Evaluating Graph Index Performance in Kernel-Based Graph Retrieval

A Study on the Interaction Between Graph Kernel and Graph Indexes

Lizheng Chen
Jade Amandine Liang

April 28, 2025

Background: Graph Kernels Meet Graph Indexes

- Graph retrieval is critical for tasks in bioinformatics, social networks, and cheminformatics.
- Graph kernels (e.g., random walk, Weisfeiler-Lehman) embed substructures of graphs into vector spaces, enabling efficient similarity search.
- For large datasets, fast nearest-neighbor retrieval on these embeddings is essential.
- Graph-based indexes such as HNSW and NSG greatly accelerate similarity search in the embedding space.

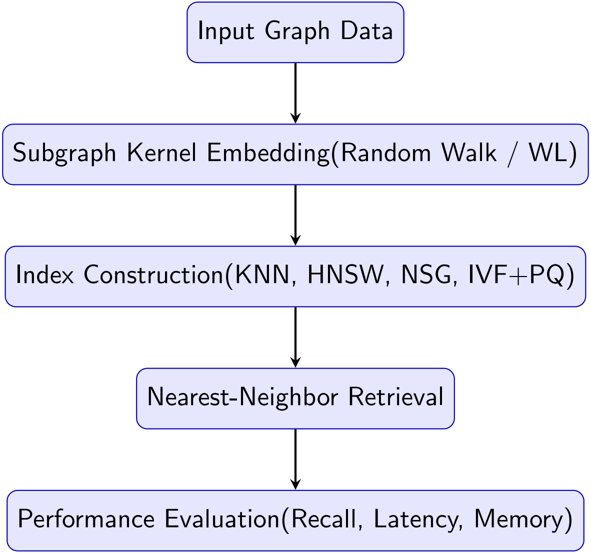
Research Aim

- **Goal:** Evaluate how random-walk-based and WL-based graph kernel embeddings interact with different graph-based indexes in retrieval tasks.
- **Focus:** Large single graph retrieval (subgraph-level embeddings), not many small graphs.
- **Investigated combinations:**
  - **Graph Kernels:** Random Walk Kernel, Weisfeiler-Lehman Kernel
  - **Graph Indexes:** KNN, HNSW, NSG, IVF+PQ
  - **Metrics:** Search recall, memory footprint, latency
- **Why it matters:** Understanding how kernel embeddings interact with index structures is crucial for building efficient retrieval pipelines.

Proposed Contributions and Innovations

- **Joint Evaluation of Kernels and Indexes**:
  Systematically evaluate the compatibility of subgraph-level graph kernels with graph indexing methods.
- **Benchmark on Large Graphs**:
  Focus on the LiveJournal social network dataset, with supplemental exploration of OGB datasets.

Pipeline Overview



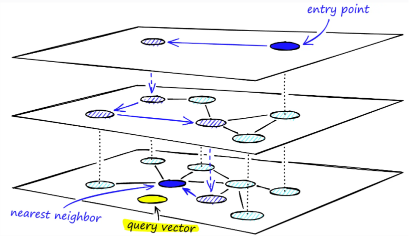
Relevant Works

- Krzysztof and Kochut: [1] – Survey on Random Walk-based Graph Embeddings.
- Fu et al. (2018): [2] – NSG: a high-recall, sparse proximity graph for fast ANN search.
- Johnson et al. (2019): [3] – FAISS library and IVF+PQ techniques.
- Malkov and Yashunin (2018): [4] – HNSW: small-world proximity graph with hierarchical search.

Scope of the Study

- **Graph-based indexes evaluated:**
  - **HNSW** (Hierarchical Navigable Small World)
  - **NSG** (Navigating Spreading-out Graph)
  - **KNN** (Brute-force search)
  - **IVF+PQ** (Inverted File with Product Quantization)

Graph Structures: HNSW and NSG



The search process through the multi-layer structure of an HNSW graph.

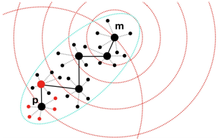


Figure 5: An illustration of the candidates of edge selection in NSG. Node p is the node to be processed, and m is the Navigating Node. The red nodes are the k nearest neighbors of node p. The big black nodes and the solid lines form a possible monotonic path from m to p, generated by the search-and-collect routine. The small black nodes are the nodes visited by the search-and-collect routine. All the nodes in the figure will be added to the candidate set of p.

Figure: (Left) HNSW multi-layer graph; (Right) NSG flat navigable graph

Datasets

- **Primary Dataset:**
  - **LiveJournal (soc-LiveJournal1)** – Real-world social network with 4.8M nodes and 69M edges. `snap.stanford.edu/data/soc-LiveJournal1.html`
- **Additional Datasets (Optional):**
  - **OGB** (Open Graph Benchmark) datasets:
  - ogbn-arxiv, ogbl-collab, ogbn-products `ogb.stanford.edu`

Evaluation Metrics

- **Recall@k**: Fraction of ground-truth nearest neighbors retrieved.
- **Latency**: Average time per query.
- **Memory Footprint**: RAM usage during index build and query.

Datasets and Tools

- **Graph Embedding Tools**:
  - **GraKeL**: for Random Walk Kernel, Weisfeiler-Lehman Kernel
- **Graph Index Libraries**:
  - **Faiss**, **HNSWLib**, **EFANNA2e/NMSLIB** (for NSG)

Ablation Study Design

- **Embedding Quality Factors**:
  - Graph kernel choice: Random Walk vs Weisfeiler-Lehman
  - Walk parameters (length, restart probability)
  - Embedding dimension size (e.g., 64, 128, 256)
- **Indexing Hyperparameters**:
  - HNSW: efConstruction, M
  - NSG: L, pruning settings
  - IVF+PQ: number of clusters, quantization precision

Project Timeline

- **Week 8–9**:
  - Survey graph kernel and index literature
  - Prepare LiveJournal and OGB datasets
- **Week 10–12**:
  - Embedding with Random Walk and WL kernels
  - Build and tune graph indexes (KNN, HNSW, NSG, IVF+PQ)
  - Measure recall, latency, memory
- **Week 12–13**:
  - Visualize trade-offs
  - Write final report and polish results

End

Thank you!

A survey on the recent random walk-based methods for embedding knowledge graphs.
arXiv:2406.07402v2 [cs.LG], 2024.

Yifan Fu, Chao Li, Yixuan Wang, Xiaogang Wang, and Hongyuan Zha.
Fast approximate nearest neighbor search with the navigating spreading-out graph.
arXiv preprint arXiv:1707.00143v9, 2018.

Jeff Johnson, Matthijs Douze, and Hervé Jégou.
Billion-scale similarity search with faiss.
IEEE Transactions on Big Data, 7(3):535–547, 2019.

Yu A Malkov and D A Yashunin.
Efficient and robust approximate nearest neighbor search using hnsw.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4):824–836, 2018.