Temporal Graph Link Prediction Using Combined Heuristics on TGBL-Wiki Dataset

Aviral Gandhi 48-256463

January 29, 2026

1 Introduction

I worked on TGB Link Prediction using heuristic-based methods on the TGBL-Wiki dataset from the Temporal Graph Benchmark (TGB). The task involves predicting the next Wikipedia page a user will edit at timestamp t, given historical co-editing patterns. I re-implement and evaluated a combined heuristic approach that integrates local and global recency with popularity metrics. The heuristic algorithm was based on the algorithm in paper cited in the reference though there was a very slight modification.

2 Task and Dataset Selection

2.1 Task Selection

I selected the **dynamic link property prediction** task from the TGB leaderboard, specifically targeting temporal link prediction where the goal is to predict future edges in an evolving graph structure.

2.2 Dataset: TGBL-Wiki

The TGBL-Wiki dataset represents Wikipedia co-editing activity with the following characteristics:

- **Nodes**: 9,227 (editors and pages)
- **Edges**: 157,474 temporal interactions
- **Time span**: January 1–31, 1970 (Unix epoch)
- **Edge features**: 172-dimensional embeddings
- **Unique (src,dst) pairs**: 18,257

2.3 Temporal Patterns

Key observations from dataset exploration:

- **Degree distribution**: Highly skewed with max out-degree of 1,603 and mean of 17.07
- **Recurrence**: Training split shows 87.71% repeated edges vs. 12.29% new edges

3 Heuristic Methodology

3.1 Combined Heuristics Approach

I implemented a composite scoring function where I combine the four components which are Local Recency(LR), Global Recency(GR) , Local Popularity(LP) and Global Popularity (GP).

3.2 Scoring Function

The composite score uses hierarchical weighting with tie-breaking order LR $\rightarrow$ GR $\rightarrow$ LP $\rightarrow$ GP: So the paper uses explicit tie breaking but I combined it with weights to make it easier for me to implement in the tutorial notebook. So the weights given were consistent to the aim of making the LR most important and so on. So, LR was weighted with 1,000,000 and then other metrics were differed by a factor or 1000, like GR was weighted by 1000 and so on for LP and GP.

3.3 Implementation Details

For each evaluation edge, we: (1) score the positive edge and all negative samples using current bank state, (2) rank candidates by composite score, (3) update bank with observed edge.

4 Baseline Training and Evaluation

4.1 Training Procedure

The bank is built chronologically from 110,232 training edges with no learnable parameters. Normalization constants are computed from the full dataset. But it was made sure that no information about the edges leak into the bank. The Algorithm was then evaluated on the Official Negatives and the metrics Mean Reciprocal Rank (MRR) and Hits@10 were recorded.

4.2 Results

Split	**MRR**	**Hits@10**
Validation	0.8415	0.9025
Test	0.8205	0.8859

Table 1: Performance of combined heuristics on TGBL-Wiki

4.3 Leaderboard Comparison

The Heuristic even with this slight modification reaches the same level as the paper which is currently the second on the leaderboard for TGBL Wiki dataset with a difference in the order of 10^{-3} in the reported MRR.

5 Key Takeaways

5.1 Dataset Insights

- **High recurrence**: 87–90% of edges are repeated interactions, making recency heuristics highly effective
- **Hub dominance**: Top user has 1,603 out-edges; hubs drive global popularity signals

5.2 Heuristic Effectiveness

Local Recency dominates predictions due to strong user-page affinity. The hierarchical weighting (10^6 for LR vs. 10^{-3} for GP) successfully prioritizes recent interactions while using popularity for tie-breaking on cold-start edges.

Normalization is critical: without global constants, raw timestamp magnitudes dwarf count-based features, degrading ranking quality. This was the part which required most debugging as accesing the global max values was a bit tricky.

5.3 Comparison Learnings

While neural temporal graph networks (TGN, TGAT) on the leaderboard incorporate edge features and learn complex temporal patterns, This algorithm get's close to the SOTA performance

1. Recency/popularity capture most signals in repetitive graphs
2. Heuristics serve as strong baselines; improvements require modeling nuanced temporal dependencies

5.4 Limitations

- **Cold-start**: New (user,page) pairs receive zero LR/LP scores
- **Feature blindness**: Ignores rich 172-dim edge embeddings
- **Static weighting**: Manual hierarchy tuning; no learned trade-offs

6 Conclusion

This implementation strengthens the claim that you always made in the class that Simple heuristics are still better performing in this domain compared to the deep modeling approaches so far. This makes it interesting to learn and increases my understanding on what types of task deep model still fail and also highlights the need for a foundational model specfic to this domain. Thank you for teaching this class.

Heuristic Reference:

Cornell, Filip, Oleg Smirnov, Gabriela Zarzar Gandler, and Lele Cao. "On the Power of Heuristics in Temporal Graphs." In Proceedings on "I Can't Believe It's Not Better: Challenges in Applied Deep Learning" at ICLR 2025 Workshops, 37-46. PMLR 296, 2025.