An User Interest Modeling with Diverse Behavior Analyses and Embeddings for Building Online News Recommendation Systems

Jiangwei Luo*
Intelligent algorithm department,
MGTV Corporation
China
luojw2007@126.com

Ye Tang
Intelligent algorithm department,
MGTV Corporation
China
tangye@mgtv.com

Shien Song
Intelligent algorithm department,
MGTV Corporation
China
shien@mgtv.com

Haibo Lu
Intelligent algorithm department,
MGTV Corporation
China
haibo2@mgtv.com

Jianpeng Long†
Intelligent algorithm department,
MGTV Corporation
China
jianpeng@mgtv.com

Abstract

Building an online news recommendation system[11] is crucial for enhancing user engagement and providing personalized content, but it comes with significant challenges. One of the key aspects is modeling user preferences based on implicit behaviors, such as reading habits and interaction patterns. This requires sophisticated algorithms to accurately capture and predict user interests.

Another important factor is the influence of the news agenda on user interests. News topics can shape what users find relevant or important, making it necessary for the recommendation system to adapt dynamically to the changing news landscape.

Additionally, managing the rapid decay of news items presents a challenge. News articles quickly lose relevance, so the system must ensure that it recommends timely and up-to-date content to maintain user engagement.

In summary, the importance of building an online news recommendation system lies in its ability to personalize user experiences, but it must effectively address the complexities of implicit user behavior modeling, the impact of news agendas, and the fast-paced nature of news relevance.

In this pager, We present an effective and industrial solution to this challenge, our recommendation pipeline is composed of six stages, which is focused on data preprocessing, candidate generation, construct training samples, build ranking models, feature engineering and blend. Finally, with our solution, our team axi2017 achieved **4th place** in ACM RecSys Challenge 2024[1] [1] among 521 participants.

*Main contribution And Corresponding author
†These authors contributed equally to this work.
[1]https://recsys.eb.dk/

RecSys Challenge '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-1127-5/24/10
https://doi.org/10.1145/3687151.3687162

CCS Concepts

• **Information systems**; • **Recommender systems**; • **Computing methodologies**; • **Natural language processing**;

Keywords

User Interest Modeling, User Behavior Analysis, Applications of news recommendation, Graph Neural Network, News content modeling, Dataset analyses and preprocessing techniques

ACM Reference Format:

Jiangwei Luo, Ye Tang, Shien Song, Haibo Lu, and Jianpeng Long. 2024. An User Interest Modeling with Diverse Behavior Analyses and Embeddings for Building Online News Recommendation Systems. In *ACM RecSys Challenge 2024 (RecSys Challenge '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3687151.3687162

1 Introduction

1.1 Background

ACM RecSys Challenge 2024[1] focuses on online news recommendation, addressing both the technical and normative challenges inherent in the design of effective and responsible recommender systems for news publishing.

The challenge will delve into the unique aspects of news recommendation. These include modeling user preferences based on implicit behavior, accounting for the influence of the news agenda on user interests, and managing the rapid decay of news items. Furthermore, this challenge also embraces the normative complexities. These involve investigating the effects of recommender systems on the news flow, and whether they resonate with editorial values. By providing participants with a comprehensive dataset and a robust news recommendation evaluation framework, our goal is to tackle these multifaceted challenges head-on. As part of the challenge, Ekstra Bladet will be releasing an anonymized dataset with approximately 2 million random users who engaged with EkstraBladet.dk over a six-week period.

1.2 Dataset Description

To support advancements in news recommendation research, organizer has constructed the Ekstra Bladet News Recommendation

Dataset (EB-NeRD). It was collected from the user behavior logs at Ekstra Bladet.

The Ekstra Bladet News Recommendation Dataset (EB-NeRD) is a large-scale Danish dataset created by Ekstra Bladet to support advancements and benchmarking in news recommendation research. EB-NeRD comprises over 2.7 million users and more than 600 million impression logs from Ekstra Bladet. Alongside, the datasets offer a collection of more than 120 thousands news articles, enriched with textual content features such as titles, abstracts, and bodies. This enables text features in a low-resource language as context for recommender systems.

1.3 Task Description

The Ekstra Bladet RecSys Challenge aims to predict which article a user will click from a list of articles that was seen during a specific impression. Utilizing the user's browsing history, session details (like time and device used), and personal metadata (including gender and age), along with a list of candidate news articles, listed in an impression log. The challenge's objective is to rank the candidate articles based on the user's personal preferences. This involves developing models that encapsulate both the users and the articles through their content and the users' interests.

To evaluate the models we use several standard metrics in the recommendation field, including the area under the ROC curve (AUC), mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG@K) for K shown recommendations. To address the normative complexities inherent in news recommendations, the test set incorporates samples specifically designed to assess models based on normative properties. This includes evaluating models on Beyond-Accuracy Objectives, such as intra-list diversity, serendipity, novelty, coverage, among others. The final result is the average of these metrics across all impression logs.

2 RELATED WORK

As depicted in Figure 1, we designed 4 stage for building online news recommendation system, which is focused on Data Preprocessing, Feature Engineering, Build Ranking Models and Blending.

2.1 Data Preprocessing

To achieve effective ranking models, we utilize a variety of features and models in both processes. In this section, we will elaborate on the generation of these commonly used data, which form the fundamental basis for our models.

2.1.1 *Covisit Matrix for Item to Item Similarity.*

First, we represent the user behavior sequence data from the competition dataset as an item-to-item co-occurrence directed graph. As shown in Figure 3. In this graph, nodes represent items, and edges represent co-occurrence relationships within a user session.

The purpose of constructing co-occurrence relationships is to calculate the similarity between edges. The similarity between edges is computed using appropriate algorithms based on the co-occurrence relationships across multiple user sessions.

The specific algorithm is represented by the following formula: I2I_Sim(i, j), which indicates the similarity between items i and j. This can be obtained by considering users who interacted with both items simultaneously. The item-to-item[9, 10, 15] similarity calculation takes into account various co-occurrence relationship weight factors.

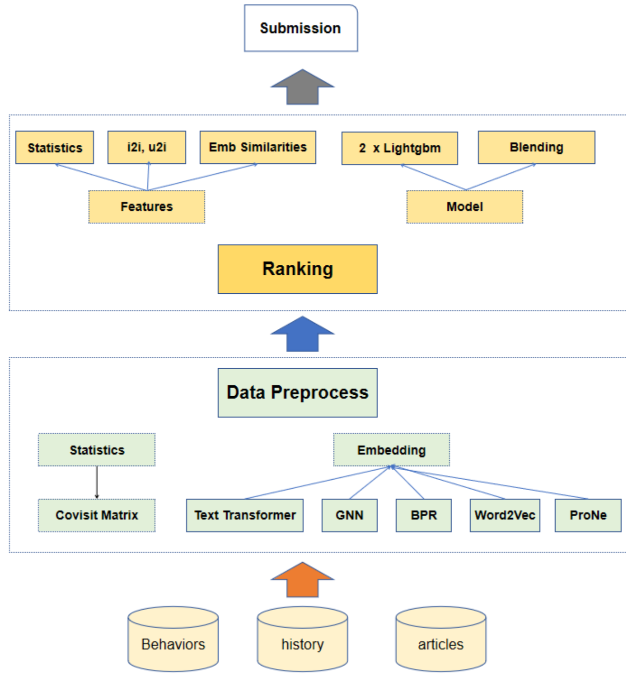

Figure 1: Solution Overall Architecture

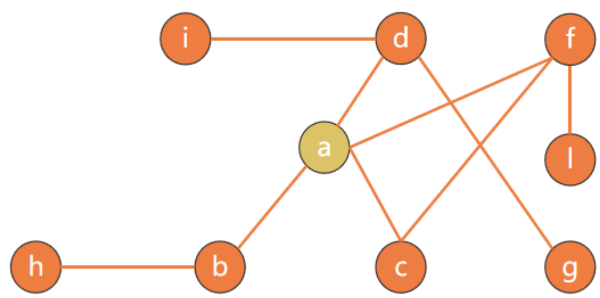

Figure 2: I2I Graph

$$I2I\_Sim(i,j) = \sum_{u \epsilon U_i \bigcap U_j} \frac{W_{pos}(i,j) * W_{sessions}(u) * W_{is\_last} * W_{is\_order}}{\sqrt{|i|} * \sqrt{|j|}}$$

$$W_{pos}(i,j) = \frac{1}{\log\left(|Pos_i - Pos_j| + 1\right)}$$

$$W_{sessions}(u) = \frac{1}{\log\left(|sessions| + 1\right)}$$

$$W_{is\_last} = \begin{cases} 1 & \text{if } is\_last \text{ is True} \\ 0.7 & \text{if } is\_last \text{ is False} \end{cases} \quad (1)$$

$$W_{is\_order} = \begin{cases} 1 & \text{if } is\_order \text{ is True} \\ 0.7 & \text{if } is\_order \text{ is False} \end{cases} \quad (2)$$

Firstly, there is a position weight factor that emphasizes the relevance between items based on their proximity in the user's interaction sequence. The closer their positions, the higher the weight.

Next, there is a purchase factor that assigns a weight of 1 if the co-occurrence relationship represents the last purchase, and 0.7 otherwise.

There is also a sequence factor that assigns a weight of 1 if the co-occurrence relationship follows the chronological order, and 0.7 if it is in reverse order.

Lastly, there is a weight decay for popular users and items. This is because highly popular users and items have a broad influence, and many items may be associated with them, which may not reflect personalized factors. Therefore, their weights are attenuated based on their popularity.

The item-to-item similarity calculation integrates these position, last purchase, sequence order, and popularity weight factors to derive a comprehensive similarity measure.

2.1.2 *Word2vec Embedding Similarity.*

Word2vec [6] is a technique for natural language processing published in 2013, The word2vec algorithm uses a neural network model to learn word associations in a large corpus of text. We combine a single user's views and purchases of the item id as a sentence to input into word2vec model. As mentioned before, most sessions are very short, so we just use the window size of 2. Considering there are only about 5 million sessions and 1 of millions of interacted items, which is not large enough to make embedding learned effective, so we use the 32 as the embedding dimension.

2.1.3 *BPR Score.*

BPR [8] (Bayesian Personalized Ranking) is a recommendation algorithm designed for collaborative filtering based on implicit feedback. It focuses on learning personalized ranking models from user-item interactions, where the feedback is often binary (e.g., whether a user has interacted with an item or not).

The algorithm utilizes a pairwise learning-to-rank approach, where it optimizes the ranking order of items based on observed user preferences. It formulates the problem as maximizing the posterior probability of item rankings given the observed user-item interactions.

BPR leverages a matrix factorization technique, typically using matrix factorization models like collaborative filtering with matrix factorization (CF-MF). It learns latent representations (i.e., embeddings) for users and items in a low-dimensional space, and then applies stochastic gradient descent (SGD) to optimize these embeddings by maximizing the posterior probability.

2.1.4 *PRONE Embedding Similarity.*

ProNE [14]: Fast and Scalable Network Representation Learning is a research paper that addresses the challenge of network representation learning on large-scale networks.

The main objective of ProNE is to provide a scalable solution for network embedding. It formulates network representation learning as an optimization problem, focusing on preserving proximity relationships between nodes in a low-dimensional embedding space.

ProNE generates high-quality network embeddings that capture structural and similarity information, enabling various downstream tasks such as node classification, link prediction, and community detection.

2.1.5 *Graph Neural Networks.*

In Recent years, Graph Neural Networks have been proposed to model graphs and have seen success on various recommender systems, even in large-style real-world production environments. We implement a GNN network based on PinSage [13] with the item node having the attribute of pivoted labels and export item node embedding to represent item info. Specifically, we encode behavior type features (behavior or history) in the construction of our graph model, so that the model can learn the embeddings discriminatingly by a different type of behavior. In addition, according to our experience of E-commence, time context constrain should be helpful to improve model accuracy. We use delta time between two behavior to regularize the learning process of embeddings.

2.1.6 *Swing.*

Swing algorithm [12] is a kind of i2u2i algorithm to capture the substitute relationships between items. It can utilize the inner structure-swing of the user-item behavior graph. It was designed to be much more stable than a single edge used in traditional CF approaches and provides much more reliable calculation propagation over a user-item bi-partite graph.

2.1.7 *Text Transformer.*

In this dataset, there are several text features associated with items, such as title, body. To leverage these textual information, we employed transformers using two approaches[5]:

- Pretrained sentence transformer[7]. We use a pretrained sentence transformer for multilingual text to encode each attribute of an item into a vector.
- Fine-tuned transformer with contrastive loss. To improve generalization in the competition dataset, we fine-tuned the transformer by referring to the method described in[3].First, we formulated an item as a "sentence" by flattening its key-value attributes described by text. Then, we used a transformer to encode this "sentence" into a vector, denoted as h_s. Finally, a contrastive loss is utilized to fine-tune the transformer:

$$L = -\log \frac{e^{sim(h_s, h_i^+)} / \tau}{\sum_{i \epsilon \beta} e^{sim(h_s, h_i)}}$$

where sim is the cosine similarity function; h_i^+ is the embedding of the next item to $item_i$ in the user session;β is all items in the batch and τ is a temperature parameter.

2.2 Feature Engineering

2.2.1 *Statistical Features.*

We create statistical features in three aspects: article, Impression & Session & User, and Impression & Session & User crossing article.

- Article Statistics:

(1) Meta Article Elements: Article id, Category ID, Premium, Article Type, Sentiment Score, Sentiment Label, Total Inviews, Total Pageviews, Total Read Time, Total Inviews /

Total Pageviews, Total Read Time / Total Inviews and so on.
(2) Article Counts: Count of Article id, Category ID, Premium, Article Type, Sentiment Score, Sentiment Label with rolling times.

- Impression & Session & User Statistic: Nunique of interacted articles and their Premium, Category, type, NER, Entities, Topics; mean, median, max, and min Sentiment Score, Sentiment Label, Total Inviews, Total Pageviews, Total Read Time, Total Inviews / Total Pageviews, Total Read Time / Total Inviews of interacted articles.
- Impression & Session & User Article Cross Statistics:

(1) Count of candidate Article's Category ID, Premium, Article Type, Sentiment Score and Sentiment Label in Impression or Session or User-interacted articles.
(2) Count of candidate Article's Category ID, Premium, Article Type, Sentiment Score and Sentiment Label in Impression or Session or User-interacted articles.
(3) Max, mean, min, std statistic of Impression or Session or User-interacted articles's Premium, Sentiment Score, Sentiment Label, Total Inviews, Total Pageviews, Total Read Time, Total Inviews / Total Pageviews, Total Read Time / Total Inviews.
(4) Max, mean, min, sum and position-weighted sum of covisit score of user's history items and candidate articles.
(5) The rank number of the candidate item under each Impression Article Cross Statistics Features.

2.2.2 *Covisit Matrix Features.*

We use the Covisit Matrix for Item to Item Similarity strategy as feature inputs to the model, which include i2i similarity, swing similarity, and the ranking scores of these similarities by grouped by Impression & Session & User.

2.2.3 *Text similarity Features.*

We calculated the Levenshtein distance, Hamming similarity, Jaccard similarity, and Jaro similarity between the titles of the items from the user's last 1, 2, and 3 history interactions and the titles of the candidate articles.

2.2.4 *Embedding Similarity Features.*

The similarity between the user and candidate item, based on the embeddings described in Section 2.1, show a very impressive effect.

- I2I embedding similarity. This refers to the similarity between the candidate article and the impression or session or user interacted articles. Equation 6 represents the process of calculating I2I similarity.

$$Sim(u,i) = Aggr(cos\_sim(e_i, e_j)|j \in u)$$

where Aggr is the aggregation function, such as max, mean, min, and sum. e_i and e_j are the embeddings of article i and article j, which can be graph embeddings, text transformer embeddings, word2vec embeddings, ProNE embeddings.
- U2I embedding similarity. As we factorize impression & session & user and article with BPR, it's natural to use the similarity of impression & session & user embedding and article embedding as a feature.

2.3 Ranking

Due to the considerable size of the dataset after the feature engineering, so we only randomly sampled 10% of the impression_id datas for model training. It requires the model used for training and predictions to be scalable and fast. We use the most popular GBDT which is Lightgbm [2], a model based on decision tree algorithm and used for ranking, classification and other machine learning tasks. It has advantage on speed, multiple loss functions, parallel training, and has been proved to be SOTA solution's tool in many challenges.We test the objective of "binary" and "lambdarank [4]" to show the difference of classification and learning to rank, and finally found the learning to rank task can get higher score in the challenge.

2.4 Blend

Finally, for increase model diversity, we performed a weighted rank average of the outputs from two models with different objectives. We assigned a weight of 0.6 to the model with a lambdarank objective and 0.4 to the model with a binary objective. As a result, we achieved an online score of 0.8692, placing us fourth on the competition leaderboard.

3 EXPERIMENT RESULTS

3.1 Final Leaderborad

Table 1 is the top 10 final leaderboard with AUC score. With our solution, our team axi2017 achieved fourth place in this competition.

Rank	Team Name	AUC
1	:D	0.8924
2	BlackPearl	0.8815
3	BlacTom3TKkPearl	0.8707
4	**axi2017**	0.8692
5	hrec	0.8667
6	FeatureSalad	0.8513
7	AmazMe	0.8347
8	SBJ Partners	0.8169
9	-.-	0.7988
10	SamNews	0.7885

Table 1: Top 10 score in acm recsys challenge 2024

3.2 Report results both with and without leaks

Table 2 is the results of both with and without leakage features. The result including leakage features achieved a final auc score of 0.8692, while the result without leakage features scored 0.8015.

Number	Model	AUC
1	With Leakage Features	0.8692
2	Without Leakage Features	0.8015

Table 2: Report results both with and without leakage features

References

[1] 2024. *RecSys Challenge 2024*. http://www.recsyschallenge.com/2024/ [Accessed: 07-18-2024].

[2] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[3] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2019. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *arXiv:2305.13731 [cs.IR]*.

[4] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[5] Jiangwei Luo, ZhouZhou He, Ye Tang, Wentao Tang, and Cheng Li. 2023. A Stacking and Transfer Learning with Diverse Similarity For Building Multilingual Session-based Recommendation Systems. In *Kdd cup 2023 workshop*.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[7] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.*

[8] Gantner Z et al Rendle S, Freudenthaler C. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. *Conference on Uncertainty in Artificial Intelligence* (2009).

[9] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[10] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[11] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2023. Personalized News Recommendation: Methods and Challenges. In *ACM Transactions on Information Systems*.

[12] Xiaoyong Yang, Yadong Zhu, Yi Zhang, Xiaobo Wang, and Quan Yuan. 2020. Large Scale Product Graph Construction for Recommendation in E-commerce. *arXiv preprint arXiv:2010.05525* (2020).

[13] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[14] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. ProNE: Fast and Scalable Network Representation Learning. (7 2019), 4278–4284. https://doi.org/10.24963/ijcai.2019/594

[15] Weijia Zhang, senkin13, Zhongshan Huang, Lu Wang, and Qiang Wang. 2023. A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation. In *Kdd cup 2023 workshop*.