

Personalized News Recommendation System Using Graph Neural Networks

Project Report by – Abde Manaaf Ghadiali (G29583342)

Problem Definition and Understanding

Problem Statement

This project focuses on creating a personalized news recommendation system using Graph Neural Networks (GNNs). It aims to suggest news articles tailored to each user's preferences by analyzing their past behavior and the relationships between users and articles. By using GNNs, the system captures complex connections in the data, offering more accurate recommendations than traditional methods.

Understanding of the Domain and Data

Personalized recommendation systems suggest content based on an individual's preferences. News recommendation systems face challenges like rapidly changing news, diverse user interests, and avoiding recommending overly similar content. Graph Neural Networks (GNNs) handle graph-structured data, making them ideal for understanding user-article interactions by analyzing both content (like article features) and relationships (like clicks or shares). This project uses two main datasets: **News Article Data** (including details like article ID, title, category, and metadata) and **User Interaction Data** (tracking user behavior, such as clicked articles, shown articles, and clicks).

Data Preparation and Preprocessing

The raw news dataset is cleaned to extract important fields and ensure consistency. The cleaned dataset includes:

- **News ID:** A unique identifier for each article.
- **Category and Sub-Category:** Broad and specific labels for classifying the article.
- **News Description:** A summary created by combining the article's title and abstract.

The graph is designed to represent relationships between users, articles, categories, and sub-categories, with the following nodes:

- **News IDs:** Represent individual articles.
- **Categories and Sub-Categories:** Represent classification levels for articles.
- **User IDs:** Represent active and new (cold start) users.

Feature embeddings for each node type are created using Sentence Transformers (all-MiniLM-L6-v2 model):

- **News Articles:** Embeddings come from the combined title and abstract.
- **Categories/Sub-Categories:** Embeddings are based on their names.
- **User Nodes:** For Active Users, embeddings are averaged from the articles they interact with while for Cold-Start Users, embeddings are averaged from the most popular articles.

Each node is then characterized by the following features:

- **Node ID:** A unique identifier for the node.
- **Node Type:** Indicates whether the node is a user, news article, category, or sub-category.

- **Node Label:** A descriptive label (e.g., user ID, news ID, category name).
- **Node Feature Embedding:** A vector representation capturing the node's semantic meaning.

Relationships between nodes are represented as edges in the graph:

- **Category-Subcategory Edges:** Connecting sub-categories to their parent categories.
- **Subcategory-News Edges:** Linking news articles to their respective sub-categories.
- **User-News Edges:** Representing interactions between users and the news articles they have read.

This structure captures the hierarchy of the news data and user-article interactions. The graph integrates all node features and edge relationships, forming a complete dataset ready for the Graph Neural Network to learn meaningful patterns.

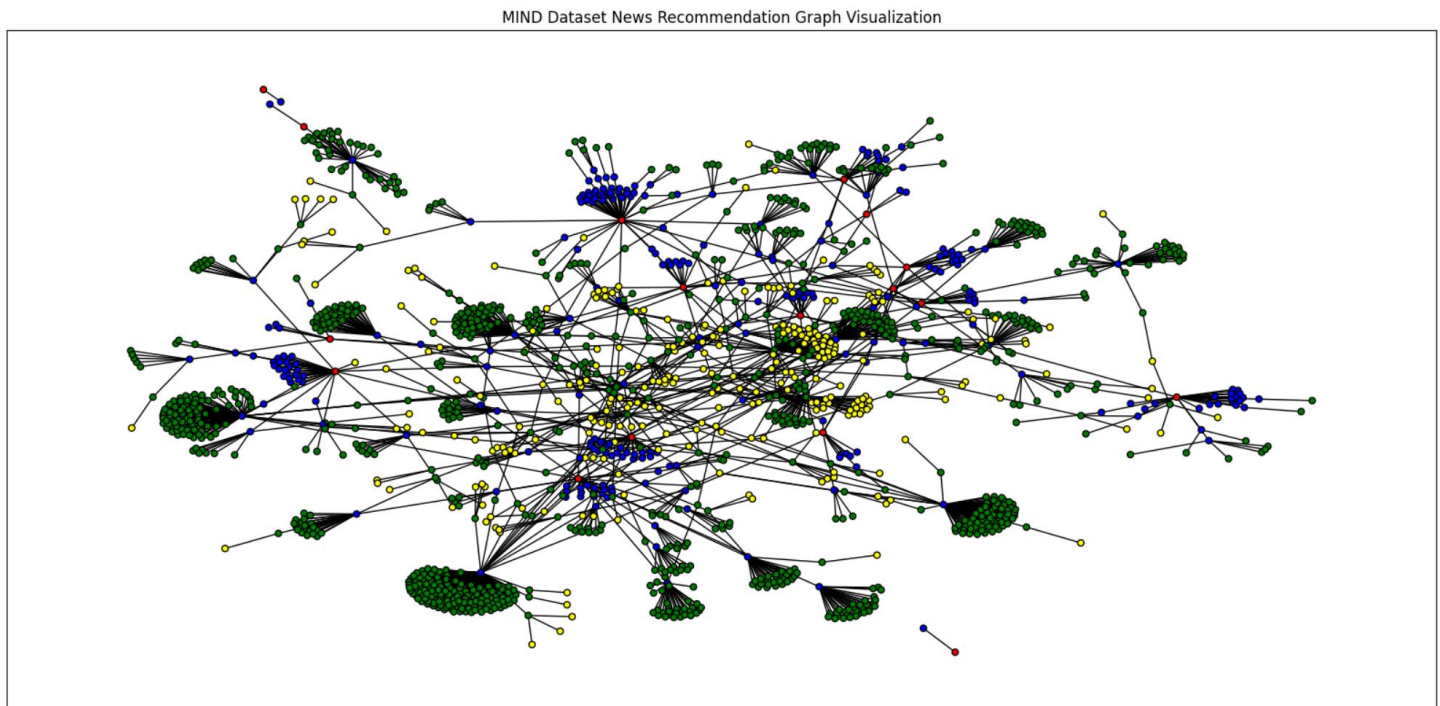


Figure 1 - Sub-Graph for our Graph Structure which shows Categories (Red), Sub-Categories (Blue), News Articles (Green), and User Ids (Yellow)

Model Selection and Development

Choice of Appropriate Model

This project uses three types of Graph Neural Networks (GNNs) to learn node embeddings and model relationships in the graph. Each model is chosen to explore its suitability for personalized news recommendations:

1. **Graph Convolutional Network (GCN):** Aggregates information from neighboring nodes to capture global patterns, making it ideal for understanding network-wide dependencies in user preferences and news relationships.
2. **Graph Attention Network (GAT):** Uses attention mechanisms to prioritize important neighbors during aggregation, which is helpful for tailoring recommendations to individual user interactions.
3. **GraphSAGE:** Scales efficiently to large graphs by sampling neighbors and supports inductive learning, making it effective for handling new users or articles in cold-start situations.

Model Architectures and Training Process

Each GNN model is implemented as a 4-layer network to generate node embeddings. **GCN** uses graph convolution layers to iteratively aggregate features, **GAT** applies multi-head attention to weigh neighboring nodes, and **GraphSAGE** samples neighbors for efficient scaling on larger graphs. The models are trained on graph data to predict links (interactions) between users and news articles using a **Link Predictor Class**. Positive edges (existing links) and random negative edges (non-existing links) are used for training and validation to ensure good generalization.

The models are trained for 100 epochs with binary classification loss (e.g., cross-entropy) to optimize link prediction. Validation involves separate positive and negative edge sets to monitor performance. Once trained, the models produce embeddings that capture node features and relationships, enabling predictions of whether a user will interact with a specific article.

Evaluation and Performance Metrics

Appropriate Choice of Evaluation Metrics

To evaluate the recommendation models, both classification and ranking metrics are used:

Classification Metrics assess how well the models predict user-article interactions:

- **Average Precision (AP):** Balances precision and recall by measuring precision at varying recall levels.
- **Average Recall (AR):** Shows the percentage of relevant links correctly identified, indicating coverage of potential interactions.
- **Average F1 Score:** Combines precision and recall into a single metric, reflecting the balance between false positives and negatives.

Ranking Metrics focus on how effectively the models rank relevant recommendations:

- **Normalized Discounted Cumulative Gain (NDCG):** Evaluates the ranking quality, prioritizing top recommendations (e.g., NDCG@5, @10).
- **Mean Reciprocal Rank (MRR):** Highlights the ranking position of the first relevant recommendation, ensuring the most relevant articles appear at the top.

Analysis of Model Performance with Respect to Metrics

Baseline Models:

Model Type	Training Data			Validation Data		
	Average Precision	Average Recall	Average F1 Score	Average Precision	Average Recall	Average F1 Score
Popularity Based Recommendations (Baseline)	0.00067	0.00366	0.00108	0.00008	0.00109	0.00016
Content-Based Recommendations (Individual Article Profiling)	0.00077	0.01356	0.00139	0.00018	0.00855	0.00034
Content-Based Recommendations (User Profiling)	0.00049	0.00183	0.00068	0.00014	0.00098	0.00023

Table 1 - Baseline Evaluation Metrics

The baseline models perform poorly across all metrics, highlighting their limitations in personalization:

- **Popularity-Based Recommendations:** With an F1 score of 0.00016 on validation, this method fails to account for individual preferences.
- **Content-Based Recommendations (Individual Article Profiling):** Shows slight improvement with a validation F1 score of 0.00034, but personalization remains limited.
- **Content-Based Recommendations (User Profiling):** Performs the worst among content-based methods, with an F1 score of 0.00023, indicating difficulty in handling nuanced user preferences.

Graph Neural Networks:

Model Type	Validation Data		
	Average Precision	Average Recall	Average F1 Score
Graph Convolutional Network	0.048	0.79	0.091
Graph Attention Network	0.048	0.651	0.09
GraphSAGE	0.041	0.485	0.075

Table 2 - Graph Neural Networks Evaluation Metrics

The GNN models outperform the baselines significantly in classification metrics:

- **Graph Convolutional Network (GCN):** Achieves the best results with a validation F1 score of 0.091 and Average Recall of 0.79, demonstrating its ability to capture global and local graph structures.
- **Graph Attention Network (GAT):** Delivers comparable F1 performance (0.09) but with slightly lower recall (0.651), leveraging attention to prioritize important node relationships.
- **GraphSAGE:** Has the lowest F1 score (0.075) among GNNs but still far surpasses the baselines.

Ranking Results:

Model	NDCG@5	NDCG@10	NDCG@15	MRR
Graph Convolutional Network	0.627	0.674	0.707	1
Graph Attention Network	0.63	0.677	0.709	1
GraphSAGE	0.624	0.672	0.705	1

Table 3 - Graph Neural Networks Ranking Metrics

All GNN models achieve nearly identical results, with NDCG@15 exceeding 0.7 and MRR consistently at 1. This indicates their strong ability to rank relevant recommendations at the top, ensuring user-focused precision.

Conclusion

The GNN models significantly outperform the baseline methods in both classification and ranking metrics, validating the graph-based approach for personalized recommendations. Among the GNNs, GCN shows the best balance between recall and ranking, making it the most promising model for this task. GAT’s competitive performance suggests its attention mechanisms may further enhance results in datasets with complex relationships.