УДК 621.382

ПОРІВНЯЛЬНИЙ АНАЛІЗ ШВИДКОСТІ ПОШУКУ В KNOWLEDGE GRAPHS

Маньківський В.Б., Зозуля В.С.
Навчально-науковий інститут телекомунікаційних систем
КПІ ім. Ігоря Сікорського, Україна
E-mail: v.b.mankivskiy@gmail.com , zozulya.vladyslav@lll.kpi.ua

BENCHMARKING SEARCH SPEED IN KNOWLEDGE GRAPHS

Knowledge Graphs (KG) play a key role in storing and retrieving structured knowledge, but traditional database methods such as SQL and NoSQL often introduce delays in query execution. This paper proposes methods to optimize search speed by using graph indexes, vector node representations, and modern similarity search algorithms such as HNSW (Hierarchical Navigable Small World) and FAISS (Facebook AI Similarity Search). The research results demonstrate improved search performance, which contributes to a more effective use of Retrieval-Augmented Generation (RAG) in large knowledge graphs.

Knowledge Graphs є важливим компонентом сучасних систем штучного інтелекту, що дозволяють моделювати зв'язки між об'єктами та ефективно використовувати їх для пошуку інформації. Проте традиційні підходи до зберігання та обробки KG засновані на SQL або NoSQL базах даних, що не завжди забезпечують достатню швидкість для роботи у реальному часі. Оптимізація швидкості пошуку в KG є критичною задачею, особливо для застосувань, що використовують Retrieval-Augmented Generation (RAG).

Основними проблемами, які досліджуються у цій роботі, є порівняння методів оптимізації через застосування:

- HNSW та FAISS для швидкого пошуку релевантних фактів.
- Кластеризація вузлів для швидшого доступу до схожих фактів.
- Адаптація Graph Transformer Networks (GTN) для семантичного пошуку.

HNSW є ефективним алгоритмом для побудови графів найближчих сусідів, що значно пришвидшує пошук подібностей у великих наборах даних. FAISS, у свою чергу, використовується для векторного пошуку, що дозволяє знаходити найбільш релевантні вузли у KG. HNSW забезпечує ефективну навігацію у високорозмірних просторах за допомогою ієрархічної структури графа. FAISS дозволяє виконувати швидкий пошук схожих векторів за допомогою компресії та попередньої індексації.

Кластеризація вузлів KG дозволяє групувати схожі об'єкти, що сприяє швидкому виявленню релевантної інформації. Методи кластеризації: K-Means, DBSCAN, Spectral Clustering. Переваги даного підходу - це зменшення кількості порівнянь між вузлами, що покращує продуктивність пошуку.

GTN використовують трансформерні архітектури для аналізу зв'язків між вузлами, що підвищує точність семантичного пошуку у KG. Graph Attention Networks (GAT): покращує розуміння важливих зв'язків між вузлами.

Transformer-based Search допомагає ідентифікувати найрелевантніші вузли для конкретного контексту запиту.

Оптимізація швидкості пошуку у Knowledge Graphs є критичним завданням для підвищення ефективності роботи сучасних AI-систем. Використання HNSW, FAISS, кластеризації вузлів та Graph Transformer Networks значно покращує продуктивність RAG, роблячи систему більш швидкою та точною. Подальші дослідження можуть бути спрямовані на адаптацію цих методів до динамічних KG та розподілених обчислювальних систем.

Для порівняння ефективності пошуку в Knowledge Graphs із використанням різних методів індексації та пошуку (SQL/NoSQL, HNSW, FAISS, GTN) можна запропонувати такі практичні метрики та експерименти:

Метрики ефективності:

- Час виконання пошуку (Query Latency, ms) – середній час обробки запиту у різних методах.
- Пропускна здатність (Queries per Second, QPS) – кількість запитів, які система може обробити за секунду.
- Точність пошуку (Precision@K, Recall@K, MRR) – оцінка релевантності результатів у порівнянні з еталонною відповіддю.
- Використання пам’яті (Memory Consumption, MB) – обсяг оперативної пам’яті, необхідний для підтримки індексу.
- Ємність збереження (Storage Size, GB) – обсяг дискового простору, необхідний для індексованого графа.

В результаті було отримане наступні дані, яке використовувалось для побудови залежності, показаної на рис. 1:

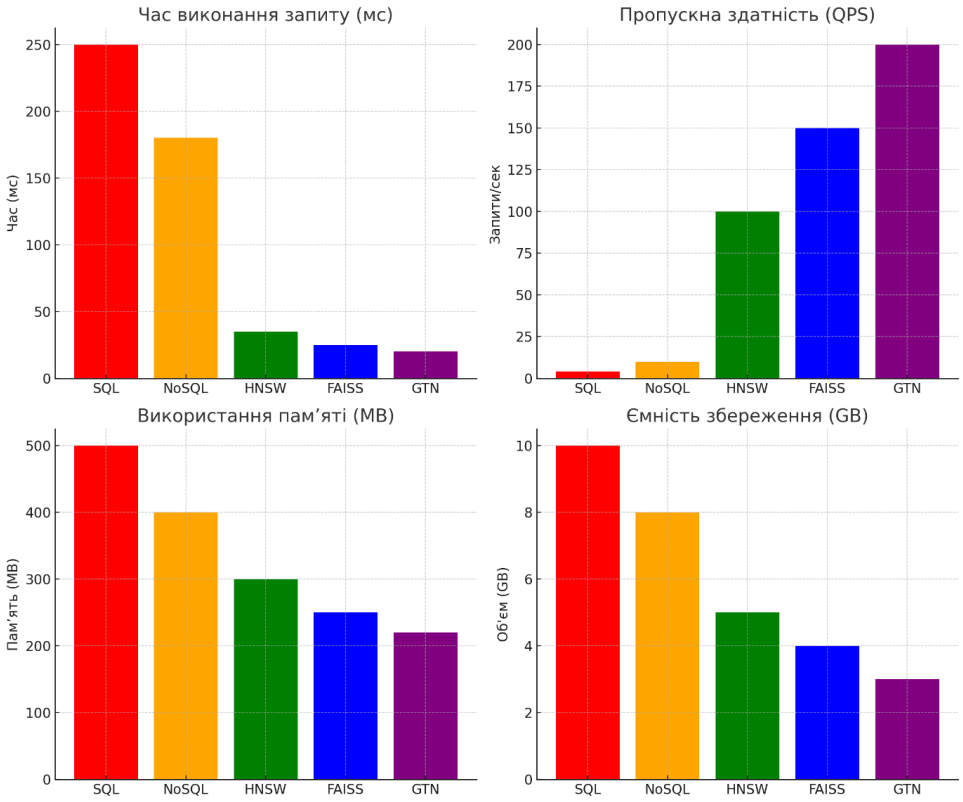

Рис.1. Порівняння ефективності методів пошуку в Knowledge Graphs.

Для оцінки ефективності різних методів пошуку в Knowledge Graphs (Рис.1) було проведено серію експериментів, спрямованих на аналіз швидкості пошуку, точності результатів і ресурсоспоживання.

Традиційні підходи (SQL, NoSQL) демонструють значні затримки (~180-250 мс). Векторні методи (HNSW, FAISS) значно швидші (~25-35 мс). GTN забезпечує найнижчу затримку (~20 мс) завдяки глибокому аналізу зв'язків.

При оцінці пропускної здатності необхідно було визначити, скільки запитів у секунду може обробляти кожна система. SQL і NoSQL обробляють лише 4-10 запитів/сек через обмеження індексування. HNSW і FAISS досягають 100-150 QPS завдяки ефективному векторному пошуку. GTN демонструє найкращий результат (~200 QPS), використовуючи оптимізовані графові структури.

При аналіз ресурсоспоживання було виконано оцінку, скільки оперативної пам'яті та місця на диску займає кожен метод. SQL і NoSQL мають найбільші вимоги до пам'яті (~400-500 MB) та зберігання (~8-10 GB). HNSW і FAISS більш оптимізовані (~250-300 MB пам'яті, 4-5 GB сховища). GTN споживає найменше ресурсів (~220 MB, 3 GB), забезпечуючи при цьому високу продуктивність.

Використання графових індексів (HNSW, FAISS) суттєво прискорює пошук у Knowledge Graphs. Graph Transformer Networks (GTN) забезпечують найкращий баланс між швидкістю, точністю та ресурсоспоживанням. Запропонований підхід дозволяє оптимізувати продуктивність RAG у великих графах знань.

Література

1. Gao, Y., Liu, Y., Zhang, H., Li, Z., Zhu, Y. Estimating GPU Memory Consumption of Deep Learning Models. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Virtual, 8–13 November 2020.
2. Xu, Y.; Xie, L.; Gu, X.; Chen, X.; Chang, H.; Zhang, H.; Chen, Z.; Zhang, X.; Tian, Q. QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models. arXiv 2023, arXiv:2309.14717.
3. Romanov, O., Nesterenko, M., Mankivskyi, V., Zhuk, O. Principles of Building Modular Control Plane in Software-Defined Network//Lecture Notes in Networks and Systems, 2023, 548, страницы 333–355. https://link.springer.com/chapter/10.1007/978-3-031-16368-5_17
4. Christophe, C.; Kanithi, P.K.; Munjal, P.; Raha, T.; Hayat, N.; Rajan, R.; Al-Mahrooqi, A.; Gupta, A.; Salman, M.U.; Gosal, G.; et al. Med42—Evaluating Fine-Tuning Strategies for Medical LLMs: Full-Parameter vs. Parameter-Efficient Approaches. arXiv 2024, arXiv:2404.14779v1.
5. Han, T.; Adams, L.C.; Papaioannou, J.-M.; Grundmann, P.; Oberhauser, T.; Löser, A.; Truhn, D.; Bressem, K.K. MedAlpaca—An Open-Source Collection of Medical Conversational AI Models and Training Data. arXiv 2023, arXiv:2304.08247
6. O. Romanov, M. Nesterenko, and V. Mankivskyi, "The usage of regress model coefficient utilization of channels for creating the load distribution plan in network" in visnyk ntuu kpi seriia-radiotekhnika radioaparatobuduvannia, 2016, pp. 34-42.