Retrieval Augmented Generation for Romanian

Alexandru C. Sasu
University of Bucharest, Romania

Abstract

In this paper, the task of performing retrieval augmented generation (RAG) for the Romanian language was tackled. The corpus used consisted of popular literary works written by Romanian authors, and the large language model used was created specifically for Romanian tasks. The best results were obtained using the prompts from the appendix.

1 Introduction

Retrieval augmented generation, or RAG, consists of enhancing large language models with textual information in order to generate more accurate, diverse, and natural responses. The first component, of retrieval, uses an information retrieval mechanism, such as vector databases, in order to index, retrieve, and perform other operations on non-parametric data. Afterwards, the augmentation component performs prompt engineering on the information retrieved by the retrieval component and a user inputted query before sending the output to the LLM. Finally, the generative step consists of feeding the aforementioned prompt to the LLM in order to obtain an answer to the query.

There is a significant number of RAG systems for English available online, though for other less used languages such as Romanian, the number of available such systems is extremely limited. This paper seeks to document the implementation of such a system for Romanian.

2 Related Work

The integration of retrieval mechanisms with generative models has been explored in order to enhance performance on knowledge-intensive natural language processing (NLP) tasks. Lewis et al. (2020) introduced Retrieval-Augmented Generation (RAG), a framework that combines pretrained sequence-to-sequence models with non-parametric memory accessed through neural retrievers. In this approach, the parametric memory is a pretrained seq2seq transformer, while the non-parametric memory is a dense vector index of Wikipedia, accessed with the help of a neural retriever. The retriever provides latent documents based on the input, and the seq2seq model conditions on these latent documents together with the input to generate the output.

The RAG framework was evaluated across various knowledge-intensive tasks, including open-domain QA and fact verification, demonstrating state-of-the-art performance. Moreover, RAG models generated more specific, diverse, and factual responses compared to parametric-only seq2seq baselines. This work demonstrates the potential of retrieval-augmented models in improving the factual accuracy and diversity of generated content.

Developing Retrieval Augmented Generation (RAG) systems for low-resource languages, such as Romanian, has been significantly constrained over the past couple years by the limited availability of high-quality datasets and specialized benchmarks. Recent advances in Romanian-specific large language models (LLMs) have addressed some of these challenges.

Masala et al. (2024) introduced the OpenLLM-Ro initiative, which released instruction-tuned LLMs tailored for Romanian by leveraging multilingual and English instruction datasets, such as CulturaX (Nguyen et al., 2024), Alpaca (Taori et al., 2023), Dolly (Conover et al., 2023), and others. To evaluate these models, the authors proposed a diverse set of benchmarks, including both standard NLP tasks (e.g., sentiment analysis, question answering, and machine translation) as well as the novel RoCulturaBench benchmark also created by them, which tests the models' understanding of Romanian cultural, historical, and social knowledge. While their work achieved state-of-the-art results in multiple domains, their methodology did not explicitly explore retrieval-augmented paradigms.

3 Method

3.1 Document Preparation

The documents used were prose literary works written by Romanian authors, and they were chosen because they

capture the nuances and semantics of Romanian in a suitable manner. More details about each document can be consulted in Appendix.

The task accepted four types of document formats: .txt, .doc, .docx, .pdf. Through the use of the LangChain library, the documents were read and then split into chunks of a maximum of 200 characters with 40 characters overlap between documents, in order to comply with the input length constraint of the LLM, as well as to produce more concise embeddings. Moreover, a shorter length of 200 characters was chosen because the user queries were also expected to be short, consisting of one or two sentences of a few words.

3.2 Preprocessing

The only preprocessing realized consisted of replacing certain unusual and possibly confusing characters for the system, with their normal counterpart, namely: “, ”, ş, Ş, ţ, Ţ; were replaced with: ", ș, Ș, ț, Ț.

Other preprocessing means were not explored so as not to exclude important information for the embedding creation model and the LLM, given how they work and their way of capturing semantic relationships.

3.3 Retrieval

For the indexing and retrieval part, ChromaDB was chosen as the vector database. Thus, the documents were fed to the database, and, with the help of a model for creating embeddings, namely the all-MiniLM-L6-v2 sentence transformer from Hugging Face, their embeddings were also stored.

ChromaDB used the hierarchical navigable small world (HNSW) graph method implemented in its library with the help of cosine distance, in order to calculate similarities between the query and the stored documents.

The IR component returned the top 15 most similar documents.

3.4 Augmentation

In order to augment the query that was inputted by the user, the relevant content retrieved by ChromaDB from its database was appended, and multiple prompt engineering strategies were employed. These strategies aimed to create a clearer input for the LLM. They contained elements such as instructions, examples regarding how the question looks like and how the answer looks like (few-shot prompting), and specially formatted tokens in order for the LLM to better recognize and understand certain parts of the input. The prompts used can be consulted in Appendix.

3.5 Generation

The LLM chosen for the generation part was the RoLlama2-7b-Instruct created by OpenLLM-Ro (Masala et al., 2024). The model used 4 bit quantization due to resource constraints posed by the development environment. The quantization was realized by Kaggle user gpreda.

3.6 Failed Approaches

Instead of the all-MiniLM-L6-v2 model used for embedding creation, at first its more advanced version, paraphrase-multilingual-MiniLM-L12-v2 was used, but it was changed due to ChromaDB retrieving documents that were not relevant. It is unknown why this happened, as all-MiniLM-L6-v2 was trained only on English, while paraphrase-multilingual-MiniLM-L12-v2 was trained on 50 languages, including Romanian and English, so it was expected that the version trained on Romanian would work better.

Before using few-shot prompting, the RAG generated worse results, by mainly copying and pasting the context and instructions included in the prompt, without actually giving and answer to the query.

4 Results

In order to compare the performance of the IR with the performance of the LLM and the performance of the RAG, each system was prompted to return a maximum of top 3 relevant documents for the inputted user query.

The addition of an IR component to the chosen LLM did bring some improvements, although not to the desired extent, as the RAG would sometimes incorrectly format the output or it would leave out the information that it was instructed to output. In comparison to the bare bones IR, the RAG managed to capture the same top of documents, and in comparison to the LLM, it managed to capture a mostly different (correct) top, as the LLM hallucinated or was inaccurate in its response.

Below are some queries along with the outputs generated by each system.

User query: "Cine este Buză-Ruptă?".

For this query, the IR returned expected results, the RAG returned good results, and the LLM hallucinated, most probably because it was not trained on the document that contains the answer to the query.

IR:

" 15 context documents

————— similarity (less is better) 0.31

document: moara_cu_noroc.pdf || context: pe Buză-Ruptă.

Obosit și înfierbântat, cum era, de-abia se mai ținea pe picioare. Pintea-și luă iar pușca și plecă să-l caute și să nu-i
—————— similarity (less is better) 0.32
document: moara_cu_noroc.pdf || context: dovedea că Buză-Ruptă spuse un neadevăr și că vrea să-l năpăstuiască.
Săilă, intrând, văzu pe Buză-Ruptă frânt, simți că treaba le stă rău și-și schimbă în buiguiala sa mărturisirea ce făcuse
—————— similarity (less is better) 0.36
document: moara_cu_noroc.pdf || context: cineva la casa lui Buză-Ruptă, poate de vreun om al lui Pintea, care le este vrăjmaș.
—————— similarity (less is better) 0.36
document: moara_cu_noroc.pdf || context: cuvintele lui. Acu intră însă Buză-Ruptă și, puțin în urma lui, Săilă Boarul, și aceștia deteră cercetărilor cu totul altă față.
...
"

LLM:
" Buză-Ruptă este un personaj fictiv din seria de cărți „Harry Potter" de J.K. Rowling.

Maximum top 3 documente relevante pentru propoziția anterioară:

- „Harry Potter și Piatra Filozofală" de J.K. Rowling
- „Harry Potter și Camera Secretelor" de J.K. Rowling
- „Harry Potter și Prizonierul din Azkaban" de J.K
"

RAG:
"
Buză-Ruptă este un personaj din romanul „Moara cu noroc" de Ioan Slavici. El este un om înalt, grăsuliu și slut, care este un dușman al lui Pintea. Buză-Ruptă este un personaj negativ în poveste și este un om care este înțeles cu Săilă Boarul.

Top 3 documente relevante:

- Moara cu noroc
- Fram ursul polar
- Baltagul
"

User query: "Cum l-a ajutat Sfânta Duminică pe Harap-Alb?".

For this query, the IR returned expected results, the RAG returned partly good results as it offered an accurate list of documents but a questionable answer, and the LLM returned mediocre results as it tried to list the answer instead of the documents which were relevant for the query, and did not return any documents.

IR:
"
15 context documents
—————— similarity (less is better) 0.25
document: povestea_lui_harap-alb.docx || context: Și când vântul a aburit, iaca și ei la Sfânta Duminică iar au sosit. Sfânta Duminică era acasă și, cum a văzut pe Harap-Alb poposind la ușa ei, pe loc l-a întâmpinat și i-a zis cu blândețe:
—————— similarity (less is better) 0.25
document: povestea_lui_harap-alb.txt || context: Și când vântul a aburit, iaca și ei la Sfânta Duminică iar au sosit. Sfânta Duminică era acasă și, cum a văzut pe Harap-Alb poposind la ușa ei, pe loc l-a întâmpinat și i-a zis cu blândețe:
—————— similarity (less is better) 0.27
document: povestea_lui_harap-alb.docx || context: Harap-Alb face cum îi zice Sfânta Duminică. Și cum ajunge în grădină, odată începe a smulge la sălăți într-ales și leagă o sarcină mare, mare, cât pe ce să n-o poată ridica în spinare. Și când să
—————— similarity (less is better) 0.27
document: povestea_lui_harap-alb.txt || context: Harap-Alb face cum îi zice Sfânta Duminică. Și cum ajunge în grădină, odată începe a smulge la sălăți într-ales și leagă o sarcină mare, mare, cât pe ce să n-o poată ridica în spinare. Și când să
"

LLM:
"
1. Sfânta Duminică l-a ajutat pe Harap-Alb prin furnizarea de sfaturi și îndrumări.
2. Sfânta Duminică l-a ajutat pe Harap-Alb prin furnizarea de sfaturi și îndrumări.
3. Sfânta Duminică l-a ajutat pe Harap-Alb prin furnizarea de sfaturi și îndrumări.
4. Sfânta Duminică l-a ajutat pe Harap
"

RAG:
"
1. povestea_lui_harap-alb.txt
2. povestea_lui_harap-alb.docx
3. povestea_lui_harap-alb.docx

—Context—:
document: povestea_lui_harap-alb.docx || context: Harap-Alb, ieșind plin de mâhnire, se duce în grajd și începe a-și netezi calul pe coamă, zicând:
"

5 Future Work

Future developments for this task may include more advanced prompting techniques, such as tree of thoughts prompting, more experimentation with the chunking of documents and different embedding models, as well as possibly fine-tuning the LLM before creating the RAG pipeline.

6 Conclusion

The implemented RAG managed to make use of the IR component, while generating decent answers to the user queries as well as returning mostly accurate lists of documents for the inputted queries, based on the database of documents.

For the used LLM, RoLlama2-7b-Instruct, it may be necessary to perform few-shot prompting in order to obtain accurate results, as without it, the experiments showed that the RAG had the tendency to just copy-paste elements from the retrieved documents or from the instructions written in the prompt.

Limitations

Hardware limitations were posed by both the Kaggle and the local environments. Neither of them had enough CUDA memory to run larger LLMs, and the local environment was bottlenecked by a modest RTX 2060 GPU in terms of model inference speed.

Regarding implementation limitations, given that the embeddings model used was designed for English, it might not perform well if the same RAG system were to be applied on other languages, or even on the same language but with lengthier queries.

Acknowledgements

Thanks to Kaggle user gpreda, the use of the RoLlama2-7b-Instruct in a Kaggle environment was made possible without exceeding the available resources, due to the 4 bit quantization realized by them.

References

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Mihai Masala, Denis C. Ilie-Ablachim, Alexandru Dima, Dragos Corlatescu, Miruna Zavelca, Ovio Olaru, Simina Terian-Dan, Andrei Terian-Dan, Marius Leordeanu, Horia Velicu, Marius Popescu, Mihai Dascalu, and Traian Rebedea. 2024. "vorbeşti româneşte?" a recipe to train powerful romanian llms with english instructions.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia. ELRA and ICCL.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

A Appendix

The following files and prompts were used:

- Files: "Baltagul" by Mihail Sadoveanu, in .pdf format, "Fram, ursul polar" by Cezar Petrescu, in .pdf format, "Moara cu noroc" by Ioan Slavici, in .pdf format, "Pădurea spânzuraților" by Liviu Rebreanu, in .pdf format, "Povestea lui Harap-Alb" by Ion Creangă, in .docx and .txt formats.

- RAG prompt:

  "—Instrucțiuni—:
  Scrie un răspuns prietenos și ușor de înțeles pentru întrebarea din cadrul câmpului —Întrebare—:
  Genereaza doar informațiile aferente câmpului —Răspuns—: fără să oferi context sau alte informații suplimentare
  Mai jos sunt trei exemple, —Exemplu 1—:, —Exemplu 2—:, și —Exemplu 3—:, care arată cum va trebui sa răspunzi întrebării din cadrul câmpului —Întrebare—:
  Încearcă să generezi răspunsul pe baza faptelor din câmpul —Context—:. Dacă în datele pe care ai fost antrenat nu se află informațiile necesare pentru a răspunde la —Întrebare—: atunci generează mesajul "Nu pot oferi un răspuns concret."

  —Exemplu 1—:
  Întrebare: Ce este schimbarea climatică?
  Răspuns: Schimbarea climatică este atunci când vremea planetei noastre se modifică pe o

perioadă lungă, de obicei din cauza activităților umane, cum ar fi arderea combustibililor fosili.

—Exemplu 2—:
Întrebare: Care este capitala României?
Răspuns: Capitala României este București.

—Exemplu 3—:
Întrebare: Cum funcționează energia solară?
Răspuns: Energia solară funcționează prin captarea luminii de la soare folosind panouri solare. Aceste panouri transformă lumina în electricitate pe care o putem folosi acasă.

—Context—:
{context}
—Întrebare—: {query} Care sunt maximum top 3 documente relevante, cu nume diferite, pentru propoziția anterioară?
—Răspuns—:"

- LLM prompt:

"—Instrucțiuni—:
Scrie un răspuns prietenos și ușor de înțeles pentru întrebarea din cadrul câmpului —Întrebare—:
Genereaza doar informațiile aferente câmpului —Răspuns—: fără să oferi context sau alte informații suplimentare
Mai jos sunt trei exemple, —Exemplu 1—:, —Exemplu 2—:, și —Exemplu 3—:, care arată cum va trebui sa răspunzi întrebării din cadrul câmpului —Întrebare—:
Dacă în datele pe care ai fost antrenat nu se află informațiile necesare pentru a răspunde la —Întrebare—: atunci generează mesajul "Nu pot oferi un răspuns concret."

—Exemplu 1—:
Întrebare: Ce este schimbarea climatică?
Răspuns: Schimbarea climatică este atunci când vremea planetei noastre se modifică pe o perioadă lungă, de obicei din cauza activităților umane, cum ar fi arderea combustibililor fosili.

—Exemplu 2—:
Întrebare: Care este capitala României?
Răspuns: Capitala României este București.

—Exemplu 3—:
Întrebare: Cum funcționează energia solară?
Răspuns: Energia solară funcționează prin captarea luminii de la soare folosind panouri solare. Aceste panouri transformă lumina în electricitate pe care o putem folosi acasă.

—Întrebare—: {query} Care sunt maximum top 3 documente relevante, cu nume diferite, pentru propoziția anterioară?
—Răspuns—:"