Evaluating Graph Transformers for Bluesky Recommendations

Aleck Wu a5wu@ucsd.edu
SangGyu An sgan@ucsd.edu
Mentor: Yusu Wang yusuwang@ucsd.edu
Mentor: Gal Mishne gmishne@ucsd.edu

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Background

Bluesky is a decentralized social media platform similar to Twitter, where the data on the network is open for access to everyone.

Compared to other recommendation datasets like Last.fm or MovieLens, social media interactions can evolve rapidly, requiring models to adapt to changing user preferences over time. Traditional collaborative filtering requires matrix factorization on the entire user-item rating/interaction matrix, requiring periodic recomputation. This may work well on music or movie recommendations, but for a social media site like Bluesky, a scalable way to compute and update embeddings for in real-time is needed. We demonstrate a simple, scalable framework for computing embeddings for a social network, and also show a Graph Transformer-based re-ranking model that refines rankings using both graph structure and temporal information.

Objective

Develop a scalable Bluesky recommendation pipeline that captures user interests in real time while balancing **personalization measured by MRR & Hit@10** and **content diversity measured by ILD@10**. The system should recommend posts from the last 24 hours.

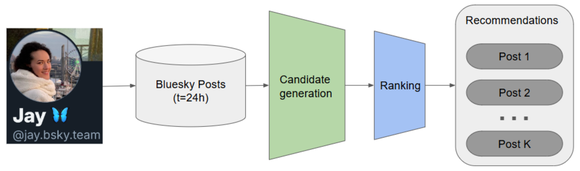

Fig. 1: Recommendation Process

- **Candidate Generation**: Filters a vast pool of posts down to a manageable set of relevant candidates.
- **Re-ranking**: Refines candidate rankings by optimizing for relevance, diversity, and engagement.

Data

Dataset Overview

We collect 6 months of Bluesky data from the beginning of the network and construct a consumer-producer bipartite follow graph. For training, we restrict like interactions to the period from 2023-06-07 to 2023-06-14. Users with ≥ 30 followers are defined as producers.

Statistic	Count
Users	23,765
Posts	365,314
Likes	1,042,739
Follows*	7,301,917

Table 1: Basic Statistics of the Dataset

Methodology

1. Embedding Creation and Candidate Generation:

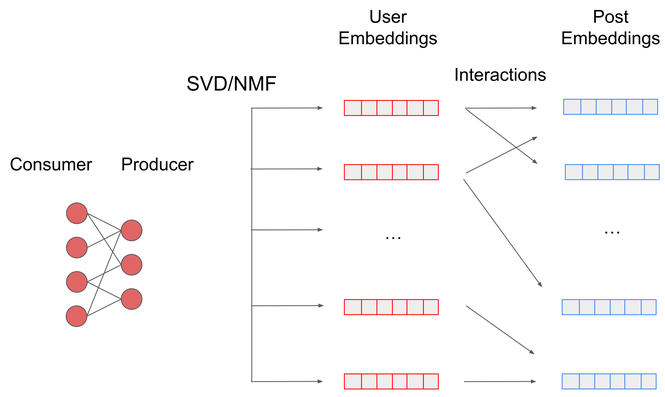

Fig. 2: User and Post Embeddings Generation Process

Instead of computing embeddings for the fast-growing user-post interaction graph, which is computationally intractable, we can use the much smaller and slower-growing consumer-producer follow graph to obtain the embeddings.

- **User Embeddings:**
  Constructed from the consumer-producer bipartite follow graph.
- **Post Embeddings:**
  Constructed in real-time by aggregating past user interactions.

2. GraphRec Pipeline (Re-ranking):

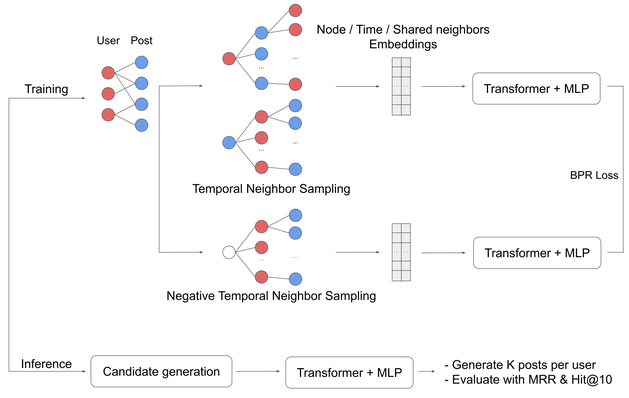

Fig. 3: GraphRec pipeline

- **Temporal Sampling:**
  Take the most recent k user–post interactions for each node. For each positive interaction, draw 4 negatives.
- **Embeddings Construction:**
  Combine node, temporal, and shared-neighbor features to form node embeddings.
- **Transformer + MLP:**
  Capture sequential/contextual patterns and optimize via BPR.

Results & Discussion

Comparison Models

- **Popularity-Based**: Ranks posts by their number of likes. Simple and fast, but lacks personalization and tends to favor older content with accumulated likes.
- **MLP-Based**: Learns user preferences via temporal neighborhood aggregation and MLP layers, incorporating time-sensitive context and user-specific behavior.

Key Metrics

- **Mean Reciprocal Rank (MRR)** (0–1): Captures how soon the true post appears in the ranked list.
- **Hit Rate** (0–1): Fraction of recommendations that include the true post in the top-k results.
- **Intra-List Diversity (ILD)** (0–1): Evaluates how different the recommended items are by measuring cosine similarity on text embeddings.

$$\mathrm{MRR} = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{\mathrm{rank}_u},$$

$$\mathrm{Hit@k} = \frac{1}{|U|}\sum_{u=1}^{|U|}\mathbb{1}(\mathrm{rank}_u \leq k).$$

$$\mathrm{ILD} = 1 - \frac{1}{N(N-1)}\sum_{i\neq j}\mathrm{sim}(i,j)$$

Model	MRR ↑	Hit@10 ↑	ILD@10 ↑	Training (1 epoch) ↓	Inference (per user) ↓
Popularity	0.0345	0.06	**0.6243**	NA	**0.118 sec**
MLP	0.02 ± 0.001	0.03 ± 0.01	0.411 ± 0.008	10:48	0.215 sec
GraphRec*	**0.228 ± 0.005**	**0.235 ± 0.01**	0.55 ± 0.02	18:41	0.283 sec

Table 2: We evaluate it on the last post 14,073 users interacted with between 2023-06-14 and 2023-06-23. Each user receives 2,000 candidate posts.

- GraphRec achieves the highest MRR and Hit@10 but has longer training and inference time, along with lower ILD@10.

How Do Different Models Rank Popular Posts?

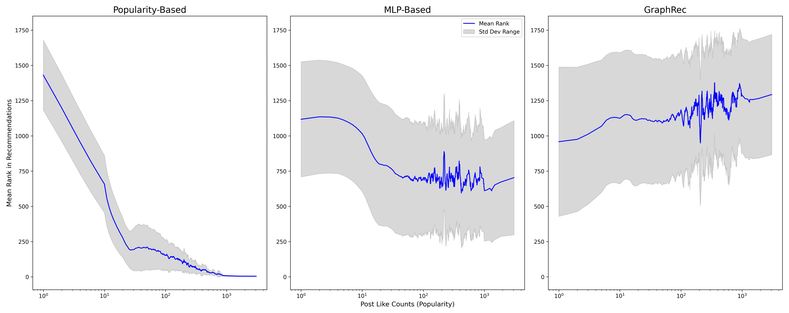

Fig. 4: Mean Rank vs. Popularity Across Different Models

Each subplot shows how the 3 models rank 2,000 candidate posts with mean rank across all users on the y-axis and number of likes a post received on the x-axis.

- **Popularity-Based** (left): Strongly favors popular content, giving poor visibility to less-liked posts.
- **MLP-Based** (middle): Incorporates post features and graph structures but still leans toward popular items.
- **GraphRec** (right): Balances personalization and popularity using user–user and user–post interactions. Unlike other models, it distributes visibility more evenly instead of prioritizing only popular posts, but at a higher computational cost.

Acknowledgments

We thank our mentors and the UCSD DSC Capstone Program for their guidance and support and the authors of DyGLib of which some of our code is based off of.