *summarize* command action signals a *focused intent* of the user. Thus, the information is modulated via a *focus-gate* $g^{(fc,t_i)}$. $g^{(fc,t_i)}$ determining the information likely to be salient (contained in $tl^{(t_i)}$) if clicked. Since the *summarize* command comes with an interest developed by the **title** of the document, $g^{(fc,t_i)}$ captures the *interest in the title* of the *d*-node to be summarized, with the *subjective focus (fc)*.

$$\mathbf{c}_{hd}^{(fc,t_i)} = g^{(fc,t_i)} \odot I_K(\mathbf{e}_{tl}^{(t_i)}; \mathbf{c}_{hd}^{(t_i)}), \tag{2}$$

[1]We set $\lambda = 0.6$ via Silverman's rule-of-thumb [Silverman 1986], $\lambda = 1.06\,\hat{\sigma}\,n^{-1/5}$, where $\hat{\sigma}$ is the sample standard deviation and n the sample size.

where $g^{(fc,t_i)} = \tanh(W_g^{fc} \cdot \mathbf{e}^{(t_i)}_{\textbf{TITLE}(tl)})$. Each *summarize* is followed by an automated *summGen* transition-action, which generates the user's expected summary s^{u_j} for the document d^{t_i} from the u-tier. Precisely, $\mathbf{c}^{(summ,t_i)}_{hd}$ is generated as a linear combination of two learnable gates: *interest-gate* $g^{(summ,t_i)}$, and *accuracy-gate* $g^{(acc,t_i)}$. The *interest-gate* $g^{(summ,t_i)}$ captures the history-influenced interest of the user, determining the content alignment of the document, and *accuracy-gate* $g^{(acc,t_i)}$ ensures the alignment of expected summary s^{u_j} with document d^{t_i}, ensuring the summary-content to be faithful to the document as:

$$\mathbf{c}^{(summ,t_i)}_{hd} = g^{(summ,t_i)} \odot I_K(\mathbf{e}^{(t_i)}_{tl}; \mathbf{c}^{(t_i)}_{hd}) + g^{(acc,t_i)} \odot I_K(\mathbf{e}^{(t_i)}_{tl}; \mathbf{e}^{(t_i)}_{hd}) \tag{3}$$

where $g^{(summ,t_i)} = \tanh(W_g^{summ} \cdot \mathbf{e}^{(t_i)}_{tl})$ and $g^{(acc,t_i)} = \tanh(W_g^{acc} \cdot \mathbf{e}^{(t_i)}_{hd})$. Although $\mathbf{e}^{(t_i)}_{b_{u_j}}$ captures fine, action-specific updates at each timestep t_i, it's a local representation, biased to immediate interactions and weak at modeling long-term shifts or separating fleeting spikes from sustained themes.

Memory Kernel Augmentation. To align users' short-term current interest with that of the longer-term persistent interests, we propose a multi-scale memory mechanism inspired by hierarchical caches [He et al. 2025] and kernelized state-space models like S4 and Mamba [Gu and Dao 2024; Gu et al. 2022]. These works affirm that fusing short-term reactivity with long-term continuity improves sequential modeling. Unlike hierarchical memory slots or implicit ODE states, we use three types of temporally kernelized (via $\mathcal{K}$ kernels) memory functions ($\mathbf{m}^{(t_i)}$): (i) short-term, (ii) long-term, and (iii) event-specific. $\mathbf{m}^{(t_i)}$ enables the past b-node embeddings $\{\mathbf{e}^{(t_k)}_{b_{u_j}}\}_{k=1:i-1}$ to shape the present b-node embeddings $\mathbf{e}^{(t_i)}_{b_{u_j}}$. The short-term memory ($\mathbf{m}^{(t_i)}_{\text{short}}$) captures recent, *reactive* behavior patterns as: $\mathbf{m}^{(t_i)}_{\text{short}} = \sum_{j=1}^{i-1} \mathcal{K}^{t_j}_{\text{short}} \odot \mathbf{e}^{(t_i)}_{b_{u_j}}$; $\mathcal{K}^{t_j}_{\text{short}} = \exp(-W_{\text{short}} \cdot \mathbf{e}^{(t_j)}_{b_{u_j}})$, where $\mathcal{K}_{\text{short}}$ encodes Markovian recency by assigning exponentially decaying weights. $\mathbf{m}^{(t_i)}_{\text{short}}$ is ideal for fast-changing interests (e.g., interest spike in soccer articles during the World Cup). However, it forgets stable preferences (e.g., a basketball lover who returns to basketball articles week after week). To address this, the long-term memory function $\mathbf{m}^{(t_i)}_{\text{long}}$ models *persistent* user interests as: $\mathbf{m}^{(t_i)}_{\text{long}} = \sum_{j=1}^{i-1} \mathcal{K}^{t_j}_{\text{long}} \odot \mathbf{e}^{(t_i)}_{b_{u_j}}$, $\mathcal{K}^{t_j}_{\text{long}} = \exp\left(\frac{-W_{\text{long}}}{||W_{\text{short}}||} \cdot \mathbf{e}^{(t_j)}_{b_{u_j}}\right)$ with $\mathcal{K}_{\text{long}}$ is modeled by W_{long}, and scaled down by norm of W_{short} to ensure slower decay than $\mathcal{K}_{\text{short}}$. Thus, $\mathbf{m}^{(t_i)}_{\text{long}}$ captures drift-aware, reinforced preferences (e.g., recurring basketball reads). However, it can miss rare yet impactful deviations (e.g., a sudden interest in chess due to some viral news). Hence, event-specific memory function is added as: $\mathbf{m}^{(t_i)}_{\text{event}} = \sum_{j=1}^{i-1} \mathcal{K}^{t_j}_{\text{event}} \odot \mathbf{e}^{(t_i)}_{b_{u_j}}$, $\mathcal{K}^{t_j}_{\text{event}} = \exp(-(W_{\text{event}} \cdot e^{(t_j)}_{b_{u_j}})^\top \mathbf{e}^{(t_i)}_{b_{u_j}})$. $\mathbf{m}^{(t_i)}_{\text{event}}$ highlights isolated yet meaningful events–interactions that are far in the past but highly influential to current user intent. W_{event} transforms a past b-node embedding. The dot product with current b-node embedding $\mathbf{e}^{(t_i)}_{b_{u_j}}$ computes the affinity score. This score quantifies the impact of a past isolated behavior on the present.

Adaptive Memory Fusion (AMF). We now fuse all three memory functions to form the **memory-injected b-node embedding** ($\mathbf{m}^{(t_i)}_{u_j}$). However, the fusion needs to be adaptive because not all the memory functions (and features) should be equally activated at every step of τ^{u_j}. To address this, we first construct a **Memory Matrix** $\mathbf{M}^{(t_i)} = \textbf{SoftMax}([\mathbf{m}^{(t_i)}_{\text{short}}; \mathbf{m}^{(t_i)}_{\text{long}}; \mathbf{m}^{(t_i)}_{\text{event}}]^\top)$ that contains the normalized weighted (in %) memory feature across all three types. We then pool the most important (i.e., maximum weight) features across the three memory functions to create the fused **memory vector** $\mathbf{m}^{(\mathbf{t_i})}_{\text{fuse}} = \text{MaxPool}_{j \in \{\text{short,long,event}\}} \mathbf{M}^{(t_i)}_{i,j}$. While $\mathbf{m}_{\text{fuse}}$ extracts only the important features (count: 3) across the memory functions at any time-step, it still gives equal importance to all the functions. To address this, we further apply a *fusion-gate* g_{fuse} to generate the corresponding **gated memory vector** $\mathbf{m}^{(t_i)}_{u_j} = g^{(t_i)}_{\text{fuse}} \odot \mathbf{m}^{(t_i)}_{\text{fuse}}$, where $g^{(t_i)}_{\text{fuse}} = \tanh(W_{\text{fuse}} \cdot \mathbf{e}^{(t_j)}_{b_{u_j}})$. We add a residual $\mathbf{c}^{(t_i)}_{tl}$ to $\mathbf{m}^{(t_i)}_{u_j}$ to generate the final memory-infused b-node embedding $\mathbf{z}^{(t_i)}_b$.

Next b-node Prediction. Once the user history (trajectory of l time-steps) is encoded in $\mathbf{c}^{(t_l)}_{tl}$ (with the final b-node embedding $\mathbf{z}^{(t_l)}_b$), we apply a next b-node embedding prediction header W_{next} on it to predict the query b-node at t_{l+1} as $\mathbf{z}^{(t_{l+1})}_b = W_{\text{next}} \cdot \text{ReLU}(W_z \cdot \mathbf{z}^{(t_l)}_b) + \mathbf{b}_{\text{next}}$.

Encoder Training. The encoder training objective ($\mathcal{L}_{\text{enc}}$) is a linear combination of two loss functions: *Alignment Loss* ($\mathcal{L}_{\text{align}}$) and the *Prediction Loss* ($\mathcal{L}_{\text{next}}$) as $\mathcal{L}_{\text{enc}} = \alpha \cdot \mathcal{L}_{\text{align}} + (1-\alpha) \cdot \mathcal{L}_{\text{next}}$. $\mathcal{L}_{\text{align}}$ makes sure that during the next-node (i.e., $\mathbf{z}^{(t_{l+1})}_b$) prediction, the encoder maintains its original positional alignment at every step on $\tau^{u_j}_l$. Specifically, at each timestep t_i, a learnable position classifier W_{pos} on $\mathbf{z}^{(t_i)}_b$ to generate positional probability distribution $\hat{\mathbf{p}}^{(t_i)}_b$ over behavior node index-vocabulary B as $\hat{\mathbf{p}}^{(t_i)}_b = \textbf{SoftMax}(W_{\text{pos}} \cdot \mathbf{z}^{(t_i)}_b)$. The total alignment loss is $\mathcal{L}_{\text{align}} = -\frac{1}{l} \sum_{i=1}^{l} \log \hat{p}^{(t_i)}_b$. $\mathcal{L}_{\text{next}}$ over behavior node vocabulary B is $-\log \hat{p}^{(t_i)}_b$, ensuring that the encoder learns to predict the next behavior of the user.

4.2.2 Task 2: Latent Summary Contextualization. The query b-node embedding $\mathbf{z}^{(t_{l+1})}_b$ predicted by the encoder represents a memory-infused entanglement of the behavior triple $\langle d_q, genSumm, s_q \rangle$ for the query document d_q. A **base `IMPerSumm` decoder (b-Decoder)** feeds this directly into an internal pre-trained frozen model (decoder of a vanilla summarizer) for personalized summarization (task-3). However, $\mathbf{z}^{(t_{l+1})}_b$ lacks explicit contextualization of d_q, which can make it hard for the summarizer to decode (see Table 10). To address this, we extract the latent summary (s-node) embedding from $\mathbf{z}^{(t_{l+1})}_b$.

Latent s-Node Extraction We reconstruct the s-node via a two-step inverse mapping. First, we recover the b-node embedding $\hat{\mathbf{e}}^{(t_i)}_{b_{u_j}} = W_k^+ . \mathbf{z}^{(t_{l+1})}_b$, where W_k^+ is a learnable pseudo-inverse approximation kernel that disentangles the memory influence $\mathbf{m}^{(t_{l+1})}_{u_j}$ from $\mathbf{z}^{(t_{l+1})}_b$ to approximate $e^{(t_{l+1})}_b$. We then infer the personalized latent

summary embedding $\hat{\mathbf{e}}^{(t_{l+1})}_{\text{summ}} = W^{+}_{\text{summ}} \cdot \mathbf{e}^{(t_i)}_{b_{u_j}}$, W^{+}_{summ} is a learnable pseudo-inverse approximation kernel. While $\hat{\mathbf{e}}^{(t_{l+1})}_{\text{summ}}$ can also be fed to the internal frozen decoder (**s-Decoder**), it lacks the query document's representation completely.

Latent s-Node Contextualization. In contrast to the b-decoder and s-decoder, the IMPerSumm decoder contextualizes the extracted $\hat{\mathbf{e}}^{(t_{l+1})}_{summ}$ with the query document embedding using cross-attention. $\hat{\mathbf{e}}^{(t_{l+1})}_{summ}$ serves as the query, and the query document embedding $\mathbf{e}^{(t_l)}_{d}$ acts as both key and value, resulting in the contextualized latent summary embedding $\mathbf{e}^{(t_{l+1})}_{\text{p-summ}}$ as follows:

$$\mathbf{e}^{(t_{l+1})}_{\text{p-summ}} = \textbf{SoftMax}\left(\frac{(W_q.\hat{\mathbf{e}}^{(t_{l+1})}_{summ})^{\top}(W_k.\mathbf{e}^{(t_l)}_{d})}{\sqrt{d}}\right) \cdot W_v.e^{(t_l)}_{d} \quad (4)$$

where W_q, W_k and W_v are learnable projection matrices for the query, key, and value, respectively.

4.2.3 Task 3: Personalized Summarization. The document contextualized s-node $\mathbf{e}^{(t_{l+1})}_{\text{p-summ}}$ is fed into a pre-trained frozen decoder of a summarization model. We use the T5-large [Raffel et al. 2020] decoder for this purpose.

Decoder Training. The decoder training objective ($\mathcal{L}_{\text{dec}}$) is a linear combination of two loss functions, *Generation Loss* ($\mathcal{L}_{\text{gen}}$) and the earlier encoder loss ($\mathcal{L}_{\text{enc}}$; see Section 4.2.1), as $\mathcal{L}_{\text{dec}} = \beta \cdot \mathcal{L}_{gen} + (1-\beta) \cdot \mathcal{L}_{\text{enc}}$. Here $\mathcal{L}_{\text{gen}}$ is the cross-entropy loss between predicted tokens $\hat{y}$ and ground-truth y^* under teacher forcing with the frozen T5 decoder. Optimizing $\mathcal{L}_{\text{gen}}$ updates W_k, W_q, W_v and inverse-mapping weights W^{+}_{summ}, W^{+}_{k}. This ensures accurate latent s-node reconstruction $\hat{\mathbf{e}}^{(tl+1)}_{\text{summ}}$ and stronger cross-attention with document embedding $\hat{\mathbf{e}}^{(tl)}_{d}$, thus improving summary relevance.

5 Evaluation

To evaluate IMPerSumm as an effective personalized summarizer, we investigate the three research questions discussed in Section 1.

5.1 Experiment Setup

Training Data. We construct UIGs from PENS, English-translated PersonalSum (via M2M-100 [Fan et al. 2020]), and OpenAI (Reddit) as $\mathcal{T}^{\text{P-D}}_{\text{train}}$, $\mathcal{T}^{\text{PS-EN}}_{\text{train}}$, and $\mathcal{T}^{\text{OAI}}_{\text{train}}$ to train IMPerSumm. We sample 52K trajectories from $\mathcal{T}^{\text{P-D}}_{\text{train}}$ (avg. 134 d-nodes, 5 s-nodes each), 21K from $\mathcal{T}^{\text{OAI}}_{\text{train}}$ (avg. 39 d-nodes, 10 s-nodes), and 700 from $\mathcal{T}^{\text{PS-EN}}_{\text{train}}$. Each trajectory is sliced before every (d–s) pair – forming training history $\tau^{u_j}_{h}$, query document d_q, and target summary $s^{*u_j}_{q}$.

Test Data. We prepare four test sets: $\mathcal{T}^{\text{P-D}}_{\text{test}}$, $\mathcal{T}^{\text{PS-EN}}_{\text{test}}$, and $\mathcal{T}^{\text{OAI}}_{\text{test}}$ for end-to-end personalized summarization, and $\mathcal{T}^{\text{M}}_{\text{test}}$ from MIND test data [Wu et al. 2020] to test sequential recommendation. For $\mathcal{T}^{\text{P-D}}_{\text{test}}$, each user's test trajectory τ^{u_j}, sampled from PENS stage-2 subset, is incrementally built by appending one (d–s) pair at a time to the stage-1 click history. At step k, $\tau^{u_j}_{h}$ includes all prior (d–s) pairs up to $k-1$, and the k^{th} d-node serves as query d_q. We also insert 50–70 randomly sampled skipped d-nodes per user. For $\mathcal{T}^{\text{PS-EN}}_{\text{test}}$ and $\mathcal{T}^{\text{OAI}}_{\text{test}}$, we slice 500 and 3000 sampled trajectories, respectively, before each ($d-s$) pair. The test UIG $\mathcal{T}^{\text{M}}_{\text{test}}$ from MIND test data is created by augmenting the positive and negative targets as the tail-node of the last behavior with same history sequence (see Appendix C).

IMPerSumm Training. IMPerSumm is trained end-to-end over each user trajectory using $\mathcal{L}_{\text{dec}}$, with optimization carried out via the Adam optimizer (PyTorch v2.0.1) at a learning rate of 1×10^{-3}. We train the full model for 10 epochs (approx. 30 hours) over 56K unique behavior sequences, using a batch size of 28. The decoder operates under teacher-forced supervision with a T5 backbone. $\mathcal{L}_{\text{dec}}$ enables the model to jointly learn dynamic behavior representations and generate summaries that are personalized, coherent, and faithful to both user history and source content.

5.2 Baseline Personalized Summarizers

(1) Personalized Summarizer Models. For RQ-1, we compare three SOTA personalized summarizers: PENS [Ao et al. 2021], GTP [Song et al. 2023], and Signature-Phrase [Cai et al. 2023]. PENS couples a pointer-generator with external user encoders; we use the strongest off-the-shelf variants–NAML (T-1), EBNR, and NRMS (T-1/T-2) [Okura et al. 2017; Wu et al. 2019a,b]. GTP integrates the TrRMIo user encoder; Signature-Phrase performs sequence modeling over user-specific keyphrases. Architecturally, NAML uses multi-view additive self-attention, EBNR is a GRU over click order, NRMS implements multi-head self-attention, and TrRMIo utilizes full-sequence transformer. All baselines are fine-tuned end-to-end for 2 epochs on $\mathcal{T}^{\text{P}}_{\text{train}}$.

(2) LLMs-as-summarizers. For RQ-2, we benchmark four **frozen** LLMs—Mistral-7B-Instruct [Jiang et al. 2023], DeepSeek-R1-14B [DeepSeek-AI et al. 2025], LLaMA-2-13B-Chat-HF [Touvron et al. 2023], and Zephyr-7B [Tunstall et al. 2023]—using the best 0-shot/2-shot prompts from Patel et al. [2024] and prompt-chaining for DeepSeek-R1-14B and Mistral-7B-Instruct (details in App. E.2).

(3) Non-personalized Summarizers as Oracles. To further probe RQ-2, we evaluate BigBird-Pegasus [Zaheer et al. 2020] and SimCLS [Liu and Liu 2021] following Vansh et al. [2023]. We substitute each article's original title with the user's gold-reference title (cueing subjective preference). If exploited, these frozen models produce "personalized" summaries, effectively acting as oracles.

(4) News Recommendation Models. For RQ-3, we test IMPerSumm-encoder on next-interaction prediction and sequential recommendation. For next b-node prediction, we use PENS-based NAML/EBNR/NRMS; for recommendation, we compare against 23 sequential news recommenders, including DGAT [Mao et al. 2022], GNEWSREC [Hu et al. 2020], FastFormer [Wu et al. 2021], and MINER [Abdulhussein and Obaid 2022].

5.3 Personalization Evaluation Metrics

Automated Personalization Metric. To evaluate IMPerSumm as a personalized summarizer, we adopt PerSEval (PSE), the only metric explicitly designed for personalized summarization [Dasgupta et al. 2024]. We report three PSE variants – PSE-SU4, PSE-METEOR, and PSE-JSD, each having strong human-judgment correlation and also computational efficiency. Details of PerSEval is in Appendix A.

Accuracy Evaluation. We also report the accuracy results of IMPerSumm based on Rouge-SU4 [Lin 2004] and Rouge-L [Lin and Och 2004] metrics.

Encoder Evaluation. IMPerSumm's encoder is validated on next interaction prediction and sequential recommendation under RQ-3 using AUC, MRR, nDCG@5/10.

Human Judgment (HJ).

To validate IMPerSumm's human-alignment, we conducted a survey-based evaluation. In the survey, each participant[2] was shown a pair of gold-reference summary and a model-generated summary for a specific news article from the PENS dataset. Model-generated summaries were drawn from IMPerSumm, along with six other best-performing baselines: PENS-NAML-T1, PENS-EBNR-T2, GTP, LLMs as DeepSeek-R1-14B and Mistal-7B under 2-shot w/history settings, and Bigbird-Pegasus as oracle. Model information was hidden to remove bias. Participants rated the model-generated summary similarity w.r.t the reference on a scale from 1 (low) to 6 (high). Each summary was rated by ≈4 respondents with a ***low average inter-agreement (Krippendorf-α = 0.3), thereby indicating subjectivity***. Hence, a higher average human rating with a low coefficient-of-variation (CV) would indicate better alignment of the model.

We also interpolate human ratings of our baseline models on the multi-domain non-news OpenAI-Reddit dataset. It contains multiple human-rated summaries of 9 models (≈ 6 raters/summary). We identify the top-rated (i.e., 7) one per user as the *human-preferred reference*. We then measure the RMSD-divergence of the model-generated summaries from the reference on the SBert-embedding-space. This leads to an average min-max range per rating map used for interpolation.

6 Results and Observations

In this section, we discuss the results in light of our three research questions. **All results have statistical significance p** < 0.01.

6.1 RQ-1: IMPerSumm Performance

IMPerSumm outperforms SOTA specialized personalized summarizers that ignore action distinctions, achieving **0.48/0.47/0.49**↑ on PSE-JSD/SU4/METEOR over the strongest baseline GTP(+TrRMIo). This demonstrates the **effectiveness of explicit action-specific information modulation** via dynamic memory kernels over uniform RNN or transformer attention (see results in Table 1). When trained on the much smaller PersonalSum train UIG $\mathcal{T}_{\text{train}}^{\text{PS-EN}}$, it achieves **0.53/0.51/0.52** on PSE-JSD/SU4/METEOR on the PersonalSum test UIG $\mathcal{T}_{\text{test}}^{\text{PS-EN}}$, underscoring its effectiveness on tailored datasets. IMPerSumm also attains average boost **48.9/43.6**↑ in terms of RG-SU4 and RG-L accuracy metrics. IMPerSumm also excels in survey-based HJ evaluation (CV = 0.08) despite inherent subjectivity, indicating stronger personalization, as well as in interpolated human ratings with low avg. RMSD from the references (Table 2).

Ablation: Effect of Memory Kernels. To assess action-specific encoding, we ablate IMPerSumm's encoder with a fixed cross-attention decoder in three settings: (i) base model w/o MI, (ii) with KDE-MI (**MI Encoder**), and (iii) KDE-MI w/ adaptively fused memory kernels (**MI+S as MI w/ $\mathbf{m}_{\text{short}}$, MI+SL as MI w/ $\mathbf{m}_{\text{short}}$ and $\mathbf{m}_{\text{long}}$, MI+SLE as MI w/ $\mathbf{m}_{\text{short}}$, $\mathbf{m}_{\text{long}}$, and $\mathbf{m}_{\text{event}}$ or IMPerSumm-Full**). The base model using cosine similarity performs worst, showing simplistic embedding-level similarity fails to capture preference shifts. Combining MI with $\mathbf{m}_{\text{short}}$ and $\mathbf{m}_{\text{long}}$ (**MI+SL**) achieves **97/96/95%** of IMPerSumm-Full performance, confirming that action-aware modulation and dynamic memory kernels are critical.

Ablation: Effect of Contextualized Injection. In order to understand the effect of contextualized injection, we use the best-performing MI+SLE encoder and compare *b*-node injection (**b-Decoder**), reconstructed *s*-node injection w/o document-context (**s-Decoder**), and w/ document-context (**IMPerSumm-Full**). We find that IMPerSumm-full outperforms b-Decoder by an avg. of **0.27**↑ and s-Decoder by an avg. of **0.29**↑ w.r.t. PSE variants; see Table 1.

[2]Demography: 139 male, 32 female graduate students from Computer Science, Humanities, Mathematics, & Natural Sciences.

Table 1: RQ-1: Personalized Summarization on PENS (20k test docs). Baselines show reported means; IMPerSumm variants show *mean ± 95% CI* from three random resampling trials (25%, 50%, 75% subsets). *IMPerSumm (Full) achieves 47–55% relative gain over the best baseline.*

Category	Model / Variant	PSE-JSD	PSE-SU4	PSE-METEOR
Personalized Summarizers	PENS-NAML-T1	0.021	0.014	0.016
	PENS-EBNR-T1	0.015	0.010	0.011
	PENS-EBNR-T2	0.011	0.008	0.009
	PENS-NRMS-T1	0.015	0.011	0.011
	PENS-NRMS-T2	0.008	0.007	0.007
	GTP	0.024	0.017	0.019
	SP-Individual	0.017	0.015	0.014
IMPerSumm (Encoder Ablations)	Base Encoder + CA-Decoder	0.241±0.02	0.013±0.01	0.107±0.02
	MI-Encoder + CA-Decoder	0.311±0.02	0.054±0.01	0.274±0.02
	MI+S-Encoder + CA-Decoder	0.357±0.02	0.056±0.01	0.334±0.02
	MI+L-Encoder + CA-Decoder	0.387±0.02	0.265±0.02	0.402±0.02
	MI+SL-Encoder + CA-Decoder	0.502±0.02	0.483±0.03	0.491±0.03
	MI+LE-Encoder + CA-Decoder	0.485±0.02	0.436±0.02	0.463±0.03
	MI+SE-Encoder + CA-Decoder	0.462±0.02	0.447±0.02	0.458±0.02
IMPerSumm (Decoder Ablations)	MI+SLE-Encoder + s-Decoder	0.325±0.02	0.096±0.01	0.202±0.01
	MI+SLE-Encoder + b-Decoder	0.331±0.02	0.106±0.01	0.237±0.02
	MI+SLE-Encoder + CA-Decoder (**Full**)	**0.518**±0.02	**0.501**±0.03	**0.514**±0.03

Table 2: Qualitative Comparison w.r.t ROUGE, Survey-based Human Eval (PENS Dataset; Krippendorf-α = 0.3; CV: Coefficient-of-Variation), & Human Ratings (OpenAI-Reddit).

Model Name	RG-SU4	RG-L	Direct Human Eval	RMSD	Interpolated Human Rating
IMPerSumm-Full	**63.61**	**66.51**	**3.15** (CV=0.08)	**0.34**	**7**
Mistral-7B	16.42	22.85	*3.05* (CV = 0.07)	0,79	5
BigbirdPegasus	17.13	21.65	3.02 (CV = 0.08)	0.81	5
DeepSeek-14B	19.57	*29.72*	3.0 (CV = 0.08)	0.78	5
PENS-EBNR-T2	12.41	20.82	2.97 (CV = 0.11)	0.93	2
PENS-NAML-T1	13.12	21.62	2.94 (CV = 0.09)	0.92	2
GTP	*21.91*	28.31	2.86 (CV = 0.09)	0.94	2

6.2 RQ-2: Effectiveness of Preference Injection as Prompts & Cues

We study whether personalization can be achieved without a dedicated behavioral encoder, either through history-aware LLM prompting or as cue-injection into vanilla baseline summarizers (thereby testing their limit as oracles).

RQ-2(a): History-aware LLM Prompting. While injecting user histories as prompts in LLMs shows improvements in personalization over traditional models [Patel et al. 2024], we assess its ability to replace a dedicated behavioral encoder by prompting LLMs with user histories (Section 5.2) via 0-shot, 2-shot+history, and prompt-chaining. IMPerSumm outperforms these approaches by **0.33/0.42/0.40**↑ on 0/2-shot and **0.44/0.47/0.49**↑ across PSE on prompt chaining across four LLMs, indicating that mid-sized LLM prompting techniques are not as effective.

Ablation: Cross-domain Generalizability. We train IMPerSumm on OpenAI-Reddit $\mathcal{T}_{\text{train}}^{\text{OAI}}$ and evaluate on $\mathcal{T}_{\text{test}}^{\text{OAI}}$ (Section 5.1) for

Table 3: Performance comparison on the PENS dataset across LLM variants and IMPerSumm under RQ-2.

Category	Model	PSE-JSD	PSE-SU4	PSE-METEOR
LLMs (0-shot history)	LLaMA-13B	0.187	0.069	0.078
	Mistral-7B	0.212	0.082	0.098
	DeepSeek-14B	0.152	0.078	0.084
	Zephyr-7B	0.211	0.081	0.089
LLMs (2-shot history)	LLaMA-13B	0.227	0.078	0.081
	Mistral-7B	0.235	0.087	0.084
	DeepSeek-14B	0.248	0.094	0.097
	Zephyr-7B	0.231	0.085	0.086
LLMs (Prompt-chaining)	Mistral-7B	0.072	0.026	0.023
	DeepSeek-14B	0.078	0.028	0.024
Vanilla (Cue Injection)	BigbirdPegasus	0.253	0.143	0.168
	SimCLS	0.157	0.032	0.116
IMPerSumm-Full	MI+SLE-Encoder+CA-Decoder	**0.518**	**0.501**	**0.514**

Table 4: RQ2: Performance comparison on the OpenAI (Reddit) dataset across LLMs w/ 2-shot user history and IMPerSumm.

Category	Model	PSE-JSD	PSE-SU4	PSE-METEOR
LLMs (2-shot history)	LLaMA-13B	0.232	0.093	0.107
	Mistral-7B	0.226	0.088	0.103
	DeepSeek-14B	0.243	0.095	0.109
	Zephyr-7B	0.214	0.087	0.104
IMPerSumm-Full	MI+SLE-Encoder+CA-Decoder	**0.576**	**0.454**	**0.494**

Table 5: RQ-3 (a): Next b-node Prediction (PENS Dataset). † NT: Originally published results were on the *news recommendation task* in contrast to next behavior prediction. For IMPerSumm, values show *mean* ± *95% CI*; Setup as in Table 1.

Category	Model	AUC	MRR	nDCG@5	nDCG@10
User Encoders	NAML†	0.498	0.001	0.0004	0.0007
(*End-to-end Trained on Summarization*)	NRMS†	*0.499*	0.0009	0.0002	0.0004
	EBNR†	*0.499*	0.0009	0.0003	0.0005
IMPerSumm	MI+SL-Encoder	0.431±0.035	*0.106*±0.015	*0.073*±0.015	*0.117*±0.020
	MI+SLE-Encoder	**0.513**±0.042	**0.216**±0.030	**0.181**±0.005	**0.236**±0.017

cross-domain generalizability[3]. IMPerSumm outperforms the best LLM baseline, DeepSeek-R1 by **0.33/0.36/0.38**↑, showing cross-domain robustness. Results are in Table 4.

RQ-2(b): Oracle Summarizers with Gold References. We evaluate oracle versions of BigBird-Pegasus and SimCLS to analyze whether they can serve as strong baselines. Despite having access to gold cues, IMPerSumm outperforms BigBird-Pegasus by **0.29**↑ and SimCLS by **0.44**↑ across PSE variants JSD/SU4/METEOR. This underscores the effectiveness of IMPerSumm's memory-gated, action-specific encoding (detailed results in Table 3).

6.3 RQ-3: IMPerSumm-Encoder Validation

We validate IMPerSumm's encoder along two axes–***Next b-node Prediction*** and ***Sequential Recommendation*** under the hypothesis that better next-interaction prediction and sequential ranking by user-encoder are expected so as to align with the observed higher personalized summarization performance.

RQ-3(a): Next b-Node Prediction. We see that user-encoder baselines (NAML/NRMS/EBNR), end-to-end trained on the PENS dataset on the summarization task along with the PENS-framework decoder, perform near-random on this task (MRR/nDCG ≈ 0). Using an *action-specific* MI encoder with multi-horizon memory, MI+SL (short+long kernels) improves performance (0.431 AUC, 0.106 MRR, 0.073/0.117 nDCG@5/10), while MI+SLE (adds event kernel) achieves the best results (0.513 AUC, 0.216 MRR, 0.181/0.236 nDCG@5/10), showing that action-conditioned MI combined with short, long, and event memory better captures evolving user (see Table 5).

[3]OAI covers 29 wide-ranging non-news topics with greater diversity (App. C).

Table 6: RQ-3(b): Sequential Recommendation on MIND-Large. (sorted by MRR) Baselines show paper-reported means. IMPerSumm results denote *mean* ± *95% CI*; Setup as in Table 1.

Methods (Venue, Year)	AUC	MRR	nDCG@5	nDCG@10
DKN (WWW'18)	64.07	30.42	32.92	38.66
GRU (Baseline, 2016)	65.42	31.24	33.76	39.47
EBNR (KDD'17)	65.46	31.26	32.18	39.04
NPA (KDD'19)	65.92	32.07	34.72	40.37
NAML (IJCAI'19)	66.46	32.75	35.66	41.40
LSTUR (ACL'19)	67.08	32.86	35.95	40.94
Linear Transformers (ICML'20)	67.76	32.94	35.91	41.97
ProFairRec (SIGIR'22)	67.64	33.08	35.32	41.67
NRCLS (Appl. Sci.'24)	68.35	33.12	36.70	43.03
Linformer (arXiv'20)	68.02	33.19	36.22	42.10
Poolingformer (ICML'21)	68.54	33.20	36.69	42.60
NRMS (EMNLP-IJCNLP'19)	67.66	33.25	36.28	41.98
BigBird (NeurIPS'20)	68.14	33.28	36.42	42.18
Transformer (NeurIPS'17)	68.22	33.32	36.35	42.23
GERL (WWW'20)	68.10	33.41	36.34	42.03
GNewsRec (IP&M'20)	68.15	33.45	36.43	42.10
FIM (ACL'20)	67.87	33.46	36.53	42.21
HieRec (ACL-IJCNLP'21)	68.33	33.86	36.83	42.65
DCAN (arXiv'22)	68.90	33.90	36.90	42.80
ANRS (arXiv'22)	69.20	34.10	37.10	43.00
TCCM (CIKM'23)	69.75	34.42	37.53	43.25
Fastformer (arXiv'21)	69.11	34.55	37.62	43.38
FUM (SIGIR'22)	69.90	34.60	37.70	43.40
CAUM (SIGIR'22)	70.04	34.71	37.89	43.57
DIGAT (Findings ACL'22)	70.08	35.20	38.46	44.15
PLM-NR (SIGIR'21)	70.64	35.39	38.71	44.38
Fastformer+PLM-NR (Hybrid)	71.04	35.91	39.16	45.03
MINER (Findings ACL'22)	71.51	36.06	39.56	45.21
CAST-Rec (TOIS'25)	*72.10*	36.90	40.20	46.30
Fastformer+PLM-NR-Ensemble (Hybrid'22)	**72.68**	*37.45*	*41.51*	46.84
MI+SL-Encoder (IMPerSumm)	48.13±0.7	*37.48*±0.4	40.81±0.5	*50.86*±0.6
MI+SLE-Encoder (IMPerSumm)	55.62±0.6	**40.03**±0.6	**45.07**±0.7	**53.42**±0.7

RQ-3(b): Sequential Recommendation. We use the standard and widely adopted MIND news recommendation test benchmark. Since the PENS train set is derived from the MIND train set, this test is **cross-task** rather than cross-domain. We observe that the PENS-trained $\mathcal{T}^{\mathrm{M}}_{\mathrm{test}}$, MI+SLE-Encoder improves head-of-list ranking over the best baseline (Fastformer+PLM-NR-Ensemble): MRR 2.58 ↑, nDCG@5 3.56 ↑, nDCG@10 6.58 ↑ (see Table 6). Lower AUC (≈ 16 ↓) indicates that although the IMPerSumm-encoder develops cross-task capability of news recommendation when exclusively trained on personalized summarization, it requires explicit training on the recommendation task for a corresponding *classification* ability.

7 Conclusion

In this paper, we propose IMPerSumm, a personalized text summarizer that models user behavior as dynamic information modulation per interaction. It encodes sequences via feature-wise mutual information and aggregates memories with learnable temporal kernels to capture evolving preferences. Unlike static profiles or prompt-based methods, it offers interpretable adaptation with gains in relevance, semantic alignment, and preference sensitivity. We see 55% gain in PSE w.r.t SOTA baselines. As future work, we are exploring adaptive kernels, advanced encoding, decoder-side personalization, and alignment reward-based reinforcement objectives.

References

Niran Abdulhussein and Ahmed Obaid. 2022. User recommendation system based on MIND dataset. *International Journal of Nonlinear Analysis and Applications* Online First (Sept. 2022). doi:10.22075/ijnaa.2022.6857

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. 2021. PENS: A Dataset and Generic Framework for Personalized News Headline Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 82–92. doi:10.18653/v1/2021.acl-long.7

Pengshan Cai, Kaiqiang Song, Sangwoo Cho, Hongwei Wang, Xiaoyang Wang, Hong Yu, Fei Liu, and Dong Yu. 2023. Generating User-Engaging News Headlines. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 3265–3280. doi:10.18653/v1/2023.acl-long.183

Junsu Cho and Hyun. 2023. Dynamic multi-behavior sequence modeling for next item recommendation. In *Proceedings of the AAAI conference on artificial intelligence*.

Maxime Darrin, Philippe Formont, Jackie Cheung, and Pablo Piantanida. 2024. COSMIC: Mutual Information for Task-Agnostic Summarization Evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand. doi:10.18653/v1/2024.acl-long.686

Sourish Dasgupta, Ankush Chander, Tanmoy Chakraborty, Parth Borad, and Isha Motiyani. 2024. PerSEval: Assessing Personalization in Text Summarizers. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=yqT7eBz1VJ

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, and Chengqi et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A General Framework for Guided Neural Abstractive Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4830–4842. doi:10.18653/v1/2021.naacl-main.384

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 1074–1084. doi:10.18653/v1/P19-1102

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond English-Centric Multilingual Machine Translation. arXiv:2010.11125 [cs.CL] https://arxiv.org/abs/2010.11125

Muhan Gao, TaiMing Lu, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into LLM Long-Context Failures: When Transformers Know but Don't Tell. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 7611–7625. doi:10.18653/v1/2024.findings-emnlp.447

Saeid Ghodratnama and Masoud Zakershahrak. 2024. SumRecom: Recommending Summaries via Interactive User Feedback. *Information Processing & Management* 61, 4 (2024), 103568.

Albert Gu and Tri Dao. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs.LG] https://arxiv.org/abs/2312.00752

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *The International Conference on Learning Representations (ICLR)*.

Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2022. CTRLsum: Towards Generic Controllable Text Summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 5879–5915. doi:10.18653/v1/2022.emnlp-main.396

Zifan He, Yingqi Cao, Zongyue Qin, Neha Prakriya, Yizhou Sun, and Jason Cong. 2025. HMT: Hierarchical Memory Transformer for Efficient Long Context Language Processing. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, Albuquerque, New Mexico, 8068–8089. doi:10.18653/v1/2025.naacl-long.410

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Montreal, Canada) *(NIPS'15)*. MIT Press, Cambridge, MA, USA, 1693–1701.

Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Inf. Process. Manage.* 57, 2 (March 2020), 10 pages. doi:10.1016/j.ipm.2019.102142

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

Frederic Kirstein, Terry Ruas, Robert Kratel, and Bela Gipp. 2024. Tell me what I need to know: Exploring LLM-based (Personalized) Abstractive Multi-Source Meeting Summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (Eds.). Association for Computational Linguistics, Miami, Florida, US, 920–939. doi:10.18653/v1/2024.emnlp-industry.69

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023. Guiding large language models via directional stimulus prompting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 2735, 27 pages.

Junhong Lian, Xiang Ao, Xinyu Liu, Yang Liu, and Qing He. 2025. Panoramic Interests: Stylistic-Content Aware Personalized Headline Generation. In *Companion Proceedings of the ACM on Web Conference 2025*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. 605–612.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173. doi:10.1162/tacl_a_00638

Yibin Liu, Ang Li, and Shijian Li. 2026. Hierarchical Interaction Summarization and Contrastive Prompting for Explainable Recommendations. In *Machine Learning and Knowledge Discovery in Databases. Research Track*. Springer Nature Switzerland, Cham, 485–501.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, 1065–1072. doi:10.18653/v1/2021.acl-short.135

Zhiming Mao, Jian Li, Hongru Wang, Xingshan Zeng, and Kam-Fai Wong. 2022. DIGAT: Modeling News Recommendation with Dual-Graph Interaction. arXiv:2210.05196 [cs.CL] https://arxiv.org/abs/2210.05196

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-Based News Recommendation for Millions of Users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 1933–1942. doi:10.1145/3097983.3098108

Zhonghong Ou, Xiao Zhang, and Zhu. 2025. LS-TGNN: Long and Short-Term Temporal Graph Neural Network for Session-Based Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhe Pan and Peng Wang. 2021. Hyperbolic Hierarchy-Aware Knowledge Graph Embedding for Link Prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2941–2948. doi:10.18653/v1/2021.findings-emnlp.251

Divya Patel, Pathik Patel, Ankush Chander, Sourish Dasgupta, and Tanmoy Chakraborty. 2024. Are Large Language Models In-Context Personalized Summarizers? Get an iCOPERNICUS Test Done!. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 16820–16842. doi:10.18653/v1/2024.emnlp-main.935

Zexuan Qiu, Jingjing Li, Shijue Huang, Xiaoqi Jiao, Wanjun Zhong, and Irwin King. 2024. CLongEval: A Chinese Benchmark for Evaluating Long-Context Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 3985–4004. doi:10.18653/v1/2024.findings-emnlp.230

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

B. W. Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall.

Yun-Zhu Song, Yi-Syuan Chen, Lu Wang, and Hong-Han Shuai. 2023. General then Personal: Decoupling and Pre-training for Personalized Headline Generation. *Transactions of the Association for Computational Linguistics* 11 (2023), 1588–1607. doi:10.1162/tacl_a_00621

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv e-prints* (2023), arXiv–2307.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct Distillation of LM Alignment. *arXiv e-prints* (2023), arXiv–2310.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748 [cs.LG] https://arxiv.org/abs/1807.03748

Rahul Vansh, Darsh Rank, Sourish Dasgupta, and Tanmoy Chakraborty. 2023. Accuracy is not enough: Evaluating Personalization in Summarizers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2582–2595. doi:10.18653/v1/2023.findings-emnlp.169

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, Copenhagen, Denmark, 59–63. doi:10.18653/v1/W17-4508

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. arXiv:2212.03533 [cs.CL] https://arxiv.org/abs/2212.03533

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural News Recommendation with Attentive Multi-View Learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019b. Neural News Recommendation with Multi-Head Self-Attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6389–6394. doi:10.18653/v1/D19-1671

Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2021. Fastformer: Additive Attention Can Be All You Need. arXiv:2108.09084 [cs.CL] https://arxiv.org/abs/2108.09084

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 3597–3606. doi:10.18653/v1/2020.acl-main.331

Wen Xiao, Yujia Xie, Giuseppe Carenini, and Pengcheng He. 2024. Personalized Abstractive Summarization by Tri-agent Generation Pipeline. In *Findings of the Association for Computational Linguistics: EACL 2024*. Association for Computational Linguistics, St. Julian's, Malta, 570–581. https://aclanthology.org/2024.findings-eacl.39/

Zhao Yang, Junhong Lian, and Xiang Ao. 2023. Fact-Preserved Personalized News Headline Generation. *2023 IEEE International Conference on Data Mining (ICDM)* (2023).

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems*, Vol. 33. 17283–17297. https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf

Lemei Zhang, Peng Liu, Marcus Tiedemann Oekland Henriksboe, Even W. Lauvrak, Jon Atle Gulla, and Heri Ramampiaro. 2024. PersonalSum: A User-Subjective Guided Personalized Summarization Dataset for Large Language Models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://openreview.net/forum?id=ETZk7lqyaF

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2021. An Unsupervised Sentence Embedding Method by Mutual Information Maximization. arXiv:2009.12061 [cs.CL] https://arxiv.org/abs/2009.12061

Yongfeng Zhang, Jin Sheng, Yueqing Xie, Xinyan Wang, and Wayne Xin Wu. 2022. P5: Pretrain, Personalized Prompt, and Predict Paradigm for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 1859–1868.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3065–3072.

Vitalii Zhelezniak, Anna Rogers, Haiming Hassan, Diana McCarthy, and Matthew Shardlow. 2020. Estimating Mutual Information Between Dense Word Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6482–6495.

Xiaoyu Zhu, Chenyang Bu, Bingbing Dong, Shengwei Ji, Yi He, and Xindong Wu. 2023. IKGN: Intention-aware Knowledge Graph Network for POI Recommendation. In *2023 IEEE International Conference on Data Mining (ICDM)*. 908–917. doi:10.1109/ICDM58522.2023.00100

Table 7: Symbol and Notation Table for `IMPerSumm`

Symbol	Description
u_j	The j-th user
$d^{(t_p)}$	Document (d-node) interacted with at time-step t_p
$s_j^{(t_q)}$	Summary (s-node) generated/read by user u_j at time t_q
$a^{(t_p)}$	User action at time-step t (e.g., click, skip, summarize)
τ^{u_j}	Trajectory of user u_j: sequence of interactions over time
$\tau_b^{u_j}$	Behavior trajectory of user u_j over the b-tier graph
$G = \langle N, E \rangle$	User-Interaction Graph (UIG) with node set N and edge set E
$b_{u_j}^{(t_i)}$	Behavior triple at time t_i: $\langle \text{head}, a^{(t)}, \text{tail} \rangle$
$e_{hd}^{(t_i)}$	Embedding of the head node at time-step t_i
$e_{tl}^{(t_i)}$	Embedding of the tail node at time-step t_i
$I^{(t_i)}$	Mutual information at time-step t_i
$\mathbf{c}_{hd}^{(t_i)}$	Content Embedding of the head node at time-step t_i
$\mathbf{c}_{tl}^{(t_i)}$	Content Embedding of the tail node at time-step t_i
$f_{\cos}$	Cosine-based alignment function
$\mathbf{g}^{(clk,t_i)}$	Learnable Click gate
$\mathbf{g}^{(skp,t_i)}$	Learnable forget gate for skip interactions
$\mathbf{g}^{(fc,t_i)}$	Learnable focus gate for summarize (genSumm) actions
$\mathbf{g}^{(summ,t_i)}$	Gating parameter for incorporating influential history in summGen
$\mathbf{g}^{(acc,t_i)}$	Gating parameter for document relevance in summGen
$m_{\text{short}}^{(t)}$	Short-term memory at time-step t
$m_{\text{long}}^{(t)}$	Long-term memory at time-step t
$m_{\text{event}}^{(t)}$	Event-specific memory at time-step t
$\mathcal{K}_{\text{short}}, \mathcal{K}_{\text{long}}, \mathcal{K}_{\text{event}}$	Learnable kernels for temporal memory aggregation
$\mathbf{g}_{\text{fuse}}^{(t_i)}$	Time-variant gating vector for fusing memories
$\mathbf{z}_b^{(t)}$	Memory-infused Behavior embedding t
$\hat{\mathbf{e}}_{\text{summ}}^{(t_i)}$	Latent summary vector w/o cross-attention
$\mathbf{e}_{\text{p-summ}}^{(t_i)}$	Document contextualized Latent summary vector from cross-attention
d_q	Query document
s_q^*	Ground-truth personalized summary for document d_q

A `PerSEval`: Measuring Degree of Personalization

Dasgupta et al. [2024] proposed `PerSEval` as arguably the only metric for direct evaluation of degree-of-personalization. It is an extension to `DEGRESS` that measures the proportionate degree of divergence (**responsiveness**) of model-generated summaries w.r.t the subjective difference in the expected summaries of the users [Vansh et al. 2023]. `PerSEval` penalizes the lack of personalization while also accounting for accuracy. Given a document d_i and user j, summary-level `PerSEval` is:

$$\texttt{PerSEval}(s_{u_{ij}}|(d_i, u_{ij})) = \texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij})) \times \texttt{EDP}(s_{u_{ij}}|(d_i, u_{ij})) \tag{5}$$

where `DEGRESS` measures responsiveness and EDP is the Effective `DEGRESS` penalty that incorporates both accuracy-drop and accuracy-inconsistency on degree-of-responsiveness. The system-level score averages over all (i, j).

PSE-SU4 uses ROUGE-SU4 skip-bigram F1 as σ, and is shown to correlate strongly with human judgment. On the other hand, **PSE-JSD** uses Jensen–Shannon Divergence between n-gram distributions of generated vs. reference summaries, while **PSE-Meteor** uses $1 - \text{METEOR}(G, R)$, where METEOR combines unigram matches with synonym/paraphrase handling.

B Information Modulation via KDE

We estimate how much information a current user action (e.g., a click or a generated summary) retains from its immediate past by computing the **mutual information (MI)** between their embeddings via **Kernel Density Estimation (KDE)**. Each behavior step $b_{u_j}^{(t_i)}$ consists of a pair of d-dimensional embeddings $(\mathbf{c}_{hd}^{(t_{i-1})}, \mathbf{e}_{tl}^{(t_i)})$, where $\mathbf{c}_{hd}^{(t_{i-1})}$ represents the head-cell content (historical context) and $\mathbf{e}_{tl}^{(t_i)}$ the tail-node embedding (current action). High MI indicates *continuity*—the current action is predictable from the user's prior behavior—while low MI suggests *novelty or drift*. The modulation process thus quantifies how information flows from the past to the present along a user's trajectory τ^{u_j}.

KDE-based density estimation. Following the KDE formulation, the marginal densities of the head and tail embeddings are given by: $\hat{p}_{hd}(\mathbf{c}_{hd}) = \frac{1}{(i-1)\lambda^d} \sum_{j=1}^{i-1} K\left(\frac{\mathbf{c}_{hd} - \mathbf{c}_{hd}^{(t_j)}}{\lambda}\right)$ & $\hat{p}_{tl}(\mathbf{e}_{tl}) = \frac{1}{(i-1)\lambda^d} \sum_{j=1}^{i-1} K\left(\frac{\mathbf{e}_{tl} - \mathbf{e}_{tl}^{(t_j)}}{\lambda}\right)$. and the joint density between the current tail-node and accumulated history is:

$$\hat{p}_{hd,tl}(\mathbf{c}_{hd}, \mathbf{e}_{tl}) = \frac{1}{(i-1)\lambda^{2d}} \sum_{j=1}^{i-1} K\left(\frac{\mathbf{c}_{hd} - \mathbf{c}_{hd}^{(t_j)}}{\lambda}\right) K\left(\frac{\mathbf{e}_{tl} - \mathbf{e}_{tl}^{(t_j)}}{\lambda}\right). \tag{6}$$

Here $K(\cdot)$ is a Gaussian kernel defined as: $K(\mathbf{u}) = \exp\left(-\frac{1}{2}\|\mathbf{u}\|^2\right)$. which assigns higher weights to semantically proximate embeddings.The bandwidth λ controls the smoothness of the estimation: a smaller λ emphasizes fine-grained local continuity, whereas a larger λ smooths over broader historical variation. In embedding space, K thus captures semantic affinity, and λ determines the scale of historical context considered.

Kernelized MI estimate. Using these densities, we estimate the mutual information between the head-cell and the tail-node embeddings as:

$$I(\mathbf{c}_{hd}; \mathbf{e}_{tl}) = \frac{1}{(i-1)} \sum_{j=1}^{i-1} \log \frac{\hat{p}_{hd,tl}(\mathbf{c}_{hd}^{(t_j)}, \mathbf{e}_{tl}^{(t_j)})}{\hat{p}_{hd}(\mathbf{c}_{hd}^{(t_j)})\, \hat{p}_{tl}(\mathbf{e}_{tl}^{(t_j)})}. \tag{7}$$

This MI score captures the degree of *information modulation*—how much of the present action's embedding is predictable from prior user context. We fix $\lambda = 0.6$, obtained via Silverman's rule-of-thumb [Silverman 1986] and validated on the development set, to maintain stable yet locality-aware estimation.

C Datasets

PENS Dataset The PENS dataset [Ao et al. 2021] contains 113,762 news articles across 15 topics, with titles (avg. 10.5 words), bodies (avg. 549 words), and WikiData-linked entities. It also includes user interaction logs derived from MIND [Wu et al. 2020]. The **training set** comprises 500k impressions sampled (June 13–July 3, 2019), recording user IDs, clicked/unclicked news, and click histories. These logs highlight strong preference shifts, underscoring temporal personalization. The **test set** has 103 students who reviewed 1,000 headlines, selected 50, and created personalized headlines for 200 unseen articles. Four annotators per article ensured coverage, and editors filtered inaccuracies. The remaining gold headlines serve as personalized references.

OpenAI (Reddit) Dataset The OpenAI Reddit dataset [Völske et al. 2017] includes 123,169 posts from 29 subreddits. It offers OpenAI- and human-written summaries, split into Comparisons (training/validation) and Axis (validation/testing). A curated set of 1,038 posts produced 7,713 summaries, later rated by 64 annotators on accuracy, coherence, coverage, and quality.

PersonalSum Dataset PersonalSum [Zhang et al. 2024] provides 441 Norwegian news articles and 1,099 personalized summaries from 39 annotators. Each article has a GPT-4-generated generic summary, plus 3 personalized summaries reflecting annotator preferences. Annotators highlight source text, and quality checks (GPT-3.5 scoring) ensure quality user-grounded outputs.

UIG Construction

We construct User-Interaction Graphs (UIGs) from two dataset types: (i) trajectory-based (e.g., PENS [Ao et al. 2021]), where clicks/skips form interaction sequences, and (ii) feedback-based (e.g., OpenAI-Reddit [Völske et al. 2017]), where posts and summaries are linked by user ratings. For PENS, we enrich click/skip logs with summary nodes from the test set, yielding $\mathcal{T}^{\text{P-D}}$. For Reddit, we treat posts as clicked if any summary scored ≥6/9, selecting the top-rated summary as its surrogate node.

MIND dataset UIG For sequential news recommendation experiments, we construct the test UIG $\mathcal{T}_{\text{test}}^{\text{M}}$ using the standard MIND-large test data [Wu et al. 2020]. Given a user trajectory in MIND, we first extract the last observed behavior sequence, which serves as the user's test history. To this history, we augment both the positive and negative targets provided in the MIND test set where **positive targets** (appended as *tail nodes* to the user's history sequence) are the news articles that the user actually clicked during the test phase, while **negative targets** (appended as *tail nodes*) are the candidate news articles that the user was exposed to but did not click (sampled negatives in MIND). This augmentation produces a unified test UIG where each user's final behavior history points to multiple candidate tail nodes—some positive (clicks) and others negative (non-clicks). Such a structure is essential for evaluating ranking-based sequential recommendation, since the model must differentiate relevant (positive) from irrelevant (negative) targets given the same history context.

D Baselines

Baseline LLMs:

1. Zephyr 7B-β [Tunstall et al. 2023] is a 7B transformer fine-tuned from Mistral-7B using DPO. It relaxes alignment constraints to boost raw performance, achieving strong MT-Bench scores, and is openly released under MIT license.

2. Mistral-Instruct 7B [Jiang et al. 2023] is a dense transformer with GQA and SWA, trained on 2T tokens. It outperforms larger models like LLaMA2-13B in many benchmarks.

3. LLaMA-2-13B [Touvron et al. 2023] is a 13B transformer trained on 2T tokens with RLHF-based chat tuning. It remains widely used though surpassed by smaller models such as Mistral 7B.

Table 8: Complete list of hyperparameters and their shapes. Includes classification weights and s/d cross-attention.

Name	Symbol	Dimension
Head-cell weight matrix	W_h	$\mathbb{R}^{192\times192}$
Head-node embedding weight	W_{hd}	$\mathbb{R}^{192\times192}$
Tail-cell weight matrix	W_a	$\mathbb{R}^{192\times192}$
b-node output weight	W_b	$\mathbb{R}^{192\times192}$
b-node output bias	$\mathbf{b}_b$	$\mathbb{R}^{192}$
Cell content (head/tail)	$\mathbf{c}_{hd}^{(t_i)}, \mathbf{c}_{tl}^{(t_i)}$	$\mathbb{R}^{192}$
Cell content (head/tail)	$\mathbf{e}_{b_{u_j}}^{(t_i)}$	$\mathbb{R}^{192}$
Cosine modulation influence factor	λ_c	Scalar
Cosine modulation influence factor	$\mathbf{c}_{hd}^{(a,t_i)}$	$\mathbb{R}^{192}$
Mutual information function	$\mathbf{I}_\mathbf{K}(\cdot;\cdot)$	$\mathbb{R}^{192}$
Click gate	$\mathbf{g}^{(\mathbf{clk},\mathbf{t_i})}$	$\mathbb{R}^{192}$
Click gate weight	$\mathbf{W}_\mathbf{g}^{\mathbf{clk}}$	$\mathbb{R}^{192\times192}$
Skip gate	$\mathbf{g}^{(\mathbf{skp},\mathbf{t_i})}$	$\mathbb{R}^{192}$
Skip gate weight	$\mathbf{W}_\mathbf{g}^{\mathbf{skp}}$	$\mathbb{R}^{192\times192}$
Focus gate	$\mathbf{g}^{(\mathbf{fc},\mathbf{t_i})}$	$\mathbb{R}^{192}$
Focus gate weight	$\mathbf{W}_\mathbf{g}^{\mathbf{fc}}$	$\mathbb{R}^{192\times192}$
Interest gate (summary)	$\mathbf{g}^{(\mathbf{summ},\mathbf{t_i})}$	$\mathbb{R}^{192}$
Interest gate weight	$\mathbf{W}_\mathbf{g}^{\mathbf{summ}}$	$\mathbb{R}^{192\times192}$
Accuracy gate (summary)	$\mathbf{g}^{(\mathbf{acc},\mathbf{t_i})}$	$\mathbb{R}^{192}$
Accuracy gate weight	$\mathbf{W}_\mathbf{g}^{\mathbf{acc}}$	$\mathbb{R}^{192\times192}$
Short-term memory kernel weight	W_{short}	$\mathbb{R}^{192}$
Long-term memory kernel weight	W_{long}	$\mathbb{R}^{192}$
Event-specific memory projection	W_{event}	$\mathbb{R}^{192\times192}$
Short-term memory vector	$\mathbf{m}_{\text{short}}^{(t_i)}$	$\mathbb{R}^{192}$
Long-term memory vector	$\mathbf{m}_{\text{long}}^{(t_i)}$	$\mathbb{R}^{192}$
Event-specific memory vector	$\mathbf{m}_{\text{event}}^{(t_i)}$	$\mathbb{R}^{192}$
Short-term kernel value	$\mathcal{K}_{\text{short}}^{t_j}$	$\mathbb{R}^{192}$
Long-term kernel value	$\mathcal{K}_{\text{long}}^{j}$	$\mathbb{R}^{192}$
Event-specific kernel value	$\mathcal{K}_{\text{event}}^{t_j}$	$\mathbb{R}$ (scalar per j)
Memory fusion gate weight	W_{fuse}	$\mathbb{R}^{192\times3}$
Memory fusion gate value	$g_{\text{fuse}}^{(t_i)}$	$\mathbb{R}^3$
Fused memory vector (before gating)	$\mathbf{m}_{\text{fuse}}^{(t_i)}$	$\mathbb{R}^3$
Gated memory output	$\mathbf{m}_{u_j}^{(t_i)}$	$\mathbb{R}^3$
Residual vector	$\mathbf{c}_{tl}^{(t_i)}$	$\mathbb{R}^{192}$
Final memory-infused b-node embedding	$\mathbf{z}_b^{(t_i)}$	$\mathbb{R}^{192}$
Cross-attention query projection	W_Q	$\mathbb{R}^{192\times192}$
Cross-attention key projection	W_K	$\mathbb{R}^{192\times192}$
Cross-attention value projection	W_V	$\mathbb{R}^{192\times192}$
Cross-attention output projection	W_O	$\mathbb{R}^{192\times192}$
History b-node classifier	$W_{\text{cls-hist}}$	$\mathbb{R}^{76{,}000\times192}$
Next b-node classifier	$W_{\text{cls-next}}$	$\mathbb{R}^{76{,}000\times192}$
s-Node Approximator	W_{summ}^{+}	$\mathbb{R}^{192\times192}$

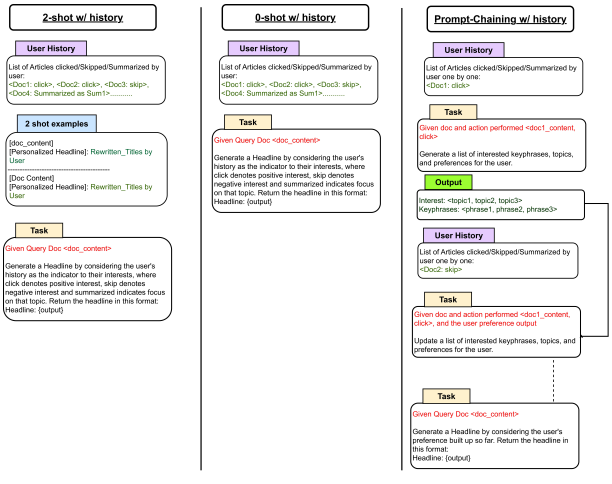

Figure 5: Prompt-Templates for LLM baselines.

4. DeepSeek-R1 14B [DeepSeek-AI et al. 2025] is a 14.8B distilled model optimized for math, code, and reasoning. Despite its size, it matches larger models on AIME/MATH benchmarks and is MIT licensed.

Baseline Generic (Oracle) Summarizers:

1. BigBirdPegasus [Zaheer et al. 2020] extends Pegasus with sparse, global, and random attention, enabling efficient long-sequence summarization.

2. SimCLS [Liu and Liu 2021] trains a Seq2Seq generator with MLE and a RoBERTa-based contrastive ranker to refine candidate summaries.

Baseline Personalized Models

1. PENS Framework uses: (i) NRMS [Wu et al. 2019b] that encodes titles and user clicks with self-attention. In Type 1, user embeddings initialize the decoder; in Type 2, they modulate attention weights; (ii) NAML [Wu et al. 2019a] that encodes multiple news views (titles, bodies, topics). Its user embedding personalizes decoding via Injection-Type 1; (iii) EBNR [Okura et al. 2017] that uses GRU-based browsing histories for embeddings. Type 1 initializes the decoder, while Type 2 personalizes via attention layers.

2. General Then Personal (GTP) GTP [Song et al. 2023] first trains a general headline generator, then applies a user-customizer with control codes. ISB and MUM modules ensure fidelity and robust personalization.

3. Signature Phrase Signature Phrases [Cai et al. 2023] distill histories into dynamic phrase-level profiles via contrastive learning. These high-level signals guide personalized headline generation.

E Implementation Details

E.1 Compute Resources

All preprocessing (graph construction, embedding lookup, probability mapping) was done on CPUs with 16GB/core. Embedding tables for bodies, headlines, and summaries were seeded from pretrained E5 embeddings [Wang et al. 2024] and applied 192-D PCA over all embeddings. Training and inference for `IMPerSumm` used mixed-precision on A10 and L40S GPUs, provided by Lightning.ai.

E.2 Prompt Details

2-shot w/ history. This setup presents the model with the user's full interaction history of clicks, skips, and summaries. Two in-context examples, each pairing an article with the user's rewritten headline, are provided before the task. Using these demonstrations together with the history, the model generates a personalized headline for the new document.

0-shot w/ history. Here, the model again receives the complete interaction history containing clicks, skips, and summaries. However, no examples are shown; instead, the prompt explains the meaning of each action type. The model must rely on zero-shot reasoning over this history to produce the personalized headline.

Prompt-Chaining w/ history. This method structures personalization as a step-by-step process. For each document and action, the model extracts topics, keyphrases, and inferred preferences, updating a running profile. When the query document appears, the accumulated profile is used to generate the personalized headline.