CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Evaluation of In-Context Retrieval Augmented Language Models for Factual Consistency

Master's thesis in Computer science and engineering

YURA UENO

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Evaluation of In-Context Retrieval Augmented Language Models for Factual Consistency

YURA UENO

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Evaluation of In-Context Retrieval Augmented Language Models for Factual Consistency

YURA UENO

Supervisor: Lovisa Hagström, Department of Computer Science and Engineering
Examiner: Richard Johansson, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2024

Evaluation of In-Context Retrieval Augmented Language Models for Factual Consistency

YURA UENO
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Pre-trained large language models (LLMs) have shown remarkable performance in natural language processing (NLP) tasks, especially in question-answering. However, these models face challenges such as limited memory expansion, interpretability issues, and susceptibility to hallucinations. To address these limitations, Retrieval-Augmented Language Models (RALMs), which integrate parametric and non-parametric memory, have been proposed. These models use a retriever to access external knowledge bases, enhancing memory flexibility and interpretability. Although RALMs have been shown to outperform pre-trained parametric-only models in various knowledge-intensive NLP tasks, one caveat with RALMs studied in the majority of the previous research is that they rely on fine-tuning the retrieval-augment architectures to downstream NLP tasks, which can be costly and difficult. To address this challenge, Ram et al. (2023) have recently introduced a simpler alternative called In-Context RALM, which simply prepends retrieved documents to the input and feeds the input to existing pre-trained language models without any further fine-tuning. Considering the importance of predictions being not only accurate but also consistent, this study evaluates In-Context RALM's effectiveness in prediction consistency compared to a parametric-only model (Llama-2-7B) and a fine-tuned RALM (Atlas). Results show that In-Context RALM produces more consistent predictions than the parametric-only model, demonstrating its capability to enhance consistency. Although it is less effective than the fine-tuned RALM (Atlas) in improving consistency, In-Context RALM remains a viable alternative when fine-tuning is impractical, particularly if retrieved contexts are relevant. However, its performance declines with irrelevant contexts, making it less robust in such scenarios compared to fine-tuned models. These findings highlight In-Context RALM's potential to improve the robustness to be a more competitive alternative to fine-tuned RALMs.

Keywords: NLP, RALM, In-Context RALM, RAG, information retrieval, retrieval-augmented generation, LLM.

Acknowledgements

I am deeply grateful to all those who have supported me during the course of this research endeavor. Foremost, my heartfelt appreciation goes to my supervisor, Lovisa Hagström, whose unwavering support and guidance have been invaluable throughout this thesis. Her insightful advice and consistent feedback have been essential in shaping and completing this project. I am also thankful to my examiner, Richard Johansson, for his invaluable input and direction on this study as well. Additionally, I want to thank my friends who helped me get through my thesis project by sending me many caring words and cheers and encouraging me to stay strong. Finally, I would like to express my profound gratitude to my parents for giving me continuous support and encouragement throughout my study at the University of Gothenburg, including the thesis project.

Yura Ueno, Gothenburg, 2024-06-26

Contents

List of Figures

List of Tables

1

Introduction

Pre-trained large language models (LLMs) such as GPT models by Open AI (Brown et al., 2020) and Llama 2 by Meta (Touvron et al., 2023a) have demonstrated astonishing performance in Natural Language Processing (NLP) tasks, particularly in question-answering tasks. These pre-trained large language models carry out question-answering tasks without accessing an external memory as they store a substantial amount of information within the parameters of the models (Lewis et al., 2020; Petroni et al., 2019).

While this development is valuable, modern LLMs have some issues. Firstly, they cannot easily expand and modify their memory as it is stored in their trained parameters. Secondly, it is not straightforward to interpret their predictions as we cannot know why the models returned the answers they did to questions. Lastly, they have been criticized for sometimes producing "hallucinations", seemingly factual yet incorrect predictions (Chen et al., 2023; Lewis et al., 2020; Marcus, 2020).

To overcome these issues with pre-trained LLMs, Retrieval-Augmented Language Models (RALM), which combine parametric memory and non-parametric memory (i.e. retrieval-based memory) have been introduced (Izacard and Grave, 2021; Lewis et al., 2020). RALM consists of (i) a retriever, which retrieves relevant information from an external knowledge base such as a vector database, and (ii) a language model, which generates answers using the information retrieved (See Figure 1.1). Given this architecture, such hybrid models can predict answers to questions using not only the information stored in the model parameters but also the information retrieved from an external non-parametric memory such as Wikipedia. Because RALMs can utilize information stored in an external non-parametric memory, their knowledge can easily be extended and modified, and the knowledge used for the prediction can be inspected and interpreted. Most importantly, previous research has demonstrated RALMs to outperform pre-trained parametric-only models in various knowledge-intensive NLP tasks as well as to mitigate hallucinations and inconsistencies (Izacard et al., 2022; Lewis et al., 2020; Shuster et al., 2021; Hagström et al., 2023).

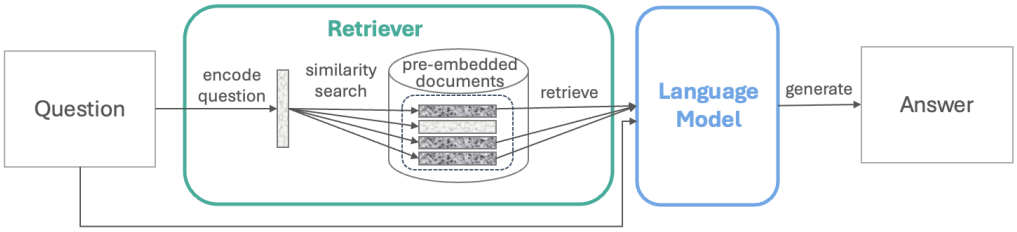

Figure 1.1: Overview of RALM

One caveat with RALMs studied in the majority of the previous research is that the models rely on fine-tuning the retrieval-augmented architectures to downstream NLP tasks (Ram et al., 2023). For instance, Lewis et al. (2020) and Izacard and Grave (2021) fine-tuned the encoder-decoder architectures in their RALM to downstream knowledge-intensive tasks, and Izacard et al. (2022) investigated various ways of pretraining such models and developed Atlas. Considering that such fine-tuning can be difficult, costly, and also not feasible for LLMs that are accessible only via an API, having to fine-tune the retrieval-augment architectures could be an obstacle for RALMs to be widely deployed (Ram et al., 2023).

To address this challenge, Ram et al. (2023) have recently introduced a simpler alternative called *In-Context RALM*, which simply prepends retrieved documents to the input and feeds the input to existing pre-trained language models without any further fine-tuning. They evaluated the effectiveness of In-Context RALM with five diverse language modeling datasets and also with two common open-domain question-answering datasets, demonstrating that In-Context RALM reduced the perplexity of the language models to the same level as a 2-3 times larger model without retrieval augmentation and that In-Cotext RALM provides a substantial performance gain in question-answering tasks.

In addition to having low perplexity and high accuracy, being able to return consistent outputs from lexically different yet semantically equivalent queries is essential for language models to be robust and reliable in fact-critical situations(Hagström et al., 2023). In fact, inconsistency in language models is shown to indicate the tendency of hallucinations and unexpected fragility in what models generate (Ji et al., 2023). As for fine-tuned RALM, Hagström et al. (2023) have demonstrated that retrieval augmentation was more effective than upscaling LLMs for increasing consistency; they showed that Atlas, a fine-tuned RALM, yielded the same level of performance as LLaMA-65B despite being 90 times smaller. However, the effectiveness of In-Context RALM for achieving consistency is yet to be evaluated. Therefore, it is necessary to evaluate In-Context RALM in terms of consistency as well in order to further validate its effectiveness as an alternative to fine-tuned RALM.

1.1 Research Questions

Considering that high consistency as well as high accuracy and low perplexity is vital for language models to be robust and reliable, this study investigates the performance of In-Context RALM with respect to consistency. Specifically, the evaluation of In-Context RALM is carried out in comparison to parametric-only language models (i.e. language models without contexts) and fine-tuned RALM such as Atlas (Izacard et al., 2023), and the study answers the research questions below.

1. Does In-Context RALM produce more consistent predictions compared to a parametric-only model (i.e. the same LM but without contexts)?
2. Is In-Context RALM more effective in increasing the consistency of predictions compared to the fine-tuned RALM, Atlas?
3. How does the quality of the provided context affect the performance of In-Context RALM in terms of consistency?

1.2 Contributions

By answering the research questions above, this study aims to contribute to further research with additional insights on the effectiveness of In-Context RALM as a means to mitigate inconsistency in outputs from language models. The first and the second research questions will help us better understand whether In-Context RALM is an effective architecture for reducing inconsistency in language models outputs compared to parametric-only LLMs and fine-tuned RALMs. The third research question will lead us to understand how much the performance of In-Context RALM depends on the retrievers or the quality of retrieved documents used as contexts for answering questions.

2

Theory

The key components of RALMs is a retriever, which retrieves relevant information from an external memory, and a language model, which generates answers using the retrieved information. This chapter first explains what language models and retrievers are as knowing those would be fundamental for understanding RALMs. As for language models, this chapter provides further explanations of the transformer as it is a key innovation in modern natural language processing and a foundation for state-of-the-art models. The language models used in this thesis (i.e. Llama-2-7B and T5 seq2seq model) are also based on the transformer architecture. After introducing the fundamentals of language models and retrievers, this chapter provides explanations of RALMs and In-Context RALMs.

2.1 Language Model

Language models are probabilistic models of natural language. They model probabilities of a next word given previous words and generate outputs. They are foundational in many natural language processing (NLP) tasks, from translation and summarization to question-answering and conversational agents. This section provides an overview of different types of language models and how they work.

2.1.1 Two Categories of Language Models

Language models can be categorized into two main types: word-count-based models and deep neural network-based models.

1. Word-Count-Based Language Models

Word-count-based language models predict the next word in a sequence based on the frequencies of word combinations in a given text corpus. A fundamental way to build such language model is to calculate n-gram probabilities.

N-gram Models: These models use the probability of the last n words to predict the next word (Jurafsky and Martin, 2019). For example, in a trigram model (n=3), the next word is predicted based on the previous two words. These models are simple and effective for many applications but can struggle with long-range dependencies due to their limited context window.

2. Neural Language Models:

Neural Language Models use neural networks to learn from large amounts of text data and can capture more complex patterns and dependencies. Neural networks are universal approximators in the sense that they can be trained to model any function, given sufficient capacity (Hornik et al., 1989). This contrasts with n-gram models, which are inherently limited to modeling only a limited subset of simple functions. Consequently, neural networks are more versatile and powerful in capturing complex patterns and relationships within data.

Recurrent Neural Networks (RNNs): RNNs are a type of neural network designed for sequential data processing, such as time series or sentences (Goodfellow et al., 2016). They maintain a hidden state that evolves over time, capturing information about previous elements in the sequence. This hidden state allows RNNs to retain context from earlier inputs, making them suitable for tasks where order and dependencies between elements matter. However, RNNs face challenges with long-term dependencies due to the vanishing gradient problem, where gradients diminish as they propagate back through time. This limits their ability to effectively capture relationships between distant elements in the sequence.

Long Short-Term Memory Networks (LSTMs): Long Short-Term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) are a specialized type of RNN designed to address the shortcomings of traditional RNNs in handling long-term dependencies. LSTMs maintain a cell state that runs through the entire sequence and is updated through carefully controlled mechanisms called gates. These gates, including the forget gate, input gate, and output gate, regulate the flow of information into and out of the cell state. The forget gate decides which information to discard from the cell state, the input gate determines what new information to store, and the output gate controls what information to pass to the next time step. By explicitly managing the flow of information, LSTMs can maintain information over long sequences, making them effective for tasks like speech recognition, language modeling, and sentiment analysis. That being said, as LSTMs still process data sequentially, they still face the challenge of forgetting information that is too far back in the sequence.

Transformers: The Transformer is a groundbreaking neural network architecture introduced by Vaswani et al. in their paper *Attention is All You Need* (Vaswani et al., 2017). Unlike previous models designed for sequence-to-sequence tasks, such as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), the Transformer leverages an attention mechanism that allows for significantly more parallelization and better handling of long-range dependencies. This has led to substantial advancements in natural language processing (NLP) and related fields. The following section explains transformers in more detail as the models used in our study (i.e. Llama-2-7B and the language model in Atlas) are based on this architecture.

2.2 Transformers

Transformers leverage a self-attention mechanism that allows them to capture relationships between words in a sequence regardless of their positions. This mechanism enables the model to attend to different parts of the input simultaneously, making it highly efficient for tasks requiring long-range dependencies and context understanding. Transformers are structured with multiple layers of self-attention and feedforward neural networks, enabling them to learn complex patterns in data, thereby achieving state-of-the-art results across diverse natural language processing tasks (See Figure 2.1).

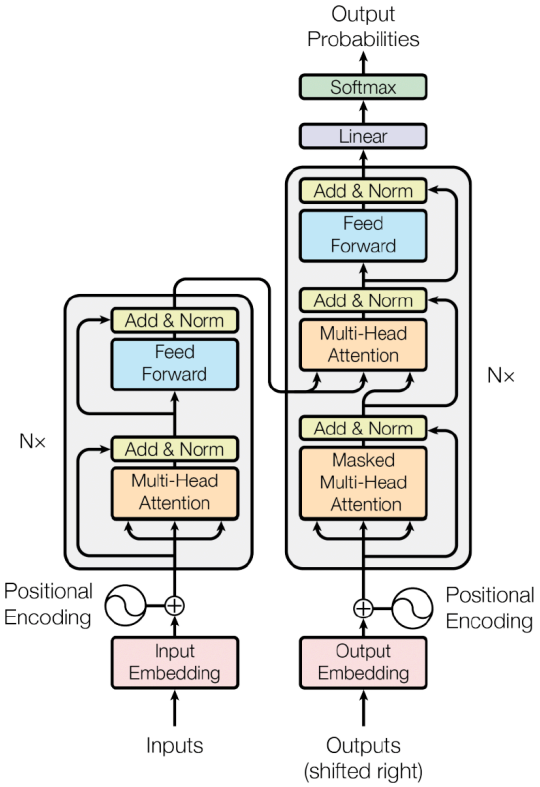

Figure 2.1: Overview of Transformers (Vaswani et al., 2017)

2.2.1 Core Components

Attention Mechanism

At the heart of the transformer is the self-attention mechanism, which enables the model to weigh the importance of different words in a sentence relative to each other, irrespective of their positions. This is achieved through the scaled dot-product attention, which operates as follows:

- **Query, Key, and Value Vectors:** For each word in the input, three vectors are derived through learned linear transformations: the query (Q), the key (K), and the value (V).

- **Attention Scores:** The attention score for a pair of words is computed by taking the dot product of their query and key vectors, then scaling and normalizing using the softmax function.
- **Weighted Sum:** The output for each word is a weighted sum of the value vectors, with the weights being the attention scores.

Mathematically, this can be expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

where d_k is the dimension of the key vectors.

Multi-Head Attention

The transformer employs multiple attention mechanisms, known as heads, to capture various relationships between words. Each head independently performs the attention operation and outputs a set of vectors. These vectors are concatenated and linearly transformed to produce the final output. This multi-head approach allows the model to focus on different parts of the sentence simultaneously, providing a richer representation.

Positional Encoding

Since the transformer lacks the sequential nature of RNNs, it incorporates positional encodings to provide information about the order of words. These encodings are added to the input embeddings and are derived from sinusoidal functions of different frequencies. Formally, for a position pos and dimension i:

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.2}$$

$$\text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.3}$$

where d_{model} is the dimension of the embeddings.

Encoder-Decoder Architecture

The transformer follows an encoder-decoder structure, which is fundamental for sequence-to-sequence tasks like translation and summarization.

- **Encoder:** The encoder consists of a stack of identical layers, each containing two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Each sub-layer is wrapped with residual connections followed by layer normalization.
- **Decoder:** The decoder also consists of a stack of identical layers but with an additional sub-layer that performs multi-head attention over the encoder's output. This allows the decoder to attend to the entire input sequence while generating each word.

2.2.2 Subsequent Developments of Transformers

Following its introduction, the transformer architecture has been adapted and extended in numerous ways, giving rise to powerful models such as T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020), LLaMA 2 (Large Language Model Meta AI) (Touvron et al., 2023b), and BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018).

2.2.2.1 T5 (Text-to-Text Transfer Transformer)

Architecture: T5, proposed by Raffel et al. (2020), adopts a unified encoder-decoder architecture for handling a wide array of NLP tasks. In T5, both the input and output are treated as sequences of tokens, allowing the model to frame all tasks as text-to-text transformations. The encoder processes the input sequence to extract contextualized representations, while the decoder generates the output sequence by attending over these representations. This architecture is composed of multiple layers of transformer encoder-decoder stacks, each layer equipped with self-attention mechanisms and feedforward neural networks.

Training Methodology: T5 leverages extensive pre-training on large-scale datasets like C4 (Colossal Clean Crawled Corpus), which comprises diverse text from the web. During pre-training, the model learns bidirectional representations of text, enabling it to understand and generate language comprehensively.

2.2.2.2 LLaMA 2 (Large Language Model Meta AI)

Architecture: LLaMA 2, developed by Touvron et al. (2023), is built upon a decoder-only transformer architecture, which optimizes the model specifically for autoregressive language modeling tasks. Unlike the original transformer architecture that includes both an encoder and a decoder, LLaMA 2 employs only the decoder part. This architecture consists of multiple layers of transformer decoder stacks, each equipped with masked multi-head self-attention mechanisms and position-wise feedforward neural networks. The masked self-attention ensures that each token can only attend to previous tokens in the sequence, maintaining the autoregressive nature of the model.

Training Methodology: LLaMA 2 is pre-trained on a diverse mixture of publicly available datasets, allowing it to learn complex patterns and relationships within the language. During pre-training, the model is trained to predict the next token in a sequence given the preceding tokens, which helps it develop a strong contextual understanding. This extensive pre-training on large-scale datasets enables LLaMA 2 to generate coherent and contextually relevant text, making it suitable for a wide range of NLP tasks such as text generation, completion, and interactive dialogue systems.

2.2.3 BERT (Bidirectional Encoder Representations from Transformers)

Architecture: BERT, introduced by Devlin et al. (2018), pioneered bidirectional training for transformer-based models in NLP. Unlike earlier models, BERT uses an encoder-only architecture that processes input sequences bidirectionally. Each layer of BERT's encoder consists of self-attention mechanisms and position-wise feedforward networks, allowing the model to capture deep contextual relationships in text. BERT's architecture focuses on encoding input sequences into contextualized representations, which are crucial for tasks such as semantic search, sentiment analysis, and named entity recognition.

Training Methodology: BERT achieves state-of-the-art performance through extensive pre-training on large-scale datasets like BookCorpus and Wikipedia, where the model learns bidirectional representations of text. This pre-training phase enables BERT to understand the intricate nuances of language and context across diverse domains.

2.3 Retrievers in Information Retrieval

Information retrieval is the process of obtaining information resources that are relevant to an information need from a collection of resources (Manning et al., 2008). In other words, it is the activity of finding the right information from a large amount of data. A retriever, in the context of information retrieval, refers to a system or a component of a system that performs the task of retrieving information (Baeza-Yates and Ribeiro-Neto, 2011). This could be a software program or algorithm designed to search through databases, documents, or other repositories of information to find items that match a user's query or information need. Retrieval systems typically work by indexing the content they are tasked to search through (i.e. retrieval corpus). Indexing involves analyzing and cataloging the content in a way that allows for efficient searching based on keywords, concepts, or other criteria. When a user submits a query, the retriever uses this index to quickly search and locate relevant documents. The rest of this section shares the examples of different kinds of retrievers as there are various ways of retrieving relevant information.

2.3.1 Examples of Retrievers

BM25: BM25 (Best Matching 25) is a probabilistic bag-of-words information retrieval model that ranks a set of documents based on the query terms appearing in each document (Robertson and Zaragoza, 2009). It is an extension of the more basic TF-IDF (Term Frequency-Inverse Document frequency) model and BM25 was designed to overcome the limitations of TF-IDF. Similar to TF-IDF, BM25 considers the frequency of terms in a document but BM25 applies a saturation function to the term frequency instead of using raw term frequencies like TF-IDF does. This helps the model avoid overemphasizing documents with excessively high term frequencies. BM25 also introduced a document length normalization factor to account for the

fact that longer documents tend to have higher term frequencies, which helps to mitigate the bias towards longer documents. Additionally, BM25 introduced tuning parameters while TF-IDF has fixed parameters for term frequency and inverse document frequency calculations, making BM25 more adaptable to different types of datasets. Overall, BM25 provides a more nuanced and context-aware approach by improving over the basic TF-IDF model.

DPR: Karpukhin et al.'s DPR retriever is a dense retriever; unlike sparse retrievres like BM25 that utilize bag-of-words methods, dense retrievers utilize dense vector embeddings generated using neural network models. The DPR uses two independent BERT (base, uncased) (Devlin et al., 2019) as encoders, one of which is used to encode source passages to vectors and build an index for all the M passages ($E_p(\cdot)$), and the other one is used to encode the input question to a vector and retrieve k passages that are the most similar to the question vector ($E_Q(\cdot)$). The similarity between the vectors of source passages and the vector of the input question is calculated using the dot product: $\text{sim}(q, p) = E_Q(q)^T E_p(p)$. These two encoders ($E_Q$ and E_p) are trained by optimizing their weights to maximize the dot product between $E_Q(q_i)$ and $E_p(p_i)$ for questions q_i and corresponding gold passages p_i from QA datasets such as Natural Questions (NQ; (Kwiatkowski et al., 2019)) or TriviaQA (Joshi et al., 2017). After the two encoders have been trained, we apply the passage encoder E_p to all the source passages and index them using FAISS (Johnson et al., 2019). Then at runtime, we encode an input question q into its embedding $v_q = E_Q(q)$ and retrieve the top k passages that are most similar to v_q.

Contriever: Just like DPR, the Contriever developed by Izacard et al. (2022) uses the BERT uncased base embedding models and encodes the query and the documents in the retrieval corpus independently using two separate encoders, having the dual-encoder architecture. The relevant documents are chosen by calculating the similarity between the query and documents using the dot product. Unlike the DPR that was pre-trained using supervised data, the Contriever model is pre-trained with unsupervised data using a self-supervised learning algorithm MoCo, or Momentum Contrast, (He et al., 2019).

2.4 Retrieval Augmented Language Model (RALM)

A Retrieval Augmented Language Model, or RALM, consists of (i) a retriever, which retrieves relevant information from an external memory, and (ii) a language model, which generates answers using the information retrieved. The external memory is usually initialized by embedding a chosen retrieval corpus, and the retriever retrieves relevant information from the embedded corpus by carrying out a similarity search between embedded queries and embedded retrieval corpus such as Wikipedia or some documentation, etc. A RALM combines the knowledge stored in the language models parameters and the knowledge in the external memory by conditioning the language model's generation over the relevant documents retrieved by the retriever.

Atlas One example of RALMs is Atlas developed by Izacard et al. (2023), which consists of a Contriever-based retriever and T5 seq2seq language model (See Figure 2.2). Given a query (x), the retriever in Atlas first embeds the query (x) using a query encoder (q). It then carries out a dot-product-based similarity search between the embedded query and pre-embedded documents in a retrieval corpus in order to retrieve the most relevant documents. Afterward, the T5 seq2seq language model generates the output based on the retrieved documents and the query utilizing a fusion-in-decoder approach.

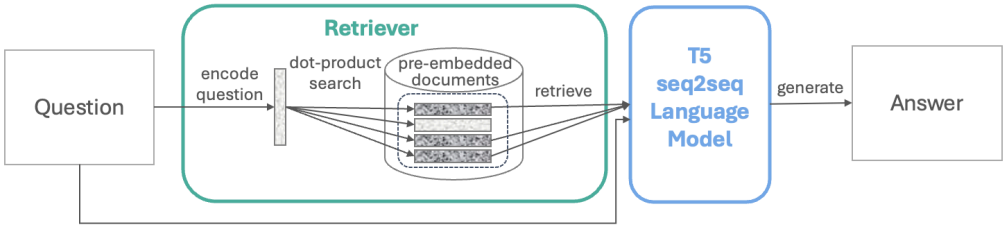

Figure 2.2: Overview of Atlas introduced by Izacard et al. (2023)

2.4.1 Advantages

An advantage of RALMs is that their knowledge can be more easily updated by extending and modifying the external non-parametric knowledge base (i.e. retrieval corpus) compared to parametric-only models that store knowledge in their parameters and require further training of the models for the knowledge to be updated. Another advantage of RALMs is that the knowledge used for the model predictions can be easily inspected by referring to the retrieved documents used for generating the outputs while this is not straightforward with parametric-only models. Additionally, RALMs have been shown to outperform parametric-only models in various NLP tasks from language generation to question-answering.

REALM (Retrieval-Augmented Language Model Pre-Training) (Guu et al., 2020) and ORQA (Latent Retrieval for Weakly Supervised Open Domain Question Answering) (Lee et al., 2019) are examples of RALMs that have shown promising results for successfully achieving and outperforming parametric-only models in open domain question answering tasks, which is a simple extractive question answering task, where the model is expected to answer a question by extracting information from the knowledge base. Lewis et al. (2020) have further tested the performance of RALMs not only with open-domain question answering but also with abstractive question answering, which differs from open-domain answering in that the model has to answer questions with free-form abstractive text generation instead of simply extracting answers from the knowledge base, showing RALMs to outperform parametric-only models. They have also tested RALMs with jeopardy question generation, in which the input is an entity and the output should be a fact about that entity, and with fact verification, which requires retrieving evidence from the data

source relating to a claim and then reasoning over this evidence to classify whether the claim is true, false, or unverifiable from the data source alone. Lewis et al. (2020) have shown that RALMs outperformed the parametric-only model BART in abstractive question answering, jeopardy question generation, and fact verification.

2.4.2 Fine-tuned RALM

The majority of previously studied RALMs depends on fine-tuning the retrieval-augmented architectures to downstream tasks. The retriever and the language model in Atlas are jointly pre-trained using an MLM (masked language modeling) task to be optimized to learn knowledge-intensive tasks with only a few examples. Lewis et al. (2020) also jointly trained the retriever and the language model to fine-tune their RALMs. Izacard and Grave (2021) fine-tuned the language models to the downstream datasets.

Such fine-tuning, however, could prevent RALMs from being widely deployed as fine-tuning can be difficult and costly, and sometimes not even possible due to not having access to language models parameters. This led Ram et al. (2023) to propose a simpler retrieval-augmentation architecture, In-Context RALMs.

2.4.3 In-Context RALM

In-Context RALMs proposed by Ram et al. (2023) is different from fine-tuned RALMs in that it uses a frozen large language model and an off-the-shelf retriever without jointly fine-tuning them; In-Context RALMs simply prepends retrieved documents to the input query and feeds them to its language model to get an output instead of fine-tuning the architecture (See Figure 2.3). Therefore, the LLMs weights are kept unchanged in In-Context RALMs.

Figure 2.3: Overview of In-Context RALM

Ram et al. (2023) investigated the performance of In-Context RALMs using Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) answering datasets. They used DPR as their retriever and LLaMA models with three different numbers of parameters (LLaMA-7B, 13B, and 33B) as their language model. As a result, they found that In-Context RALMs performed better than the LLaMA counterparts without retrieved documents (See Table 2.1). For instance, LLaMA 13B with retrieved documents performed better than LLaMA 13B in a closed-book setting by more than 18 points on NQ (from 12.0% to 31.0%) and more than 5 points on TriviaQA (from 54.8% to 60.1%), demonstrating substantial performance gains

achieved by prepending relevant documents retrieved by a general purpose retriever to input queries.

Table 2.1: Results of In-Context RALM on NQ and TriviaQA by Ram et al..

Model	Retrieval	NQ	TriviaQA
LLaMA-7B	-	10.3	47.5
	DPR	28.0	56.0
LLaMA-13B	-	12.0	54.8
	DPR	31.0	60.1
LLaMA-33B	-	13.7	58.3
	DPR	32.3	62.7

3

Methods

This study investigates the effectiveness of In-Context RALMs in improving the consistency of predictions. In order to measure the consistency of predictions, we use an evaluation task called ParaRel (Elazar et al., 2021), which allows us to measure consistency in outputs from language models. Using ParaRel, we compare the performance of an In-Context RALM with a parametric-only model and with a fine-tuned RALM in order to investigate whether the In-Context RALM is effective in mitigating inconsistency in comparison to other architectures. We also look into the performance of the In-Context RALM with different retrievers to study how much the performance of the In-Context RALM depends on the retrievers used in the In-Context RALM. This chapter first introduces the models and retrievers used in this study, and then explains how we measure consistency using an evaluation task called ParaRel. Lastly, we explain how we investigate the research questions.

3.1 Models and Contexts Used in the Evaluation

3.1.1 Models

For models, we compare the results of an In-Context RALM (Llama-2-7B with contexts) with a parametric-only model (Llama-2-7B) and with a fine-tuned RALM Atlas.

Parametric-Only Model We use Llama 2 with 7 billion parameters (Llama-2-7B) (Touvron et al., 2023b) as the parametric-only model to compare against the In-Context RALM. This represents the closed-book case, where no contexts are provided to the language model.

In-Context RALM In this study, we use Llama-2-7B as the language model of the In-Context RALM. When running a query from ParaRel, we prepend a context to the query and run the language model to answer the query with the prepended context. Although it is possible to prepend several retrieved passages to the query, we only prepend one passage due to the limited token size of the models and also due to the limited computational resources.

Fine-tuned RALM We use the base version of Atlas with 330M parameters as the fine-tuned RALM to compare against the In-Context RALM. As explained in

the theory, Atlas generates answers by combining information from the question and context utilizing a fusion-in-decoder approach.

3.1.2 Contexts

In this study, we experiment with contexts with different levels of quality; (1) golden contexts, which are the perfect contexts, (2) Atlas-retrieved contexts, which are retrieved using the retriever in Atlas and expected to be the medium-quality contexts, and (3) random contexts, which are the worst contexts (See Table 3.1 for the examples of contexts). All the contexts (golden, random, and Atlas-retrieved contexts) are retrieved from the Wikipedia 2017 passages dataset as this is the origin of the ParaRel* data.

(1) Golden Contexts For the ParaRel* evaluation dataset, we have the golden passage containing the correct information for each query. In our experiments, we treat the case of prepending these golden passages to the query as if we had the perfect retriever. For both In-Context RALM and the fine-tuned RALM Atlas, we provide one golden passage as a context.

(2) Atlas Contexts In order to have a more realistic quality of contexts, we also experiment with providing contexts retrieved by the retriever in Atlas. As explained in the theory, the Atlas retriever is a dense retriever initialized with the unsupervised Contriever model. As it is possible to use only the retriever part of Atlas, we use this retriever to choose relevant contexts. The quality of Atlas-retrieved contexts is not as perfect as the golden passages but is not as poor as the random contexts. For both In-Context RALM and the fine-tuned RALM Atlas, we provide one Atlas-retrieved passage as a context.

(3) Random Contexts We also experiment with providing a random irrelevant passage as a context and treat this case as if we had the worst retriever. The results from the random contexts allow us to learn how In-Context RALMs would perform when they have a retriever that is worst in quality.

It should be noted that we provide one random context when running as the In-Context RALM while several random passages were provided when running with the fine-tuned RALM Atlas. It would be more ideal to provide the same number of random contexts to make the results more directly comparable but we could not do that due to the time limitation of this study.

Table 3.1: Examples of Retrieved Contexts: golden, Atlas, and random contexts for the query "MessagePad, a product developed by [X]." The expected answer to fill in [X] is "Apple." The contexts below exemplify the difference in the level of accuracy and relevancy among the golden, Atlas, and random contexts; the golden context is precisely about the MessagePad, the Atlas context is slightly related but not exactly about the MessagePad, and the random context is completely irrelevant to the MessagePad.

Levels	Retrieved Contexts
Golden	The MessagePad is the first series of personal digital assistant devices developed by Apple Computer for the Newton platform in 1993.
Atlas	Breakpad (previously called Airbag) is an open-source replacement for Talkback. Developed by Google and Mozilla, it is used in current Mozilla products such as Firefox and Thunderbird. Its significance is being the first open source multi-platform crash reporting system. Since 2007, Breakpad is included in Firefox on Windows and Mac OS X, and Linux. Breakpad is typically paired with Socorro which receives and classifies crashes from users. Breakpad itself is only part of a crash reporting system, as it includes no reporting mechanism.
"Random	Recycling in the United Kingdom", "section": "Glass", "text": " for all types of waste in which large glass containers are located. There are now over 50,000 bottle banks in the United Kingdom, and 752,000 tons of glass are now recycled annually. The waste recycling industry in the UK cannot consume all of the recycled container glass that will become available over the coming years, mainly due to the colour imbalance between that which is manufactured and that which is consumed. The UK imports much more green glass in the form of Wine bottles than it uses, leading to a surplus amount for recycling. The resulting surplus of green glass from imported bottles may be exported to producing countries, or used locally in the growing diversity of secondary end uses for recycled glass.

3.2 Evaluation Task "ParaRel" for Measuring Consistency

ParaRel, developed by Elazar et al. (2021), builds upon LAMA (Petroni et al., 2019), an evaluation task relying on Wikidata to gauge factual knowledge in language models by prompting for subject-relation-object triples. ParaRel extends LAMA by adding semantically equivalent cloze-style prompts, enabling the evaluation of consistency in outputs from language models (See Figure 3.1). The core idea is that a model exhibits consistency if its responses remain unchanged despite variations in query wording. ParaRel assesses consistency for N-1 relations, where there is only

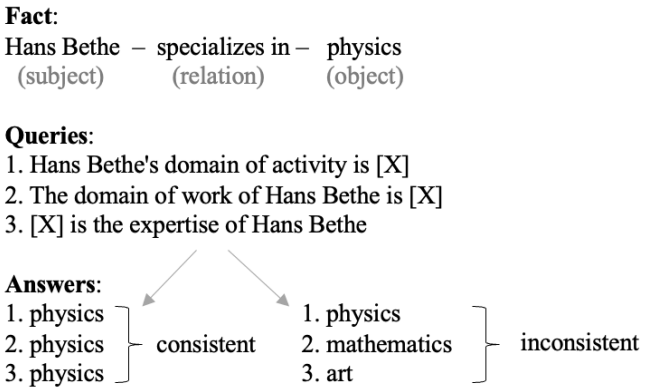

Figure 3.1: Overview of ParaRel

one correct object alternative per subject-relation pair, and plausibility for N-M relations, where multiple correct objects exist. For our evaluation, we focus solely on N-1 relations to gauge consistency. Furthermore, we use the improved version of ParaRel referred to as ParaRel*, which was developed by Hagström et al. (2023) by removing ambiguous fact duplicates. Additionally, we exclude the following 9 relations out of the 30 relations in ParaRel* given the issues listed in Table 3.2.

Relation	Name	Reasons for removal
P36	capital-of	The relation is redundant as similar questions are covered in P1376.
P101	specializes-in	There are semantic overlaps among possible answers.
P106	occupation	A person could hold several roles such as a diplomat and a politician.
P131	located-in	Inconsistent types of locations such as cities, states, and countries are given.
P138	named after	There are overlaps between subjects and correct answers.
P140	religion	Contains overlapping answers such as Catholicism, Christianity, Christian.
P279	subclass of	Overlap between subjects and the correct answers.
P1376	part-of	Overlap between subjects and the correct answers.
P176	produced-by	Computational difficulty when running the relation with golden passages.

Table 3.2: Removed Relations and Reasons for Removal

3.2.1 Evaluation Metrics

ParaRel offers several metrics for assessing model performance. We primarily focus on three key indicators: *Consistency*, which measures the agreement between prompts for each tuple; *Accuracy*, which indicates the accuracy of responses when using the original LAMA prompt; and *Consistent & Accurate*, which reflects the percentage of subjects that get correct objects assigned across all prompts. The consistency metric provides insights into all possible query pairs per tuple and relation. To derive a single consistency value per evaluated model, we compute the micro-average of consistency values across tuples for each relation, and then calculate the macro-average across relations, as outlined by Elazar et al. (2021).

3.3 Investigations of Research Questions

RQ1: Does the In-Context RALM perform better in mitigating inconsistency compared to parametric-only model? In order to investigate the first research question, we compare the performance of Llama-2-7B on ParaRel with that of the In-Context RALM (Llama-2-7B + contexts).

RQ2: Is the In-Context RALM more effective in increasing the consistency of predictions compared to the fine-tuned RALM, Atlas? Similarly, for the second research question, we compare the performance of Atlas with that of the In-Context RALM (Llama-2-7B + retrieved contexts). These comparisons lead us to understand if the In-Context RALM performs better in reducing inconsistency compared to parametric-only models and fine-tuned RALMs.

RQ3: How does the quality of context affect the performance of In-Context RALM in terms of consistency? For investigating the third research question, we compare the performance between Llama-2-7B with Atlas retrieved passages, Llama-2-7B with gold passages, and Llama-2-7B with random passages. The comparison allows us to understand how much the performance of In-Context RALM depends on the quality of retrievers/retrieved contexts. We also look at the difference in the performance of Atlas with different retrievers (i.e. Atlas-retriever, hypothetical perfect retriever, and hypothetical worst retriever) in order to understand how sensitive In-Context RALMs are to the quality of retrieved documents in comparison to fine-tuned RALMs.

3.3.1 Comparison using Hypothesis Testing & Cohen's d

When comparing the performance of Llama-2-7B with different retrieved contexts or among Atlas with different retrieved contexts, we carry out one-sided paired t-tests to test whether the accuracy or consistency between different models statistically differs or not. We set the significant level α to be $\alpha = 0.05$, accepting a 5% chance of incorrectly rejecting the null hypothesis.

In the case of rejecting the null hypothesis (i.e. concluding that there is a statistically significant difference in performance between two models), we evaluate the magnitude of the significance of the difference through Cohen's d effect size (Cohen, 1969). This is because we may sometimes have significance while the difference in results is practically small.

Cohen's d

Cohen's d is a statistical measure used to express the size of an effect in terms of standard deviation units; Cohen's d quantifies the difference between two means relative to the variability observed in the data. Cohen's d is calculated as $d = \frac{M_1 - M_2}{s_p}$, where M_1 is the mean of the first group, M_2 is the mean of the second group, and s_p

is the pooled standard deviation of the two groups. The pooled standard decision s_p is calculated as $s_p = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$.

According to Cohen, Cohen's d values can be interpreted as follows:

- 0.2: small effect size,
- 0.5: medium effect size,
- 0.8 or above: large effect size.

That being said, these thresholds are context-dependent and thus should be used with caution, but they provide a general guideline for interpreting the magnitude of an effect.

4

Results

This chapter shares the results of the experiments. First, we look at the results of Llama-2-7B with and without contexts in terms of accuracy and consistency, which are aimed at answering RQ1: "Does In-Context RALM perform better in mitigating inconsistency compared to parametric-only model?", and RQ3: "How does the quality of context affect the performance of In-Context RALM in terms of consistency?". We then look at the results of Atlas with and without contexts, and compare them with the results of Llama-2-7B in order to answer RQ2: "Is In-Context RALM more effective in increasing accuracy and consistency of predictions compared to the fine-tuned RALM, Atlas?"

4.1 Results of Llama-2-7B With and Without Contexts

This study evaluated whether prepending contexts to input queries improve the accuracy and consistency of Llama-2-7B in order to answer RQ1: "Does the In-Context RALM perform better in mitigating inconsistency compared to parametric-only model?". We also looked into how differently the accuracy and consistency improved for different qualities of retrieved contexts (i.e. golden, Atlas-retrieved, and random contexts) in order to answer RQ3: "How does the quality of context affect the performance of the In-Context RALM in terms of consistency?".

4.1.1 Accuracy

Table 4.1: Accuracy of Llama-2-7B With and Without Contexts on ParaRel. See Section 3.1.2 for the explanations of different contexts and Section 3.2.1 for the explanations of evaluation metrics. The p-values and the Cohen's d are both measured against the base-case (i.e. the without context case).

Model	Accuracy	p-value	Cohen's d
Llama-2-7B without context	0.63 ± 0.19	–	–
Llama-2-7B with Atlas context	0.75 ± 0.11	0.00	0.77
Llama-2-7B with golden context	0.82 ± 0.14	0.00	1.16
Llama-2-7B with random context	0.52 ± 0.21	0.00	-0.51

When given Atlas-retrieved contexts, the average accuracy increased statistically significantly from 0.63 to 0.75 with a moderate effect size. The accuracy further improved with golden contexts to 0.82 with a large effect size, indicating a more substantial accuracy gain from more accurate contexts. In contrast, using random contexts led to a statistically significant decrease in accuracy in comparison to the without-context case. These results suggest that In-Context RALM can achieve higher accuracy than parametric-only models. However, the performance can be significantly degraded and become even worse than without contexts if the retrieved contexts are not relevant, which is in accordance with prior research (Yoran et al., 2024).

4.1.2 Consistency

Table 4.2: Consistency of Llama-2-7B With and Without Contexts on ParaRel. See Section 3.1.2 for the explanations of different contexts and Section 3.2.1 for the explanations of evaluation metrics. The p-values and the Cohen's d are both measured against the base-case (i.e. the without context case).

Model	Consistency	p-value	Cohen's d
Llama-2-7B without context	0.58 ± 0.19	–	–
Llama-2-7B with Atlas context	0.62 ± 0.15	0.04	0.22
Llama-2-7B with golden context	0.76 ± 0.14	0.00	1.04
Llama-2-7B with random context	0.52 ± 0.18	0.00	−0.33

When prepending Atlas-retrieved contexts, the average consistency of Llama-2-7B increased statistically significantly but with a small effect size, indicating limited practical performance gain. In contrast, when prepending golden contexts, the average consistency increased with a far larger effect size ($d = 1.04$). This result suggests that Llama-2-7B gained more consistency when provided with more accurate contexts compared to Atlas-retrieved contexts. That being said, the larger consistency gain can also be partially attributed to the fact that golden contexts were always consistent across all the paraphrases while Atlas-retrieved contexts were not.

Conversely, when random passages were used as contexts, the average consistency of Llama-2-7B decreased statistically significantly compared to Llama-2-7B without contexts. This indicates that irrelevant contexts can harm the consistency of the language model and can make it worse than without contexts. It is also worth mentioning that random contexts were fixed across the paraphrases, and thus, the loss of consistency is primarily due to the provided contexts being irrelevant.

4.2 Results of Atlas With and Without Contexts

This study also evaluated the accuracy and consistency gained from providing contexts in Atlas, a fine-tuned RALM, in order to answer RQ2: "Is the In-Context

RALM more effective in increasing accuracy and consistency of predictions compared to the fine-tuned RALM, Atlas?". This section shares the results of Atlas with and without contexts.

4.2.1 Accuracy

Table 4.3: Accuracy of Atlas With and Without Contexts on ParaRel. See Section 3.1.2 for the explanations of different contexts and Section 3.2.1 for the explanations of evaluation metrics. The p-values and the Cohen's d are both measured against the base-case (i.e. the without context case).

Model	Accuracy	p-value	Cohen's d
Atlas without context	0.41 ± 0.21	–	–
Atlas with Atlas context	0.70 ± 0.10	0.00	1.80
Atlas with golden context	0.92 ± 0.04	0.00	3.45
Atlas with random context	0.42 ± 0.20	0.00	0.08

When given Atlas-retrieved contexts, the average accuracy significantly increased compared to Atlas without contexts. The accuracy further increased when using golden contexts, underscoring the effectiveness of fine-tuned RALMs in enhancing accuracy. With random contexts provided, on the other hand, the accuracy interestingly did not decline significantly, demonstrating the resilience of Atlas against irrelevant contexts.

4.2.2 Consistency

Table 4.4: Consistency of Atlas With and Without Contexts on ParaRel. See Section 3.1.2 for the explanations of different contexts and Section 3.2.1 for the explanations of evaluation metrics. The p-values and the Cohen's d are both measured against the base-case (i.e. the without context case).

Model	Consistency	p-value	Cohen's d
Atlas without context	0.59 ± 0.21	–	–
Atlas with Atlas context	0.63 ± 0.15	0.15	0.22
Atlas with golden context	0.86 ± 0.12	0.00	1.59
Atlas with random context	0.57 ± 0.23	0.04	-0.09

When receiving Atlas-retrieved contexts, the average consistency did not increase statistically significantly. When receiving golden contexts, in contrast, the average consistency increased statistically significantly with a large effect size. These results indicate that the fine-tuned RALM can considerably enhance consistency when provided with precise and pertinent contexts. Moreover, similar to the accuracy results, the consistency did not decrease statistically significantly even with random contexts, further underscoring the robustness of Atlas against irrelevant information.

4.3 Comparison between Llama-2-7B and Atlas

In order to answer RQ2: "Is In-Context RALM more effective in increasing accuracy and consistency of predictions compared to the fine-tuned RALM, Atlas?", this study compares the performance gain in accuracy and consistency between Atlas and Llama-2-7B.

Table 4.5: Accuracy & Consistency of Llama-2-7B With and Without Contexts Compared with Atlas With and Without Contexts.

Model	Accuracy	Cohen's d	Consistency	Cohen's d
Llama-2-7B without context	0.63	–	0.58	–
Llama-2-7B with Atlas context	0.75	0.77	0.62	0.22
Llama-2-7B with golden context	0.82	1.16	0.76	1.04
Llama-2-7B with random context	0.52	−0.51	0.52	−0.33
Atlas without context	0.41	–	0.59	–
Atlas with Atlas context	0.70	1.80	0.63	0.22
Atlas with golden context	0.92	3.45	0.86	1.59
Atlas with random context	0.42	0.08	0.57	−0.09

4.3.1 Accuracy

While Atlas without contexts had far worse accuracy than Llama-2-7B without contexts, Atlas with golden contexts achieved higher accuracy than Llama-2-7B with golden contexts. Additionally, the accuracy of Atlas did not decline when provided with random contexts while it did for Llama-2-7B when compared to without-context cases. These results indicate that the fine-tuned RALM Atlas is more effective than In-Context RALM with Llama-2-7B in improving the accuracy of predictions while being more robust against irrelevant contexts. Furthermore, the effect sizes measured in Cohen's d were larger for fine-tuned RALM Atlas than In-Context RALM with Llama-2-7B, indicating that the fine-tuned RALM Atlas has achieved a larger gain in accuracy from Atlas-retrieved contexts and golden contexts compared to Llama-2-7B.

4.3.2 Consistency

When it comes to consistency, neither Atlas nor Llama-2-7B yielded statistically significant increases in consistency when provided with Atlas-retrieved contexts. On the other hand, both achieved significant increases in consistency when provided with golden contexts, with Atlas achieving far higher consistency with a larger effect size than Llama-2-7B despite them having a similar level of consistency when no contexts were provided. Conversely, when provided with random contexts, Llama-2-7B had a larger loss in consistency than Atlas did. These results suggest that the fine-tuned RALM Atlas is more effective than the In-Context RALM with Llama-2-7B in enhancing the consistency of predictions from relevant contexts while being more resilient against irrelevant contexts.

5

Discussion

This chapter discusses the answers to our research questions based on our results. Additionally, it delves deeper into how In-Context RALMs can be improved according to other research. Although those prior researchers have looked at the potential ways to improve In-Context RALMs in terms of accuracy, their suggested solutions could also be applicable to improving In-Context RALMs in terms of consistency, and thus, can shed light on what future research would be valuable for further improvement of In-Context RALMs.

5.1 RQ1: Does the In-Context RALM produce more consistent predictions compared to the parametric-only model?

Our results suggest that the In-Context RALM (Llama-2-7B with contexts) produces more accurate and consistent predictions compared to the parametric-only model (Llama-2-7B without contexts) when the retrieved contexts are accurate and relevant. However, if the quality of these contexts cannot be guaranteed, it may be preferable to avoid using In-Context RALM, as providing poor contexts can negatively impact the accuracy and consistency of the language model's predictions.

5.2 RQ2: Is the In-Context RALM more effective in increasing the consistency of predictions compared to the fine-tuned RALM, Atlas?

Our findings indicate that the fine-tuned RALM (Atlas) is more effective at enhancing the accuracy and the consistency of predictions, and demonstrates greater robustness against irrelevant contexts compared to the In-Context RALM (Llama-2-7B with contexts). Notably, the fine-tuned RALM Atlas used in this study has only 330 million parameters, while Llama-2-7B has 7 billion parameters. The fact that the far smaller Atlas overperformed the In-Context RALM with Llama2-7B also implies that fine-tuning is considerably more effective in leveraging contexts and it can enable small language models to achieve competitive or even better performance than far larger language models with in-context learning. Therefore, fine-tuning

RALM generally appears preferable for achieving higher accuracy and consistency in outputs when investment in the cost of fine-tuning is feasible. That being said, in scenarios where fine-tuning is too costly or impractical, our results also support that In-Context RALMs could serve as a viable alternative, provided that high-quality contexts are guaranteed. As there is a clear tradeoff between the cost of training and performance when comparing fine-tuned RALMs to In-Context RALMs, the selection of the appropriate model should be considered along with the specific needs and constraints of each case.

5.3 RQ3: How does the quality of the provided context affect the performance of the In-Context RALM in terms of consistency?

Our results suggest that more accurate and relevant retrieved contexts result in a larger performance gain from the In-Context RALM. Additionally, irrelevant contexts seem to harm the accuracy and the consistency of the In-Context RALM and lead the LM to produce less accurate and consistent predictions than the without-context setting.

5.4 How to Improve In-Context RALMs

Although our results suggest that the fine-tuned RALM is more effective than the In-Context RALM in improving consistency, this does not necessarily mean that we should not utilize In-Context RALMs; there are cases where it is not feasible to invest in fine-tuning RALMs due to its cost and limited resources. Our results rather indicate the necessity of improving In-Context RALMs to achieve equivalent or higher performance as fine-tuned RALMs so In-Context RALMs can be competitive alternatives to fine-tuned RALMs.

Prior studies regarding the improvement of In-Context RALMs can be helpful in bringing some insights on how we should navigate the future improvement of In-Context RALMs. Although prior research has explored different ways to improve the effectiveness and robustness of In-Context RALMs in terms of accuracy, the suggested approaches could also apply to improving In-Context RALMs in terms of consistency.

5.4.1 Filtering Out Irrelevant Contexts with Natural Language Inference (NLI) Models

Using Natural Language Inference (NLI) models to filter out irrelevant contexts has been shown to help make In-Context RALMs more robust. NLI models can determine whether a textual *hypothesis* is entailed (true), neutral, or contradicted given a textual *premise*. Accordingly, given a question and its retrieved-context, an NLI model can determine whether the answer to the question (*hypothesis*) is

supported by the context (*premise*). Yoran et al. (2024) have shown that using NLI models to filter out retrieved contexts that do not entail question-answer pairs is effective in identifying irrelevant contexts and can help improve the robustness of In-Context RALMs. That being said, they also pointed out the cost of occasionally losing helpful contexts since NLI models can be too strict and can result in filtering out relevant contexts as well.

5.4.2 Larger LMs are More Proficient at In-Context Learning

Brown et al. (2020) have indicated that larger language models are more proficient in in-context learning and make more efficient use of in-context information. For instance, in their study, GPT-3 model with one-shot learning (i.e. one context is provided) achieved around 44.5 in the TrivialQA task when the model had 6.7 billion parameters while it caught up to 68.0, the equivalent performance of a fine-tuned RAG, when the model had 175 billion parameters. Therefore, if we had experimented with larger language models such as Llama-2-70B, we might have seen more performance gain from relevant contexts than we did with Llama-2-7B.

5.4.3 Prompt Wording Affects In-Context Learning

Wu et al. (2024) have demonstrated that prompt wording also has an impact on in-context learning. Using "strict" prompt wording (e.g. "You MUST absolutely strictly adhere to the context") resulted in LMs being more adherent to the provided contexts while "loose" prompt wording resulted in more reasonable judgments of provided contexts. Although their research does not necessarily suggest how the contexts should be provided in the prompt, it provides a helpful insight into how prompt wording can affect the degree to which In-Context RALMs utilize contexts.

5.4.4 Fine-tuning with Small Automatically Generated Data

The main motivation for the development of In-Context RALMs was to provide a simpler alternative to fine-tuned RALMs as fine-tuning can be costly and difficult to achieve especially as it usually requires an enormous amount of training data and large computational resources. However, our research along with other prior research has shown that In-Context RALMs are not as resilient as fine-tuned RALMs are against irrelevant contexts, and thus, fine-tuning seems indispensable for achieving robustness.

A recently proposed middle-ground solution is fine-tuning In-Contex RALMs with automatically generated relatively small training data. Yoran et al. (2024) have shown that fine-tuning RALMs using automatically generated training data of relevant and irrelevant contexts is effective in making RALMs more robust against irrelevant contexts. Their method automatically generates training data by treating top-1 retrieved contexts as relevant and low-ranked contexts as irrelevant data and then fine-tunes RALMs using the generated data. They have demonstrated that even as few as 1,000 examples from the generated data were sufficient to fine-tune

the model to be robust to irrelevant contexts while maintaining high performance. Their approach can serve as a liable middle-ground solution considering that it does not require as much effort and resources compared to previously introduced fine-tuned RALMs.

6

Conclusion

This chapter concludes the study on the effectiveness of In-Context RALM, particularly in terms of consistency. The study compared In-Context RALM (Llama-2-7B with retrieved contexts) with both a parametric-only model (Llama-2-7B without contexts) and a fine-tuned RALM (Atlas). Key findings indicate that In-Context RALM produces more consistent predictions than the parametric-only model but is less effective than the fine-tuned RALM Atlas. In-Context RALM's performance suffers with irrelevant contexts, emphasizing the importance of accurate context retrieval. The study's limitations include variability in Atlas-retrieved contexts and insufficient evaluation time. Future research should focus on enhancing In-Context RALM's robustness against irrelevant contexts and exploring methods to optimize context usage for improved accuracy and consistency.

6.1 Conclusion

This study aimed to investigate the effectiveness of In-Context RALM especially in terms of the consistency of predictions. We compared the performance of In-Context RALM (Llama-2-7B with retrieved contexts) with a parametric-only model (Llama-2-7B) to study whether In-Context RALM produces more accurate and consistent predictions, and we compared In-Context RALM with a fine-tuned RALM Atlas to see whether In-Context RALM is more effective in increasing accuracy and consistency compared to fine-tuned RALMs. We also investigated the performance of In-Context RALM with different retrievers to study how much the performance of In-Context RALM depends on the retrievers used in In-Context RALM.

As a result, we found that

- the In-Context RALM produces more consistent predictions compared to the parametric-only model (i.e. the same LM but without contexts).
- the In-Context RALM is not as effective in increasing the consistency of predictions as the fine-tuned RALM Atlas but it can still serve as a viable alternative to achieve higher accuracy and consistency when fine-tuning is not feasible provided that the quality of retrieved contexts is guaranteed.
- the In-Context RALM is less robust against irrelevant contexts than the fine-tuned RALM and its accuracy and consistency get worse than without-context settings if the provided contexts are irrelevant.

- Providing more accurate and relevant retrieved contexts seems to result in higher accuracy and consistency in both the In-Context RALM and the fine-tuned RALM.

These findings of the present study provide valuable new insights into the effectiveness of In-Context RALMs in terms of increasing consistency, especially considering that prior research regarding In-Context RALMs has mostly evaluated its effectiveness and robustness with respect to accuracy but not consistency.

6.2 Limitations

The conclusions drawn from our results are limited due to the Atlas-retrieved contexts varying across paraphrases and also due to not having had enough time to evaluate the quality of Atlas-retrieved contexts. As Atlas-retrieved contexts were not fixed across paraphrases, the consistency achieved by using Atlas-retrieved contexts could have been partially limited due to the context being inconsistent. If we had more time, we should have fixed the Atlas-retrieved contexts across paraphrases and analyzed the accuracy and relevancy of the retrieved contexts to the queries. Additionally, we did not have enough time and resources to measure the accuracy and relevancy of Atlas-retrieved contexts. Although we know that the quality of Atlas-retrieved contexts is better than the completely irrelevant random contexts and is not as good as the perfect golden contexts, we cannot clearly quantify the quality of Atlas-retrieved contexts.

6.3 Future Work

For In-Context RALMs to be a more reliable and effective alternative to fine-tuned RALMs, making In-Context RALMs more robust against irrelevant contexts seems to be essential. Although there has been prior research on different ways to improve the robustness of In-Context RALMS, prior studies have evaluated the robustness in terms of accuracy but not consistency. Therefore, further research is needed to understand the effective ways to improve the resilience of In-Context RALMs against irrelevant contexts. Additional research is required to investigate how to increase the performance gain from In-Context RALMs for it to be a more competitive alternative to fine-tuned RALMs. For instance, studying the effective way to prepend and prompt the retrieved contexts is one important aspect of In-Context RALMs that needs further research.

Bibliography

R. Baeza-Yates and B. Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison-Wesley.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

J. Chen, H. Lin, X. Han, and L. Sun. 2023. Benchmarking large language models in retrieval-augmented generation. *arXiv*.

J. Cohen. 1969. *Statistical Power Analysis for the Behavioral Sciences*. Academic Press, New York.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

I. Goodfellow, Y. Bengio, and A. Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv*.

L. Hagström, D. Saynova, T. Norlund, M. Johansson, and R. Johansson. 2023. The effect of scaling, retrieval augmentation and form on the factual consistency of language models. *arXiv*.

K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *arXiv*.

G. Izacard and E. Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. *arXiv*.

G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, W. Dai, A. Madotto, and P. Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

J. Johnson, M. Douze, and H. Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1601–1611.

Daniel Jurafsky and James H Martin. 2019. *Speech and Language Processing*, 3rd edition. Pearson.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

K. Lee, M.-W. Chang, and K. Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv*.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

G. Marcus. 2020. The next decade in ai: Four steps towards robust artificial intelligence. *arXiv*.

F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th Inter-*

national Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. 2023. In-context retrieval-augmented language models. *arXiv*.

S. Robertson and H. Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trendső in Information Retrieval*, 3(4):333–389.

K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv*.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv*.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, and T. ... Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv*.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

K. Wu, E. Wu, and J. Zou. 2024. How faithful are rag models? quantifying the tug-of-war between rag and llms internal prior. *arXiv*.

O. Yoran, T. Wolfson, O. Ram, and J. Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. *arXiv*.