University of Auckland

Engineering Part IV Project

# Data-Centric Methods For Improving RAG-like Architectures

Jaime Wu

442689185

jwu153

October 15, 2021

## Acknowledgments

I would like to thank Shamane Siriwardhana for being a very supportive and friendly supervisor throughout our project. He was always willing to answer any queries we had whilst providing guidance throughout our research. Furthermore, his knowledge and expertise in deep neural language models is invaluable, and we are very grateful for his supervision. Overall, I had a really great experience working with Shamane. I would also like to thank Suranga Nanayakkara for pointing us in the right direction during our meetings, and motivating us to work with Shamane more frequently and closely. Lastly, I would like to thank Andrew Wang for being a great project partner. We had many insightful discussions and coding sessions together.

## Declaration of Authorship

I hereby declare

- that the following work has been written solely by me without any external assistance
- that the use of external documents, work, tools, and etc have been stated and cited correctly according to $APA^{7th}$ rules.
- that the exception of the contributions indicated below, all contributions to this project are exclusively the work Jaime Wu and Andrew Wang:
  - The development of our second RAG Pipeline (training of RAG full end-2-end) was accomplished by Shamane Siriwardhana.
  - The following data was supplied by Shamane Siriwardhana - 100K synthetic QA pairs generated using the traditional extractive method.
  - The development of fine-tuned transformer-like models and open-source tools such as Google Colab. Where these tools have been used, they have been referenced according to $APA^{7th}$ rules.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Jaime Wu, October 15, 2021

# Abstract

Due to the emergence of transformer-based deep learning neural networks, factoid open-domain question-answering (ODQA) has recently seen significant progress (Vaswani et al., 2017). For the task of question-answering, we have seen the emergence of single end-to-end systems that fine-tune both the information retrieval (IR) and machine reading comprehension (MRC) components simultaneously. Retrieval Augmented Generation (RAG) is one such end-to-end approach. Recently, Siriwardhana et al. published a paper introducing novel architectural changes to RAG, improving RAG's performance for ODQA.

The focus of our research is mainly on data-centric methods to further improve RAG's ODQA performance. Our first primary contribution was generating unambiguous synthetic question-answer pairs using an abstractive summarisation approach instead of the traditional extractive approach. Our second primary contribution was to enhance the input question by appending the most relevant passage using a lexical model. Finally, our two minor contributions are introducing dense vector embeddings to the round trip consistency method and implementing an efficient RAG pipeline to conduct our research. Our main results showed that longer sequences (regardless of input or output) lead to the retriever collapsing in RAG – the same set of passages were selected regardless of the question. However, enhancing the input with additional context or information improved RAG's performance, but we still observed retrieval collapse. Therefore, we showed that additional research on more sophisticated data-centric methods is required to improve RAG's ODQA performance without the retriever collapsing.

# Table of Contents

**Abstract** **i**

**Table of Contents** **ii**

**List of Figures** **iv**

**List of Tables** **iv**

**Code Listings** **iv**

**Glossary** **v**

**1 Introduction** **1**

1.1 Background . . . 1

1.2 Project Scope and Research Objectives . . . 1

**2 Literature Review** **3**

2.1 History of Question-Answering (QA) . . . 3

2.2 Open-Domain Question-Answering (ODQA) . . . 3

2.2.1 Open-book ODQA Frameworks . . . 4

2.2.2 Closed-book ODQA Frameworks . . . 4

2.3 NLP Models for ODQA Frameworks . . . 5

2.3.1 Pre-training and Fine-tuning . . . 5

2.3.2 The State-of-the-art Transformer . . . 5

2.3.3 BERT . . . 7

2.3.4 GPT . . . 7

2.3.5 BART . . . 7

2.3.6 Dense Passage Retrieval (DPR) . . . 8

2.3.7 Retrieval Augmented Generation (RAG) . . . 8

2.4 NLP Data Methods . . . 8

2.4.1 Synthetic Data . . . 8

2.4.2 Roundtrip Consistency . . . 9

**3 Related Work and Tools** **10**

3.1 RAG Partial End-2-End and RAG Full End-2-End . . . 10

3.2 Benchmark Paper (End-to-End QA on COVID-19) . . . 10

3.3 Open-source Python Libraries and Tools . . . 11

**4 Methodology** **12**

4.1 Adopted ODQA Framework and Motivations . . . 12

4.2 High Level Overview of Methodology . . . 12

4.3 RAG Pipeline Implementation . . . 13

4.3.1 RAG Pipeline 1 - Original RAG . . . . . . . . . . . . . . . . . . . . . . . . . 13
4.3.2 RAG Pipeline 2 - RAG full end-2-end . . . . . . . . . . . . . . . . . . . . . 15
4.4 Dense Vector Round Trip Consistency . . . . . . . . . . . . . . . . . . . . . . . . 16
4.5 Abstractive Summarisation for Unambiguous Generation of Synthetic QA Pairs 17
4.6 Question Enhancement For Better Information Retrieval . . . . . . . . . . . . . 20

**5 Results and Discussions 22**
5.1 Abstractive Summarisation for Unambiguous Generation of Synthetic QA Pairs 22
5.2 Question Enhancement For Better Information Retrieval . . . . . . . . . . . . . 23
5.3 Limitations of Research . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
5.4 Other Insights and Next Steps . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

**6 Conclusion 26**

**References 27**

## List of Figures

Fig. 1: Open Domain Question-Answering Systems (Weng, 2021) . . . . . . . . . . . 4
Fig. 2: Transformer Architecture . . . . . . . . . . . . . . . . . . . . . . . . . 6
Fig. 3: Domain adaptation of RAG full end-2-end vs. RAG original . . . . . . . . . . 10
Fig. 4: Retrieval Collapse . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

## List of Tables

Tab. 1: Examples of unambiguous synthetic QA pairs. . . . . . . . . . . . . . . . . . . . . 19
Tab. 2: Examples of enhanced synthetic questions. . . . . . . . . . . . . . . . . . . . . 21

## Listings

1 RAG Data Pipeline Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 14
2 DPR Training . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
3 Model Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
4 Synthetic QA Generation Algorithm . . . . . . . . . . . . . . . . . . . . . . . 17
5 Synthetic QA Generation Algorithm . . . . . . . . . . . . . . . . . . . . . . . 18
6 Question Enhancement Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 20

# Glossary

**BERT** Bidirectional Encoder Representation from Transformers.

**DPR** dense passage retriever.

**end-2-end** end to end.

**GPT** Generative Pre-trained Transformer.

**IR** information retrieval.

**MRC** machine reading comprehension.

**NLP** natural language processing.

**ODQA** open-domain question-answering.

**QA** question-answering.

**RAG** Retrieval Augmented Generation.

**seq-2-seq** sequence to sequence.

**TF-IDF** term frequency–inverse document frequency.

# 1 Introduction

## 1.1 Background

Factoid open-domain question-answering (ODQA), a fundamental challenge in natural language processing (NLP), has recently made significant progress, and this is attributed to the introduction of transformer-based deep learning neural architectures (Vaswani et al., 2017). Fundamentally, ODQA frameworks are made up of an information retrieval (IR) component and machine reading comprehension (MRC) component (Semnani & Pandey, 2020). Traditionally, the two components have been independently trained and fine-tuned to their respective downstream tasks of retrieving documents and answering questions (Chen et al., 2017). As a result, the neural IR models lag behind traditional lexical matching approaches like term frequency–inverse document frequency (TF-IDF) in more specific and specialised target domains such as COVID-19, consequently impacting the MRC model's performance (Reddy et al., 2020). However, with recent advancements, we have seen developments of single end-2-end neural models that consist of both the IR and MRC components jointly trained and fine-tuned together in an end-2-end manner. Retrieval Augmented Generation (RAG) is one such novel single end-2-end neural architecture – it jointly fine-tunes a Dense Passage Retriever (DPR) for IR and BART seq-2-seq for MRC (P. Lewis, Perez, et al., 2020). The RAG architecture, designed to retrieve information from an indexed Wikipedia knowledge base, performed exceptionally well on ODQA tasks (P. Lewis, Perez, et al., 2020). Furthermore, researchers have also shown that RAG has desirable features such as interpretability and minimal hallucination, which is vital to the NLP research field (P. Lewis, Perez, et al., 2020). The focus of our research is on RAG.

## 1.2 Project Scope and Research Objectives

Despite all of the advances, improving the performance of the RAG using data-centric strategies is currently under-explored, setting our motivation for the following research. Our contributions will predominantly relate to the development of synthetic data, which can be used to fine-tune RAG for ODQA.

Our two primary contributions can be summarised as follows:

1. We observed that in some ODQA settings, the input question is too ambiguous, which makes it difficult for the model to retrieve documents and answer the question. Motivated by this, our first major contribution introduces a novel synthetic question-answering (QA) mechanism to reduce the ambiguity of synthetically generated question-answer pairs.

Instead of taking the traditional approach of extracting the most likely answer from a passage using a seq-2-seq model, we use an abstractive summarisation model to generate the answer.

2. Furthermore, the uncertainty of some question-answer pairs motivated us to increase the amount of information in the input question. For instance, BlenderBot 2.0, a novel open-source chatbot, significantly improved long term memory performance using an internet search query prior to the input (Komeili et al., 2021). Similarly, our second contribution aims to improve RAG's retrieval performance by concatenating extra information (e.g. passages) to the input question using a lexical matching approach. We therefore provide the retriever with an improved, information-rich, input.

Furthermore, we also had two minor contributions which can be summarised as follows:

1. Given the need to train and fine-tune the RAG architecture multiple times, we are motivated to build an efficient and effective RAG pipeline with easily exchangeable components. The pipeline should accommodate our needs for exchanging the training data quickly, fine-tuning the RAG model end-2-end, and providing an evaluation pipeline.
2. The current round trip consistency method uses an extractive strategy to filter data, which we believe is not optimal, since responses might have the same meaning but differ in linguistic structure. Motivated by this, this minor contribution explores improvements to the current round trip consistency method using dense vector embeddings and sentence similarity.

# 2 Literature Review

## 2.1 History of Question-Answering (QA)

Early research and development of QA systems began in the 1960s and 1970s, where simple QA systems consisted of a natural-language front-end that interfaced to a back-end database of answers (Prager, 2006). LUNAR, for example, transforms a natural language question into a set of database queries to retrieve an answer (Woods, 1973). While these QA systems were functional, they were limited to the database's knowledge, rendering them impractical or narrow in use-case (Prager, 2006). Between the late 1990s and the early 2000s, there few advances in the QA field, but over the last decade, the QA field has seen tremendous growth, attributed to two major driving forces: (1) advances in deep learning neural network models, and (2) increase in the number of large-scale benchmark data sets for question-answering (Liu et al., 2019).

## 2.2 Open-Domain Question-Answering (ODQA)

With the recent advancements in the QA field, QA systems are now more commonly referred to as open-domain question-answering (ODQA) since questions and answers are drawn from a vast knowledge base, spanning several subject areas (Liu et al., 2019) (Figure 1). Fundamentally, ODQA frameworks are made up of two main components (Semnani & Pandey, 2020):

1. An IR model, which is capable of retrieving passages from an external knowledge base that are closely related to the input question.
2. A MRC model, which utilises the retrieved passages to construct an answer to the input question, either by text extraction or text generation.

An ODQA system that leverages information retrieval (IR) from an external knowledge corpus such as Wikipedia is referred to as open-book, and in contrast, a system without access to any external knowledge base is referred to as closed-book (Figure 1) (Liu et al., 2019).

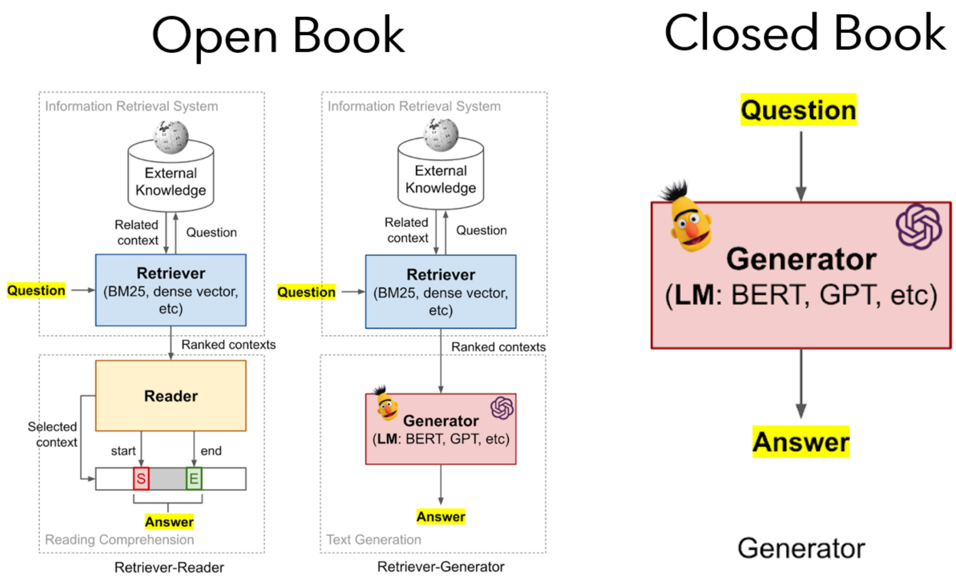


Figure 1: Open Domain Question-Answering Systems (Weng, 2021)

### 2.2.1 Open-book ODQA Frameworks

Existing open-book systems are based on the information retrieval (IR) and machine reading comprehension (MRC) paradigm (Figure 1, Left). An IR model initially finds a relevant set of documents from a large knowledge base, such as Wikipedia, and then passes them on to a reader or generator model, which extracts or generates answers, much like an open-book exam (Chen et al., 2017). The retriever and reader components can be trained independently or jointly trained together in an end-2-end fashion.

- Independent retriever and reader refers to the IR and MRC components being fine-tuned independently. This framework of independently training each component was first adopted by DrQA (Chen et al., 2017).
- Joint retriever and reader refers to combining both the IR component and MRC component as a single overall model, which is then fine-tuned for a specific task such as question-answering in an end-2-end manner. The ORQA system was the first to adopt end-2-end joint learning, with both the retriever and reader components being BERT derivatives (Lee et al., 2019).

### 2.2.2 Closed-book ODQA Frameworks

Closed-book models solely rely on the knowledge it has gained during pre-training and does not leverage an information retrieval component since they store knowledge within the parameters rather than in an external knowledge base (P. Lewis, Stenetorp, et al., 2020) (Figure 1, Right).

These models typically consist of seq-2-seq transformer models that are directly fine-tuned on question answering pairs. For example, one could fine-tune a pre-trained BART model on QA pairs to produce a closed-book model (P. Lewis, Stenetorp, et al., 2020).

## 2.3 NLP Models for ODQA Frameworks

The aim behind NLP models is to teach a machine to learn and interpret language in the same manner that a human does. Many non-neural traditional methods use lexical matching techniques such as TF-IDF, which quantifies the frequencies of terms across documents, computing a weight for each word and signifying the word's importance in the set of documents (Qaiser & Ali, 2018). TF-IDF was widely used in traditional IR and MRC models.

Advancements in NLP led to the development of neural-based NLP models such as convolution neural networks, recurrent neural networks, and LSTMs. Since the past decade, we have seen rapid improvements of these neural-based NLP models, with a breakthrough in 2017, where a novel neural-based architecture called the transformer set the new state of the art standard for NLP models (Vaswani et al., 2017).

### 2.3.1 Pre-training and Fine-tuning

For the following neural NLP models, the concepts of ”pre-training” and ”fine-tuning” are widely used in the literature. Pre-training refers to performing a conventional supervised training process, commonly on an extensive knowledge corpus such as the BooksCorpus (800M words) or the English Wikipedia (2,500M words) (Devlin et al., 2018). Pre-training allows the deep learning model to understand the English language, such as semantics and contextual information (Devlin et al., 2018). The pre-trained model may be fine-tuned with only one additional output layer for various down-streamed tasks, such as question answering, text summarisation, and translation, without requiring significant task-specific architectural changes (Devlin et al., 2018).

### 2.3.2 The State-of-the-art Transformer

The underlying architecture underpinning the development of most recent neural ODQA systems is based on the novel transformer model introduced in “Attention Is All You Need” (Vaswani et al., 2017). Based solely on the addition of attention mechanisms, the transformer achieved a new state of the art performance in language translation tasks, outperforming previous recurrent neural networks such as LSTMs. The use of the attention mechanisms allowed transformers to be more parallelisable, requiring less time to train, while also being deeply bidirectional (able to learn the context of words left to right, vice versa) (Vaswani et al., 2017). The transformer architecture can be broken down into two main components: (1) An encoder stack and (2) A decoder stack, as seen in figure 2.

1. Encoder Stack: Comprises of six identical layers, with each layer consisting of a multi-head attention mechanism that connects to a standard feed-forward neural network (Figure 2). The encoder self-attention mechanism captures contextual information via the relevance of each input token (e.g. word) to every other input token across the whole sequence (e.g. sentence), outputting a set of attention vectors (Vaswani et al., 2017). The first input is layered with contextual information given by positional encoding (Vaswani et al., 2017).
2. Decoder Stack: Comprises of six identical layers, with each layer consisting of a masked multi-head attention layer (Figure 2). The decoder layers also have an additional encoder-decoder attention mechanism. Because the decoder attends to the encoder stack, masked multi-head attention layers prevent the decoder from copying information from the encoder. The final outputs of the decoder stack are converted into a probability distribution via a softmax function, indicating the most likely output sequence (Vaswani et al., 2017).

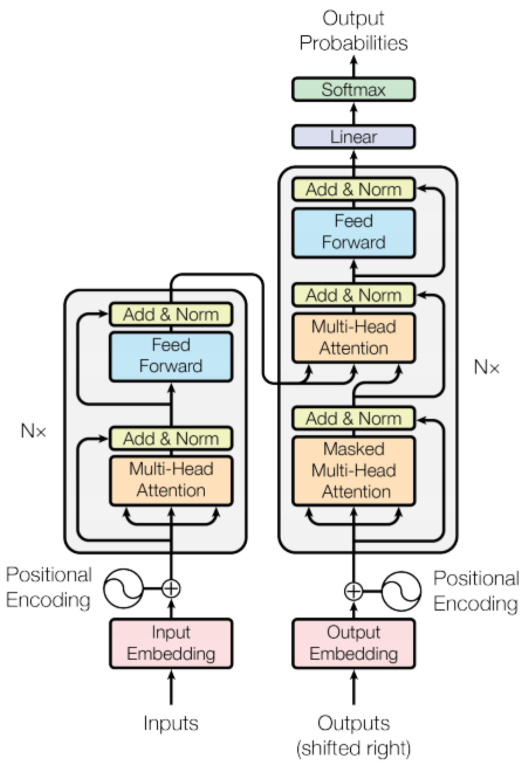


Figure 2: Transformer Architecture

The original transformer was explicitly designed for neural machine translation i.e. translating one language to another, for example, English to French (Vaswani et al., 2017). Since the introduction of the transformer, new neural models deriving from the original transformer architecture emerged throughout the literature.

### 2.3.3 BERT

Unlike the original transformer architecture, Bidirectional Encoder Representation from Transformers (BERT) is made up purely of a stack of encoders. BERT was designed to pre-train deep bidirectional representations, relying on masked language modelling and next sentence prediction techniques to achieve this joint conditioning on both left and right context across all layers (Devlin et al., 2018). Masked language modelling remove tokens in the input sequence at random, forcing the model to learn to decode the original token using the left and right tokens as context, resulting in bidirectional context learning (Devlin et al., 2018). Next sentence prediction, on the other hand, trains the model to predict if the second sequence in a pair of input sequences belong to the original document in the correct order (Devlin et al., 2018). The result is a pre-trained BERT model that is efficient at predicting masked tokens but is not optimal for text generation. The pre-trained BERT model may be fine-tuned with only one extra output layer to provide state-of-the-art models for various NLP tasks, without requiring significant task-specific architectural changes.

### 2.3.4 GPT

Similar to BERT, Generative Pre-trained Transformer (GPT) is made up purely of a stack of decoders that can be pre-trained and fine-tuned to down streamed tasks such as text summarisation (Brown et al., 2020). GPT, a left-to-right decode designed for text generation, was trained with the casual language modelling technique, which is more powerful at predicting the next token (word) in a sequence, which allows GPT to generate syntactically coherent text compared to BERT.

### 2.3.5 BART

BART, a denoising autoencoder for pretraining seq-2-seq models, has a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT) and thus is the most similar to the original transformer architecture (M. Lewis et al., 2019). BART was trained by reordering corrupted text that were randomly shuffled using an arbitrary noising function, and a novel in-filling scheme to reconstruct the original passage where spans of text were replaced with a single mask token (M. Lewis et al., 2019). BART is particularly effective for text generation but also works well for comprehension tasks. Therefore, it is particularly effective for machine reading comprehension tasks such as answer generation.

### 2.3.6 Dense Passage Retrieval (DPR)

As mentioned earlier, ODQA systems rely on efficient information retrieval to select relevant passages that aid question answering. Traditional approaches to IR is to use sparse vector space models (weighted bag-of-words vectors) such as TF-IDF or BM25. However, with the emergence of transformer-like models, IR can be practically implemented using dense vector representations. The DPR model, for example, can create dense vector embeddings from a limited number of questions and passages using a basic dual-encoder framework in which each encoder is a conventional BERT model (Karpukhin et al., 2020). In terms of top-20 passage retrieval accuracy, the DPR outperformed a powerful Lucene-BM25 system by 9%-19% in absolute terms (Karpukhin et al., 2020). As a result, DPR and other dense vector embedding models set the current state-of-the-art IR performance in ODQA systems.

### 2.3.7 Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG) is a novel end-2-end architecture that combines a DPR as the IR component with a seq-2-seq BART generator as the MRC component (P. Lewis, Perez, et al., 2020). RAG's main feature enables the ability to utilise pre-trained parametric knowledge, present within the parameters, and non-parametric memory found in an external knowledge base for better language generation (P. Lewis, Perez, et al., 2020). Furthermore, this means that RAG's internal knowledge may be readily changed on the fly, allowing researchers and engineers to manage what RAG's knowledge without spending time or computing power retraining the entire model (P. Lewis, Perez, et al., 2020). RAG can be fine-tuned in an end-2-end fashion, resulting in models that outperform current parametric seq2seq models, establishing a new state-of-the-art performance for knowledge-intensive NLP tasks (P. Lewis, Perez, et al., 2020).

## 2.4 NLP Data Methods

### 2.4.1 Synthetic Data

To train the above models for the downstream task of question-answering, we require QA pairs from a large knowledge base. It is well accepted in the machine learning community that developing deep learning models for any application depends on the availability of high-quality, unbiased labelled data; however, obtaining such labelled data sets, complied by annotators, to train a model, is often expensive and inefficient (Puri et al., 2020).

Weakly supervised learning is a typical method for addressing this need, in which a model is trained on existing data and then used to classify new data or generate new data for further training. One such self-supervised learning technique, explored by Puri et al., 2020, is to generate synthetic QA data. A general approach is to sample answer candidates from paragraphs based on a probability model and then use the answer along with the paragraphs to generate a question.

Puri et al. showed that the model using solely synthetic data achieved higher accuracy than using the SQuAD1.1 training data set alone. Likewise, much prior work explores techniques based on synthetic data to improve the domain adaptation of QA systems (Shakeri et al., 2020). However, there is still a relatively large quality gap between synthetic and human-annotated data, which is yet to be investigated within the research community.

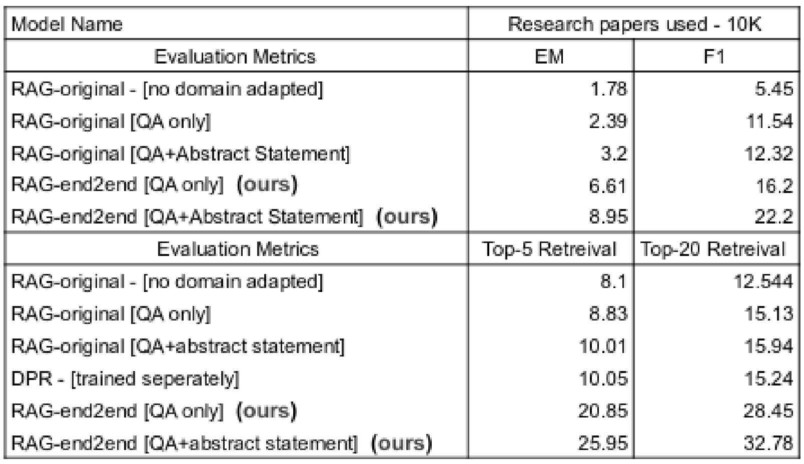
### 2.4.2 Roundtrip Consistency

As previously noted, the most typical method for generating synthetic QA pairs involves first generating the answer from a selected text and then using a generative model to create the synthetic question. Because using a synthetic answer to generate a synthetic question may result in low-quality data, a roundtrip consistency technique may be used to fact-check the synthetic question-answer pair (Alberti et al., 2019). A roundtrip consistency method consists of using a neural extractive MRC model to answer the synthetic question given the passage that was used to generate it (Alberti et al., 2019). If the extracted fact-checked response matches the extracted synthetic answer, the synthetic QA pair is retained; otherwise, it is discarded.

# 3 Related Work and Tools

## 3.1 RAG Partial End-2-End and RAG Full End-2-End

RAG was originally designed to be fine-tuned in a partial end-2-end manner since only the input encoder and output generator is trained end-2-end, while the DPR's weights are frozen and trained independently (P. Lewis, Perez, et al., 2020). The authors of the original paper justified that fine-tuning the whole system end-2-end is too computationally expensive and not easily achievable (P. Lewis, Perez, et al., 2020). However, a related piece of work by Siriwardhana et al. showed that fine-tuning the overall RAG architecture (all three components including the DPR) in a complete end-2-end manner, intelligently, resulted in considerable improvements (Siriwardhana et al., 2021) (Figure 3). Based on these findings, our research aims to develop novel data-centric techniques, rather than architectural modifications (as this has already been achieved by Siriwardhana et al.).



| Model Name | Research papers used - 10K | |
|---|---|---|
| Evaluation Metrics | EM | F1 |
| RAG-original - [no domain adapted] | 1.78 | 5.45 |
| RAG-original [QA only] | 2.39 | 11.54 |
| RAG-original [QA+Abstract Statement] | 3.2 | 12.32 |
| RAG-end2end [QA only] **(ours)** | 6.61 | 16.2 |
| RAG-end2end [QA+Abstract Statement] **(ours)** | 8.95 | 22.2 |
| Evaluation Metrics | Top-5 Retreival | Top-20 Retreival |
| RAG-original - [no domain adapted] | 8.1 | 12.544 |
| RAG-original [QA only] | 8.83 | 15.13 |
| RAG-original [QA+abstract statement] | 10.01 | 15.94 |
| DPR - [trained seperately] | 10.05 | 15.24 |
| RAG-end2end [QA only] **(ours)** | 20.85 | 28.45 |
| RAG-end2end [QA+abstract statement] **(ours)** | 25.95 | 32.78 |

Figure 3: Domain adaptation of RAG full end-2-end vs. RAG original

## 3.2 Benchmark Paper (End-to-End QA on COVID-19)

Reddy et al. explored the application of synthetically generating QA pairs to improve the performance of end-2-end QA systems in the COVID-19 domain. They used an end-2-end QA system with various DPR and BART models for IR and MRC. Using synthetic QA pairings to train and fine-tune their models, Reddy et al. presented substantial improvements over multiple COVID-19 QA data sets. In our research, the work in "End-to-End QA on COVID-19" will set our benchmark, as we will be evaluating our models using similar COVID-19 data sets.

## 3.3 Open-source Python Libraries and Tools

Given time and computational constraints, it is not feasible to develop and train the above transformer-like models. As a result, we will leverage Huggingface Transformers, a free and open-source Python library, to provide pre-existing frameworks and models as needed (HuggingFace, 2021). We will also be utilising the Haystack AI library, an end-to-end framework that enables the easy development of ODQA pipelines (DeepsetAI, 2021). Finally, we will use Google Colab, a free browser-based platform that allows us to create and execute Python code within a Jupyter Notebook, to address our restricted computing hardware (Google, 2021). Despite Google Colab offering free CPU and GPU use, certain usage limitations will inhibit our research (Google, 2021).

# 4 Methodology

## 4.1 Adopted ODQA Framework and Motivations

According to a review of the current literature, RAG establishes the current state-of-the-art performance for end-2-end ODQA systems (P. Lewis, Perez, et al., 2020). Furthermore, Siriwardhana et al. introduced a novel architectural modification to allow for full end-2-end joint training of the entire RAG model (including the DPR). Despite all of the advances, improving the performance of RAG is currently under-explored, setting our motivation for the following research. In contrast to prior work, we are focusing on improving RAG's performance in the area of ODQA, whether this is in a generic domain such as Wikipedia or a more domain-specific setting such as COVID-19. It is challenging to improve ODQA systems across multiple domains with limited or no annotated data. It is also not trivial to create a large, unbiased annotated data set using human labour due to cost and time constraints (Puri et al., 2020). Therefore, our work primarily focuses on developing novel RAG-friendly, data-centric strategies. It is common in the literature that techniques based on synthetic data have been used for fine-tuning ODQA systems; thus, our data-centric strategies will mostly relate to synthetic data.

## 4.2 High Level Overview of Methodology

The following work aims to explore and establish novel data-centric methods to improve the performance of open domain question answering (ODQA) for RAG. We follow a sequential methodology for conducting our research and adapt our research of interest as required, given intermediate results and findings. As a starting point, we have identified two main areas of improvement for RAG.

1. Generate RAG-friendly synthetic question-answering pairs via abstractive summarisation.
2. Improve the information retrieval component of RAG by concatenating additional input information to the input question with lexical matching methods.

Furthermore, we have also identified the need to establish the following:

1. Implement an efficient and effective RAG pipeline to quickly train and fine-tune various models and provide a consistent method for model evaluation.
2. Improve round trip consistency to better filter synthetic data by using dense vector representation of the answers instead of a lexical (extractive) approach.

## 4.3 RAG Pipeline Implementation

As a preliminary step, a RAG pipeline will be implemented. This RAG pipeline should allow us to exchange the training data easily, fine-tune RAG in an end-2-end fashion, and provide a simple mechanism to assess an accuracy score (such as an exact match score) using a benchmark data set (COVID QA). As mentioned in our related works section, there are currently two RAG-like architectures: (1) original RAG (partial end-2-end), (2) RAG full end-2-end (Siriwardhana et al., 2021). Our research will employ two different RAG pipelines to leverage both the partial and full end-2-end RAG architectures. We took such an approach to ensure valid and consistent results.

### 4.3.1 RAG Pipeline 1 - Original RAG

We will first implement a pipeline that uses the original RAG model (partially end-2-end) using the Haystack library. Our pipeline can be broken down further into three sub-pipelines as follows: (1) data preparation, (2) model fine-tuning, and (3) model evaluation.

**Data Preparation**

Before fine-tuning RAG, we created an information retrieval (IR) data set in a format suitable for training. This IR data set contains three main attributes: (1) a question, (2) a positive context, and (3) a set of hard negative contexts. The question is simply the input question from a question-answering pair. There is a positive context for each question, which is the target passage the DPR should retrieve. Finally, there is a set of hard negative contexts for each question, which are passages that should not be retrieved by the DPR. We created this IR data set as follows:

1. Given a question-answering training data set with the columns: (1) question, (2) positive context, and (3) answer, we formulate a large knowledge base using all the positive contexts.
2. Iterating through each data set, we create a set of hard negative contexts for each question. To achieve this, we used a BM25 lexical matching model to query the top 10 most relevant passages from our large knowledge base. As the BM25 model will likely choose the positive context, we chose to ignore that selected document from the set of hard negatives (Sparck Jones et al., 2000).
3. Finally, we took an 80/20 split of our final IR data set for training and validation.

The pseudocode for our data preparation algorithm is:

```python
topK = 10 # Get top 10 best matching documents to use as the hard negatives
hard_negative_ctxs = []

# Iterate each synthetic QA pair in QA.csv and convert into DPR data format
for idx, r in qa.iterrows():

    # Clean the question
    question = qa.question[idx]
    question = preprocess(question)

    # Use BM25 to get hard negatives
    # Remove the same context to current question (so we don't pick it)
    # Reverse order and get best ones (best hard negatives)
    docScores = BM25.get_scores(question)
    docScores = np.delete(docScores, idx)
    idxTopKDocs = np.argsort(docScores)[::-1][:topK]
    hard_negatives = qa.iloc[idxTopKDocs]

    # Format hard negatives
    positive_ctxs = {"title": r.title, "text": r.context}
    hard_negative_ctx = [{"title": r.title, "text": r.context} for _, r in
  hard_negatives.iterrows()]
    hard_negative_ctxs.append({"question": r.question, "answers": [r.answer
  ], "positive_ctxs": [positive_ctxs], "negative_ctxs": [], "
  hard_negative_ctxs": hard_negative_ctx})
```

Listing 1: RAG Data Pipeline Algorithm

### Model Fine-tuning

Using the prepared data set, we fine-tuned a DPR using the Haystack API (DeepsetAI, 2021). To summarise our approach, we first created a FAISS document store that contains our knowledge base, and then we leveraged the API to initiate the training process. Following this, we fine-tuned a BART model using the question-answer pair in an end-2-end manner with our fine-tuned DPR. Finally, we leveraged Haystack's pipeline API to create the RAG pipeline from our fine-tuned DPR and BART model, which can be used for question-answering (DeepsetAI, 2021).

Similarly, we also used the RAG fine-tuning script available in the Huggingface Transformers library to fine-tune the original RAG model (HuggingFace, 2021). Both methods produced similar results.

```python
# Start training our model and save it when it is finished
retriever.train(
    data_dir="",
    train_filename=train_filename,
    dev_filename=dev_filename,
    test_filename=dev_filename,
    n_epochs=1,
    batch_size=16,
    grad_acc_steps=8,
    save_dir=save_dir,
    evaluate_every=3000,
    embed_title=True,
    num_positives=1,
    num_hard_negatives=1
)
```

Listing 2: DPR Training

### Model Evaluation

The final part of our original RAG pipeline evaluates the performance of our fine-tuned RAG model using the COVID-QA data set as a benchmark (Reddy et al., 2020).

The pseudocode for our model evaluation algorithm is:

```
# Evaluating the retrieval recall
for question in QUESTIONS:
    # Retrieve related documents from retriever
    retriever_results = retriever.retrieve(
        query=question
    )

    # Now generate answer from question and retrieved documents
    predicted_result = generator.predict(
        query=question,
        documents=retriever_results,
        top_k=1
    )

    # Print you answer
    answers = predicted_result["answers"]
    print(f'Generated answer is \'{answers[0]["answer"]}\' for the question
   = \'{question}\'')
```

Listing 3: Model Evaluation

### 4.3.2 RAG Pipeline 2 - RAG full end-2-end

Finally, we will leverage the full end-2-end pipeline created by Siriwardhana et al.. As this pipeline fine-tunes RAG fully end-2-end, it will provide more insightful results; therefore, our final results will be from this pipeline. The first pipeline will be used to verify our results and ensure consistency across the original RAG model and the full end-2-end RAG model. Siriwardhana et al. will be conducting model training and evaluation using the full RAG end-2-end pipeline as this requires access to AHLab's virtual machine cluster, to which we do not have access.

## 4.4 Dense Vector Round Trip Consistency

Once the RAG pipelines have been implemented, we then explored improvements to the round trip consistency method. From the literature review and recent developments, the round trip consistency method consists of using a neural extractive MRC model to answer the synthetic question given the passage used to generate it (Alberti et al., 2019). If the extracted fact-checked response matches the extracted synthetic answer, the synthetic question-answer pair is retained; otherwise, it is discarded. However, this approach neglects fact-checked answers that are similar to the synthetic answer but have different lexical representations. Some of the fact-checked answers might be extracted from a different section of the passage and thus have different words but can still be interpreted to have the same meaning as the synthetic answer. In such a case, the original round trip consistency method would discard the pair of data. We argue that we can avoid such situations with our dense vector round trip consistency method.

Using a sentence BERT similarity model, our version of the round trip consistency method represents the fact-checked answer and the synthetic answer as dense vector embeddings. A similarity score can then be calculated using the dense vector embeddings, and above a certain threshold, we could retain that QA pair in our version of round trip consistency. Our method for creating such a dense vector round trip consistency pipeline can be summarised as follows:

1. For each pair of synthetic question-answer and its passage, we use a standard DistilBERT model, fine-tuned for QA using knowledge distillation on SQuAD v1.1, to produce a fact-checked answer from the synthetic question and passage.
2. We then used a neural sentence BERT model to convert the fact-checked answer and synthetic answer into a dense vector embedding.
3. Using the dense vector embeddings, we then calculate a cosine similarity score, and if it is above a threshold value of 0.8, then this pair of synthetic question-answer will be retained in the filtered data set.

Step 2 of the algorithm is crucial as this distinguishes our method from the current round trip consistency method, where a more lexical approach has been taken. We employ our version of round trip consistency to filter out poor quality QA pairs.

The pseudocode for our round trip consistency algorithm is:

```python
# Read each synthetic QA pair
for row in reader:

    # Generate answer from question and context using distillbert
    answer = None
    try:
        answer = GenerateAnswer(row['question'], row['context'])
    except Exception as e:
        print(e)

    # Convert synthetic answer and distillbert answer into dense embedding
  using sentence transformer
    # And calculate cosine similarity
    sentence_embeddings = modelSentence.encode([answer, row['answer']])
    cos_sim = 1 - spatial.distance.cosine(sentence_embeddings[0],
  sentence_embeddings[1])
    # print(cos_sim)

    # If similarity of answers are above a threshold we accept generated
  synthetic QA pair
    # Need to optimise threshold
    if cos_sim > 0.8:
        synthetic_QA_round_trip_writer.writerow([row['context'], row['
  question'], row['answer']])
```

Listing 4: Synthetic QA Generation Algorithm

## 4.5 Abstractive Summarisation for Unambiguous Generation of Synthetic QA Pairs

Current methods for generating synthetic QA pairs are based on a copy-paste mechanism (Puri et al., 2020). In the literature, the answer is usually obtained from an extractive model, whereby the final output layer produces a log-likelihood distribution of the most likely answer locations within a passage (Puri et al., 2020). The extracted answer is then fed into a seq-2-seq language model to generate the question (Puri et al., 2020). We argue that this approach can lead to ambiguous question-answer pairs because some extracted answers may neglect the context and overall meaning of the passage. Therefore, in this part of our work, we mainly explore how to synthetically generate QA pairs that are more specific and less ambiguous.

As the first contribution, we introduce a novel synthetic QA mechanism that reduces the ambiguity of question-answer pairs using an abstractive summarisation technique. A synthetic answer is generated by summarising a given passage. A synthetic question is then generated with the synthetic answer (summary) and the original passage. We hypothesise that this approach reduces the ambiguity of synthetically generated QA pairs. A summary of a passage is likely to be more information-rich and, therefore, more likely to contain factual information than the current extractive approaches. For this section of our work, we set our domain as COVID19 as there is an open research data set, CORD-19, with over 500,000 scholarly articles which we can leverage to generate our data. Our algorithm for generating such unambiguous synthetic QA pairs process is as follows:

1. Process the CORD-19 data into 100 word passages.
2. Using a neural summarisation model such as SCI-TDLR, trained specifically for scientific papers, we summarise the 100 word passages to obtain a synthetic answer (Cachola et al., 2020).
3. We then use a T5-based question generator to generate a synthetic question given the summary and the 100 word passage (iarfmoose, 2021).
4. With the synthetic QA pair, we employ the above dense vector round-trip consistency method to filter the data.

The pseudocode for our synthethic QA generation algorithm is:

```python
def generate(data, split, chunk_size = 5):
    '''generate takes a list of passages and generates a list of synthetic
   QA pairs
    :param data: [list]
    :param split: str
    :param chunk_size: int
    '''
    # Create synthetic qa generator object from class
    # The class contains a SCI-TLDR model for answer sumarisation
    # The class contains a T5 model for question generation
    qa_gen = SyntheticQAGenerator()

    # Open files for saving QA pairs
    source = open('{}.source'.format(split), 'w')
    target = open('{}.target'.format(split), 'w')

    # Process passages in parallel given chunk_size
    numberChunks = math.ceil(len(data) / chunk_size)
    for i in tqdm(range(numberChunks), position = 0, leave = True):
        passages = data['text'][i*chunk_size:(i + 1)*chunk_size]

        # Generate synthetic QA pair
        answers = qa_gen.generate_answers(passages)
        questions = qa_gen.generate_questions(passages, answers)

        # Remove empty results
        for j in zip(answers, questions):
            if j[0] == '' or j[1] == '':
                continue

            # Save synthetic QA pairs
            source.write(j[1] + '\n')
            target.write(j[0] + '\n')
```

Listing 5: Synthetic QA Generation Algorithm

| Question | Answer | Context |
|---|---|---|
| What is an example of a virus that initiates sg mRNA synthesis internally? | brome mosaic virus | Some viruses, such as brome mosaic virus, initiate sg mRNA synthesis internally on the full-length minus strand RNA template... |
| What is the purpose of this study? | to identify domains with homology to AlkB in viral genomes | The purpose of this study has been to identify domains with homology to AlkB in viral genomes, in order to... |
| How many new antibodies were isolated? | 46 | After immunisation and screening 2,000 hybridomas, we isolated 46 new monoclonal antibodies... |

Table 1: Examples of unambiguous synthetic QA pairs.

In our experiment, we use the above algorithm to generate three different scenarios for which to compare and test our hypothesis:

1. 100K synthetic QA pairs: traditional extractive method to use for baseline comparison (this data has been provided by Shamane Siriwardhana)
2. 100K synthetic QA pairs: abstractive summarisation for unambiguous generation
3. 100K synthetic QA pairs: traditional method + 50K synthetic QA pairs: abstractive summarisation

With the three scenarios, we then fine-tune RAG, assessing the performance using the COVID-QA data set. In the first scenario, we will be fine-tuning RAG using the original synthetic QA data (provided by Shamane Siriwardhana), and this will act as a baseline benchmark. In the second scenario, we will be fine-tuning RAG using the synthetic QAs generated using abstractive summarisation. In the third scenario, we will fine-tune RAG with a mixture of 100K QA pairs from the original method and 50K QA pairs from the abstractive summarisation method. To evaluate performance, we will compare the exact match (EM) scores of the last two scenarios with the first scenario.

## 4.6 Question Enhancement For Better Information Retrieval

Siriwardhana et al., along with our research, have found that in a domain-specific setting, RAG under-performs when the question is very ambiguous (Siriwardhana et al., 2021). As a start, RAG has to retrieve passages to answer questions, and this is a much more challenging task since little context other than the question, and we rely on the model itself to make good judgements on which passages to retrieve. Given that RAG has been trained on general domain data sets such as SQuAD, where questions are generic and passages based on articles found in Wikipedia, it is not surprising that RAG suffers in performance when it is used in more specific domains such as COVID19 (P. Lewis, Perez, et al., 2020).

To address the challenge of retrieving supporting documents, we explored the second RAG-friendly signal to enhance the amount of relevant information available to the input question. This is an attempt to improve the performance of the DPR in RAG by supplementing the input question with additional signals/context. Our approach is similar to the internet-augmented dialogue generation used by BlenderBot 2.0, whereby the chat-bot can access information via an internet search (Komeili et al., 2021). It was shown that taking a simplistic approach improved the long term memory performance of the chat-bot, and likewise, we hypothesise that our approach will lead to similar improvements for RAG. To enhance the input question for better information retrieval, we used a lexical language model to retrieve the most relevant passage from an external knowledge base, appending it to the question with an unique token. The longer input is then used by RAG for question-answering. It is important to note that this external knowledge base does not need to be RAG's knowledge base; rather, it could be more general, such as the Google search engine database. Our algorithm is as follows:

1. Create a knowledge base to retrieve documents. For our experiment, we used the same knowledge base as RAG.
2. For each question-answer pair, we used a standard Okapi BM25 model to retrieve the most relevant document to the question. In information retrieval, Okapi BM25 is a ranking function used by search engines to estimate the relevance of documents to a given search query (Sparck Jones et al., 2000).
3. Concatenate the retrieved passage to the question with a <BM25> token as follows: question <BM25> context.

The pseudocode for our question enhancement algorithm is:

```
for idx, r in tqdm(data.iterrows(), total = data.shape[0]):  # Iterate
    through a existing data set of synthetic QA pair
     question = preprocess(r['question']) # Clean the question

    docScores = BM25.get_scores(question) # Retrieve most relevant passage
using BM25
    idxTopKDocs = np.argsort(docScores)[::-1][:topK]

    textfile.write(r['question'] + ' <BM25> ' + knowledgebase['text'][
idxTopKDocs].to_list()[0] + "\n") # Save new data
```

Listing 6: Question Enhancement Algorithm

| Question | Enhanced Question |
|---|---|
| What is the main cause of HIV-1 infection in children? | What is the main cause of HIV-1 infection in children? <BM25> Fifty-one children had primary dengue infection and 45 had secondary dengue infection... |
| What plays the crucial role in the Mother to Child Transmission of HIV-1 and what increases the risk? | What plays the crucial role in the Mother to Child Transmission of HIV-1 and what increases the risk? <BM25> DC-SIGNR plays a crucial role in MTCT of HIV-1 and that impaired placental DC-SIGNR expression increases risk of transmission... |
| How many children were infected by HIV-1 in 2008-2009? | How many children were infected by HIV-1 in 2008-2009? <BM25> gastroenteritis in breast-fed children infected with E. coli serogroup O 111 compared to that in formula-fed infants, similarly infected... |

Table 2: Examples of enhanced synthetic questions.

In our experiment, we processed a pre-existing synthetic data set, consisting of about 175K QA pairs, to create an enhanced version. We then fine-tuned two RAG models, one with the original synthetic data set and the enhanced version. The original data set will act as a baseline benchmark to compare against the enhanced version. Finally, we processed a validation data set, consisting of about 2K QA pairs, in a similar way. We evaluated our model's performance using this validation data set.

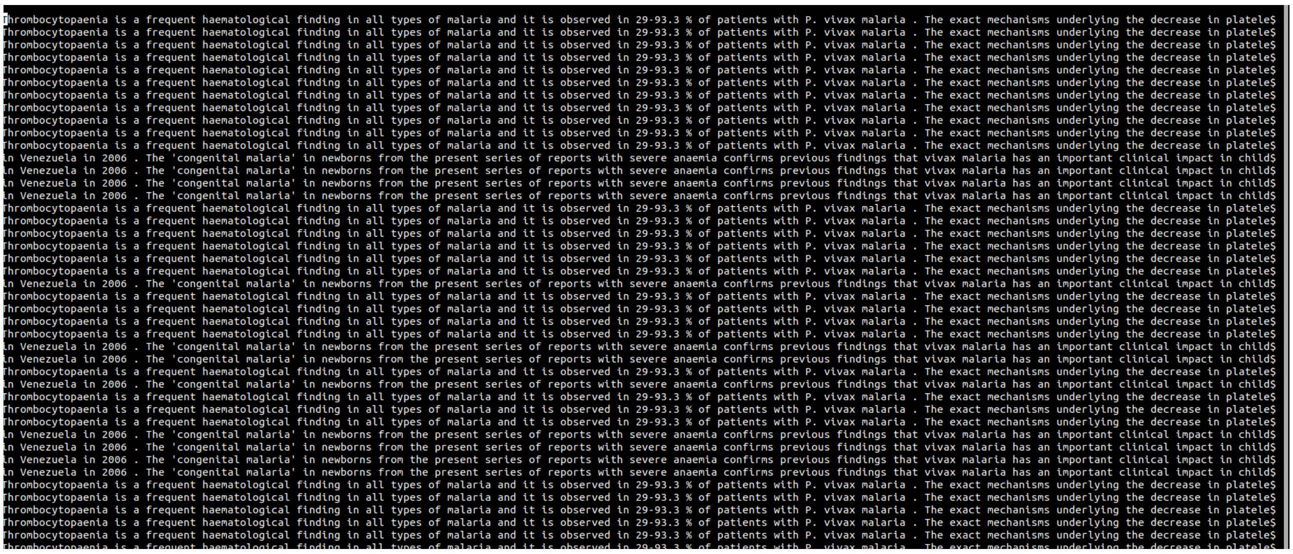
# 5 Results and Discussions

## 5.1 Abstractive Summarisation for Unambiguous Generation of Synthetic QA Pairs

As the first major contribution, we explored a novel synthetic QA mechanism that uses an abstractive summarisation model instead of the traditional extractive models to generate unambiguous question-answer pairs. With experimented with three scenarios:

1. 100K synthetic QA pairs: traditional method to use for baseline comparision
2. 100K synthetic QA pairs: abstractive summarisation for unambiguous generation
3. 100K synthetic QA pairs: traditional method + 50K synthetic QA pairs: abstractive summarisation

The first scenario, containing 100K synthetic QA pairs (generated using the traditional extractive method), sets a baseline exact match score of 34%-40% on the benchmark COVID-QA dataset. Our hypothesis for the remaining two scenarios was that using unambiguous QA pairs would improve RAG's exact match score on the same COVID-QA dataset. However, we discovered that RAG suffered from retrieval collapse, a phenomenon observed in the original RAG paper (Figure 4) (P. Lewis, Perez, et al., 2020). Retrieval collapse refers to the case where the generator would learn to ignore the documents in the knowledge base, which means that the same set of documents would be retrieved for every question-answer pair (P. Lewis, Perez, et al., 2020). Our results showed that the fine-tuned RAG model would perform equivalently to BART because it did not use the retrieval model.



```
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
in Venezuela in 2006 . The 'congenital malaria' in newborns from the present series of reports with severe anaemia confirms previous findings that vivax malaria has an important clinical impact in child$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
Thrombocytopaenia is a frequent haematological finding in all types of malaria and it is observed in 29-93.3 % of patients with P. vivax malaria . The exact mechanisms underlying the decrease in platele$
```

Figure 4: Retrieval Collapse

According to the original paper, retrieval collapse could be due to three main reasons (P. Lewis, Perez, et al., 2020):

1. There is a less-explicit requirement for factual knowledge in some tasks.
2. Longer target sequences, which could result in less informative gradients for the retriever.
3. Spurious retrieval results when optimising a retrieval component in order to improve performance on downstream tasks.

From our work and understanding of the problem, we suspect that the second reason (longer target sequences) made it extremely difficult for the RAG model to find relevant documents for question-answering. The summarised passages as the synthetic answers were usually quite long, spanning up to 30 characters in length, which meant less informative gradients for the retriever during backpropagation. Therefore, the gradients would not change significantly, and the model gets trapped in a local optimum. As a result, the model learns to ignore the retrieval component, defaulting to a similar set of documents for every QA pair, even if there is no similarity. We believe that more informative synthetic data is required to train the model effectively, without additional architecture modifications. For example, our second contribution appends additional information to the input question to potentially aid the retrieval component of RAG.

## 5.2 Question Enhancement For Better Information Retrieval

As the second significant contribution, we explored a novel technique to enhance the information of the input question. Our approach parallels the approach taken in BlenderBot 2.0, where instead of an internet search, we index our knowledge base for the most relevant passage using a BM25 model (Sparck Jones et al., 2000). We then append this passage to the input question with a unique <BM25> token. Our experiment used an existing set of synthetic QA pairs, where we ran our algorithm to generate an enhanced version for better information retrieval. We required computational aid from Siriwardhana et al. to fine-tune RAG fully end-2-end.

Our experiments found that fine-tuning RAG fully end-2-end on the original data set gave us an exact match score between 34%-40% on the validation data set, and on the enhanced data set, we obtained an exact match score between 54%-66% on the validation data set. The exact match score measures the percentage of predictions that match any one of the ground truth answers exactly (Rajpurkar et al., 2016). Although we see a measurable increase in the exact match score of RAG using the enhanced data, we again observed the retriever collapsing.

Overall, our hypothesis that increasing the amount of input information for RAG increases overall performance is significant – appending the most relevant passage to the question through lexical matching does in fact improve the exact match score. However, the reoccurring retrieval collapse suggests that longer sequences, regardless of input or output, causes the model to ignore information retrieval. From this, we can infer that the increase in performance of RAG is merely a result of increasing the context of the question through an initial BM25 search. Furthermore, this suggests the model learns that it is not essential to search for relevant passages to answer the question.

## 5.3 Limitations of Research

Our first contribution used dense vector embeddings for round trip consistency, and this required a powerful computer to process a data set within a reasonable amount of time. Given Google Colab usage limits, we could not run our round trip consistency method over a long period. Furthermore, running the code locally on our computers was also infeasible since it would take many hours to process a few thousand data samples. Because we generated over 495K synthetic QA pairs, it was difficult to assess the effectiveness of our dense vector round trip consistency method. Nevertheless, our contribution has been used by Siriwardhana et al. and was shown to improve data quality.

Similarly, our second contribution for creating unambiguous synthetic QA pairs was also limited by our computational hardware and pre-existing models. We relied on pre-existing models found within the Hugging Face Transformer's library for our summarisation model and question generation model. While these models functioned perfectly fine, they may not have been optimal for our use case. For instance, the SCI-TLDR model was trained on scientific documents, whereas our T5 question generator model was fine-tuned on a series of generic QA data sets such as SQuAD. Because the question generator model was trained on generic QA pairs, it would have been difficult to generate unambiguous questions even if the summary from the SCI-TLDR model was specific; therefore, resulting in poor data quality.

Our third contribution was limited by only having access to one lexical information retrieval model (Okapi BM25). We were unable to explore the use of other types of non-neural information retrieval models. Furthermore, our knowledge base was also limited since we use the same one as RAG. In BlenderBot 2.0, an internet search was made to improve the model's performance. Likewise, a more extensive knowledge base such as the internet may also provide similar or better benefits to RAG. A potential next step would be to explore a larger knowledge base for our question enhancement technique.

Lastly, our final limitation would be that we could not fine-tune and train RAG full with our hardware. We leveraged external help from Siriwardhana et al. to train RAG, which meant that we could not modify specific parameters and experiment with other ideas, so our results were limited.

## 5.4 Other Insights and Next Steps

Siriwardhana et al. have already demonstrated a performance increase to RAG by fine-tuning RAG fully end-2-end, so any additional architectural changes are unlikely to be made. We noticed that RAG's performance is mainly limited by the retriever being unable to find relevant passages to answer ambiguous questions in a more domain-specific setting such as COVID-19. Therefore, from our understanding of our results, we suggest that there needs to be more research into data-centric methods for improving RAG.

As mentioned as one of our limitations above, the reliance on free and open-source models from the Huggingface Transformer's library may have led to sub-par results. Ideally, we will train and fine-tuned our own text summarisation and question generation model as our next step. Doing so will ensure that our synthetically generated QA pairs are of the highest possible quality.

We found that our question enhancement approach improved RAG's EM score by about 20%, but this was not a result of RAG making more informed decisions in information retrieval. Therefore, more research is required to discover additional unsupervised techniques that improve the input question's context. A potential method for enhancing the input question is to employ a topic modelling technique (such as Latent Dirichlet Allocation) to discover "topics" that occur for each passage in the knowledge base (Blei et al., 2003). After classifying the knowledge base, we can apply a similar technique to the question. Therefore, with a topic associated with each question and passage, the DPR might be able find more relevant documents and not result in the retriever collapsing.

Finally, our next step is to combine our unambiguous synthetic QA generation method with our question enhancement method to prevent the retrieval collapse. As stated above, the retriever collapsing from our synthetic QA was likely due to the longer summarised answers, making it hard for RAG to utilise the DPR fully. If the input sequence was also long (using our enhanced questions), perhaps this would supply the backpropagation algorithm with additional information to select relevant passages and produce longer answers.

# 6 Conclusion

Our research presented two data-centric contributions to improve the performance of RAG, with a focus on the COVID-19 domain, and they were: (1) generation of unambiguous synthetic QA pairs using an abstractive summarisation method, and (2) enhancement of information to the input question for improving the retriever component in RAG. We also had two minor contributions: (1) using dense vector embeddings for round trip consistency, and (2) implementing a RAG pipeline. For each contribution, we established a baseline RAG model for comparison. We found that an abstractive approach to generate unambiguous synthetic QA pairs resulted in retrieval collapse, a phenomenon that occurs due to a longer target sequence. Finally, we found that appending related passages to the input question improves RAG's exact match score by about 20%, but this is not a result of the retriever making better judgments as we still observed retrieval collapse. Overall, our research showed that further efforts focusing on data-centric techniques are required to improve RAG's ODQA performance without the issue of retrieval collapse.

## References

Alberti, C., Andor, D., Pitler, E., Devlin, J., & Collins, M. (2019). Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416.*

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165.*

Cachola, I., Lo, K., Cohan, A., & Weld, D. S. (2020). TLDR: Extreme summarization of scientific documents. *arXiv:2004.15011.*

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051.*

DeepsetAI. (2021). Haystack – an end-to-end framework that enables you to build powerful and production-ready pipelines for different search use cases. https://github.com/deepset-ai/haystack

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

Google. (2021). Google colab – colab notebooks allow you to combine executable code and rich text in a single document, along with images, html, latex and more. https://colab.research.google.com/

HuggingFace. (2021). Hugging face – the ai community building the future. https://huggingface.co/

iarfmoose. (2021). Iarfmoose/t5-base-question-generator. https://github.com/AMontgomerie/question_generator

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906.*

Komeili, M., Shuster, K., & Weston, J. (2021). Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566.*

Lee, K., Chang, M.-W., & Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300.*

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461.*

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401.*

Lewis, P., Stenetorp, P., & Riedel, S. (2020). Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637.*

Liu, S., Zhang, X., Zhang, S., Wang, H., & Zhang, W. (2019). Neural machine reading comprehension: Methods and trends. *Applied Sciences*, *9*(18). https://doi.org/10.3390/app9183698

Prager, J. (2006). Open-domain question-answering. foundations and trends in information retrieval. *Foundations and Trends in Information Retrieval(Vol. 1, Issue 2).*

Puri, R., Spring, R., Patwary, M., Shoeybi, M., & Catanzaro, B. (2020). Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599.*

Qaiser, S., & Ali, R. (2018). Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, *181.* https://doi.org/10.5120/ijca2018917395

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250.*

Reddy, R. G., Iyer, B., Sultan, M. A., Zhang, R., Sil, A., Castelli, V., Florian, R., & Roukos, S. (2020). End-to-end qa on covid-19: Domain adaptation with synthetic training. *arXiv preprint arXiv:2012.01414.*

Semnani, S. J., & Pandey, M. (2020). Revisiting the open-domain question answering pipeline. *arXiv preprint arXiv:2009.00914.*

Shakeri, S., Santos, C. N. d., Zhu, H., Ng, P., Nan, F., Wang, Z., Nallapati, R., & Xiang, B. (2020). End-to-end synthetic data generation for domain adaptation of question answering systems. *arXiv preprint arXiv:2010.06028.*

Siriwardhana, S., Weerasekera, R., Wen, E., & Nanayakkara, S. (2021). Fine-tune the entire rag architecture (including dpr retriever) for question-answering. *arXiv preprint arXiv:2106.11517.*

Sparck Jones, K., Walker, S., & Robertson, S. (2000). A probabilistic model of information retrieval: Development and comparative experiments: Part 1. *Information Processing & Management*, *36*(6), 779–808. https://doi.org/https://doi.org/10.1016/S0306-4573(00)00015-7

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998–6008.

Weng, L. (2021). How to build an open-domain question answering system? https://lilianweng.github.io/lil-log/2020/10/29/open-domain-question-answering.html

Woods, W. A. (1973). Progress in natural language understanding: An application to lunar geology. *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, 441–450. https://doi.org/10.1145/1499586.1499695