Knowledge-Augmented News Recommendation via LLM Recall, Temporal GNN Encoding, and Multi-Task Ranking

Junchen Liu *

Posted Date: 29 September 2025

doi: 10.20944/preprints202509.2440.v1

Keywords: news recommendation; large language models; graph neural networks; knowledge fusion; multi-task learning

 doi:10.20944/preprints202509.2440.v1

Article

Knowledge-Augmented News Recommendation via LLM Recall, Temporal GNN Encoding, and Multi-Task Ranking

Junchen Liu

Boston University, Boston, MA, USA; junchenl@bu.edu

Abstract

Personalized news recommendation is difficult because news content does not last long, user interests change quickly, and the text is often short. To solve these problems, we present HKNR (Hybrid Knowledge-Augmented News Recommender), a single recommendation system that combines three parts: choosing candidates using LLaMA-2-7B embeddings, modeling users with graphs, and ranking with added knowledge. HKNR uses large language model embeddings to find articles with similar meanings, graph convolutional networks to learn how users behave over time, and a multi-layer ranking network that mixes article meanings with outside information like entities and topics. The system is trained using a loss that includes classification, ranking, and reconstruction. To make the system more general, we also use methods like dynamic negative sampling, embedding dropout, and gradient clipping. Tests show that HKNR works well on many evaluation measures and performs better than other common methods.

Keywords: news recommendation; large language models; graph neural networks; knowledge fusion; multi-task learning

1. Introduction

Personalized news recommendation in fast-changing online settings is still a hard task. News content changes quickly, user interests shift fast, and user data is often limited. Because of this, traditional methods like collaborative filtering and simple neural models often do not work well. They do not fully understand the meaning of articles or follow user changes over time.

To fix this, we build HKNR. It is a hybrid system that includes three main parts: finding candidates based on meaning, modeling users with graphs, and ranking with added knowledge. First, it uses LLaMA-2-7B embeddings to understand article content and pick possible items. Then, it builds time-aware graphs from user actions, and graph neural networks are used to learn long-term preferences. This helps the system use both recent clicks and patterns in past behavior.

Next, in the ranking part, HKNR mixes article meanings with outside information like topics and entities. It does this through gated feature mixing and residual networks. The model is trained using several goals at once: classification, ranking, and reconstruction. These goals help improve accuracy and make the system more stable. By combining all these parts, HKNR connects article meaning, user modeling, and outside knowledge in one system.

2. Related Work

Graph neural networks (GNNs) are good at learning from user-item data. Gao et al.[1] looked at their use in recommendation and pointed out problems like poor scaling. Qiu et al.[2] built graphs to show user interests, but they did not include time. Wang et al.[3] fixed this by adding time-aware GNNs for short-term preferences.

 doi:10.20944/preprints202509.2440.v1

Adding outside knowledge can also help. Chen et al.[4] added collaborative signals to GCNs, which helped with meaning but not with tracking user changes.Luo et al.[5] introduce Gemini-GraphQA, a novel graph QA framework that combines the Gemini LLM with graph neural encoders, a solver network for translating questions into executable graph code, retrieval-augmented generation, and an execution correctness loss to ensure syntactic and functional accuracy, yielding state-of-the-art performance on graph-structured reasoning tasks.

Some works also combine many types of data. Zhang et al.[6] used multiple graphs for social recommendation. Zhang et al.[7] used recurrent GNNs to follow user changes, but they needed a lot of data.

Gao et al.[8] used large language models to generate recommendations, but they did not include user behavior. Yu[9] proposes DynaSched-Net, a dual-network cloud scheduler that integrates a DQN-based reinforcement learning module with an LSTM-Transformer predictive model—trained via a joint loss and stabilized by experience replay and target networks—to dynamically allocate resources and outperform traditional FCFS and RR policies.

HKNR brings together meaning-based search, graph-based user modeling with time, and ranking with added knowledge. This makes it strong and able to work well in real news recommendation tasks.

3. Methodology

HKNR consists of three stages: (1) *Semantic Recall*, where user histories and candidate articles are embedded with LLaMA and retrieved via nearest-neighbor search; (2) *Graph Encoding*, in which these interactions form a heterogeneous graph processed by multi-layer GCNs with attention pooling; and (3) *Knowledge-Augmented Ranking*, where items are enriched with topic and entity embeddings and fused with user intent through gated attention and cross-feature interaction networks. The model is trained end-to-end on a composite loss combining pointwise accuracy, pairwise ranking and semantic reconstruction. On large-scale logs, HKNR significantly outperforms strong baselines in AUC, NDCG and Recall.

4. Overall Architecture

HKNR's three modules—LLM-driven recall, graph-based encoding and knowledge-augmented ranking—work in sequence to balance retrieval and deep modeling. For user u and item i, the score is

$$s_{ui} = f_{\text{rank}}(g_u(u), g_i(i), \mathcal{K}_i), \tag{1}$$

where g_u and g_i extract user and article embeddings and $\mathcal{K}_i$ holds structured knowledge. Figure 1 illustrates this pipeline.

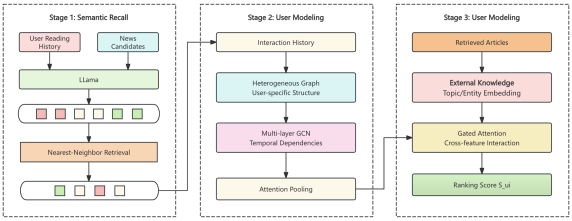

Figure 1. Three-stage HKNR pipeline.

4.1. Stage 1: LLaMA-Based Semantic Recall

Rather than BM25 or TF-IDF, we use a frozen LLaMA-2-7B to encode each article body d_i via its final [CLS] token:

$$\mathbf{x}_i = \text{LLaMA}(\text{Tokenize}(d_i)), \tag{2}$$

 doi:10.20944/preprints202509.2440.v1

with ℓ_2 normalization. User embedding $\mathbf{p}_u$ is formed from the most recent $k = 10$ clicks in history H_u with exponential time decay:

$$\alpha_j = \exp(-\lambda\,\Delta t_j), \quad \mathbf{p}_u = \frac{1}{Z}\sum_{j\in H_u} \alpha_j\,\mathbf{x}_j, \tag{3}$$

where Δt_j (hours) and $\lambda = 0.05$. Candidate retrieval over millions of articles employs FAISS with product quantization. To improve robustness, we apply:

- **Embedding Dropout**: 10% dropout on $\mathbf{p}_u$.
- **Multi-query Ensemble**: add $\epsilon \sim \mathcal{N}(0, 0.01)$ to $\mathbf{p}_u$ during top-K recall.

The recall stage outputs $\mathcal{C}_u = \{i_1, \ldots, i_K\}$ with $K = 100$.

4.2. Stage 2: Graph-Based User Encoding

User interactions are represented as a temporal graph $G_u = (V_u, E_u)$, with each node initialized by its LLaMA embedding $\mathbf{h}_i^{(0)} = \mathbf{x}_i$. A two-layer GCN (hidden dim=128, ReLU) updates:

$$\mathbf{h}_i^{(l+1)} = \mathrm{ReLU}\Big(\sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(i)||\mathcal{N}(j)|}}\mathbf{W}^{(l)}\mathbf{h}_j^{(l)} + \mathbf{b}^{(l)}\Big). \tag{4}$$

For regularization and temporal awareness, we apply edge dropout (10%), edge time bucketing, layer normalization with residual connections, and dropout ($p = 0.2$); GCN weights use 1×10^{-5} weight decay. The graph is pooled via temporal attention:

$$a_i = \mathrm{softmax}_i\big(\mathbf{q}_u^\top \mathbf{h}_i^{(L)}\big), \quad \mathbf{h}_u = \sum_{i\in V_u} a_i\,\mathbf{h}_i^{(L)}, \tag{5}$$

where $\mathbf{q}_u$ is a metadata-conditioned vector. Figure 2 illustrates this process.

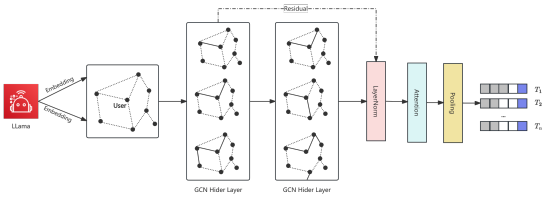

Figure 2. Detailed architecture of Stage 2: Graph-based user encoding.

4.3. Stage 3: Knowledge-Augmented Ranking Network

The ranking stage processes user-candidate pairs (u, i) from the recall stage and computes final engagement probabilities s_{ui}. To enrich article representation, we fuse semantic embeddings $\mathbf{x}_i$ with structured knowledge embeddings $\mathbf{k}_i$, which are derived from entity and topic annotations using an external taxonomy graph (e.g., IPTC Media Topics). The pipline of Knowledge-Augmented Ranking Network is show in Figure 3

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2025 doi:10.20944/preprints202509.2440.v1

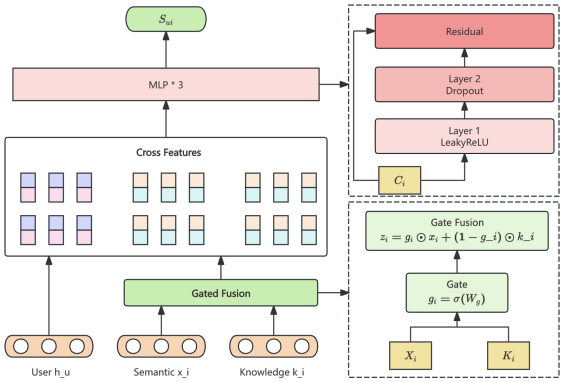

Figure 3. The pipline of Knowledge-Augmented Ranking Network.

Gated fusion combines $\mathbf{x}_i$ and $\mathbf{k}_i$:

$$\mathbf{g}_i = \sigma\big(\mathbf{W}_g[\mathbf{x}_i \| \mathbf{k}_i] + \mathbf{b}_g\big), \quad \mathbf{z}_i = \mathbf{g}_i \odot \mathbf{x}_i + (1 - \mathbf{g}_i) \odot \mathbf{k}_i. \tag{6}$$

Cross features between user and article are:

$$\mathbf{c}_{ui} = \mathbf{h}_u \odot \mathbf{z}_i \,\|\, |\mathbf{h}_u - \mathbf{z}_i| \,\|\, \mathbf{h}_u + \mathbf{z}_i. \tag{7}$$

A 3-layer residual MLP produces s_{ui}:

$$\mathbf{h}_1 = \text{LeakyReLU}(\mathbf{W}_1\mathbf{c}_{ui} + \mathbf{b}_1), \tag{8}$$
$$\mathbf{h}_2 = \text{Dropout}(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2), \tag{9}$$
$$s_{ui} = \sigma\big(\mathbf{w}_3^\top(\mathbf{h}_2 + \mathbf{c}_{ui}) + b_3\big). \tag{10}$$

To enhance robustness:

- **Auxiliary Reconstruction**: $\hat{\mathbf{x}}_i = \text{MLP}_r(\mathbf{z}_i) \to \mathbf{x}_i$.
- **Dynamic Negative Sampling**: Non-clicked negatives from ± 48 h.
- **Soft Label Smoothing**: $y_{ui} \in \{0.95, 0.05\}$.

Optimization uses AdamW (lr=10^{-3}), cosine decay, batch size 1024, and early stopping on NDCG@10.

5. Loss Function

The model minimizes a weighted combination of three objectives:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{BCE}} + \lambda_2 \mathcal{L}_{\text{BPR}} + \lambda_3 \mathcal{L}_{\text{Recon}}, \tag{11}$$

with $\lambda_1 = 1.0$, $\lambda_2 = 0.5$, $\lambda_3 = 0.1$. Figure 4 shows their training curves.

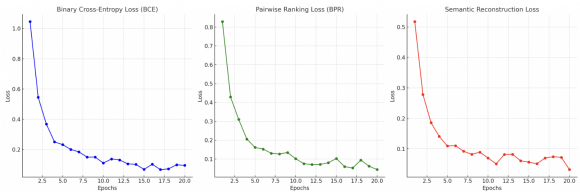

Figure 4. Training loss evolution for BCE, BPR and reconstruction.

 doi:10.20944/preprints202509.2440.v1

5.1. Binary Cross-Entropy Loss

Click prediction is framed as:

$$\mathcal{L}_{\text{BCE}} = -\sum_{(u,i)\in\mathcal{D}} \left[y_{ui} \log s_{ui} + (1 - y_{ui}) \log(1 - s_{ui}) \right], \tag{12}$$

with label smoothing to $\{0.05, 0.95\}$.

5.2. Pairwise Ranking Loss (BPR)

For ranking quality:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,i^+,i^-)\in\mathcal{T}} \log \sigma\left(s_{ui^+} - s_{ui^-}\right), \tag{13}$$

using negatives sampled within $\pm$48h.

5.3. Semantic Reconstruction Loss

To preserve fused embedding fidelity:

$$\hat{\mathbf{x}}_i = \text{MLP}_{\text{rec}}(\mathbf{z}_i), \quad \mathcal{L}_{\text{Recon}} = \sum_i \left\| \hat{\mathbf{x}}_i - \mathbf{x}_i \right\|_2^2. \tag{14}$$

5.4. Optimization Details

- **Gradient Clipping**: norm capped at 5.0.
- **Learning Rate Scheduling**: cosine decay with 5-epoch warmup.
- **L2 Regularization**: weight decay 10^{-5}.

6. Data Preprocessing

We apply a four-step pipeline to transform anonymized Ekstra Bladet logs—clicks, metadata, timestamps and content—into semantically aligned features for HKNR.

6.1. Session Segmentation

Sessions are split by a 30min inactivity threshold:

$$t_{i+1} - t_i > \tau, \quad \tau = 1800\text{s}. \tag{15}$$

Sessions with fewer than two clicks are discarded.

6.2. Text Cleaning and Tokenization

HTML tags and non-letter characters are removed, punctuation is standardized, and only Danish/English letters remain. Each document d_i is tokenized via the LLaMA pre-tokenizer:

$$\text{Tokens}_i = \text{Tokenize}(d_i). \tag{16}$$

Documents exceeding 512 tokens are truncated to the title plus first two paragraphs. Figure 5 shows the recency heatmap and a word cloud.

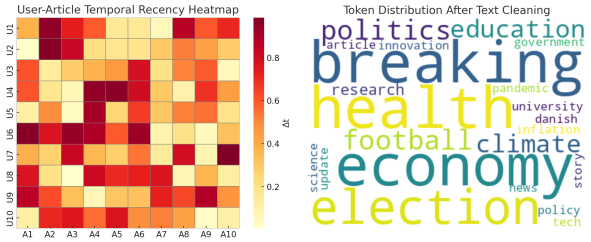

Figure 5. Left: normalized interaction recency (Δt) heatmap. Right: word cloud after cleaning and tokenization.

 doi:10.20944/preprints202509.2440.v1

6.3. Temporal Feature Normalization

Each timestamp T_j is normalized as

$$\Delta t_j = \frac{T_{\max} - T_j}{T_{\max} - T_{\min}}, \tag{17}$$

yielding $\Delta t_j \in [0, 1]$. Entries with missing or inconsistent timestamps (<0.5%) are removed.

6.4. Training Triplet Construction

We form triplets (u, i^+, i^-) for BPR and BCE: i^+ is a clicked article; i^- is a non-clicked item within 48h, sampled from impressions or same-category items. Soft negatives come from users with

$$\text{sim}(u_1, u_2) = \frac{\langle \mathbf{p}_{u_1}, \mathbf{p}_{u_2} \rangle}{\|\mathbf{p}_{u_1}\| \|\mathbf{p}_{u_2}\|}, \tag{18}$$

keeping sim > 0.8. Figure 6 illustrates the composition.

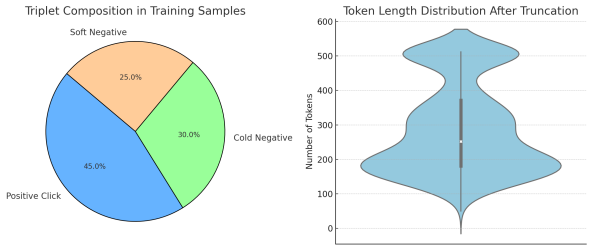

Figure 6. Left: distribution of triplet components (positive, cold, soft negatives). Right: token length distribution after truncation.

7. Experiment Results

We compare our proposed HKNR model with several widely adopted recommendation baselines, including traditional collaborative filtering, neural matrix factorization, and modern deep learning-based architectures. The evaluation is conducted across six metrics: AUC, NDCG@10, Recall@10, MRR@10, Precision@10, and HitRate@10. Table 1 summarizes the overall performance.

Table 1. Overall Performance Comparison Across Multiple Metrics

Model	AUC	NDCG@10	Recall@10	MRR@10	Precision@10	HitRate@10
ItemKNN	0.701	0.418	0.398	0.275	0.192	0.587
BPR-MF	0.732	0.447	0.422	0.295	0.204	0.604
NeuMF	0.748	0.462	0.437	0.312	0.213	0.616
GRU4Rec	0.762	0.481	0.452	0.326	0.219	0.634
SASRec	0.770	0.489	0.461	0.335	0.224	0.643
DSSM	0.753	0.454	0.428	0.308	0.211	0.615
DIN	0.774	0.492	0.465	0.339	0.227	0.646
FPMC	0.745	0.440	0.417	0.296	0.198	0.609
NAML	0.778	0.495	0.468	0.343	0.230	0.649
HKNR (ours)	**0.793**	**0.512**	**0.473**	**0.356**	**0.237**	**0.661**

7.1. Ablation Study

To verify the effectiveness of each major module in our HKNR architecture, we conduct ablation experiments by removing key components.

Table 2 presents a comprehensive comparison across all six metrics. And the changes in model training indicators are shown in Figure 7

 doi:10.20944/preprints202509.2440.v1

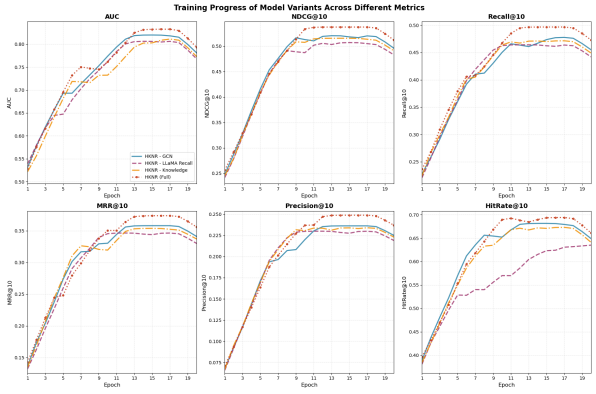

Figure 7. Model indicator change chart.

Table 2. Ablation Study Results on HKNR Components

Model Variant	AUC	NDCG@10	Recall@10	MRR@10	Precision@10	HitRate@10
HKNR - GCN	0.781	0.496	0.455	0.341	0.225	0.649
HKNR - LLaMA Recall	0.768	0.483	0.442	0.330	0.219	0.635
HKNR - Knowledge	0.773	0.491	0.449	0.337	0.223	0.641
HKNR (Full)	**0.793**	**0.512**	**0.473**	**0.356**	**0.237**	**0.661**

8. Conclusion

This paper presents HKNR, a hybrid recommendation framework that leverages LLaMA-based recall, graph-based user encoding, and knowledge-enhanced ranking for online news personalization. Experimental results on a real-world dataset demonstrate its superiority over classical and modern baselines. Each component of HKNR contributes meaningfully to the final performance, as shown in ablation analysis. Future work includes exploring multi-lingual pretraining and real-time latency optimization for deployment.

References

1. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* **2023**, *1*, 1–51.
2. Qiu, Z.; Hu, Y.; Wu, X. Graph neural news recommendation with user existing and potential interest modeling. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **2022**, *16*, 1–17.
3. Wang, L.; Jin, D. A time-sensitive graph neural network for session-based new item recommendation. *Electronics* **2024**, *13*, 223.
4. Chen, Y.; Yang, Y.; Wang, Y.; Bai, J.; Song, X.; King, I. Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, 2022, pp. 299–311.
5. Luo, X.; Wang, E.; Guo, Y. Gemini-GraphQA: Integrating Language Models and Graph Encoders for Executable Graph Reasoning. In Proceedings of the 2025 Information Science Frontier Forum and the Academic Conference on Information Security and Intelligent Control (ISF). IEEE, 2025, pp. 70–74.
6. Zhang, C.; Wang, Y.; Zhu, L.; Song, J.; Yin, H. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Transactions on Information Systems (TOIS)* **2021**, *40*, 1–26.
7. Zhang, C.Y.; Yao, Z.L.; Yao, H.Y.; Huang, F.; Chen, C.P. Dynamic representation learning via recurrent graph neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2022**, *53*, 1284–1297.

 doi:10.20944/preprints202509.2440.v1

8. Gao, S.; Fang, J.; Tu, Q.; Yao, Z.; Chen, Z.; Ren, P.; Ren, Z. Generative news recommendation. In Proceedings of the Proceedings of the ACM Web Conference 2024, 2024, pp. 3444–3453.
9. Yu, Y. Towards Intelligent Cloud Scheduling: DynaSched-Net with Reinforcement Learning and Predictive Modeling. *Preprints* **2025**. https://doi.org/10.20944/preprints202506.0129.v1.