Literature Review: REPLUG: Retrieval-Augmented Black-Box Language Models

Gilles de Waha | 330072 | gilles.dewaha-baillonville@epfl.ch
MoreNectarLessPoison

1 Summary

This paper introduces REPLUG, a new framework to do Retrieval Augmentation for Language Models. Compared to the previous contributions, it treats the LM as a black-box and improves its performances with a frozen or trainable retriever.

RAG (Retrieval Augmented Generation) has emerged as an effective way to improve the performance of LLMs on knowledge intensive NLP tasks (Patrick Lewis, 2020). In addition to parametric memory, RAG models have a non-parametric memory that consists of retrieved passages from an external dataset according to the prompt and a pre-trained retriever.

In many of the past approaches, the LM needed to be retrained to integrate the retrievals (that are frozen) with chunked cross-attention (Sebastian Borgeaud, 2022). This makes RAG really costly to use with large models.

REPLUG (REtrieve and PLUG) takes a different approach by having a retriever that can be fine-tuned and a LM that is used as a black box. Given a prompt, the retrieved passages are concatenated to it and fed to the model to generate the prediction. Moreover, REPLUG adresses the problem of having a limited input size by a method that encodes in parallel all the retrieved passages with the same black-box LM. Concretely, given a prompt, the retriever collects the top-k most similar documents, by mapping the prompt and each document to an embedding and comparing their cosine similarity. Then since all documents can not be directly concatenated to the input, they are separately fed to the black-box model with the prompt. This gives k probability distributions over the vocabulary for the next token that are averaged with the similarities between the prompt and the documents as coefficients.

Finally, the paper presents REPLUG LSR (REPLUG with LM-Supervised Retrieval) that adapts the retriever, that outputs the most similar documents, by using the LM itself. Instead of only considering the prompt likelihood of each document, REPLUG LSR feeds the LM with the input concatenated with each document separately, and takes the probability that the LM predicts the ground truth. This way, we also have the LM likelihood for the document. Bringing everything together, the retriever is fine-tuned with the objective of minimizing the KL divergence between the prompt and the LM likelihood.

REPLUG can be integrated with any LM and consistently enhanced their performance. It has also shown to improve the accuracy on Open Domain QA tasks.

2 Strengths

The paper is clear and well written. Moreover, it is detailed enough to be able to reproduce it.

The main contribution is the development of a framework that allows to use RAG with very large language models at a low cost by using them as black-box (and not having to fine-tune them). The fact that REPLUG can be used with almost any retriever and LM is definitely a strength.

It is also interesting to see how they managed to get around the fixed input size of the LM by inputting each retrieved document in parallel. Moreover, the fine-tuning of the retriever not only according to the similarity between the prompt and the documents, but also to how much the document reduces the perplexity of the LM, is an innovative approach.

The study conducts comprehensive evaluations of the framework, with experiments on various models and tasks. Furthermore, a robust qualitative and quantitative analysis of the results has been made by the researchers to ensure that the improvements are due to REPLUG.

3 Weaknesses

As stated in the paper, the first weakness is the lack of interpretability of REPLUG. It is unclear when the LM bases its prediction on retrieved or parametric knowledge. From the qualitative analysis, we can guess that retrieved documents are the most useful with rare tokens but it is not enough.

Then, a major weakness is the lack of comparison with other RAG models of the same size that use white-box LM and fine-tune them. In the paper, the only comparison made is between Atlas (a RAG model with 11B parameters) and Codex+REPLUG (175B parameters). It would have been interesting to see which methods works best when having models of the same size, without considering the training cost.

Furthermore, it would have been interesting to assess the performance of the retriever before and after its training. Either qualitatively by reviewing the retrieved documents with the same prompt in the two cases or quantitatively by analyzing the similarity between the two. This way, it could highlight even more (or not) the usefulness of REPLUG and its strategy of only fine-tuning the retriever instead of the LM.

Finally, a key aspect of REPLUG that is not discussed is its computational cost. When training the retriever, REPLUG LSR feeds each document to the LM to determine by how much it decreases its perplexity. This severely restricts the size of the external dataset used to retrieve documents. Moreover, at inference, the LM makes k predictions for the k retrieved documents that are concatenated to the prompt. So given that REPLUG is meant to be used with large LM, it would have been necessary to evaluate how well it scales with the size of dataset we can retrieve from.

In summary, the paper is a significant contribution to the field of RAG models. It provides a framework to use RAG with large models at a lower cost. However, it lacks comparison with previous RAG models of the same size, more interpretation and an evaluation on how it scales with the number of retrievable documents.

References

Aleksandra Piktus Fabio Petroni Vladimir Karpukhin Naman Goyal Heinrich Küttler Mike Lewis Wen-tau Yih Tim Rocktäschel Sebastian Riedel Douwe Kiela Patrick Lewis, Ethan Perez. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Arthur Mensch Sebastian Borgeaud. 2022. Improving language models by retrieving from trillions of tokens. *DeepMind*.