

Graph Enhanced Representation Learning for News Recommendation on EB-NeRD

YIGIT CAN OZKAYA, Universiteit van Amsterdam, Netherlands

TIM VAN GELDER, Universiteit van Amsterdam, Netherlands

RORY GULIKER, Universiteit van Amsterdam, Netherlands

CCS Concepts: • **Mathematics of computing** → *Graph theory; Probabilistic algorithms;*

Additional Key Words and Phrases: Graphs, Recommender systems, Graph Neural Network

ACM Reference Format:

Yigit Can Ozkaya, Tim van Gelder, and Rory Guliker. 2024. Graph Enhanced Representation Learning for News Recommendation on EB-NeRD. *J. ACM* 37, 4, Article 111 (August 2024), 8 pages. <https://doi.org/XXXX.XXXXXXX>

1 Introduction

Recommendation systems have, for a while now, played a crucial role in the delivery of news articles to their audiences, helping users navigate the overwhelming volume of information that is now available online. Accurate and diverse recommendations are essential, keeping people up to date on matters that interest or concern them, without having to sift through irrelevant information that might turn someone away. Traditional recommendation approaches, such as collaborative filtering and content-based filtering, have been widely used by a multitude of platforms, but are frequently found lacking in their ability to capture the more complex relations between users and the news articles they read. These often methods face challenges such as the cold-start problem and a lack of diversity in recommendations, which can unfortunately lead to issues such as filter bubbles and reduced user satisfaction.

To address these challenges, the paper (./cite GERL) proposed a novel approach leveraging Graph Neural Networks (GNNs) and enhanced representation learning. The GERL framework demonstrated significant improvements in recommendation accuracy and diversity by effectively modeling the intricate relationships between users and news articles through graph structures. The framework utilized 1-hop and 2-hop neighbor information to capture both direct and broader contextual user interests, achieving state-of-the-art performance on real-world datasets like MIND and Adressa.

This work aims to reproduce the authors' work, and to apply it to the Ekstra Bladet News Recommendation Dataset (EB-NeRD), a large-scale Danish dataset specifically prepared for the RecSys2024 challenge. The EB-NeRD dataset provides a unique opportunity to test the robustness and generalizability of the GERL framework in a new linguistic context, as it operates in an entirely

Authors' Contact Information: Yigit Can Ozkaya, Universiteit van Amsterdam, Amsterdam, Netherlands, yigit.ozkaya@student.uva.nl; Tim van Gelder, Universiteit van Amsterdam, Amsterdam, Netherlands, tim.van.gelder@student.uva.nl; Rory Guliker, Universiteit van Amsterdam, Amsterdam, Netherlands, rory.guliker@student.uva.nl.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1557-735X/2024/8-ART111

<https://doi.org/XXXXXXX.XXXXXXX>

different language. Furthermore, the usage of this dataset in particular, due to the challenge, allows the GERL framework's performance to be directly compared to its contemporaries.

In addition to reproducing the original GERL framework, we extend the work by comparing the use of several pre-trained EB-NeRD document embeddings without the Transformer title encoder to FastText Danish pre-trained word embeddings with the original GERL architecture, including the title encoder. Furthermore, we evaluate the impact of using user and news histories (one-hop neighbors) sorted by read time, combined with user two-hop neighbors ranked by commonly clicked articles, against the more naive method of randomly sampling one-hop neighbors and not ranking two-hop neighbors. Lastly, we initiated the incorporation of image embeddings into the news representations to capture visual information alongside textual data and began exploring different aggregation methods such as Multi-Layer Perceptron (MLP) and Multi-Head attention.

Our findings reveal that the choice of pre-trained embeddings (e.g., document vs. word) has a significant impact on recommendation performance. Moreover, leveraging pre-trained embeddings instead of the Transformer title encoder in the GERL architecture can potentially reduce computational costs. Additionally, sorting user and news histories by read time does not lead to substantial improvements, suggesting that this approach requires further investigation.

2 Related Works

2.1 Graph Enhanced Representation Learning for News Recommendation (GERL)

The GERL framework leverages advanced graph neural networks (GNNs) to model complex relationships between users and news articles. This approach utilizes both 1-hop and 2-hop neighbor information to capture direct and broader contextual user interests. In this framework, users and news articles are represented as nodes, and interactions such as clicks, reads, or likes are represented as edges. The GNNs can capture high-order connectivity patterns, propagating information across the network and learning rich, contextualized representations of both users and articles. Evaluated on datasets like MIND and Adressa, GERL outperformed state-of-the-art baseline models such as DKN, NRMS, LSTUR, and NAML in terms of accuracy and diversity of recommendations [GERL2020].

GERL incorporates two key components: 1. Graph Neural Networks (GNNs): Utilizes graph structures to model user-item interactions, capturing both local (1-hop) and broader (2-hop) contextual information. 2. Enhanced Representation Learning: Combines GNNs with pre-trained language models like BERT for news encoding and dynamic user profiles for user encoding.

2.2 Collaborative Filtering (CF)

Collaborative Filtering has been a cornerstone in recommendation systems, relying on user-item interaction data to make recommendations. Techniques such as Matrix Factorization [Koren2009] decompose the interaction matrix into latent factors representing users and items. This method predicts user preferences by finding patterns in user-item interactions. However, CF often struggles with the cold-start problem and sparsity issues in datasets.

2.3 Hybrid Recommendation Systems

To address the limitations of individual methods, hybrid recommendation systems combine collaborative and content-based techniques. Burke [Burke2002] provides a comprehensive overview of hybrid systems, highlighting their ability to leverage the strengths of both approaches. These systems can offer improved performance but often require more complex integration strategies.

2.4 Deep Knowledge-Aware Network (DKN)

Another important work preceding ours was DKN, which integrates knowledge graphs with news recommendation by incorporating entity embeddings into a deep learning framework. This method enhances the semantic understanding of news content and improves recommendation accuracy by leveraging structured knowledge. DKN uses a CNN-based architecture to combine semantic embeddings from a knowledge graph with textual content, allowing for a more enriched representation of news articles [DKN].

2.5 Neural News Recommendation with Multi-Head Self-Attention (NRMS)

NRMS employs multi-head self-attention mechanisms to capture diverse aspects of news articles and user preferences. This approach allows the model to attend to different parts of the news content, providing a richer representation for recommendation. The self-attention mechanism in NRMS enables the capture of both global and local contextual information in news articles, leading to improved recommendation quality [NRMS].

2.6 Long- and Short-term User Representations (LSTUR)

Long- and short-term user interests were effectively addressed in LSTUR by using two separate encoders. This method captures the dynamic nature of user preferences, allowing the recommendation system to adapt to recent user behavior while considering historical interactions. LSTUR uses a GRU-based model to handle short-term interests and an embedding-based model for long-term interests, integrating both to provide balanced recommendations [LSTUR].

2.7 Neural News Recommendation with Attentive Multi-View Learning (NAML)

NAML learns from different views of news information, including title, body, and entities, using attentive mechanisms. This multi-view approach enables the model to integrate various aspects of news articles, leading to more comprehensive recommendations. By employing an attention mechanism, NAML can weigh the importance of different views, tailoring recommendations more closely to user interests [NAML].

2.8 Graph Neural Networks in Recommendations

The integration of GNNs in recommendation systems has shown promising results in capturing high-order connectivity patterns. GraphRec [GraphRec2019] is one of the notable works in this area, utilizing user-item interaction graphs to enhance recommendation performance. GNN-based methods can propagate information across the graph, capturing intricate relationships and providing richer user and item representations. These methods are particularly effective in scenarios where interactions are sparse, as they can infer latent connections through multi-hop propagation.

2.9 Enhanced Representation Learning

The use of pre-trained language models like BERT [BERT2018] for news encoding has revolutionized text representation in recommendation systems. These models capture deep contextual information from text, significantly improving the semantic understanding of news articles. BERT uses a transformer-based architecture to generate embeddings that reflect the context of words within a sentence. When combined with user encoding strategies that aggregate interaction histories, as seen in GERL, the result is a more accurate and personalized recommendation system. This approach allows for capturing nuances in user preferences and news content, leading to higher recommendation relevance.

3 Methodology

3.1 Graph Enhanced Representation Learning (GERL) Framework

In this work, we largely reproduce the exact method proposed by the original authors, taking the code provided by the unofficial GitHub repository (cite github link here) as the foundation for our work. The GERL framework is designed to capture complex user-news interactions by constructing a user-news bipartite graph and leveraging graph neural networks (GNNs) for representation learning. The framework, as described by the original authors, operates as described below:

3.1.1 Graph Construction. We construct a bipartite graph where nodes represent users and news articles. Edges between user and article nodes indicate user interactions with news articles, such as clicks or views. This graph captures both the user-item interactions and the content features of the news articles.

3.1.2 Graph Neural Networks. The GERL framework employs graph convolutional networks (GCNs) to propagate information across the graph and learn robust user and news representations. The GCNs aggregate information from a node's local neighborhood, allowing the model to capture higher-order connectivity patterns in the graph.

The model uses two layers of GCNs:

- **1-hop information:** This captures immediate interactions between users and news articles.
- **2-hop information:** This captures interactions through intermediate nodes, effectively capturing the influence of other users on a given user's preferences.

The node representations are updated by aggregating features from neighboring nodes, which helps in learning both user and article embeddings that encapsulate the underlying interaction patterns.

3.1.3 Model Architecture. The general outline of the GERL framework's architecture can be described as follows:

- **Embedding Layer:** This layer generates the initial embeddings for users and articles based on the features related to them.
- **Graph Convolutional Layers:** Two layers of GCNs that are used to propagate and aggregate information from the neighboring nodes.
- **Prediction Layer:** A fully connected layer that utilizes the learned embeddings to predict user preferences for news articles.

The model is then trained to minimize the cross-entropy loss between the predicted and actual user interactions with news articles.

3.2 Ekstra Bladet News Recommendation Dataset (EB-NeRD)

Unlike the work performed by the original authors, this implementation operates with the Ekstra Bladet News Recommendation Dataset (EB-NeRD), a large-scale dataset collected from user behavior logs at the Danish news website, Ekstra Bladet. The dataset contains user interaction logs, news article information, and several precomputed features.

3.2.1 Dataset Description. The EB-NeRD dataset comprises several main components:

- **behaviors.parquet:** Contains seven days of impression logs, detailing user interactions with news articles.
- **history.parquet:** Includes 21 days of user click histories prior to the behavior logs, providing a historical context for user interactions.

- **articles.parquet:** Provides detailed information about the news articles, including titles, abstracts, bodies, categories, and additional features like named entity recognition tags and article embeddings.
- **artifacts.parquet:** Contains precomputed embeddings for articles and images, using models like multilingual BERT, RoBERTa, and a proprietary contrastive-based model.

3.2.2 Model Training. This version of the GERL model is trained using the training set of the EB-NeRD dataset. The behaviors.parquet file is used to construct the user-news graph, whilst the history.parquet file provides historical context for user interactions. The articles.parquet file is used to incorporate article content features into the model. The training process optimizes the model parameters to minimize the cross-entropy loss, so that the learned embeddings accurately predict user preferences for news articles.

3.2.3 Evaluation. For evaluation, we use the test set of the EB-NeRD dataset. The performance of the GERL model is assessed using the following metrics:

- **Group Area Under the Curve (group_AUC):** Measures the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance.
- **Mean Reciprocal Rank (mean_MRR):** Evaluates the rank of the first relevant item, with a higher value indicating better performance.
- **Normalized Discounted Cumulative Gain at rank 5 (NDCG@5):** Evaluates the ranking quality considering the top 5 recommendations, taking into account the position of relevant items.
- **Normalized Discounted Cumulative Gain at rank 10 (NDCG@10):** Similar to NDCG@5, but considers the top 10 recommendations.

4 Experiments and Results

4.1 Experiment 1: Document vs. word embeddings

In this experiment, we compare the performance of EB-NeRD pre-trained document embeddings (word2vec, Facebook Roberta, Google Bert Multilingual) against FastText pre-trained Danish word embeddings coupled with the Transformer title encoder as used in the original GERL paper. We initialize a learnable parameter matrix with the pre-trained EB-NeRD document embeddings and use these representations directly as input to the self-attention layers, bypassing the Transformer title encoder for Neighbor News, Clicked News, and Candidate News. For the word embeddings, we adhere to the architecture described in the original paper.

Model Configuration	group_auc	mean_mrr	ndcg@5	ndcg@10
Google Bert Multilingual doc-emb	0.4987	0.3163	0.3541	0.4368
word2vec doc-emb	0.5059	0.3256	0.3637	0.444
Facebook Roberta doc-emb	0.5443	0.3441	0.3881	0.4636
FastText word-emb + Transformer	0.5412	0.3454	0.3889	0.4644
NRMS Baseline	0.547	0.341	0.375	0.454

Table 1. Comparison of word2vec, Facebook Roberta, and Google Bert Multilingual pre-trained document embeddings from the EB-NeRD dataset, and FastText Danish word embeddings coupled with the original Transformer title encoder architecture on the EB-NeRD Small dataset.

As shown in Table 1, the FastText Danish word embeddings with the Danish word tokenizer and the original Transformer title encoder architecture outperform the word2vec document embeddings across all metrics. Specifically, the `group_auc` improves from 0.5059 to 0.5412, the `mean_mrr` increases from 0.3256 to 0.3454, the `ndcg@5` improves from 0.3637 to 0.3889, and the `ndcg@10` improves from 0.444 to 0.4644. The Facebook Roberta document embeddings also show significant improvements over the word2vec document embeddings, achieving a `group_auc` of 0.5443, `mean_mrr` of 0.3441, `ndcg@5` of 0.3881, and `ndcg@10` of 0.4636. Furthermore, the Google Bert Multilingual document embeddings perform the worst across all metrics, while the FastText word embeddings, Facebook Roberta document embeddings, and NRMS Baseline demonstrate comparable performance.

These results indicate that if powerful pre-trained document embeddings are available, it is possible to achieve comparable performance while reducing computational costs by omitting the Transformer title encoder. Furthermore, the results underscore the significant influence of the type of pre-trained embeddings on the performance of the GERL model.

4.2 Experiment 2: Ranking one-hops and two-hops by read-time and common clicks

In this experiment, we enhanced the user and news one-hop histories by sorting them based on read time. Specifically, for each user, the articles they spent the longest time reading were prioritized in the user one-hops (up to 50 articles). For the news one-hops, we included users who spent the most time on a given news article (up to 50 users). Additionally, user two-hops were ranked by the number of commonly clicked news articles (up to 15 users), a method originally proposed in the official GERL paper but not implemented in the unofficial GERL version we used, which randomly sampled one-hops and two-hops.

The rationale behind these modifications is to improve the quality of the graph structure. For example, an article with a very short read time may not be relevant, as the user quickly clicked away. By sorting histories in this manner, we aim to ensure that only the most relevant articles are included. Similarly, a user who spends only a few seconds on an article is less likely to be a representative neighbor compared to one who spends several minutes. The underlying assumption is that a longer read time indicates greater user interest. However, this may not always hold true, such as when a user leaves a page open without reading it.

Moreover, ranking user two-hops by common clicks is intuitive, as users who share many commonly clicked articles likely have similar interests and are "closer" neighbors.

4.2.1 EB-NeRD Demo Dataset. As shown in Table 2, using ranked one-hop user and news histories yields a slight improvement in performance across all metrics on the EB-NeRD Demo dataset. The `group_auc` increases from 0.5463 to 0.5488, the `mean_mrr` improves from 0.3464 to 0.3498, the `ndcg@5` improves from 0.3889 to 0.3923, and the `ndcg@10` improves from 0.465 to 0.4679.

FastText word-emb + title encoder	group_auc	mean_mrr	ndcg@5	ndcg@10
Random sampling	0.5463	0.3464	0.3889	0.465
Ranked one- and two-hops	0.5488	0.3498	0.3923	0.4679

Table 2. Performance on the EB-NeRD Demo dataset with and without sorted one-hop user and news neighbors and ranked two-hop user neighbors.

4.2.2 EB-NeRD Small Dataset. In contrast, Table 3 shows a slight decrease in performance when using ranked one-hop user and news histories on the EB-NeRD Small dataset. The `group_auc` decreases from 0.5412 to 0.5392, the `mean_mrr` drops from 0.3454 to 0.3433, the `ndcg@5` decreases from 0.3889 to 0.3863, and the `ndcg@10` decreases from 0.4644 to 0.463.

FastText word-emb + title encoder	group_auc	mean_mrr	ndcg@5	ndcg@10
Random sampling	0.5412	0.3454	0.3889	0.4644
Ranked one- and two-hops	0.5392	0.3433	0.3863	0.463

Table 3. Performance on the EB-NeRD Small dataset with and without sorted one-hop user and news neighbors and ranked two-hop user neighbors.

Overall, the results indicate that while ranking user and news histories and user two-hop neighbors may lead to improvements in some cases, it does not consistently enhance performance across different dataset sizes.

4.3 Experiment 3: Image Embeddings

In this experiment, we try to enhance the GERL model by incorporating EB-NeRD pre-trained image embeddings for each news article to capture visual information alongside textual data. We developed code to index into the learnable image embedding matrix to retrieve the image embeddings for Neighbor News (news two-hops), Clicked News (user one-hops), and Candidate News. These embeddings are then projected down to 128 dimensions using a learnable linear projection.

To effectively utilize these image embeddings, we employ the `SelfAttendLayer` from the base code for both the news image two-hops and the user image one-hops. The purpose of this layer is to attend to the various image embeddings, resulting in a 128-dimensional representation for both news two-hop images and user one-hop images. We then append the news two-hop image representation and the candidate news image embedding to a list of existing news representations, originally containing three elements, and aggregate these five 128-dimensional vectors using summation. Similarly, we add the user one-hop image representation to the list of user representations and aggregate these four 128-dimensional vectors using the same method.

In addition to summation, we introduce alternative aggregation methods for the different user and news representations. These methods include passing the representations through a Multi-Layer Perceptron (MLP) to enable learned aggregation, and utilizing a Multi-Head attention network coupled with a projection layer to project the user or news representation to the appropriate size (128 dimensions). The final step involves taking the dot product between the news and user representations.

However, due to time and compute constraints, particularly the blocking of the super-cluster we utilized, we were unable to fully complete the implementation and evaluation of these enhancements. Consequently, we leave the extension and completion of this work to future research. The current results reflect the performance of the model using the summation method for aggregation.

5 Conclusion

We successfully reproduced the Graph Enhanced Representation Learning (GERL) framework for news recommendation and applied it to the EB-NeRD dataset. Our findings indicate that the type of pre-trained embeddings (e.g., document vs. word embeddings) significantly impacts recommendation performance. Specifically, we demonstrated that EB-NeRD pre-trained Facebook

Roberta document embeddings and pre-trained FastText word embeddings outperform EB-NeRD word2vec and Google Bert Multilingual document embeddings. This suggests that one can reduce computational costs by bypassing the Transformer title encoder in the GERL architecture when pre-trained document embeddings are available.

We also introduced a method to sort user and news histories by read time to refine the graph structure, under the assumption that articles with shorter read times are less relevant. However, this approach did not yield significant improvements, suggesting that further investigation is needed to validate this method across different datasets and settings.

Additionally, we explored the inclusion of image embeddings in the GERL model to capture visual information alongside textual data. We also experimented with alternative aggregation methods, such as Multi-Layer Perceptron (MLP) and Multi-Head attention networks, in addition to summation. Due to time and computational constraints, we were unable to fully evaluate these enhancements, leaving their completion and assessment for future research.

Future research directions could include excluding news and users with read times below a certain threshold (e.g., 10 seconds) from the neighbor formation, as these are likely not relevant. Another promising approach is to incorporate read time and other features (e.g., scroll percentage) into the graph structure, ensuring the model accounts for the time users spend on different articles. For instance, read times could be concatenated with the news article representations obtained from the Transformer before passing them to the self-attention layer, or the representations could be weighted based on read time or scroll percentage.

Received 28 June 2024