Development of Approach to Reranking Language Model Responses using Knowledge Graphs in Accordance with Factual Correctness

Scientific supervisor:
Mariya Khodorchenko, PhD
Student:
Ekaterina Tsaplina, J4233c

2024

Formulation of the problem

The central challenge:

LLMs in QA systems suffer from: lack of actual knowledge, hallucinations, recognition of complex terms and jargonisms, similarity of documents on narrow topics. As a consequence, answers are not always complete and relevant to the question.

Goal:
To improve the quality of LLM responses by reranking based on facts from knowledge graphs (KGs).

Practical relevance:
To refine the user experience of QA systems.

Objectives:

- Preparation of data for the experiment: generation of questions for dataset texts, text processing, preparation of ontologies.
- Development of Approach to Reranking Language Model Responses using Knowledge Graphs in Accordance with Factual Correctness.
- Construction of two pipelines for comparison: RAG and RAG+KG reranking.

Role of KGs in challenging the limitations of LLM & RAG

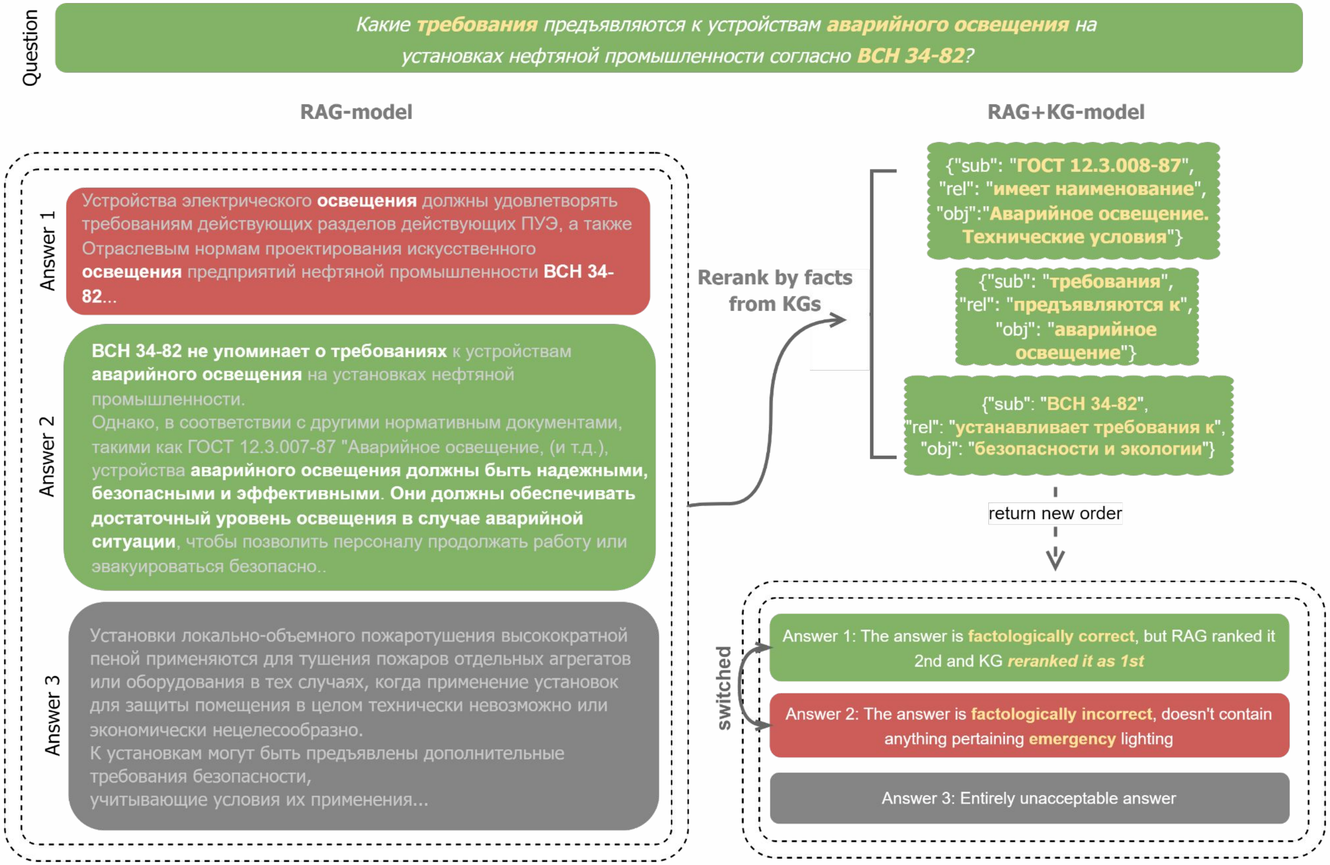
LLM & RAG	Knowledge Graphs (KGs)
Cons: • Vector similarity is not transitive • Lots of context vs details: a problem of text chunking strategy • Lack of accuracy in search for complex topics • Hallucinations are not completely avoided	Pros: • Structural knowledge • Accuracy • Interpretability • Domain-specific knowledge • Evolving knowledge
Pros: • General knowledge • Language processing • Generalizability	Cons: • Incompleteness • Lacking language understanding • Non-triviality of the design

The complementary nature of RAG and KGs

Knowledge Graphs has been cited as one of **the most impactful technology to be relevant in coming years** [1]

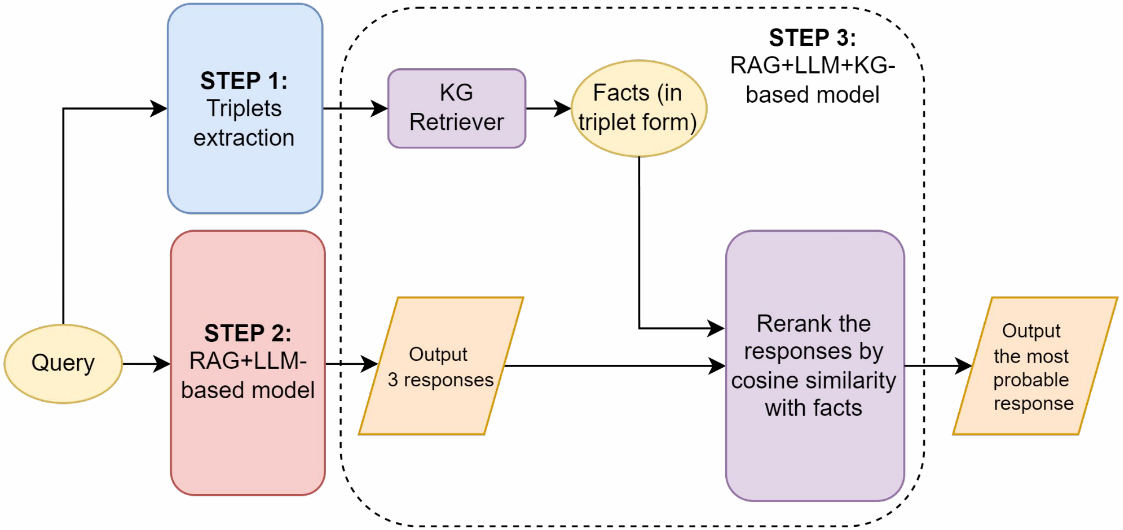
[1] Gartner, Impact Radar for Generative AI, 2024

Illustration of the KGs Performance



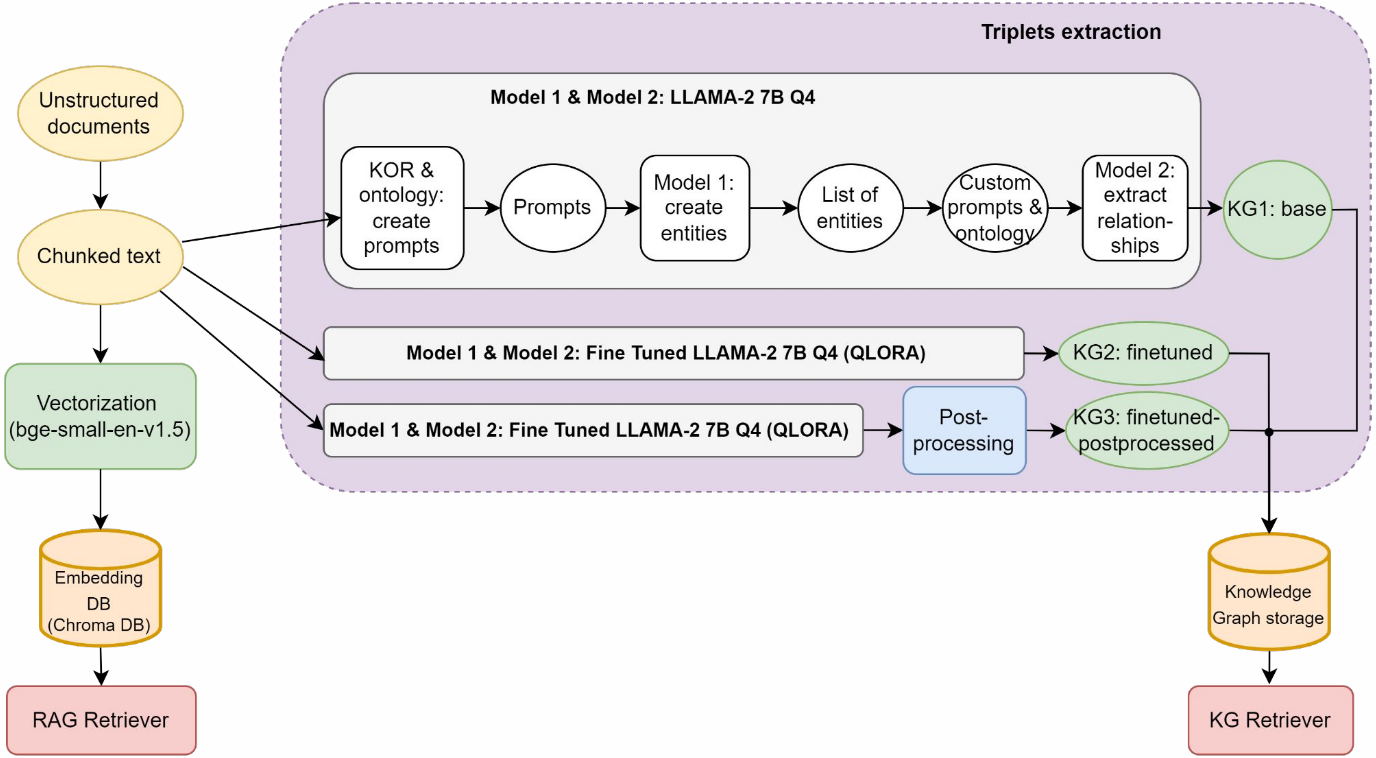
Methodology: Overall pipeline

- The key idea is to use KG in order to rerank the answers of the RAG model
- First, 3 responses are received from the RAG model, then the reranking with KG is performed
- If the most significant answers of two models (RAG model and RAG+KG) differ, they are compared by the measure of relevance to the query



Overall pipeline of the algorithm

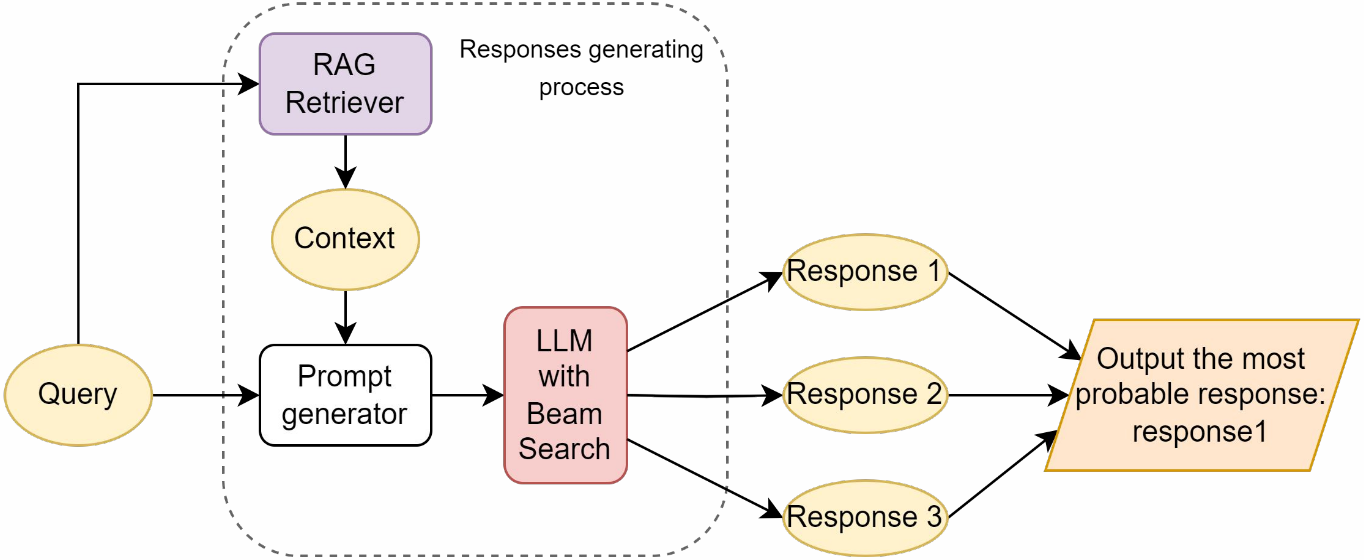
Methodology: Step 1 - KG generation



Text processing and storing procedure

Methodology: Step 2 - RAG-based model

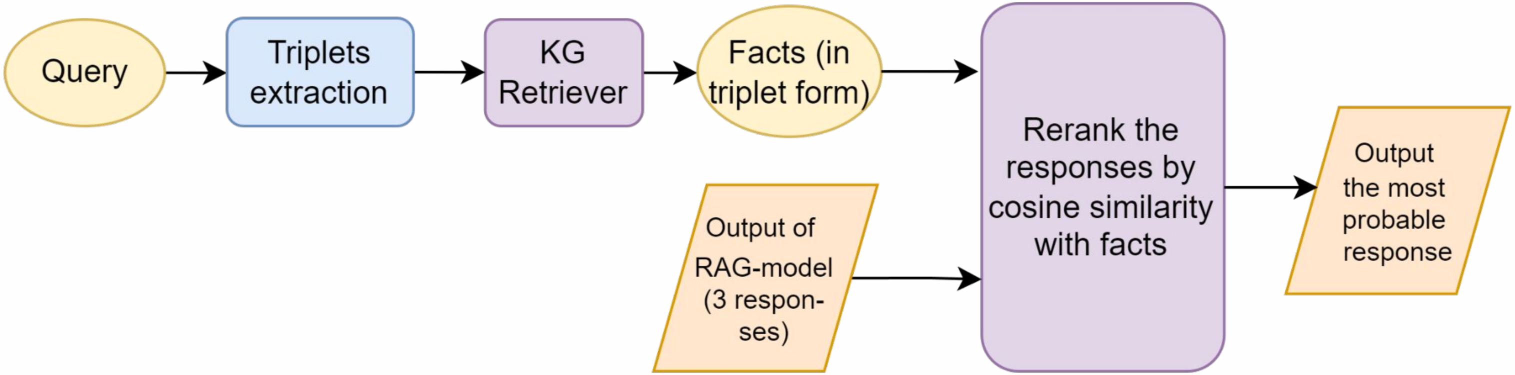
- RAG retriever provides the context for LLM
- The generated responses have probabilities assigned to them by Beam Search
- The model outputs the response with the highest probability



Pipeline of the model

Methodology: Step 3 - RAG+KG-reranking model

- The KG retriever extracts triplets from query
- The cosine similarity between vectorized facts and responses (from RAG-based model) is calculated
- The responses are reranked by similarity measure with facts

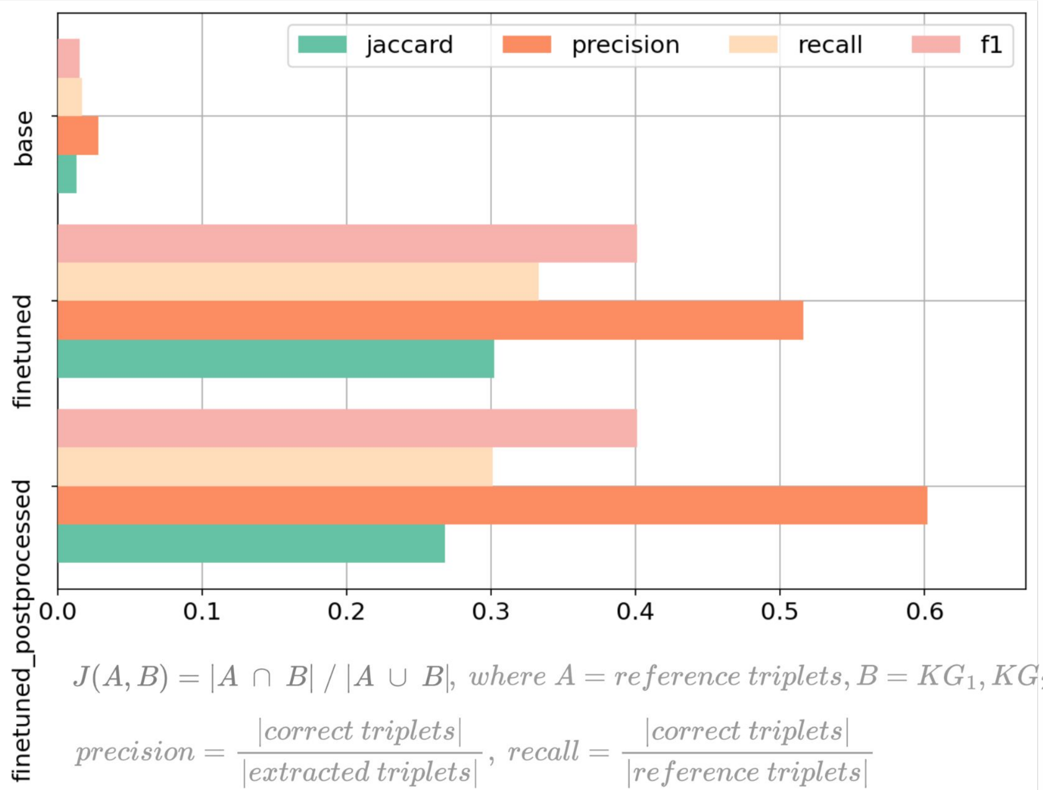


Pipeline of the RAG+KG-based model

Experiment design

- The **two models** - **RAG-based** and **RAG+KG rerank** model are compared
- Get **3 answers** to each question using **RAG**
- **Filter** the **answers** and leave only those cases where reranking makes sense - i.e. 3 not the same answers to 1 question
- **Reranking** answers **by KG**
- Counting statistics: the most important thing is how many times the **reranking occurred**, and then in **how many** of those cases the **RAG** or **RAG+KG Rerank** is **better**.
- Repeat for all domains (computer, nature, movie, GOST)

Experiments: Results of KGs generation



$$J(A,B) = |A \cap B| \,/\, |A \cup B|, \textit{ where } A = reference\ triplets, B = KG_1, KG_2 \textit{ or } KG_3$$

$$precision = \frac{|correct\ triplets|}{|extracted\ triplets|},\ recall = \frac{|correct\ triplets|}{|reference\ triplets|}$$

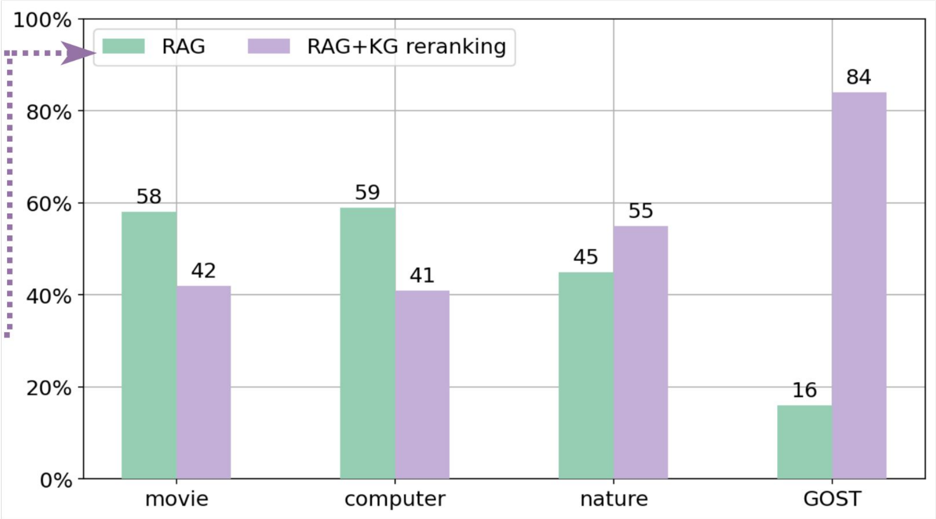
The **first stage** of experiments was **selection** of a **model** (base, fine-tuned or fine-tuned with post-processing).

From the diagram we can conclude that the **fine-tuning** had a **positive effect** in **comparison** with **base**

The **post-processing** of **triplets** after fine-tuning **improved precision** and slightly **reduced recall**, which is expected, since postprocessing is filtering

Experiments: Overall results

Reranking took place in about 1/3 of all responses; in 2/3 of responses both models ranked the answers in the same order



Metric	movie	computer	nature	GOST
FRES/Text complicacy	64.5	65.14	52.51	96.36
Mean sentence length	19.6	20.35	16.30	34.70

$$-106.835 + 1.015\frac{total\ words}{total\ sentences} + 84.6\frac{total\ syllables}{total\ words}$$

In the **majority of cases (2/3)** reranking did not actually take place.

The chart shows the percentage of cases **where reranking took place** and which model performed better: either **RAG** (green) or **RAG+KG rerank** (purple).

The table shows the **complexity** of **domains.**

Value added from KG reranking: nature - 13.5%, GOST - 42.5% compared to the average between movie and computer (41.5%)

Conclusions

In this work the **KG reranking** was implemented in attempts to **improve** the **responses** of **LLM**.
It was discovered that:

- LLM pre-training and triplet post-processing is important for knowledge graph construction.
- In more than half of the cases, no reranking was performed because the RAG performed well.
- Thus, although RAG is a powerful tool, combining it with knowledge graphs helps to improve the quality of answers in complex and semantically confusing texts:
    a. Knowledge graphs capture rich relationships between objects, **providing more accurate reasoning**.
    b. The **RAG+KG reranking** approach works best for documents with **clear relations and complex terms**, while basic RAG works best for simpler contexts.
    c. **Added value from KG reranking**: **nature - 14%**, cases mentioning "complex terms" - geo-coordinates.
    d. **Added value from KG reranking**: **GOST - 43%**, knowledge graphs corrected RAG errors according to real facts.

Thank you for your attention!

ITsMOre than a UNIVERSITY