Enhancing Retrieval-Augmented Generators with Symbolic Query Definitions

Filippo Bianchini[1], Marco Calamo[1], Marco Console[1], Massimo Mecella[1] and Laura Papi[1]

[1]*Sapienza, Università di Roma*

Abstract

Retrieval-Augmented Generation is as a paradigm that integrates the parametric knowledge of a Large Language Model with external non-parametric information through a retrieval component. While RAGs have shown strong performances on several tasks, their answers may still be prone to errors as of the approximate nature of the retrieval process they utilize. In this paper, we argue that the output of a dense retriever can be interpreted as a set of noisy samples that could be retrieved via a query that it is implicitly executed over the data. A symbolic representation of a query that leads to the same results as the set obtained via the dense retriever can then be reconstructed from the samples and used to compute the final augmented result. We instantiate this idea in a concrete retrieval architecture, which we call Symbolic-Assisted Dense Retriever, and present a preliminary use case scenario and experimental evaluation.

1. Introduction

Recent years have seen a surge of AI systems capable of processing *unstructured natural-language requests*, matching them against their *prior knowledge*, and returning a *meaningful multimodal answer*. These abilities are usually achieved using a *large language model* (LLM) [1], i.e., a large generative machine learning model trained to produce answers from natural language input.

Despite their impressive linguistic skills, LLMs present several well-known limitations that hamper their adoption in mission-critical tasks: a propensity to *hallucinate*, i.e., mention false information in their answers; a reliance on *outdated knowledge*, due to a training process too costly to repeat every time relevant information changes; and an *intrinsic opaqueness* of their reasoning, rooted in their large neural architectures.

The AI architecture known as *Retrieval-Augmented Generation* (RAG) has recently been introduced to overcome these limitations [2]. In simple words, a RAG system aims to combine the parametric knowledge stored in an LLM with the non-parametric knowledge stored in an external data storage system. To this end, the standard RAG pipeline augments an LLM with a *dense retriever*, i.e., a system that uses the user request to select relevant items from an external data source, typically by ranking them according to similarity [3]. More concretely, a RAG consists of three components: an LLM $\mathscr{L}$; a data source D, containing documents; and a *dense retriever* Ret. For a given user request σ, the RAG runs Ret over D and σ to extract the set A of relevant documents, combines A with σ to obtain an augmented request σ', and returns the result of asking σ' to $\mathscr{L}$. This way, the linguistic skills of $\mathscr{L}$ are applied directly to the relevant information Ret extracts from D, allowing easier access to updated information, fewer hallucinations, and a clearer view of the reasoning process.

While RAGs have proved effective on multiple tasks, they still suffer from several limitations [4], chief among them the approximate nature of the retrieval component. Dense retrieval is usually achieved by some embedding-based procedure that ranks the documents in D by their similarity to the user request σ and returns the top k documents in the ranking, for some parameter $k \in \mathbb{N}$ (top-k similarity). When the data source is structured or partially structured, e.g., in the case of knowledge graphs [5], the embedding procedure may also take the data's structure into account, or try to produce a query

Joint Workshop on Statistics and Knowledge Integration for Logic, Learning, Ethical Decisions, and LLMs (SKILLED-LLMs 2026), co-located with FLoC 2026, Lisbon, Portugal

bianchni@diag.uniroma1.it (F. Bianchini); calamo@diag.uniroma1.it (M. Calamo); console@diag.uniroma1.it (M. Console); mecella@diag.uniroma1.it (M. Mecella); papi@diag.uniroma1.it (L. Papi)

0009-0006-3278-9853 (F. Bianchini); 0009-0006-2602-9604 (M. Calamo); 0009-0004-5526-019X (M. Console); 0000-0002-9730-8882 (M. Mecella); 0009-0003-2281-9500 (L. Papi)

expression in some language via generative AI methods [6, 7]. All such methods are known to yield a noisy retrieval process, in terms of both false negatives, i.e., missed relevant items, and false positives, i.e., retrieved irrelevant ones.

Because of these issues, the output of a dense retriever should not be regarded as the exact answer *to the query implicitly defined by the user request*, but rather as a set of noisy samples of it. Under this assumption, these samples could be used to *synthesize a symbolic representation of the underlying query*, which could then either explain the retriever's output or improve it by being executed directly over the data source. The literature has studied the problem of reconstructing symbolic query representations from samples in various settings, e.g., query definability [8], query-by-example [9], and query fitting [10].

The remainder of this paper presents Symbolic-Assisted Dense Retrievers. Section 2 gives a formal account of symbolic-assisted dense retrievers in terms of their black-box components. Section 3 introduces a concrete use case where our architecture is embedded in a RAG system and reports a preliminary experimental evaluation. Finally, Section 4 concludes and discusses future work.

The anonymized repository, with full experiment results, is available on Zenodo at: https://zenodo.org/records/19891828.

2. Symbolic-Assisted Dense Retriever

From the technical standpoint, our retrieval architecture is based on the idea that the process performed by a *dense retriever* can be reconstructed in terms of the expression of a symbolic query language. Such an expression can then be further refined and processed to improve the results of the retriever. The goal of this section is to present a brief formal account on the notions underlying this idea and use them to provide a formal definition of our retrieval architecture.

To this end, we start by formalizing the notion of graph-shaped data that our architecture assumes. We fix two disjoint and countable sets of symbols Γ and Δ that we call the set of *attribute names* and the *text alphabet*, respectively. As customary, we use Δ^* for the set of all strings over Δ, i.e., finite sequences of symbols in Δ. A *textual attribute* is a pair $\alpha = (A, v)$ where $A \in \Gamma$ (the key of α) is an attribute name and $v \in \Delta^*$ (the value of α) is a string. We will use $\mathbf{A}$ for the set of all such pairs.

In the context of this work, a *Text-Attributed Graph* (TAG) [5] is a tuple of the form $G = \langle N, E, \lambda_N, \lambda_E \rangle$ where N is a set called the *set of nodes* of G, $E \subseteq N \times N$ is called the *set of edges* of G, and $\lambda_N : N \to 2^{\mathbf{A}}$ and $\lambda_E : E \to 2^{\mathbf{A}}$ are called *node-attribute* and *edge-attribute* functions, respectively. We further assume that, for every TAG defined as above, the sets N, $\lambda_N(n)$, for each $n \in N$ and $\lambda_E(e)$, for each $e \in E$ are finite. Intuitively, TAGs are a simplified form of property graphs [11] in which every attribute is a string of text and every ordered pair of nodes is connected by at most one edge.

A central notion in our architecture is that of *queries for TAGs*. Let $\mathbf{G}$ be the set of all possible TAGs, a *TAG query* is simply a function $q : \mathbf{G} \to \mathbf{G}$. Given a TAG G and a query q, we call the TAG $q(G)$ the *answer for q over G*. Our architecture will mostly deal with queries that ask for specific sets of nodes from a TAG. A *node query* is simply a TAG query q such that, for every $G = \langle N, E, \lambda_N, \lambda_E \rangle$, $q(G) = \langle N_{ans}, \emptyset, \lambda_{ans}, \emptyset \rangle$ with $N_{ans} \subseteq N$ and $\lambda_{ans}(n) = \lambda_N(n)$, for each $n \in N$. TAG queries can be specified either procedurally, i.e., via an algorithm that defines their results, or declaratively, i.e., via a formal language whose expressions define TAG queries. In the latter case, we talk about a *query language*. A query language is a possibly infinite set $\mathbb{Q}$ such that, for each $\varphi \in \mathbb{Q}$ there is an associated TAG query $[\![\varphi]\!]^{\cdot}$ that we call the *query semantics*.

To obtain a query language for TAGs, we adapt the standard notion of *graph patterns* [12] to TAGs. A *unary TAG pattern* (simply, TAG pattern, TP) is a pair $\langle P, a \rangle$ where P is a TAG and a is a node in P, while a *union of TAG patterns* (UTP) is a finite set of TAG patterns. Given a set $\mathscr{A} \subseteq \Gamma$, we say that an UTP Q is *an $\mathscr{A}$-UTP* if $\mathscr{A}$ contains all the attribute names A_Q occurring in the TPs of Q ($A_Q \subseteq \mathscr{A}$). Semantics for (unions of) TAG patterns is given by the notion of matches. Given a TAG $G = \langle N, E, \lambda_N, \lambda_E \rangle$ and a TAG pattern $\pi = \langle P, a \rangle$ with $P = \langle N_P, E_P, \lambda_N^P, \lambda_E^P \rangle$, a *match μ for π over G* is a label-preserving homomorphism from P to G, i.e., a mapping $\mu : N_P \to N$ such that $(\mu(u), \mu(v)) \in E$, for each $(u, v) \in E_P$; $\lambda_N^P(u) \subseteq \lambda_N(\mu(u))$,

for each $u \in N_P$; and $\lambda_E^P((u,v)) \subseteq \lambda_E(\mu(u), \mu(v))$, for each $(u,v) \in E_P$. A *result for a match μ for π over G* is the node $n \in N$ of G such that $\mu(a) = n$, and a *result for a union of TAG patterns Q over G* is a result of a match μ for a $\pi \in Q$ over G. Finally, *the answer for Q over G* is the TAG $[\![Q]\!]^G = \langle N_{ans}, \emptyset, \lambda_{ans}, \emptyset \rangle$ where N_{ans} is the set of all results for Q over G and λ_{ans} is the restriction of λ_N to N_{ans}. UTPs can be easily encoded into standard query languages such as Cypher [11].

Procedural query specifications in our architecture will be given by *dense retrievers*. In our context, we understand a dense retriever simply as a function $\mathsf{Ret} : \Delta^* \times \mathbf{G} \to \mathbf{G}$ that maps every pair of string $\sigma \in \Delta^*$ and TAG G into a TAG G' understood as the answer for the *question* σ[1] over the TAG G. To present our architecture, we abstract away from the details of how the retriever is trained and focus only on the representation we have just presented.

An important problem for our architecture is to reconstruct, as accurately as possible, the query defined by a dense retriever using a symbolic query. In our context, a *TAG query synthesis technique for a query language* $\mathbb{Q}$ is a function $\mathsf{Synth} : \mathbf{G} \times \mathbf{G} \to \mathbb{Q}$ such that, given a pair $(G, q(G))$ where G is a TAG and q is a TAG query, $\mathsf{Synth}(G, q(G)) \in \mathbb{Q}$ is a query expression such that $[\![\mathsf{Synth}(G, q(G))]\!]^G$ best approximates $q(G)$, according to some criteria. For our experiments, we will use a simple query synthesis technique for UTPs. Given the result $q(G)$ of a node query q over a TAG G and a set of attributes $\mathscr{A}$, we say that an $\mathscr{A}$-UTP Q is a complete $\mathscr{A}$-UTP-approximation for q over G if $[\![Q]\!]^G$ contains all the nodes of $q(G)$. Usually, we will be interested in those UTPs that best approximate the original query. We say that an $\mathscr{A}$-UTP Q is a minimally complete $\mathscr{A}$-UTP-approximation for a query q if there exists no complete $\mathscr{A}$-approximation Q' for q whose nodes are strictly contained in those of Q. Intuitively, Q is the $\mathscr{A}$-UTP that can capture every node in $q(G)$ adding as little extra noise as possible.

With the notions presented so far we are now ready to introduce the formal foundations of our architecture that we call *Symbolic-Assisted Dense Retriever*.

Definition 1. *Let* $\mathsf{Ret} : \Delta^* \times \mathbf{G} \to \mathbf{G}$ *be a dense retriever and let* $\mathsf{Synth} : \mathbf{G} \times \mathbf{G} \to \mathbb{Q}$ *be a TAG query synthesis technique. A Symbolic-Assisted Dense Retriever based on* Ret *and* Synth *is simply a function* $\mathsf{Ret}_{\mathsf{Synth}} : \Delta^* \times \mathbf{G} \to \mathbf{G}$ *such that, for each $\sigma \in \Delta^*$ and $G \in \mathbf{G}$, we have*

$$\mathsf{Ret}_{\mathsf{Synth}}(\sigma, G) = [\![\mathsf{Synth}(G, \mathsf{Ret}(\sigma, G))]\!]^G.$$

This approach replaces standard similarity-based retrieval by first using a dense retriever to extract candidate nodes, which are then treated as noisy examples to synthesize a symbolic query (specifically, a minimally complete $\mathscr{A}$-UTP approximation) representing the user's latent intention. We subsequently present a preliminary RAG-based experimental evaluation to assess whether evaluating this inferred symbolic query effectively improves the retriever's recall and the system's overall downstream performance.

3. Use-Case and Experiments

A natural use case for the presented architecture is to employ Symbolic-Assisted Dense Retrievers within RAG systems. Specifically, we are interested in investigating whether improving the retrieval by reconstructing its implicitly defined query can improve the generated answer. To this end, we implement the Symbolic-Assisted Dense Retrieval and present two examples of its application within a RAG system. In both the examples we start by fixing a user question σ, and then describe the full question answering pipeline under three distinct configurations: *i)* the *naive* approach, which directly prompts the generator component with the question σ, avoiding the retrieval step, hence without providing the generator with additional relevant knowledge; *ii)* the *basic retrieval* approach, where retrieval is based solely on embedding similarity; *iii)* the *advanced retrieval* approach, which employs the Symbolic-Assisted Dense Retriever to improve the results of the basic retriever. All approaches generate a response to the initial user question σ, which we compare to assess the benefits of integrating the Symbolic-Assisted Dense

[1]The literature often calls σ a query. We use question here to avoid to overload the term.

Retriever architecture into the RAG use case. We tested our approach using as a TAG G an extended version (120k total nodes) of the Neo4j recommendation dataset[2], containing data about movies, actors, directors, and reviews from IMDb users. To conduct the experiments, we decided to use only open-source models to ensure maximum reproducibility. In particular, we used Gemma 4[3] (E4B) as a LLM for generating the answers. For embedding the graph nodes, we adopted the BGE Flag Embeddings small v1.5 [13] (English version) that maps textual strings to a 384-dimensions latent vector space. To obtain the embedding vectors, we concatenated, with a whitespace in between, all $\lambda_N(n)$ for each node $n \in N$ in a string called `retrieval_text`, excluding only the meta-attributes, such as data of creation or data of last update of the node. For example, for a `Movie` type node, `retrieval_text` includes the plot, the country of origin and the languages. The `retrieval_text` is then encoded using the embedding model and later retrieved using cosine similarity with respect to the embedded user question σ. In this way, the output of the basic retriever $\mathrm{Ret}(\sigma, G)$ consists of the top-k nodes with higher cosine similarity to σ. To evaluate the advanced retrieval approach, we implemented the Symbolic-Assisted Dense Retriever as defined in Section 2. We designed an algorithm that computes $\mathrm{Ret}_{\mathsf{Synth}}(\sigma, G)$ for a given user question σ and a TAG G. Specifically, $\mathrm{Ret}(\sigma, G)$ is implemented as the embedding-based retriever described above, while Synth computes the minimally complete $\mathscr{A}$-UTP approximation of the query implicitly defined by $\mathrm{Ret}(\sigma, G)$. For this purpose, $\mathscr{A}$ is defined as the set of textual attributes that are shared by at least βk nodes among the top-k nodes. Here, $\beta \in [0, 1]$ is a tunable parameter. Intuitively, β defines a threshold that determines whether a textual attribute is sufficiently frequent in $\mathrm{Ret}(\sigma, G)$ to be considered relevant to its implicitly defined query, and thus included in the reconstruction produced by Synth. For these experiments, the minimally complete $\mathscr{A}$-UTP approximation computed by Synth is produced in the Cypher query language, and evaluated over G using Neo4j. As for the execution times, we note that the additional computational cost introduced by our architecture is negligible in practice, with execution times consistently low enough to render detailed timing analysis uninformative.

3.1. Example One

User Question σ_1

What are the best adventure animated movies from Japan?

Below are reported the responses to σ_1, generated under the three different settings described above: *naive*, *basic retrieval*, and *advanced retrieval* approaches. The *naive* approach (Figure 1) produces a generic, hedged answer with no concrete recommendations, reflecting the model's tendency to over-qualify when lacking grounded context.

Naive Approach (no retrieval)

That is a wonderfully massive question, because "best" is incredibly subjective! Japanese animation (Anime) is an enormous, diverse industry that covers every genre imaginable—from whimsical family comedies to complex, dystopian sci-fi epics. [...]

Figure 1: Response generated for the user question σ_1 through the naive approach

On the other hand, the *basic retrieval* approach is able to surface relevant nodes via embedding similarity. However, ranking solely by vector proximity causes the retriever to return lower-rated or niche titles ahead of critically acclaimed ones, consequently affecting the generated response. Condensed versions of the top-k nodes and response are shown in Figure 2; full versions are reported in the Appendix.

Finally, the response generated by the *advanced retrieval* approach (Figure 3) demonstrates how the

[2] https://github.com/neo4j-graph-examples/recommendations
[3] https://ai.google.dev/gemma/docs/core/model_card_4

Basic Retrieval Approach

(a) Top-k Retrieved Nodes

Here we only report the titles of the movies; however, the complete retrieval includes all textual attributes, as shown in the Appendix.

- "The Great Yokai War"
- "One Piece: Strong World"
- "Majokko shimai no Yoyo to Nene"
- "Inuyasha the Movie: Affections Touching Across Time"
- "Bloody Spear at Mount Fuji"

(b) Generated Response

Highly Rated Options:

- **One Piece: Strong World** (2009) IMDb: 7.4
  Action, Adventure, Animation
- **Inuyasha the Movie** (2001) IMDb: 7.1
  Action, Adventure, Animation
- **Majokko shimai no Yoyo to Nene** (2013) IMDb: 6.6
  Adventure, Animation, Comedy
- **The Great Yokai War** (2005) IMDb: 6.3
  Adventure, Family, Fantasy

Figure 2: Basic retrieval approach for σ_1: (a) embeddings-based retrieval result; (b) generated response.

Symbolic-Assisted Dense Retriever is effective in reconstructing the query implicitly defined by σ_1, and is able to retrieve a more complete and informative set of nodes. Specifically, for this experiment we set $\beta = 0.7$. Intuitively, $\mathscr{A}$ contains all the textual attributes that appear in at least 70% of the top-k nodes. The resulting query in this experiment consists of a Cypher query that matches movies whose countries contain "Japan", and which among its genres have "Adventure" (see Appendix for the complete query). This allows the generator to acquire knowledge about higher rated movies, thereby improving its recommendations.

Advanced Retrieval Approach

Here are the top recommendations for *best adventure animated movies from Japan*, categorized by type:

- **Attack on Titan the Movie: The Last Attack** IMDb: 9.1
- **Ramayana: The Legend of Prince Rama** IMDb: 9.1
- **Solo Leveling: ReAwakening** IMDb: 8.7
- **Demon Slayer: Kimetsu no Yaiba - Infinity Castle** IMDb: 8.5

Figure 3: Response generated for the user question σ_1 using the advanced retrieval approach

3.2. Example Two

User Question σ_2

Horror Japanese Movies all time by rating

Similarly to the experiment described in Section 3.1, we report a comparison of the three proposed strategies, here applied to the user query σ_2. The *naive* approach produces a culturally informed but ultimately generic overview of J-Horror as a genre, offering no concrete titles or rankings (Figure 4).

Naive Approach (no retrieval)

Japanese horror (often referred to as J-Horror) is one of the most distinctive and influential branches of genre cinema. It tends to favor psychological dread, deeply ingrained cultural superstitions, and atmospheric suspense over explicit gore (though gore is certainly present in some subgenres). [...]

Figure 4: Response generated for the user question σ_2 through the naive approach

The *basic retrieval* approach retrieves thematically related films via embedding similarity, but lacks any ranking by IMDb score, surfacing titles purely by semantic proximity to the query rather than

by quality or relevance. Therefore, it fails to retrieve movies with high IMDb score, favoring lower rated ones instead. Furthermore, embedding similarity alone also fails to capture structured metadata such as nationality: the basic pipeline incorrectly retrieves *Over Your Dead Body*, a French-Japanese co-production, as it appears thematically relevant despite not being a purely Japanese production. Figure 5 shows a condensed version of the retrieved nodes and the consequently generated response.

Basic Retrieval Approach

(a) Top-k Retrieved Nodes

Here we only report the titles of the movies; however, the complete retrieval includes all textual attributes, as shown in the Appendix.

- "Yokai Monsters: 100 Monsters"
- "Yokai Monsters: Along with Ghosts"
- "Love Ghost"
- "Reincarnation"
- "Over Your Dead Body"

(b) Generated Response

Yōkai and Supernatural Horror:

- **Yokai Monsters: 100 Monsters** (1968) IMDb: 6.5
  Horror Dir: K. Yasuda
- **Yokai Monsters: Along with Ghosts** (1969) IMDb: 6.3
  Action, Horror, Mystery Dir: Y. Kuroda
- **Love Ghost** (2001) IMDb: 6.2
  Horror Dir: K. Shibuya
- **Reincarnation** (2005) IMDb: 6.1
  Horror, Mystery Dir: T. Shimizu

Figure 5: Basic retrieval approach for σ_2: (a) embeddings-based retrieval result; (b) generated response.

The *advanced retrieval* approach computes the $\mathscr{A}$-UTP approximation of the top-k retrieved nodes with $\beta = 0.7$. The reconstructed Cypher query matches movies in which "Japan" appears in the countries text attribute, and "Horror" appears in the connected genres. This reconstructed query retrieves a much richer set of movie nodes, which allows the generator to produce a a substantially more useful result. Also this allows the generator to correctly filter by nationality, and therefore exclude *Over Your Dead Body* from the recommendation. The generated response is partially shown in Figure 6, while the complete response and the reconstructed query are deferred to the Appendix due to space limitations.

Advanced Retrieval Approach

Based on the information retrieved, here are the Japanese horror and supernatural-themed movies listed, ranked by their IMDb rating:

- **Demon Slayer: Kimetsu no Yaiba - Tsuzumi Mansion Arc** IMDb: 8.5
- **Demon Slayer: Kimetsu no Yaiba - Infinity Castle** IMDb: 8.5
- **Kwaidan** IMDb: 7.9
- **Three... Extremes** IMDb: 6.9

Figure 6: Response generated for the user question σ_2 with the advanced retrieval approach

4. Conclusions and Future Work

We introduced the Symbolic-Assisted Dense Retriever, an architecture that improves dense retrieval output by synthesizing a minimally complete $\mathscr{A}$-UTP approximation of the augmented implicit query underlying the initial dense retrieval. Preliminary experiments on a Neo4j dataset show that our approach retrieves a richer set of relevant nodes, enabling the generator to produce more accurate, better-informed responses. Future work will target larger-scale experiments across diverse domains to assess scalability, and benchmarking against existing Knowledge Graph RAG methods, particularly direct Text2Cypher approaches [6]. The current notion of minimally-complete best approximation lets the retrieval include nodes missed by embedding-based retrievers, but offers no mechanism to identify and filter out undesired ones. To address this, we also plan to refine the synthesized query using axioms of intensional domain knowledge, improving both the inclusion of previously missed nodes and the exclusion of undesired ones, with potential benefits for RAG effectiveness as well.

Acknowledgements

M. Calamo, M. Mecella and F. Bianchini were partially funded by the EU Projects GAINAfrica (ID 101298559).

Declaration on Generative AI

During the preparation of this work, the authors used Claude 4.7 in order to: Grammar and spelling check. The authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[3] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 6769–6781. URL: https://doi.org/10.18653/v1/2020.emnlp-main.550. doi:10.18653/V1/2020.EMNLP-MAIN.550.

[4] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, Q. Li, A survey on rag meeting llms: Towards retrieval-augmented large language models, KDD '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 6491–6501. URL: https://doi.org/10.1145/3637528.3671470. doi:10.1145/3637528.3671470.

[5] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph retrieval-augmented generation: A survey, ACM Trans. Inf. Syst. 44 (2025). URL: https://doi.org/10.1145/3777378. doi:10.1145/3777378.

[6] I. Mandilara, C. M. Androna, E. Fotopoulou, A. Zafeiropoulos, S. Papavassiliou, Decoding the mystery: How can llms turn text into cypher in complex knowledge graphs?, IEEE Access 13 (2025) 80981–81001. doi:10.1109/ACCESS.2025.3567759.

[7] F. Bianchini, M. Calamo, F. De Luzi, M. Macrì, M. Mecella, Enhancing complex linguistic tasks resolution through fine-tuning llms, rag and knowledge graphs (short paper), in: International Conference on Advanced Information Systems Engineering, Springer, 2024, pp. 147–155.

[8] P. Barceló, M. Romero, The complexity of reverse engineering problems for conjunctive queries, in: M. Benedikt, G. Orsi (Eds.), 20th International Conference on Database Theory, ICDT 2017, Venice, Italy, March 21-24, 2017, LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 7:1–7:17. URL: https://doi.org/10.4230/LIPIcs.ICDT.2017.7. doi:10.4230/LIPICS.ICDT.2017.7.

[9] R. Krishnamurthy, S. P. Morgan, M. M. Zloof, Query-by-example: Operations on piecewise continuous data (extended abstract), in: M. Schkolnick, C. Thanos (Eds.), 9th International Conference on Very Large Data Bases, October 31 - November 2, 1983, Florence, Italy, Proceedings, Morgan Kaufmann, 1983, pp. 305–308. URL: http://www.vldb.org/conf/1983/P305.PDF.

[10] B. ten Cate, M. Funk, J. C. Jung, C. Lutz, Fitting algorithms for conjunctive queries, SIGMOD Rec. 52 (2023) 6–18. URL: https://doi.org/10.1145/3641832.3641834. doi:10.1145/3641832.3641834.

[11] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor, Cypher: An evolving query language for property graphs, in: G. Das, C. M. Jermaine, P. A. Bernstein (Eds.), Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018, ACM, 2018, pp. 1433–1445. URL: https://doi.org/10.1145/3183713.3190657. doi:10.1145/3183713.3190657.

[12] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. Reutter, D. Vrgoč, Foundations of modern query languages for graph databases, ACM Comput. Surv. 50 (2017). URL: https://doi.org/10.1145/3104031. doi:10.1145/3104031.

[13] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, J.-Y. Nie, C-pack: Packed resources for general chinese embeddings, in: Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval, 2024, pp. 641–649.

A. Appendix

A.1. Example One

This section reports the details of both the basic and advanced retrieval approaches for the user question σ_1: *"What are the best adventure animated movies from Japan?"*.

A.1.1. Basic Retrieval

The result of $\text{Ret}(\sigma_1, G)$ consists of the following top-k nodes ($k = 5$):

Retrieved Nodes (Embeddings-based)

- **The Great Yokai War** (2005) — Similarity: 0.831
  Adventure, Family, Fantasy
- **One Piece: Strong World** (2009) — Similarity: 0.830
  Action, Adventure, Animation
- **Majokko shimai no Yoyo to Nene** (2013) — Similarity: 0.827
  Adventure, Animation, Comedy
- **Inuyasha the Movie: Affections Touching Across Time** (2001) — Similarity: 0.825
  Action, Adventure, Animation
- **Bloody Spear at Mount Fuji** (1955) — Similarity: 0.824
  Adventure, Drama

The response generated exploiting the context obtained from the above nodes is the following:

Generated Response (Basic Retrieval)

Based on the knowledge provided, several anime movies fit the criteria of being animated, adventurous, and originating from Japan. Here are some options:

- **One Piece: Strong World (2009)** — IMDb Rating: 7.4
  Genres: Action, Adventure, Animation
- **Inuyasha the Movie: Affections Touching Across Time (2001)** — IMDb Rating: 7.1
  Genres: Action, Adventure, Animation
- **Majokko shimai no Yoyo to Nene (2013)** — IMDb Rating: 6.6
  Genres: Adventure, Animation, Comedy
- **The Great Yokai War (2005)** — IMDb Rating: 6.3
  Genres: Adventure, Family, Fantasy

Recommendation: Among the titles listed, One Piece: Strong World (2009) has the highest IMDb rating (7.4) among the available selections.

A.1.2. Advanced Retrieval

The reconstructed query produced by Synth for σ_1 is the following:

Reconstructed Cypher Query

```
MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n)-[:IN_GENRE]-(n4) MATCH (n0:Person&Actor&Director)
MATCH (n1:Person&Actor) MATCH (n2:Person&Actor) MATCH (n3:Genre)
MATCH (n4:Genre) WHERE n4.genre = "Adventure"
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n)-[:IN_GENRE]-(n4) MATCH (n0:Person&Director)
MATCH (n1:Person&Actor) MATCH (n2:Person&Actor) MATCH (n3:Genre)
WHERE n3.genre = "Adventure" MATCH (n4:Genre)
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:ACTED_IN]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:IN_GENRE]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n0:Person&Actor) MATCH (n1:Person&Actor)
MATCH (n2:Genre) MATCH (n3:Genre)
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n)-[:IN_GENRE]-(n4) MATCH (n0:Person&Director)
MATCH (n1:Person&Actor) MATCH (n2:Person&Actor) MATCH (n3:Genre)
WHERE n3.genre = "Adventure" MATCH (n4:Genre)
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n0:Person&Director) MATCH (n1:Person&Actor)
MATCH (n2:Person&Actor) MATCH (n3:Genre) WHERE n3.genre = "Adventure"
RETURN n
```

The response generated exploiting the context obtained from the advanced retrieval is the following:

Generated Response (Advanced Retrieval)

Based on the films retrieved, there are several excellent choices for high-stakes, adventure-filled Japanese animated movies, ranging from modern fantasy epics to historical adventures.

- **Attack on Titan the Movie: The Last Attack** IMDb: 9.1
  Modern & Supernatural Action Epics
- **Ramayana: The Legend of Prince Rama** IMDb: 9.1
  Grand Historical/Mythological Epics
- **Solo Leveling: ReAwakening** IMDb: 8.7
  Modern & Supernatural Action Epics
- **Demon Slayer: Kimetsu no Yaiba- The Movie - Infinity Castle** IMDb: 8.5
  Modern & Supernatural Action Epics
- **Ugetsu (1953)** IMDb: 8.1
  Grand Historical/Mythological Epics
- **Kwaidan** IMDb: 7.9
  Cautionary/Supernatural Adventure

A.2. Example Two

This section reports the details of both the basic and advanced retrieval approaches for the user question σ_2: *"Horror Japanese Movies all time by rating"*.

A.2.1. Basic Retrieval

The result of $\text{Ret}(\sigma_2, G)$ consists of the following top-k nodes ($k = 5$):

Retrieved Nodes (Embeddings-based)

- **Yokai Monsters: 100 Monsters** (1968) — Similarity: 0.885
  Horror
- **Yokai Monsters: Along with Ghosts** (1969) — Similarity: 0.876
  Action, Horror, Mystery
- **Love Ghost** (2001) — Similarity: 0.868
  Horror
- **Reincarnation** (2005) — Similarity: 0.866
  Horror, Mystery
- **Over Your Dead Body** (2014) — Similarity: 0.866
  Drama, Horror, Thriller

The response generated exploiting the context obtained from the above nodes is the following:

Generated Response (Basic Retrieval)

Based on the retrieved knowledge-graph context, here is a list of Japanese horror movies ranked by their IMDb rating.

- **Yokai Monsters: 100 Monsters (1968)** — IMDb Rating: 6.5
  Genre: Horror
- **Yokai Monsters: Along with Ghosts (1969)** — IMDb Rating: 6.3
  Genre: Action, Horror, Mystery
- **Love Ghost (2001)** — IMDb Rating: 6.2
  Genre: Horror
- **Reincarnation (2005)** — IMDb Rating: 6.1
  Genre: Horror, Mystery

A.2.2. Advanced Retrieval

The reconstructed query produced by Synth for σ_2 is the following:

Reconstructed Cypher Query

```
MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n0:Person&Director) MATCH (n1:Actor) MATCH (n2:Actor&Person)
MATCH (n3:Genre) WHERE n3.genre = "Horror"
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:DIRECTED]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:ACTED_IN]-(n3)
MATCH (n)-[:IN_GENRE]-(n4) MATCH (n)-[:IN_GENRE]-(n5)
MATCH (n0:Person&Director) MATCH (n1:Person&Director)
MATCH (n2:Actor&Person) MATCH (n3:Actor) MATCH (n4:Genre)
WHERE n4.genre = "Horror" MATCH (n5:Genre)
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n0:Person&Director) MATCH (n1:Actor&Person)
MATCH (n2:Actor&Person) MATCH (n3:Genre) WHERE n3.genre = "Horror"
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n)-[:IN_GENRE]-(n4) MATCH (n0:Person&Director)
MATCH (n1:Actor) MATCH (n2:Actor&Person) MATCH (n3:Genre)
WHERE n3.genre = "Horror" MATCH (n4:Genre)
RETURN n

UNION

MATCH (n:Movie) WHERE "Japan" IN n.countries
MATCH (n)-[:DIRECTED]-(n0) MATCH (n)-[:ACTED_IN]-(n1)
MATCH (n)-[:ACTED_IN]-(n2) MATCH (n)-[:IN_GENRE]-(n3)
MATCH (n)-[:IN_GENRE]-(n4) MATCH (n0:Actor&Director&Person)
MATCH (n1:Actor&Person) MATCH (n2:Actor&Person) MATCH (n3:Genre)
MATCH (n4:Genre)
RETURN n
```

The response generated exploiting the context obtained from the advanced retrieval is the following:

Generated Response (Advanced Retrieval)

Based on the information retrieved, here are the Japanese horror and supernatural-themed movies listed, ranked by their IMDb rating:

- **Demon Slayer: Kimetsu no Yaiba - Tsuzumi Mansion Arc** IMDb: 8.5
  Genre: Supernatural/Action (Japanese)
- **Demon Slayer: Kimetsu no Yaiba- The Movie - Infinity Castle** IMDb: 8.5
  Genre: Supernatural/Action (Japanese)
- **Kwaidan** IMDb: 7.9
  Genre: Supernatural Horror (Japanese)
- **Three... Extremes** IMDb: 6.9
  Genre: Horror (Cross-Cultural)