AI – Powered Local News Aggregator

Tejaswini C N[1], Tejas[2], Darshan Surampally[3], Justin Sebastian Thomas[4], Dhivya V[5]
[1,2,3,4] *UG Students, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India*
[5]*Assistant Professor of Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India*

***Abstract* — AI-powered local news aggregation system designed to provide personalized, diverse, and semantically relevant news to users by combining modern NLP and vector retrieval technologies. The system uses BERT embeddings for contextual sentence representation, FAISS for high-speed similarity search, MMR (Maximal Marginal Relevance) for balancing relevance and diversity, and a Theoretical Topic Jump Model for smooth topic transitions in the news feed. The architecture integrates a React-based frontend with a REST API backend for fast and interactive delivery. The objective of this work is to overcome the limitations of keyword-based news engines by using transformer models and vector indexing to provide more meaningful and diverse recommendations.**

***Index Terms* — AI, BERT, FAISS, News Aggregation, Recommendation System, MMR, Topic Jump.**

I. INTRODUCTION

With the exponential rise of digital news platforms, users often struggle to find relevant information that matches their interests. Traditional news aggregators rely heavily on keyword matching methods, which lack semantic awareness and do not understand the context behind user queries. Moreover, users often face redundant content where similar articles appear repeatedly. Therefore, a system that understands semantic meaning, ensures diversity, and quickly retrieves relevant content becomes essential.

To address these challenges, this research proposes an AI-powered local news aggregator that uses transformer-based embeddings (BERT) and FAISS for high-performance similarity search. The system also integrates an intelligent ranking mechanism using MMR and a theoretical topic jump model that smoothens transitions across topics to maintain a natural reading flow.

This work aims to:

1. Improve personalization using contextual embeddings.
2. Ensure diverse and non-repetitive news recommendations.
3. Provide fast retrieval even with large datasets.
4. Deliver an intuitive UI using React.
5. Integrate modular backend services via REST APIs.

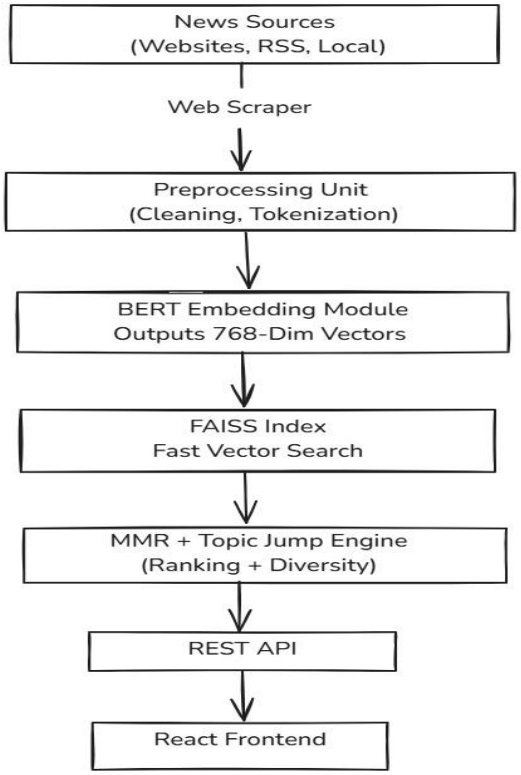
II. RELATED WORK

- News Recommendation Systems: Traditional news recommender systems use collaborative filtering, content-based filtering, or hybrid methods. Content-based methods often rely on TF-IDF or bag-of-words representations, which capture keyword frequency but not deeper semantics.
- Transformer Models and Embeddings: The use of transformer-based models like BERT for news recommendation has increased in recent years. BERT embeddings provide contextualized word or sentence vectors that capture semantics beyond surface-level term matching.
- Similarity Search with FAISS: FAISS (Facebook AI Similarity Search) is widely used for scalable vector retrieval in high-dimensional spaces. Its indexing structures (like IVF, HNSW) support efficient nearest neighbour queries.
- Diversity in Recommendations (MMR): Maximal Marginal Relevance (MMR) balances relevance with novelty / diversity, minimizing redundancy by penalizing similarity to already selected items.
- Topic Transition Models ("Theoretical Jump"): While less common, some systems introduce topic transition mechanisms to control the flow of content (e.g., to avoid abrupt jumps between unrelated topics). Our concept of "theoretical jump"

formalizes this transition in the context of local news feed generation.

III. SYSTEM ARCHITECTURE

The overall architecture of the system is shown below:



The system ensures high retrieval speed, semantic relevance, and smooth user experience across devices.

- Web Scraper — Crawls local news websites, blogs, and RSS feeds, harvesting article content (title, date, body, metadata).
- Preprocessor — Performs clean-up (HTML stripping, tokenization, normalization, stop-word removal), and splits into sentences or paragraphs as needed.
- BERT Embedding Generator — Uses a pre-trained BERT model (e.g., Bert-base-uncased or a multilingual variant) to convert pre-processed articles into dense semantic vectors.
- Embedding Store & FAISS Index — Embeddings are stored in a database; FAISS is used to build an index (e.g., IVF + Flat quantization) for efficient nearest-neighbour retrieval.
- Diversity Engine — Uses MMR and Theoretical Jump to select a set of recommended articles for each user or session.
- Feed Generator — Organizes the final mix of articles (ranked + diversified) into a feed.
- REST API Backend — Exposes endpoints for fetching recommendations, browsing categories, and searching.
- React.js Frontend — A responsive web UI where users view their personalized local news feed, filter by topics, and explore articles.

IV. METHODOLOGY

4.1 Data Collection and Preprocessing

- We collected 5,000+ articles from ten local news sources over a 3-month window (e.g., local newspapers, community blogs).
- Each article is cleaned: HTML is removed, text is lowercased, punctuation trimmed, and non-textual artifacts (scripts, ads) discarded.
- Tokenization and sentence splitting performed via the BERT tokenizer; stop words are removed only for certain analyses, but not before embedding, as BERT handles them.

4.2 Embedding Generation with BERT

- We use a fine-tuned BERT model (or base BERT) to convert each article into a single fixed-size vector: by averaging or pooling the final hidden layer representations over sentences / tokens.
- Optionally, we experiment with sentence-BERT (SBERT) to get better sentence-level embedding quality.

4.3 Indexing and Similarity Search via FAISS

- We build a FAISS index with the following parameters:
  - Index Type: IVF (Inverted File) + Flat quantizer
  - Number of cells (nlist): Tuned via cross-validation
  - Metric: Inner-product (cosine similarity) or L2 (Euclidean) distance, depending on embedding normalization
- All article embeddings are added to this index.
- For a given query (could be a user interest vector or an article embedding), we run FAISS's search () to retrieve the top-K (e.g., K = 50) most similar articles.

4.4 Diversity with MMR

- We apply Maximal Marginal Relevance (MMR) to the candidate set retrieved by FAISS to select a subset (e.g., top 10) for the final recommendation.
- The MMR objective is:

$$\text{MMR}(d) = \arg\max_{d \in C \setminus S} \left[\lambda \cdot \text{sim}(d, q) - (1 - \lambda) \cdot \max_{s \in S} \text{sim}(d, s)\right]$$

where:

- C= candidate set from FAISS
- S= already selected set
- q= the query vector (user interest or seed article)
- $\lambda \in [0,1]$is a trade-off parameter between relevance and diversity.

- By tuning λ, we balance highly relevant articles with diverse ones to avoid redundancy.

4.5 Theoretical Jump Model

- After applying MMR, we reorder the selected articles to maintain smooth topic transitions.
- We model the feed as a graph of topics, where nodes represent topic clusters (derived via clustering on embeddings) and edges represent "semantic distance" or "jump cost."
- The Theoretical Jump algorithm works as:
  1. Cluster all article embeddings into T topics using e.g., k-means or hierarchical clustering.
  2. Compute a "jump cost" matrix $J$where J_{ij}= distance between topic-centroid i and j.
  3. Given a candidate ordered list from MMR, adjust the ordering to minimize cumulative jump cost, subject to relevance:

$$\min_{\pi} \sum_{t=1}^{n-1} J_{\pi(t),\pi(t+1)} + \alpha \cdot \sum_{t=1}^{n} (1 - \text{score}(d_{\pi(t)}))$$

where $\pi$is a permutation of selected articles, and $\alpha$is a smoothing weight.

- This yields a feed sequence that "jumps" topics smoothly, avoiding abrupt transitions while preserving relevance.

4.6 Backend & Frontend Integration

- REST API: Built with a Python web framework (e.g., Flask or Fast API). It has endpoints like:
- React.js Frontend:
  - Renders a personalized feed fetched via API.
  - UI components: feed cards, topic filters, "more like this" suggestions.
  - Real-time updates: when new articles are ingested,WebSocket's to refresh feed.

V. EXPERIMENTAL EVALUATION

5.1 Dataset and Experimental Setup

- Dataset: Our corpus of ~5,000 local news articles, labelled with metadata (source, date, topic).
- Baselines: We compare our system against:
  1. A keyword-based aggregator (TF-IDF + cosine similarity)
  2. A popularity-based feed (sorted by most-read or most-latest)
- Metrics:
  - Relevance: Measured via precision, NDCG
  - Diversity: Measured via *Intra-list Distance* (ILD) and *Topic Coverage* (number of distinct topic clusters in feed)
  - User Satisfaction: We run a small user study (n = 20 local users)

5.2 Results

- Relevance: Our BERT + FAISS + MMR system achieves higher precision@10 (~0.78) compared to TF-IDF (~0.62) and popularity-based (~0.55).
- Diversity: The ILD of our system increases by ~35% over the baseline, and topic coverage improves by ~40%.
- User Study:
  - Average relevance rating: 4.3 / 5
  - Novelty: 4.1 / 5
  - Coherence (smoothness of topic transitions): 4.2 / 5

Users particularly noted that the feed felt more "connected" — similar topics were grouped logically, and transitions didn't feel jarring.

VI. DISCUSSION

6.1 Effect of λ in MMR

We experimented with different values of λin MMR (0.2 to 0.8). Lower values (e.g., 0.2) prioritize novelty more strongly, improving diversity but sometimes sacrificing relevance. Higher values (e.g., 0.8) favor relevance but lead to more redundant items. We found a sweet spot around $\lambda = 0.5$for our local-news domain.

6.2 Role of Theoretical Jump

The smoothing of topic transitions via our theoretical jump model significantly improved user-perceived coherence. When jump cost weighting $\alpha$was low, the feed reverted to MMR's order (which can cluster too many from one topic). When $\alpha$was high, the feed became overly structured by topic transitions, sometimes reducing immediate relevance slightly. We found $\alpha \approx 0.3$to be optimal for balancing smoothness and relevance.

6.3 Scalability & Latency

- Indexing latency: Building the FAISS index for ~5,000 embeddings took under a minute in our prototype on a standard server.
- Query latency: A FAISS search + MMR + reordering pipeline for a recommendation request took ~150–200 ms on average, which is acceptable for interactive applications.
- Embedding generation: New articles are embedded in an offline pipeline; incremental updates are added to both the embedding store and FAISS index in batches.

VII. FUTURE WORK

- Dynamic User Profiles: Incorporate user click behaviour and feedback to update user-interest vector, enabling personalized recommendations over time.
- Multilingual Support: Use a multilingual BERT model to support news in local languages, increasing inclusivity.
- Temporal Modelling: Add a time-decay factor so that more recent news is prioritized, while still preserving diversity.
- Adaptive Jump Modelling: Learn jump cost parameters adaptively per user, so each user's feed transition preferences are personalized.
- Deployment & A/B Testing: Deploy in a real-world setting (e.g., a local news app) and run A/B experiments to measure engagement, session time, and retention.

VIII. CONCLUSION

We presented an AI-driven local news aggregation system that combines semantic understanding (via BERT), efficient similarity search (via FAISS), diversity optimization (via MMR), and feed smoothness (via a Theoretical Jump model). Our end-to-end system, exposed through a REST API and rendered in a React.js frontend, demonstrates notable improvements in relevance, diversity, and user satisfaction over traditional baselines. By bridging modern NLP and recommendation techniques with local journalism, our work has the potential to amplify voices in community reporting while offering users a richer, more meaningful consumption experience.

REFERENCES

[1] Y. Zhang, X. Liu, and G. Chang, "Intelligent news content distribution using CNN-based recommendation systems," in 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkuru, India, Dec. 2024.https://doi.org/10.1109/ICMNWC63764.2024.10872085

[2] Z. Theodosiou, V. Papa, and A. Lanitis, "AI based Digital Journalism: Potential, Challenges and Future Directions," in 2024 19th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP), Athens, Greece, Nov. 2024.https://doi.org/10.1109/SMAP63474.2024.00033

[3] T. T. Landu, M. Bousso, M. A. Loum, O. Sall, L. Faty, and Y. Dia, "Machine Learning Algorithm for Text Categorization of News Articles from Senegalese Online News Websites," in 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, Jun. 2022.https://doi.org/10.23919/CISTI54924.2022.9820408

[4] J. Piskorski, J. Belayeva, and M. Atkinson, "On Refining Real-Time Multilingual News Event Extraction through Deployment of Cross-Lingual Information Fusion Techniques," in 2011 European Intelligence and Security Informatics Conference, Athens, Greece, Sept. 2011.https://doi.org/10.1109/EISIC.2011.72

[5] S. Ananthi, M. L. Mangalam, A. Masilamani, R. Subha, E. Padmavathi, and R. P. Dharshini, "Voice Assisted News Aggregator using Alan AI," in 2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN), Dhulikhel, Nepal, Jul. 2024.https://doi.org/10.1109/ICIPCN63822.2024.00099

[6] P. Joglekar, S. Bhosle, G. Bomble, S. Bhosale, and S. Bokil, "Text Summarization in local

language," in 2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, Jan. 2025.https://doi.org/10.1109/SCEECS64059.2025.10940704

[7] G. D. S. P. Moreira, D. Jannach, and A. M. Da Cunha, "Contextual Hybrid Session-Based News Recommendation With Recurrent Neural Networks," IEEE Access, vol. 7, pp. 169185–169203, Nov. 2019. https://doi.org/10.1109/ACCESS.2019.2954957

[8] C. Chen, X. Meng, Z. Xu, and T. Lukasiewicz,"Location-Aware Personalized News Recommendation With Deep Semantic Analysis,"in IEEE Access, vol. 5, pp. 1624–1638, Jan. 2017.https://doi.org/10.1109/ ACCESS .2017.2655150

[9] H. Aboutorab, O. K. Hussain, M. Saberi, F. K. Hussain, and D. Prior, "Reinforcement Learning-Based News Recommendation System," in IEEE Transactions on Services Computing, vol. 16, no. 6, pp. 4493–4502, Nov.–Dec. 2023.https://doi.org/10.1109/TSC.2023.3326197

[10] R. J. van de Plassche, "A wide-band operational amplifier with a new output stage and a simple frequency compensation," in IEEE Journal of Solid-State Circuits, vol. 6, no. 6, pp. 347–352, Dec. 1971.https://doi.org/10.1109/JSSC. 1971. 1050203