KAGS: Knowledge graph Augmented Generation System with enriched-context retrieval (Supplemental Material)

No Author Given

No Institute Given

Abstract. Large Language Models (LLMs) have become a dominant force across diverse domains, from customer interaction to drug discovery. However, the reliance of LLMs on static training data restricts their ability to incorporate new information beyond their cutoff date. Retrieval-Augmented Generation (RAG) systems have emerged as a solution to this limitation but conventional RAG systems depend heavily on unstructured text retrieval, often lacking deeper reasoning capabilities. In this work, we propose the Knowledge graph Augmented Generation System (KAGS), a Type-I neurosymbolic architecture that integrates symbolic reasoning through Knowledge Graphs (KGs) with the neural adaptability of LLMs. Its strength lies in its Knowledge Graph Construction (KGC) strategy that includes a unique way of triplet representation and its novel retrieval strategy that employs metadata filtration and h-hop graph traversal altogether, ensuring faithfulness, reasoning and knowledge grounding. Experimental evaluation across nine metrics demonstrates the efficacy of KAGS over competing RAG variants with an answer Relevance of 0.4701, outperforming state-of-the-art RAG techniques. In structural knowledge evaluation, KAGS achieves the highest token containment accuracy (28.29%) and fuzzy containment accuracy (51.56%), confirming its enhanced comprehension and retrieval efficiency. These findings indicate that KAGS not only grounds neural generation in symbolic knowledge but also improves reasoning, faithfulness, and context alignment.

Keywords: RAG · KAGS · Knowledge Graphs · Retrieval · Knowledge Augmentation · Neurosymbolic AI · Type-I system · AI · NLP · Metadata

1 Preliminaries

In this section, we'll firstly define the Knowledge Graphs (KG), Large Language Models (LLMs) and then, we finally discuss about the reasoning in-depth including its types and relation with KAGS.

Knowledge Graph (KG) is means of knowledge representation where the knowlede is represented in the form of triplets. These triplets consist of entities which are connected by relations. Given a KG G, it can be represented as collection of these triplets given as,

$$G = \{\tau_i |\ \tau_i \in T\} \tag{1}$$

such that τ_i represents the i^{th} triplets, given $i \in n$, where n is total number of triplets in the triplet set T.

$$\tau_i = (h_i, r_i, t_i) \tag{2}$$

Also, $h_i \& t_i \in E$ and $r_i \in R$ where h_i represents the head entity, t_i represents the tail entity and r_i represents the relationship between them, E represents the set of all the entities, and R represents the set of relationships among the entities. Depending upon the type of information they store, KGs can be named as encyclopedic KG [21, 1, 4], commonsense KG [9, 19, 11], domain-specific KG [3, 13, 24, 16] and multi-model KG [23, 8, 14].

Large Language Models (LLMs) are parameterised probabilistic models that approximate the conditional distribution $P(w_p \mid w_{<p}; \theta)$, where w_p is the token at position t, $w_{<p} = (w_1, \ldots, w_{p-1})$ is the preceding context, and $\theta \in R^n$ represents the learnt parameters (often numbering in the billions). These models are typically implemented as deep neural networks based on the Transformer architecture, which uses self-attention mechanisms. The self-attention at layer l computes context-dependent embeddings

$$h_i^{(l)} = \sum_{j=1}^{T} \text{softmax}\left(\frac{(Q_i^{(l)} K_j^{(l)\top})}{\sqrt{d_k}}\right) V_j^{(l)} \tag{3}$$

where $Q, K, V \in R^{T \times d_k}$ are linear projections of the input representations. The training objective minimizes the Negative Log-Likelihood (NLL) loss

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log P_\theta(w_t \mid w_{<p}) \tag{4}$$

which is equivalent to maximising the cross-entropy between predicted and true token distributions. Optimization is performed via stochastic gradient descent or its variants (e.g., Adam), using backpropagation to update parameters according to

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta) \tag{5}$$

where η is the learning rate. In essence, LLMs approximate a high-dimensional function $f_\theta : \mathcal{W}^* \rightarrow [0,1]^{|\mathcal{V}|}$, mapping token sequences to probability distributions over a vocabulary $\mathcal{V}$, with the attention mechanism providing an $O(T^2 d)$ computational complexity per layer, where d is the embedding (or hidden) dimension per token,

Reasoning is a basic concept of intelligence, and various forms of reasoning have been achieved differently across the AI domain [2]. The taxonomy of reasoning and its types is illustrated in Fig. **??**. Broadly, reasoning can be categorised into *deductive*, *inductive*, and *abductive* forms. Deductive reasoning involves deriving specific conclusions from general premises or known facts, i.e., applying logical inference rules such that if the premises are true, the conclusion necessarily follows. Inductive reasoning, on the other hand, generalises from specific

observations to broader hypotheses or patterns, often employed by LLMs to infer likely continuations or generalised answers from contextual data. Abductive reasoning seeks the most plausible explanation for an observation, typically under incomplete information, and is central to hypothesis generation and contextual inference in AI.

The proposed **KAGS** embodies all three forms of reasoning in an integrated manner. Its *retrieval process*, which employs metadata filtering and graph-based traversal, aligns with **deductive reasoning** as it narrows down relevant knowledge from structured premises (knowledge graph and triplet metadata). The *generation phase*, powered by the LLM, exhibits **inductive reasoning** by synthesising generalised answers from the retrieved contextual evidence. Finally, the dynamic interplay between retrieval and generation where the system infers the most plausible knowledge paths, reflects **abductive reasoning**, enabling KAGS to hypothesise the most likely sources and explanations supporting an answer. Together, these reasoning modalities make KAGS a comprehensive framework that operationalises logical inference,

2 Knowledge graph Augmented Generation System

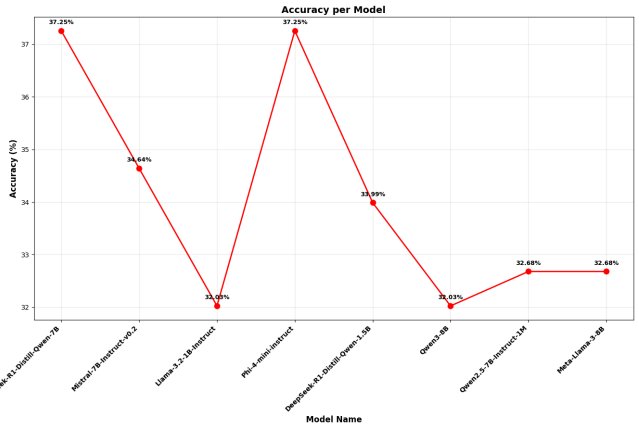

Fig. 1. Accuracy comparison of top 8 LLMs for the KGC phase

The LLM and the prompt were chosen based on a meticulous evaluation of a sample of the dataset containing 15 chunks and 153 manually created triplets. Based on the top-trending LLMs for text generation on the Hugging Face Leaderboards [1], we selected 8 models for the triplet extraction, and then the metrics for these models were matched against the human-made standard

[1] https://huggingface.co/

of 153 triplets. Considering the time, accuracy and the number of parameters, Phi-4-mini-instruct performed optimally.

Fig. 1 and Fig. 2 represents the accuracy and the average time taken per model for the triplet extraction. The accuracy metrics was calculated based on the number of triplets each model extracted. For example, say if model $\mathcal{M}_a$ extracts $\mathcal{T}_{ad}$ triplets from chunk d where $d \in \mathcal{D}$, then the accuracy of the system is defined as,

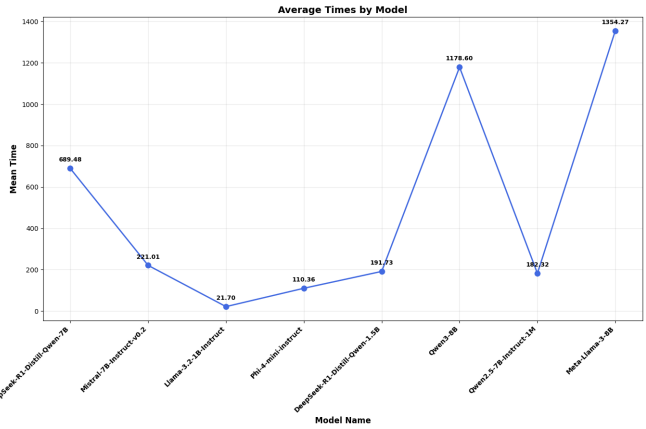

Fig. 2. Extraction time comparison of top 8 LLMs for the KGC phase

$$accuracy_{kgc} = \frac{\sum_{d=1}^{\mathcal{D}} \mathcal{T}_{ad}}{Total\ number\ of\ triplets} \tag{6}$$

Likewise, the average time taken per model could be defined as,

$$average_{time} = \frac{\sum_{d=1}^{\mathcal{D}} time_{ad}}{Total\ number\ of\ chunks} \tag{7}$$

where $time_{ad}$ indicates the time taken by model $\mathcal{M}_a$ to extract triplets from chunk d.

It can be seen that the DeepSeek-R1-Distil-Qwen-7B leads in terms of accuracy, followed by Phi-4-mini-Instruct and Mistral-7B-Instruct-v0.2. However, Phi-4-mini-Instruct takes the lowest amount of time among the three, before Mistral and DeepSeek. Also, considering the aspect that Phi-4-mini-Instruct has the smallest number of parameters (3.8B), it becomes the optimal choice for triplet extraction.

KGC in-depth: Fig. 3 shows the triplets extracted directly from the context using a prompt. In the case of sentences that involve reasoning and explanation, the LLM either tends to skip those sentences or breaks the sentence into 4 or 5 parts, questioning the sanctitiy of the triplet.

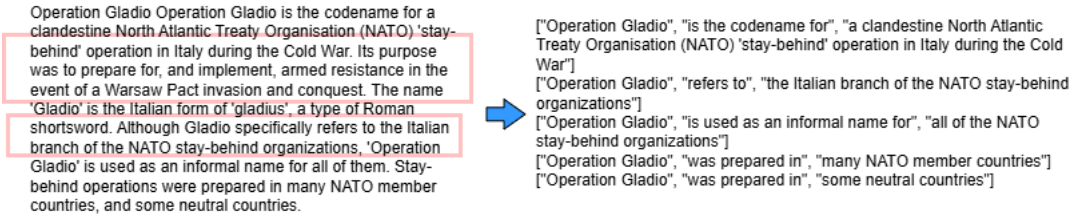

Fig. 3. LLM-based triplet extraction (Sentences in red boxes have not been utilized by the LLM)

Hence, we propose the Knowledge Graph Creation (KGC) phase that also stores the metadata of the triplets, which allows us to also capture the sentences that involve reasoning and explanation. Fig. 4 displays an example of the knowledge graph triplets in addition to the source text of the context metadata. It shows that this metadata allows us to effectively capture the nuanced sentences that involve intricate patterns of reasoning or explanation.

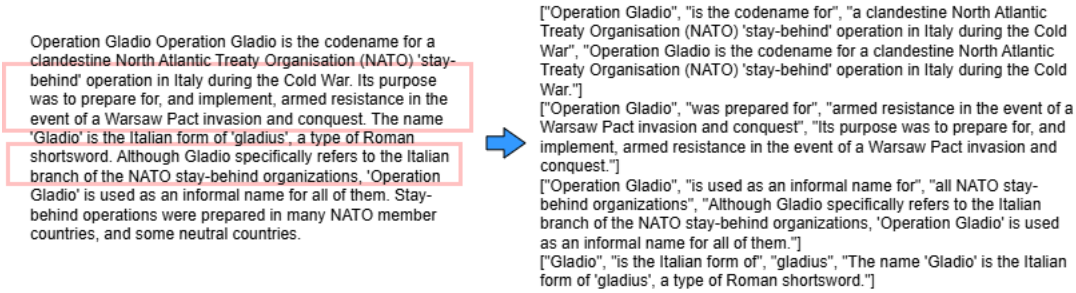

Fig. 4. Knowledge Graph creation through KGC in KAGS

3 Evaluation Metrics

Precision [17] is a standard metric that indicates the relevance of the the retrieved context with respect to the ground truth whereas recall [17] indicates how well the context is being supported by the ground truth. On the other hand, the F1-score [10] explains the number of overlapping words between the predicted answer and the ground truth answer. In simpler terms, it's balanced score between the precision and recall.

Faithfulness [17] (a form of precision) tells us how much of an AI's answer can actually be confirmed using the given context. In simple terms, it checks

whether the AI is "sticking to the facts" rather than making things up. It tests the generator's ability and works as a hallucination checker, thus increasing the trust on the LLM. Faithfulness is calculated in two main steps:

- Breaking Down the Answer (Statement Extraction): We first ask a language model to split the AI's answer into smaller, clear statements. This makes it easier to verify each part of the answer.
- Checking Each Statement (Statement Verification): Then, for each statement, we ask the model whether that statement can be supported by the original context. The model gives a short explanation and then decides "Yes" (supported) or "No" (not supported).

Finally, the faithfulness score is calculated as the proportion of supported statements to the total number of statement and then averaged over the number of questions.

$$\Phi = \frac{1}{S} \sum_{q=1}^{S} \frac{\nu_q}{\sigma_q} \tag{8}$$

Where S is the total number of test samples Φ is the faithfulness score ν_q is the number of verified (supported) statements for the q^{th} query and σ_q the total number of extracted statements for q^{th} the query.

Answer relevance (a form of recall) measures how well an AI's response actually addresses the original question regardless of whether the answer is factually correct. It helps spot answers that might be incomplete, off-topic, or only loosely connected to what was asked. Steps to calculate answer relevance is given as,

- Generate a question using the LLM based on the predicted answer.
- Next, create the embeddings of the actual and predicted answers.
- Finally, calculate the cosine-similarity between both the embeddings.

Formally, it can be calculated as:

$$\mathcal{A}_{\mathcal{R}} = \frac{1}{S} \sum_{q=1}^{S} \text{sim}(\theta_q, \bar{\theta}_q) \tag{9}$$

Where $\mathcal{A}_R$ is the answer relevance score $\text{sim}(\theta, \theta_i)$ is the cosine similarity between the embedding of the original question θ_q and the embedding of the q^{th} generated question $\bar{\theta}_q$.

4 Related Work

Paper	Dataset	Indexing / Context Storage	Retrieval Strategy	LLMs	Notes
Tree-RAG [7]	Private organisation documents	Tree structure for entity hierarchies, BM25	Dense Passage Retriever (DPR) with entity-guided negatives + ChromaDB	LLaMA2-7B	Hybrid RAG using hierarchical entity metadata. Limitation: expert curation required for tables/charts.
Adaptive RAG [10]	SQuAD v1.1, NQ, TriviaQA, MuSiQue, HotpotQA, 2WikiMultiHopQA	External KBs (unspecified)	BM25 or iterative multihop + ChromaDB	GPT-3.5-Turbo-Instruct, FLAN-T5-XL, FLAN-T5-XXL	Adaptive retrieval controlled by a complexity classifier. Limitation: depends on classifier performance.
GraphRAG [6]	Podcasts, news articles	GPT-4 Turbo to extract graph; Leiden algorithm for communities	Graph traversal / QFS, Map-Reduce	GPT-4 Turbo	Structured graph-based retrieval; no embeddings. Limitation: domain-specific prompt tuning required.
Hybrid RAG [17]	CRAG Benchmark	Pinecone vector DB, ADA-002 embeddings, LangChain + networkx KG	Dense + BM25 + KG retrieval	LLaMA3-70B-Instruct	Hybrid dense/sparse/KG retrieval. Limitation: KG underused; large context overhead.
KG^2-RAG [22]	HotpotQA variants	Triplet generation via LLM (unspecified); chunks linked to triplets	KG-guided chunk expansion, mxbai-embed-large, bge-reranker-large	LLaMA3-8B	KG-guided retrieval with MST expansion. Limitation: expensive KG creation for large corpora.
KGQA [18]	WebQ, ComplexWebQ, Mintaka, LC-QuAD	Static Wikidata dump → differential KG using sparse matrices	Rigel KGQA with up to 2-hop reasoning	FLAN-T5-Small/XL/XXL, OPT-13B, Alexa TM, T0-3B/11B	Efficient KG-based reasoning. Limitation: assumes gold entity spans.
IrCoT [20]	HotpotQA, 2WikiMultiHopQA, MuSiQue, IIRC	Wikipedia + ElasticSearch	CoT-driven iterative retrieval	FLAN-T5-XXL/XL, GPT-3	Reasoning-aware retrieval cycle. Limitation: CoT increases latency and may overflow input limits.
HyPA-RAG [12]	NYC Local Law LL144; synthetic legal QA	Pattern-based chunking, GPT-4o triplets	Adaptive hybrid (dense + sparse + KG)	GPT-4o, GPT-3.5-Turbo	Legal-domain hybrid strategy. Limitation: evaluated on synthetic data only.
RecipeRAG [15]	Recipes from Food.com	KG embedding models	Custom KG-based retriever	DeepSeek-R1-0528, Mistral Small 3.2 24B	Introduces RecipeKG for recipe retrieval. Limitation: domain-restricted and hard to generalize.
Graph -based QA pipeline [5]	World Literature Knowledge Graph + Manually curated questions for evaluation	FAISS, entites are extracted through NuExtract-1.5 model	SPARQL query based KG retrieval and vector similarity search for initial entities	Llama 3.1 8B for triple verbalization and generator LLM is not specified	Presents a meticulous evaluation strategy for KG creation and QA system analysis Limitation: Efficiency of the system is concerning point as it evaluates every step leading to computation efficiency issues.

Table 1. Comparison of RAG Variants

Bibliography

[1] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: international semantic web conference. pp. 722–735. Springer (2007)

[2] Bhuyan, B.P., Ramdane-Cherif, A., Tomar, R., Singh, T.: Neuro-symbolic artificial intelligence: a survey. Neural Computing and Applications **36**(21), 12809–12844 (2024)

[3] Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. Nucleic acids research **32**(suppl_1), D267–D270 (2004)

[4] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250 (2008)

[5] Draetta, L., Stranisci, M.A., Corallo, F., Balestrucci, P.F., Oliverio, M., Damiano, R., Mazzei, A.: Beyond the metrics: an investigation into the reliability of evaluation metrics for domain specific graph-based question answering (2025)

[6] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R.O., Larson, J.: From local to global: A graph rag approach to query-focused summarization. arXiv preprint arXiv:2404.16130 (2024)

[7] Fatehkia, M., Lucas, J.K., Chawla, S.: T-rag: lessons from the llm trenches. arXiv preprint arXiv:2402.07483 (2024)

[8] Ferrada, S., Bustos, B., Hogan, A.: Imgpedia: a linked dataset with content-based analysis of wikimedia images. In: International Semantic Web Conference. pp. 84–93. Springer (2017)

[9] Ilievski, F., Szekely, P., Zhang, B.: Cskg: The commonsense knowledge graph. In: European Semantic Web Conference. pp. 680–696. Springer (2021)

[10] Jeong, S., Baek, J., Cho, S., Hwang, S.J., Park, J.C.: Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. arXiv preprint arXiv:2403.14403 (2024)

[11] Ji, H., Ke, P., Huang, S., Wei, F., Zhu, X., Huang, M.: Language generation with multi-hop reasoning on commonsense knowledge graph. arXiv preprint arXiv:2009.11692 (2020)

[12] Kalra, R., Wu, Z., Gulley, A., Hilliard, A., Guan, X., Koshiyama, A., Treleaven, P.: Hypa-rag: A hybrid parameter adaptive retrieval-augmented generation system for ai legal and policy applications. arXiv preprint arXiv:2409.09046 (2024)

[13] Liu, Y., Zeng, Q., Ordieres Meré, J., Yang, H.: Anticipating stock market of the renowned companies: A knowledge graph approach. Complexity **2019**(1), 9202457 (2019)

[14] Liu, Y., Li, H., Garcia-Duran, A., Niepert, M., Onoro-Rubio, D., Rosenblum, D.S.: Mmkg: multi-modal knowledge graphs. In: European Semantic Web Conference. pp. 459–474. Springer (2019)
[15] Loesch, J., Durmuş, E., Celebi, R.: Reciperag: A knowledge graph-driven approach to personalized recipe retrieval and generation (2025)
[16] Safavi, T., Belth, C., Faber, L., Mottin, D., Müller, E., Koutra, D.: Personalized knowledge graph summarization: From the cloud to your pocket. In: 2019 IEEE International Conference on Data Mining (ICDM). pp. 528–537. IEEE (2019)
[17] Sarmah, B., Mehta, D., Hall, B., Rao, R., Patel, S., Pasquali, S.: Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In: Proceedings of the 5th ACM International Conference on AI in Finance. pp. 608–616 (2024)
[18] Sen, P., Mavadia, S., Saffari, A.: Knowledge graph-augmented language models for complex question answering. In: Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE). pp. 1–8 (2023)
[19] Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
[20] Trivedi, H., Balasubramanian, N., Khot, T., Sabharwal, A.: Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. arXiv preprint arXiv:2212.10509 (2022)
[21] Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
[22] Zhu, X., Xie, Y., Liu, Y., Li, Y., Hu, W.: Knowledge graph-guided retrieval augmented generation. arXiv preprint arXiv:2502.06864 (2025)
[23] Zhu, X., Li, Z., Wang, X., Jiang, X., Sun, P., Wang, X., Xiao, Y., Yuan, N.J.: Multi-modal knowledge graph construction and application: A survey. IEEE Transactions on Knowledge and Data Engineering **36**(2), 715–735 (2022)
[24] Zhu, Y., Zhou, W., Xu, Y., Liu, J., Tan, Y.: Intelligent learning for knowledge graph towards geological data. Scientific Programming **2017**(1), 5072427 (2017)