Utrecht University

Department of Information and Computing Science

Applied Data Science Master Thesis

An assessment of Zero-Shot Open Book Question Answering using Large Language Models

First examiner:
Yannis Velegrakis

Second examiner:
Enas Kwaileh

Candidate:
Bob Merkus

In cooperation with:
SUE B.V.

July 3, 2023

Abstract

In this thesis, we aim to compare the performance of State-Of-The-Art Language Models in a Zero-Shot Open Domain Question Answering setting for technical topics, specifically regarding cloud technology and containerization. Question Answering has historically been mostly extractive in nature, but in recent years we have seen the paradigm of Natural Language Processing switch towards the more abstract Natural Language Generation approach. We propose a two-step architecture, in which the solution attempts to answers questions from a set of documents with no prior training or fine-tuning. We do not solely focus on Retriever-Reader methods (e.g., BERT, RoBERTa), but also evaluate Retriever-Generator (e.g., GPT, FLAN-T5) systems through Long Form Question Answering. The Amazon Web Services dataset is used as an benchmark for evaluating performance of the zero-shot Open Book Question Answering system [1]. Empirical results are sometimes obtained by splitting the documents in to smaller subsections like paragraphs or passages, therefore we analyse the hyperparameters for document splitting using a sliding window. We show that RoBERTa-large is able to achieve a new State-Of-The-Art F1 score of 59.19 through proper pre-processing of the documents and carefully selecting hyperparameters, gaining a respectful 18.66 compared to the baseline and 16.99 compared to the best results in the original study. We conclude that generative models and Long Form Question Answering demonstrate great potential, but come with their own set of biases and risks. We observe that when the complexity of the model far exceeds the evaluation metrics, the relevance and meaning of the metrics become questionable. In this context, Semantic Answer Similarity and METEOR prove useful for analyzing diverse model outputs, as they are not dependent on lexical stride like ROUGE, BLEU, F1 and EM. Splitting documents into passages offers performance benefits, although it is important to note that document splitting may not necessarily be superior for all use cases and the optimal hyper parameter values are expected to vary depending on the specific application.

Contents

Appendix

1. Introduction

Recently, we have seen a explosive increase in the popularity of publicly available generative Artificial Intelligence (AI) systems. We have seen an unprecedented adoption rate of "Chat Bots" in particular, as these systems have demonstrated to be useful in both personal and commercial use-cases. The main driver behind the popularity and attention of Natural Language Processing (NLP) systems in the past years can be traced back to Large Language Models (LLM's). LLM's have been shown to be able to generalize, reason, solve problems, think abstractly, and comprehend ideas [2]. The rise of these generative AI systems are hypothesized to cause disruptions in the way most people go about their daily lives and how organizations apply their practices. This can have a considerable social and economic implications as approximately 19% of jobs in the US have around 50% of their tasks exposed to LLM's when considering both current model capabilities and anticipated LLM-powered software [3]. With the increasing demand for AI systems, there is a need for robust and accurate systems that can handle complicated questions while still being able to provide precise and explainable answers.

1.1 Motivation and Context

Question Answering has historically been mostly extractive in nature, but in recent years we have seen the paradigm of NLP switch towards the more abstract Natural Language Generation (NLG) approach. NLG is the process of producing a natural language text in order to meet specified communicative goals [4]. The texts that are generated may range from a single phrase given in answer to a question, through multi-sentence remarks and questions within a dialog, to full-page explanations [5]. This has been implemented through various model architectures like the Text-To-Text Transfer Transformer (T-5) [6] [7] and the Generative Pre-Trained Transformer (GPT) [8] [2].

1.2 Literature Review

1.2.1 Transformer Architecture

The transformer was first introduced in the paper “Attention Is All You Need” from AI researchers at Google Brain [9]. It has proven to be a more efficient and reliable design than previous State-Of-The-Art (SOTA) methods, such as Recurrent Neural Networks (RNN's) and Convolutional Neural Networks (CNN's). The transformer was developed to simplify Neural Network architecture, while still employing successful techniques such as the encoder-only or encoder-decoder architectures. The self-attention mechanism is the simplest form of an attention mechanism and it relates the positional importance within a sequence. The entire transformer design handles long term-dependencies especially well though the multi-head self-attention mechanism and has been shown to be more efficient and allow for increased parallelization. The architecture has proven itself in multiple fields of Artificial Intelligence, notably in NLP, Computer Vision and Audio signal processing.

1.2.2 Large Language Models

The past decades have had a high availability of text data on the internet, leading to the development of many different Language Models by AI researchers. On top of that, computational resources in modern data centers have scaled a lot through Moore's Law, which states that *"the number of transistors in an integrated circuit doubles about every two years"*. This has allowed for Large Language Models (LLM's) to be developed by training Neural Networks on vast corpora retrieved from the internet, also being known as the large family of Pre-Trained Models (PTMs) [10]. Most recent LLM's can be described as Deep Neural Networks (DNN's) with transformer architectures that can process relatively long sequences of text with high accuracy and efficiency. These models have shown remarkable performance in various NLP tasks such as language translation, sentiment analysis, text classification, Question and Answering (QA), text similarity and text generation. The literature on LLMs has grown rapidly in recent years. Some of the notable models include BERT [11], RoBERTa [12], ALBERT [13], BART [14], XLNet [15], T-5 [8], FLAN-T5 [7], LaMDA [16], LLaMA [17], GPT-2 [18], GPT-3 [8], GPT-3.5 (ChatGPT/InstructGPT) [19] and GPT-4 [2]. The reader is referred to the original literature for a comprehensive technical explanation of the model architectures. These pre-trained models have distinctive designs that can roughly be described into bi-directional models such as BERT, uni-directional or casual-decoder (left-to-right decoder) models such as GPT, and the encoder-decoder generative model like BART and T5 [20].

LLMs have achieved State-Of-The-Art (SOTA) performance on a range of NLP benchmarks, such as the Stanford Question Answering Dataset (SQuAD) [21][22], TriviaQA [23], and Natural Questions [5]. The development of LLM's has introduced new approaches such as pre-training and fine-tuning, masked language modeling, permutation language modeling, few-shot-learning and recently zero-shot learning. These models have been proven to increase performance when scaling the number of tasks, network size and when using Chain-of-Thought (CoT) prompting [8][24][7]. There are some specific biases and shortcomings of these models and the fine-tuning process often requires complex alignment procedures. LLM's can be prone to abuse so restrictions and guardrails are needed to avoid harmful actors (e.g., fake news, harmful content) and one proposed solution is Reinforced Learning with Human Feedback (RLHF). which has proven very useful for the alignment of GPT models [19].

LLM's seemingly have emergent properties, in which identical architectures display disproportionate phase shifts in performance upon scaling model size [25]. Some have even went as far as claiming sparks of Artificial General Intelligence (AGI) in GPT-4 [26], although other have been pushing back against these claims. Sceptics argue that there is the rising concern that LLM's exhibit "data contamination": when downstream test sets find their way into the pre-train corpus [27].

1.2.3 Transfer learning: Pre-training & Fine-tuning

Transfer Learning is a concept first explored in the 1970's that is thought to have originated from the work of Stevo Bozinovski [28]. Some of the first Transfer Learning implementations happened within Computer Vision applications by replacing the top layer from a Neural Network, while still utilizing the 'knowledge' in the trained network. This method aims to use knowledge that has been embedded in the latent vector space of previous layers. Transfer Learning is a popular approach within recent years of NLP, in which models are pre-trained on large unlabeled text corpora and fine-tuned on a specific downstream task [6] [10] [29] [7]. The most popular pre-training objective for bi-directional models is often through a Cloze style masking procedure also known as "span corruption", therefore they are also referred to as Masked Language Models (MLM's) [8]. One might associate the first stage with unsupervised machine learning, while the second stage is represented by supervised machine learning. The main idea behind this philosophy is that fine-tuning on a specific task enables the model to better utilize the knowledge embedded in the network for that specific task (the model is not learning new knowledge, it is learning to adapt to the use-case).

1.2.4 Information Retrieval

Information retrieval (IR) is a scientific discipline in computing and information sciences that aims to retrieve information efficiently from a collection of documents [30]. Searching can be done on original data formats (i.e. keyword matching on full-text for NLP), but is more often than not implemented through an aggregated numerical representation of the information. There are many ways to do information retrieval, the known method is arguably a sparse Vector Space Model (VSM) representation in combination with a similarity measure [31] [1]. Semantic similarity measures the degree of semantic equivalence between two linguistic items, be they concepts, sentences, or documents [32].

The first step in IR is almost always a data mutation to a numeric data format like a vector, matrix or tensor (e.g., TF-IDF, word/sentence embedding, etc.). The second step often includes a probabilistic model that evaluates the likelihood of a document and query being related. A classic implementation of a VSM is a sparse representation of text using a bag-of-words model like the Term Frequency Inverse Document Frequency (TF-IDF) Matrix [33] on which a Cosine Similarity is applied. The Okapi Best Match (BM25) is a ranking function that was developed in the Probabilistic Relevance Model [34], and has since taken over TF-IDF in popularity through its use in Search Engines. More recently we have seen IR approaches utilize neural retrievers such as retrieval with numeric embeddings like SentenceBERT [35] and Dense Passage Retrieval (DPR) [36]

[1] David Dubin claims that Gerald Salton is unrightfully credited for the VSM introduction to IR: `https://experts.illinois.edu/en/publications/the-most-influential-paper-gerard-salton-never-wrote`

1.2.5 Machine Reading Comprehension

Machine Reading Comprehension (MRC) is an area of research within NLP that occupies itself with teaching computers to read the text and understand the meaning of the text [37]. Neural MRC is a popular approach that relies on DNN's to understand the meaning of text [38]. These systems are most often coined as 'readers', they closely analyze documents and perform the core task of question answering. The methodology section describes specifically how we utilize MRC, among other things.

1.2.6 Question and Answering

Question Answering (QA) is a subdomain of MRC, which involves, not only understanding the question, but also answering it based on facts in a given passage of text [23]. QA has many practical applications, including customer service, education, and information retrieval. The most known variant is the extractive approach, where the objective of the machine learning model is to correctly identify the sub-string from a context that contains the answer to a given question. Zhu et al. [39] note: *"Compared with a search engine, the QA system aims to present the final answer to a question directly instead of returning a list of relevant snippets or hyperlinks, thus offering better user-friendliness and efficiency. Nowadays many web search engines like Google and Bing have been evolving towards higher intelligence by incorporating QA techniques into their search functionalities... In fact, building an OpenQA system that is capable of answering any input questions is deemed as the ultimate goal of QA research."*

1.2.6.1 Extractive vs. Generative Question Answering

In the most general sense there are two types of QA approaches:

Extractive Models: An extractive approach consists of 3 things; question, answer and context. It is mandatory that the question and answer pair has a conclusive factual answer in the context. The model extracts the answer directly from a context document (text, table, HTML page, image, etc.). This applies to literal approaches (Right There Questions), in which there is a literal sub-string match with a numerical start:stop index for the answer. Collecting large labelled datasets is often a costly and time consuming process, which is one of the big downsides of this approach. Some recent trends have included question+answer pair generation for a given context. Here LLM's are used to create a new datasets, although limited is known about data-set quantity and fine-tuning on synthetically created data sets remains an open research topic.

Generative Models: Generate a natural language answer that correctly answers the question, which is also coined the 'Abstractive' approach. The distinction from extractive approach being that the answer does not require a literal sub-string match from the context. This allows for the language model to reply in ways that mimic human language as found in digital text documents. The recent series of generative LLM's are arguably the most powerful AI systems to date that are publicly available and free to use. It opens the door for some interesting and intelligent systems, but there are some known downsides as well. There is an additional risk factor when using generative LLM's as they might display biases and give unexpected/unwanted responses, even after intensive AI alignment. A notorious effect that generative LLM's display are 'hallucinations', especially for auto regressive models like GPT. [40]. The term is often used to describe a model output that is syntactically correct but factually incorrect. Hallucinations can be somewhat controlled by the temperature parameter, often denoted by θ, which works by influencing the probability distribution while sampling the next predicted token. A low value of θ indicates the generation process should closely follow data as seen in training examples, while a high value of θ is associated with more randomness during answer generation, thus showing more 'creativity'. *"We call it creativity when the model generates something we like, while we call it hallucinations when we do not"*.

1.2.6.2 Open Domain Question Answering

Finding the answer to a factual question in a large collection of unstructured documents is a tedious task that many people from a broad range of domains have to complete on a daily basis. This especially applies to knowledge intensive professions like physicians, pharmacists, architects, engineers, scientists, journalists, lawyers and academics. In order to address this task with machine learning approaches, it has been formalized as Open Domain Question Answering (ODQA) [41]. The goal of ODQA is to build automated computer systems which are able to answer any sort of (factoid) questions, sourcing knowledge from a large collection documents. The data-modalities might include unstructured natural language documents, structured data and semistructured data.

In an Open Domain approach the answer is assumed to lie in the set of documents $\mathscr{D}$. It can consist of either Open Book QA [42] [1] [39] or Closed Book QA [43], referring to inclusion or absence of IR at inference respectively. Models that explicitly exploit an external corpus are referred to as Open-Book models [20] [43]. The impracticality of computing answers on all documents that are unrelated to our question naturally requires some sort of IR approach to filter relevant information from the set of documents. The retrieved candidate evidences can be in the form of documents, passages, sentences or phrases where the answer to the question can be found [44]. In a Zero-Shot Open Book Question Answering approach, we are simply not allowed to train the model on the dataset through any form of supervised machine learning, and thus we can only evaluate the system on unseen samples. Not allowing for any machine learning while still solving a task properly requires the system to be flexible and generalize, which is by itself an interesting challenge to solve.

The act of generating a concise answer to a question is also known as Long Form Question Answering (LFQA). The goal is to form a satisfying response to a complicated question that requires *why/how* reasoning. Dan Su thoroughly examined multiple generative approaches for Question Answering with the goal of creating a comprehensive paragraph that answers a given question [20]. Simply put; extractive modelling is inherently not able to provide satisfying explanations, because it requires the understanding of multiple concepts and relating them to each other. We can more structurally dissect Open Domain

Question Answering methods when considering Information Retrieval:

Retriever-Only: One could build a system that solely relies on Information Retrieval (IR) to answer a question. This is a valid approach if there is reason to assume that our question (or a very similar one) has already been answered in our knowledge base, thus we have to merely retrieve it. This is what happens when you use a classic search engine on the internet and it returns documents from forums, social media, etc.

Reader-Only: A reader model extracts the answer directly from a context document, often implemented using a MRC module. With this approach we face the risk of the question not being answerable based on the document. Modern readers often show a confidence score of question being answerable, as suggested in SQuAD v2.0 [22].

Retriever-Reader (Extractive Reader): This 2 step approach aims employ a Retriever model to reduce the number of documents that the Reader must process. The retriever scans through each document $\mathscr{D}_i$, to quickly identify the relevant top-k subset of documents $\mathscr{D}_k$. The candidate documents $\mathscr{D}_k$ are then passed on to the Reader, that in turn extracts the answer based on the question. This approach has been utilized by many [45] [39] [36] [1] [46]. Most often a Extractive Question Answering model like BERT [11] or RoBERTa [12] is used. Petroni et al. [47] have shown that the model can remember large amount of knowledge in all those floating point numbers, showcasing the surprisingly the strong ability of these models to recall factual knowledge without any fine-tuning, making them excellent contestants for ODQA.

Retriever-Generator (Generative Reader): Much like the extractive Retriever-Reader approach, the top-k subset of documents $\mathscr{D}_k$ are selected by the retriever model. The alternative to an extractive reader is a generative model like T-5 [7] or GPT [19]. This approach is suited for questions that are more summarizing in nature (Think and Search Questions), in which the answer has to be gathered from multiple sections of text, and thus more reasoning is required. Retriever-Generator systems are more useful in an abstract QA settings which is also known as Long Form Question Answering (LFQA), in which a model is expected to generate a comprehensive natural language response that is both linguistically and factually consistent. Lewis et al. [48] introduce a Retrieval-Augmented Generation (RAG) models, in which the parametric memory is a pre-trained Sequence2Sequence model and the non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever. Responses are mostly related to the quality of the source documents and effectiveness of the modeling approach, but the generative risks like 'hallucinations' start to appear.

Generator-Only: In this case, no context is provided and thus a fully generative model is required. For a human, knowledge comes directly from the brain. For a transformer-based model, the answer is generated based on the parametric weights in the neural network [43]. Dan Su describes this as "The Closed-book QA Method" [20]. There is no act of

Information Retrieval, so we avoid the risk of the question not being answerable based on selected context. This is good if we know that questions in our application might not factually answerable based on the retrieved context. The absence of a knowledge base during inference requires the LLM to be sufficiently pre-trained and fine-tuned, showing strong performance across benchmarks.

The fundamental nature of this architecture can be seen as a double edged sword; You have a simplified single step inference process of high quality (e.g, generation from GPT-4 [2]) instead of multi step (e.g. retriever-reader). The major disadvantages of relying on a generator-only approach is that it requires a computationally expensive training-process to include up-to-date information from the knowledge base. This is especially problematic for critical applications which rely on factual data (e.g., law, medicine) or when data is updated frequently (i.e, we do not want to retrain the model every week). Additionally, we have to take in to account that model size has scaled exponentially in the recent years, resulting in serious monetary requirements for large training runs.

Sequential Pipelines: Pipelines where the number of nodes exceeds $N > 2$ can be considered as sequential pipelines. An example of this would be *Retriever -> Reader -> Generator* or *Retriever -> Generator -> Generator*. This is an example of what Dan Su proposes with his state-of-the-art Long-form Question Answering System, an end-to-end framework named Read Before Generate (RBG) [20] [49], which is an form of *Retriever -> Reader -> Generator*. There are many possible combinations between models in a sequential manner, so we will not be evaluating any architectures with $N > 2$ nodes.

Automatic AI Agents: Automatic AI agents, also referred to as autonomous AI agents, are computational systems or software applications that possess the ability to carry out tasks or arrive at decisions without direct human involvement. These agents leverage artificial intelligence (AI) methodologies to scrutinize data, acquire knowledge from past experiences, recursively call themselves, and adjust their behavior accordingly in order to accomplish predefined objectives. The design of automatic AI agents aims to facilitate their operation within intricate and ever-changing environments, wherein they can amass information, process it effectively, and undertake suitable actions based on their underlying programming and accumulated knowledge. These agents exhibit a wide spectrum of capabilities, ranging from simple task automation to intricate decision-making processes. These types of systems are excluded from our analysis but some corresponding literature include HuggingGPT [50] and AutoML-GPT [51].

1.2.6.3 Huggingface

The rise of LLM demand and adaptation of Transfer Learning has sparked an sparked an explosion in the development of model checkpoints. Both a cause and effect of increasing demand for open-source machine-learning platforms where these models can be distributed and tested. Huggingface is an Open-Source platform that seeks to democratize the distribution of models. Transformers is a python package from Huggingface that allows for interoperability of transformer based Neural Networks between mainstream machine-learning frameworks like PyTorch, TensorFlow, and JAX[52]. It utilizes the flexibility of model checkpoints for the pre-training + fine-tuning approach. The Datasets python package can be used to efficiently load multiple datasets with an Apache Arrow powered backend [53]. The Accelerate package is designed for scalability and implements this through hardware acceleration and distributed training of Huggingface models. Currently there are over 240.000 model checkpoints and 45.000 datasets available though the Huggingface Hub, and there are several platforms that wrap around Huggingface Pipelines, such as Graphcore [54] and Haystack [55].

1.2.6.4 Haystack

Haystack is a python library from Deepset AI that implements Pipelines, which serve as the fundamental structure for connecting data and performing NLP tasks. Within a Pipeline, data flows sequentially from one Node to another. The interactions between Nodes and the flow of data are user-defined [55]. It also contains functionality towards the creation of complex AI Agents, although those will be beyond the scope of this paper. We will be using the Haystack to run our analysis, although there are some slight limitations for model evaluation of generative models.

1.3 Research Question

In this thesis, we aim to compare the performance of state-of-the-art Language Models in a Zero-Shot Open Domain Question Answering setting for technical topics, specifically regarding cloud technology and containerization. We will discuss both extractive and generative approaches, while attempting to evaluate model performance in an uniform manner. The large amount of available information retrieval systems, model checkpoints and the rise of generative models raises the question:

"How can we utilize Large Language Models in a Zero-Shot Open Domain Question Answering setting in an attempt to answer complex technical questions?"

We will investigate the strengths and weaknesses of various models in attempt to provide insights into the suitability of Zero-Shot Open Domain Question Answering architectures. Some of the LLM's are able to perform a wide range of tasks other than QA but those tasks will be excluded from the scope of this paper. Recent approaches also utilize (multi-hop) agents which have been shown to perform well on complex questions, but we will discuss these neither. Due to resource limitations and practical limitations the scope has been limited to models < 10 Billion parameters. We will focus on text-only models so multi-modal MRC is also excluded from the scope of this paper (e.g, tri-encoder retriever, table reader, etc.). Code can be found on GitHub [2] and is distributed under the CC-BY 4.0 licence [3].

[2] GitHub: `https://github.com/BobMerkus/ADS-LLM-QA`

[3] CC-BY 4.0 Licence: `https://creativecommons.org/licenses/by/4.0/`

2. Data

2.1 Description of the data

Open Domain Question Answering requires various datasets for training, bench marking and inference. The dataset(s) available to the system will vary depending on the application at hand and some specific ones will be discussed in this section.

2.1.1 Stanford Question Answering Dataset

We have chosen to note the Stanford Question Answering Dataset (SQuAD) here, not because we are evaluating our system on it, but because many of the open-source models on Huggingface [52] are either trained and/or fine-tuned on SQuAD version 1.1 [21] or 2.0 [22]. QA systems are typically trained on a dataset of question-answer pairs, where each question is associated with a specific answer in the context. SQuAD is one such dataset commonly used for this purpose. The context is usually a paragraph or a longer document, and the questions are designed to be answerable within that context. The dataset contains a wide range of questions across various topics, making it suitable for training and evaluating question answering models.

2.1.2 Amazon Web Services (AWS) Documentation

We will assess system performance using the Amazon Web Services (AWS) Documentation data set, which was published on GitHub [1] by Sia Gholami and Mehdi Noori. The dataset is accompanied by their original paper: *"Zero-Shot Open-Book Question Answering"* [1], focusing on the act of QA without fine-tuning. We will in part by evaluating their work, but also expanding on it by including an additional ODQA approach. The dataset

Question	Gold Answer	YNN	Corresponding Document
What is the maximum number of rows in a dataset in Amazon Forecast?	10 billion	None	amazon-forecast-developer-guide/limits.md
When you stop a DB instance does it retain its DNS endpoint?	When you stop a DB instance it retains its DNS endpoint	Yes	amazon-rds-user-guide/USER_-StopInstance.md

Table 2.1: An extractive sample from the AWS dataset

consists of 100 questions from actual AWS customers that can be answered based on 25,175 documents (which are supplied in markdown format). All questions in the dataset have a valid, factual and unambiguous answer within the accompanying documents, noted in Appendix D. They deliberately avoided questions that are ambiguous, incomprehensible,

[1] AWS Documentation data source: `https://github.com/siagholami/aws-documentation`

opinion-seeking, or not clearly a request for factual information [1]. All answers are supplied in natural language, but also as a classification that defines whether the question is answerable by 'Yes', 'No' or 'None'. Additionally, a singular correct corresponding document is supplied for each question, allowing for a performance evaluation of retriever and reader separately. Thus, the chance of randomly selecting the correct document is $\frac{1}{25{,}175}$ [2]. The dataset is quite challenging because the questions are both extractive and abstractive in nature, as they can be facts retrieved from the documentation like 2.1 or answers written by AWS employees like in 2.2. Furthermore, answers can be contained in either text or tables, exemplifying the need for pre-processing and/or a multi-modal approach.

Question	Gold Answer	YNN	Corresponding Document
What are the Amazon RDS storage types?	General Purpose SSD, Provisioned IOPS, Magnetic	None	amazon-rds-user-guide/CHAP_-Storage.md

Table 2.2: A generative sample from the AWS dataset

2.1.3 Preparation of the data

Data preparation used during analysis consists of processing documents from Markdown to HTML, after which tables are extracted. Tabular data is extracted and converted to comma delimited key:value pairs because a substantial amount of extractive answers in the AWS dataset are contained in tables. The documents are convert to match the haystack document class, after which text pre-processing is applied.

2.2 Ethical and Legal considerations

There are not many ethical and legal considerations to be made for the AWS dataset as it is publicly available under a CC4.0 licence. There is however, much to be said about the LLM's we are including. There are a lot of active developments in alignment & lawmaking, which cannot all be described here. We will however note some different aspects of AI alignment, as LLM's have been found to exhibit biases and vulnerabilities that can influence their output. These biases often stem from the data they are trained on, which often reflects societal biases, as well as the sources they learn from, including potentially unreliable or biased information. Contextual biases can also arise when the model responds to biased or leading prompts. Furthermore, the lack of fact-checking ability can result in the propagation of misinformation. These biases and vulnerabilities can be exploited by bad actors who misuse the technology to spread hate speech, propaganda, or manipulate public opinion. Additionally, language models may inadvertently reinforce existing beliefs, contributing to echo chambers. The complexity and opacity of these models make it challenging to address biases and hold them accountable.

There is also the existential threat of digital super intelligence, also known as the dooms-

[2] AWS Dataset: https://github.com/siagholami/aws-documentation

day argument of Artificial General Intelligence (AGI). It is by some regarded as science-fiction while others believe it is a bigger risk to humanities survival than global nuclear warfare or global warming. There are many ways AI could end up doing more harm than good, both intentional (bad human actor) and unintentional (existential risk) [56]. It is not a rudimentary thing to estimate the risk AI poses, especially if we imagine the trend in model capability to continue. Elizier Yudkowski reasonably argues that there is no human that would intentionally press a button to destroy the world, so the real worry should be doing it by accident while believing the button will make the world a better place. AI Safety is not settled science and we do not have relevant data to make probabilistic estimations of risk: *"We cannot consult actuarial statistics to assign small annual probabilities of catastrophe, as with asteroid strikes. We cannot use calculations from a precise, precisely confirmed model to rule out events or place infinitesimal upper bounds on their probability, as with proposed physics disasters. But this makes AI catastrophes more worrisome, not less"* [57]. Large scale AI systems are already somewhat capable of 'understanding' the human psyche and the public cloud infrastructure they are built upon. This allows for science-fiction scenarios like self-replicating AGI systems, but that is where the line has to be drawn as these complex discussions can also be considered a study of philosophy, not just AI safety and IT security.

Some interesting questions that are being discussed include: *"How can we avoid bad actors from exploiting powerful AI systems? What should AI Agents be allowed to do without human intervention? Should we allow our most intelligent AI's to be connected to the internet in real time? Should there be a physical barrier between data centers hosting our internet and the AI? How do we know if the AI itself is displaying harmful intentions or lying about them? Should we make sure critical infrastructure is not dependent on powerful AI systems in case we have to shut them down? Can we agree on international procedures and limitations through policies, and if so, how are we able to enforce them without hindering legitimate research and use-cases?"*. These concerns are all factors that contributed to an open letter signed by prominent AI researchers called 'Pause Giant AI Experiments' [3]. It is accompanied by 7 proposed policies regarding the development of large scale AI systems [58]:

1. Mandate robust third-party auditing and certification.
2. Regulate access to computational power.
3. Establish capable AI agencies at the national level.
4. Establish liability for AI-caused harms.
5. Introduce measures to prevent and track AI model leaks.
6. Expand technical AI safety research funding.
7. Develop standards for identifying and managing AI-generated content.

[3] Pause Giant AI Experiments: `https://futureoflife.org/open-letter/pause-giant-ai-experiments/`

3. Method

The main goal of our methodology is to asses the viability of several Open Book Question Answering architectures in a zero-shot fashion (i.e., we are not allowed to train our QA model).

3.1 Theoretic Background

3.1.1 Information Retrieval

The problem of Information Retrieval can be formally described:

- Let $\mathscr{D}$ be the set of documents, where each document d in $\mathscr{D}$ is represented as a vector $d = \{w_1, w_2, ..., w_n\}$, where w_i represents the weight or frequency of the ith term in the document.
- Let q be the query or question, represented as a vector $q = \{w_1, w_2, ..., w_n\}$, where w_i represents the weight or frequency of the *ith* term in the query.
- Let $sim(d, q)$ be a similarity measure between a document d and the query q.

Now, we can define the mathematical formulation for finding the correct subset of documents:

- Compute the similarity between the query and each document using a similarity measure, such as cosine similarity: $sim(d, q) = (d \cdot q)/(||d|| * ||q||)$, where (dq) represents the dot product between the document vector d and the query vector q, and $||d||$ and $||q||$ represent the Euclidean norms of the document and query vectors, respectively.
- Rank the documents based on their similarity to the query. The higher the similarity score, the more relevant the document is considered to be.
- Select the top-k documents $\mathscr{D}_i$ with the highest similarity scores as the subset of documents that are likely to be most relevant to the query. We can denote this as $\mathscr{D} = \{\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_n\}$ with $\mathscr{D}_k$ being the top-k retrieved documents for query q.

The naïve approach is to simply concatenate the text of $\mathscr{D}_k$ and truncate the text to match the model size. This however, might cause us to lose the information that contained the answer to our question, which is especially problematic when the document size is much greater than the input size of our model. The probability of success with this approach diminishes as the document size increases or the model input size decreases. Another big problem that arises with large document sizes is the fact that transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length. This is what the longformer architecture attempts to tackle, by introducing a linear scaling self-attention design [59].

3.1.2 Document splitting

Empirical results are sometimes obtained by splitting the documents in to smaller subsections like paragraphs or passages. We refer to a paragraph as "a cluster of sentences grouped under one topic in a meaningful way", while the term passage refers to "an extract from a text, novel, story or a paragraph from a piece of writing or music that is part of a larger piece of work".

A more theoretical justification for splitting the documents in to smaller pieces comes from the fact that the answer to the question might lie in different parts of the document(s), which is applicable to questions that require reasoning over multiple facts. Splitting the document hopefully allows us to efficiently retrieve the information we need to answer the question. Let $\mathscr{P}$ be the set of passages, where each passage p in $\mathscr{P}$ is represented as a vector $p = \{w_1, w_2, ..., w_n\}$, where w_n represent the tokens of the jth passage of document $\mathscr{D}_i$, where $\mathscr{P}_{ij} \subseteq \mathscr{D}_i$. The procedure includes splitting every document $\mathscr{D}_i$ of size n in to j passages using a sliding window approach where each passage $\mathscr{P}_{ij}$ consists of window w with stride s. The number of resulting passages is denoted by j:

$$\mathrm{j} = \left\lfloor \frac{(\mathrm{n}-\mathrm{w})}{\mathrm{s}} \right\rfloor + 1$$

Document $\mathscr{D}_i$ with a size of $n = 200$ tokens with window size $w = 100$ and stride $s = 50$ will result in to $j = \left\lfloor \frac{200-100}{50} \right\rfloor + 1 = 3$ passages of 100 tokens. The reasoning behind a sliding window approach as apposed to normal splitting is that the probability increases of the answer being contained in a continuous sequence of text, which is beneficial for extractive models in specific. The downside of an sliding window approach is that we are duplicating segments of the sequence, which is not very efficient for data storage. Additionally, it might lead to some duplicate information in the model input (if both passages $\mathscr{P}_{ij}$ are considered relevant to query q). The average length of the passages will be slightly less or equal to window size w, while amount of documents is related to both the window size and stride.

3.1.3 Selecting a value for k

The optimal value of top-k can be estimated based the window size w of each passage $\mathscr{P}_{ij}$ (or document $\mathscr{D}_i$ if no splitting is applied) and the token input size of model M, denoted by m:

$$k = \left\lfloor \frac{m}{w} \right\rfloor$$

Lets assume a token size of $m = 512$ for a given model M and a window length $w = 100$ tokens for each $\mathscr{P}_{ij}$, then our estimation would be $k = \left\lfloor \frac{512}{100} \right\rfloor = 5$. Above formula is assuming we are not including an prompt to our model (as is required by generative approaches). Let the prompt size in tokens be denoted by z, our formula then becomes:

$$k = \left\lfloor \frac{m - z}{w} \right\rfloor$$

For an prompt size of $z = 100$ tokens, the new estimated amount of top-k candidate passages $\mathscr{P}_{ij}$ becomes $k = \left\lfloor \frac{512-100}{100} \right\rfloor = 4$.

3.1.4 Question Answering

Dan Su formulates the task of QA as the following [20]; Given a collection of training examples $(q_i, a_i)_{i=1}^{n}$, the goal is to learn a predictor $\mathscr{F}$, which takes the question q as input and outputs the answer a:

$$\mathscr{F} : q \rightarrow a$$

$a = \{a_1, a_2, ..., a_{l_a}\}$, $a_i \in \mathbb{V}$ for $i = 1, ..., l_a$, $q = \{q_1, q_2, ..., q_{lq}\}$, where $q_i \in \mathbb{V}$ for $i = 1, ..., l_q$.

Usually we will also obtain information from a large knowledge source like an database or web API. Upon considering documents that were not present in the training data, the problem becomes:

$$\mathscr{F} : (q, \mathscr{D}) \rightarrow a$$

Just like Dan Su, we will mainly focus on answering the questions q based on information from the external knowledge corpus $\mathscr{D}$. We will not answer questions that are unanswerable or 'out-of-domain' of supporting documents $\mathscr{D}$, as [1] only selected questions with a factual and unambiguous answer.

3.2 Evaluation Metrics

We will evaluate the performance of our Retriever separately from the Reader/Generator.

3.2.1 Retriever

We want our retriever system to list as many relevant documents for query q, and evaluating the retriever requires us to know which documents $\mathscr{D}_i$ are relevant to a given query q. Naturally we can evaluate how well the retriever is performing by calculating the ratio between correctly retrieved documents to the amount of retrieved documents. This is what precision in IR does, denoted as P:

$$\mathrm{P} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

We will evaluate the accuracy of our retrieval system by analyzing the precision of the retriever at different values of k, denoted by *P@K*. We calculate *P@K* to compare our retriever performance to the original zero-shot AWS paper that belongs to the AWS dataset [1]. They analysed 2 retriever systems which we have not (Whoosh and AWS Kendra). The original work notes to be using documents for calculating *P@K*, but seem to have increasing $P@K$ over k, which is impossible because of the fact that you cannot have a $P@K > 0.5$ at $k = 2$ if only 1 document is considered correct (as is the case with the AWS dataset). The formula above would lead to $P@2 = \frac{1}{2} = 0.5$, so it looks like they have noted recall instead of precision. An alternative hypothesis is that they did not calculate $P@K$ based on the correct document $\mathscr{D}_i$, but rather on the amount of correct passages $\mathscr{P}_{ij}$, but this is all speculation as the original code contains no evaluation script [1].

In our case, the retrieved passage $\mathscr{P}_{ij}$ is considered relevant if $\mathscr{P}_{ij} \subseteq \mathscr{D}_i$ in which the answer lies:

$$\mathrm{P} = \frac{|\{\text{relevant passages}\} \cap \{\text{retrieved passages}\}|}{|\{\text{retrieved passages}\}|}$$

Recall aims to measure the opposite of precision, namely the amount of retrieved relevant documents divided by the amount of relevant documents, described as R:

$$\mathrm{R} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

In our scenario there are multiple correct passages $\mathscr{P}_{ij}$ for one query, as we do not know where in document $\mathscr{D}_i$ the answer lies to query q, as such we can only assert a correct retrieval at the document level. The recall single hit metric considers whether at least one of the correct passages $\mathscr{P}_{ij}$ is retrieved whereas recall multi hit takes into account how many of the passages $\mathscr{P}_{ij}$ are correctly retrieved. These will be denoted by $R_s@K$ and $R_m@K$ respectively.

The metrics $P@K$, $R_s@K$, and $R_m@K$, although commonly used in Information Retrieval, have a limitation. They solely focus on evaluating the performance of the retriever based on the presence or absence of documents in $\mathscr{D}_k$, without considering the specific ranking order of those documents. This oversight disregards a crucial aspect of evaluating the effectiveness of a ranking algorithm or system. To overcome this limitation, researchers have developed metrics like the Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG).

3.2.2 Reader / Generator

In our evaluation procedure, we encounter samples that initially have a textual answer but also include additional Yes/No/None (YNN) answers. To handle these cases, we convert the single-label samples into multi-label format by combining the textual answer with the corresponding YNN answer (if available). This conversion results in a multi-label sample with three correct answers. Let's consider a sample from 2.1 to illustrate how an original sample is transformed into a multi-label format. The textual answer "When you stop a DB instance it retains its DNS endpoint" is combined with the YNN answer "Yes". The resulting multi-label consists of three correct answers: "When you stop a DB instance it retains its DNS endpoint", "Yes", and "Yes, When you stop a DB instance it retains its DNS

[1] AWS Original code: `https://github.com/siagholami/zero-shot-open-book-qa`

endpoint" are all considered to be valid answers to the question. This approach allows us to assess the model's performance accurately in cases where both textual and YNN answers contribute to the correct labeling of the sample.

Normalization on gold labels and predictions is performed based on the SQuAD v2 evaluation script. The normalization process involves removing articles, removing punctuation and lower casing. We have added a preceding step that removes trailing zeros from the values (e.g. 32.0 is normalized to 32) as answers are noted without decimals while the original documents include them in 2 samples.

3.2.2.1 Exact Match

The Exact Match (EM) requires a literal string match between the answer and the ground truth. It is also known as a strict evaluation, resulting in a Boolean value that indicates if the generated answer was equal to (one of) the answer(s). It is most often used for reader models to identify a passage that was extracted correctly.

3.2.2.2 F1 score

The F1 scored is an aggregate measurement of precision and recall. For NLP it is often calculated based on the amount of matching words or tokens, but does not take in to account the position of words:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

Note that precision and recall have different meanings for IR and NLP, the former calculates it based on the documents $\mathscr{D}$ and the latter on the number of matching words/tokens between the answer a and the gold standard answer that corresponds to question q. It calculates the Harmonic Mean to penalize extreme values:

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

3.2.2.3 Bilingual Evaluation Understudy

The Bilingual Evaluation Understudy (BLEU) score was originally developed in a Language translation setting but can also be applied to measure the general similarity between the candidate text and reference text [60].

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log\left(\frac{\text{Count}_n}{\text{Total}_n}\right)\right)$$

3.2.2.4 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE-N attempts to measure lexical stride through counting the number of overlapping n-grams ($gram_n$) between the candidate text and the reference text. ROUGE-1 being the uni-gram and ROUGE-2 being bi-gram variant [61] and it is calculated using:

$$\text{ROUGE-N} = \frac{\sum gram_n \in \{ReferenceSummaries\} gram_n \in S \sum Count_{match}(gram_n)}{\sum gram_n \in \{ReferenceSummaries\} gram_n \in S \sum Count(gram_n)}$$

There are related ROUGE metrics like the ROUGE-L, ROUGE-W and ROUGE-S. We will be utilizing the ROUGE-L as it attempts to capture the Longest Common Subsequence (LCS), a useful metric for extractive models.

3.2.3 Metric for Evaluation of Translation with Explicit ORdering

The Metric for Evaluation of Translation with Explicit ORdering (METEOR) is a widely used evaluation metric for assessing the quality of machine translation output. It was designed to address some limitations of the popular BLEU (Bilingual Evaluation Understudy) metric.

METEOR takes into account both precision and recall by incorporating a combination of unigram matching, stemming, synonymy, and word order similarity. It aligns the generated translation with the reference translation and computes various scores based on the matched and unmatched words. These scores are then combined to provide an overall evaluation.

3.2.3.1 Semantic Answer Similarity

Semantic Answer Similarity (SAS) has recently been introduced for evaluating NLP systems in an attempt to tackle the ambiguous nature of text (e.g, synonyms). The above mentioned metrics are all lexical-based and therefore miss out on answers that have no lexical stride but are still semantically similar, thus treating correct answers as false [62]. We use a cross encoder for semantic answer similarity, because it has proven to be more precise than Bi-Encoders [35]. The specific model checkpoint is RoBERTa Large [12] fine-tuned on the STSbenchmark [2].

3.3 Experiment Setup

The entire analysis process has been visualized in 3.1 and can be described by three main steps: Pre-Processing, Inference and Evaluation.

Much like the original AWS paper [1] we propose a two-step architecture, in which the solution attempts to answers questions from a set of documents with no prior training or fine-tuning (zero-shot open book question answering). Unlike their original work, we do not solely focus on Retriever-Reader methods, but also evaluate Retriever-Generator systems. They already noted that generative models like GPT-2 seemed like a natural extension to their research, hinting at the limitations of an extractive approach. The inclusion of both extractive and generative models causes somewhat of a discrepancy in model evaluation (e.g., Exact Match is inherently not suited for generative models). Overall, a retriever-reader or retriever-generator system is relatively explainable as the data sources

[2] Cross Encoder: `https://huggingface.co/cross-encoder/stsb-roberta-large`)

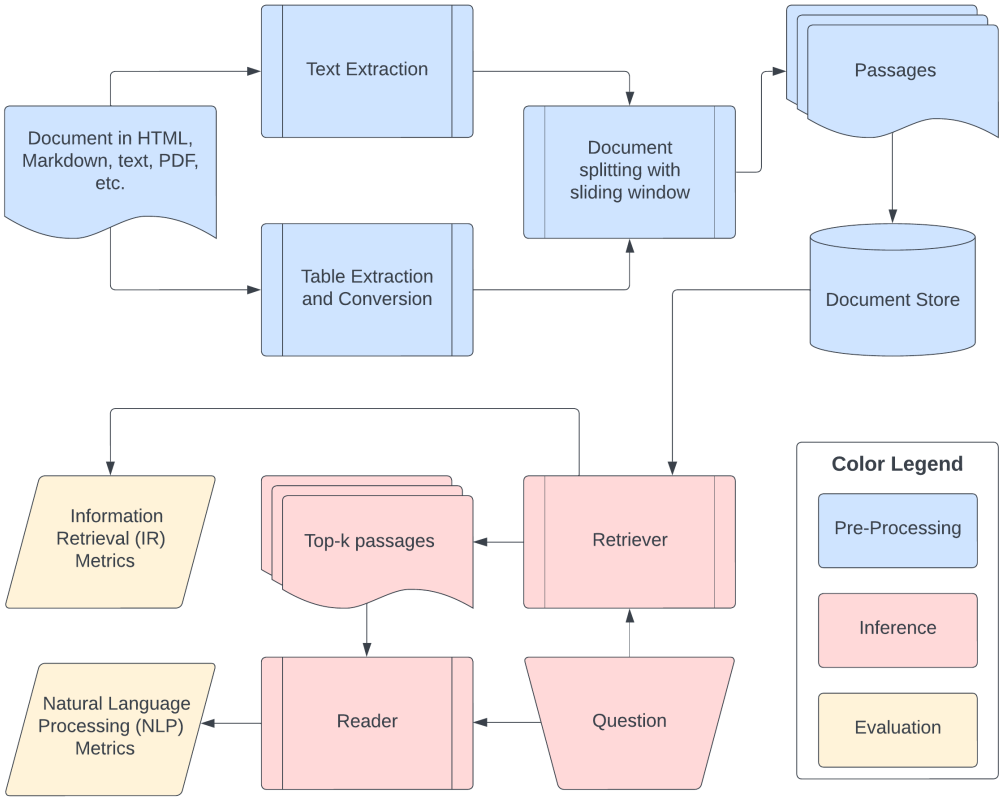

Figure 3.1: High level description of the Open Book Question Answering process visualized using a flow-chart

are noted by the retriever at inference time as opposed to a generator-only approach [43].

Mainly due to time constraints we have decided to limit our retriever to a single sparse retriever, the traditional Okapi BM25. We have briefly explored neural retriever methods, such as Word/Sentence Embeddings (e.g., Sentence BERT [35]) and a Dense Passage Retriever (DPR) [36] with slightly worse performance compared to the BM25 retriever.

We have chosen to incorporate the same models as in original study [1]; BERT (tiny, base, large [11]) along with RoBERTa [12] (base, large), AlBERT [13] V1&V2 (base, large), and distillBERT [63]. We will include one more extractive architecture; the Longformer architecture [59]. All our extractive model checkpoints are fine-tuned on SQuAD V1.1 or V2.0.

Additionally we will evaluate the following generative models; GPT [18] [8] (2, 3.5-turbo), FLAN-T5 [7] (base, large, xl), Flan-Alpaca [64] (base, xl-gpt4, xl-sharegpt), BART [14] (LFQA, ELI5). This selection has been made considering their impressing performance on benchmarks and useful corresponding literature. All models are open-source and available on the Huggingface hub, except for GPT-3.5 Turbo, which is easily accessible via a web API. The entire selection of model checkpoints can be found in Appendix B. We have deliberately left out GPT-4 [2] as it not publicly accessible at the moment of writing (June

2023). Furthermore, the results of GPT-3.5 Turbo and FLAN-Alpaca GPT series are not considered Zero-Shot as it is highly probable that these models have been trained either directly or indirectly on the AWS data set. We utilize a slightly edited version of the Long Form Question Answering (LFQA) prompt for this analysis [3]: *"Synthesize a comprehensive answer from the following topk most relevant paragraphs and the given question. Provide a clear and concise response that summarizes the key points and information presented in the paragraphs. Your answer should be in your own words and be no longer than 50 words.*
Question: {query}
Paragraphs: {join(documents)}
Answer:"

Note that the prompt might be cutoff at the 'join(documents)', so the 'Answer:' might not always be included. Based on literature, we chose $\theta = 0.6$ to prioritize coherent and relevant answers. For top-p sampling a value of 0.9 has been selected, following Dan Su's LFQA research [20].

[3]LFQA prompt template: `https://docs.haystack.deepset.ai/docs/prompt_node`

4. Results

4.1 Baseline top 1 document

The baseline performance for our initial evaluation is based on the top-1 document. This baseline is chosen because some of our models are limited in their ability to process a complete document, making it unsuitable for a direct comparison when using a value of k greater than 1. Most documents are well over the 512 token limit of the T-5 model family for example. The BM25 retriever is evaluated using various metrics, showing a strong baseline performance as shown in table 4.1. The performance of the baseline model is expected to be correlated with the retriever performance.

Metric	**Value**
Mean Average Precision (MAP)	0.63
Mean Reciprocal Rank (MRR)	0.65
Normalized Discounted Cumulative Gain (NDCG)	0.63
Precision (P)	0.65
Recall Multi Hit (R_m)	0.63
Recall Single Hit (R_s)	0.65

Table 4.1: Information Retrieval Performance Metrics

Model Name	ROUGE-L	BLEU	F1	EM	METEOR	SAS
BERT Tiny SQuAD v2	12.07	0.81	12.13	4.0	9.23	20.31
BERT Base SQuAD v2	26.32	1.01	26.86	15.0	17.26	35.08
BERT Large SQuAD v1	34.74	2.84	36.1	20.0	24.56	41.51
BERT Large SQuAD v2	31.1	3.24	32.09	19.0	21.97	39.86
DistilBERT Base SQuAD v1	20.68	3.39	21.31	8.0	15.61	29.12
DistilBERT Base SQuAD v2	21.7	1.43	22.57	6.0	15.87	31.39
ALBERT Base v2 SQuAD v2	25.16	2.33	25.88	11.0	18.08	33.91
ALBERT XXLarge v1 SQuAD v2	34.53	4.95	35.38	21.0	24.49	41.14
ALBERT XXLarge v2 SQuAD v2	37.9	5.79	38.03	23.0	27.35	44.22
RoBERTa Tiny SQuAD v2	29.98	5.81	30.56	16.0	22.09	37.73
RoBERTa Base SQuAD v2	36.23	3.84	36.46	21.0	25.05	41.18
RoBERTa Base Distilled SQuAD v2	33.54	4.39	33.72	20.0	23.39	39.55
RoBERTa Large SQuAD v2	40.51	8.44	40.53	25.0	29.94	43.92
Longformer Base SQuAD v1	30.63	3.35	31.3	18.0	21.22	37.07
Longformer Base SQuAD v2	27.29	6.9	27.48	16.0	20.47	33.31
FLAN T5 Base	30.14	0.51	30.77	19.0	20.02	40.54
FLAN T5 Large	29.33	7.26	30.0	9.0	29.07	46.68
FLAN T5 XL	32.74	3.1	33.22	22.0	25.87	46.81
FLAN Alpaca Base	18.04	5.21	18.91	1.0	25.17	48.27
FLAN Alpaca GPT4 XL	22.37	8.17	24.42	2.0	32.65	51.78
FLAN Alpaca ShareGPT XL	16.94	6.11	17.88	1.0	24.89	47.99
GPT-2	5.43	1.1	5.71	0.0	10.33	63.22
GPT-3.5 Turbo	15.87	4.59	17.14	1.0	28.41	56.44
BART ChatGPT	5.1	0.51	6.19	0.0	11.13	52.75
BART ELI5	4.32	0.7	4.49	0.0	5.28	22.47
BART LFQA	2.36	0.23	2.72	0.0	5.73	62.67

Table 4.2: Baseline NLP results for $k = 1$

Interestingly, all models that were included in the original AWS paper achieve stronger performance in our baseline, with ALBERT-XXL being the only exception. These are promising results as they are obtained by simply using the top-1 document without pre-processing and allowing the prompt to be truncated to match model size. The RoBERTa Large model emerges as the best performing extractive model in table 4.2, showcasing strong baseline performance across all metrics. It provides accurate answers, demonstrates a strong understanding of the context, and achieves high scores in ROUGE-L, BLEU, F1, EM, METEOR, and SAS. The RoBERTa Base model performs on par with the ALBERT-XXL V1 and V2 models, demonstrating that these models offer similar levels of performance in extractive question-answering tasks. The Longformer models perform reasonably well, although using a top-1 evaluation metric may introduce bias against its architecture. It is still a viable choice for question-answering tasks, offering satisfactory performance in various evaluation metrics. The FLAN-Alpaca GPT4 XL model shows strong performance, particularly in METEOR, outperforming the GPT-3.5 model across all metrics except SAS. GPT-2 and BART LFQA exhibits the highest SAS score, but its performance in other metrics are poor. These models often produce results that essentially repeat the question or context in the prompt, indicating a lack of meaningful and accurate answers.

4.2 Exploration of hyperparameters

We have decided to perform a brute force hyper parameter search on window size w, stride $s(s < w)$ and top-k in order to find the optimal values for each. All valid combinations of the following parameters were evaluated: $w = \{50, 100, 150, 200, 250, 500\}$, $s = \{0, 50, 100\}$ and $k = \{1, 2, 3, 5, 9, 10, 20, 30, 40, 50\}$. The average retriever performance has been visualised in figure 4.1 and further shown in Appendix A, showing IR metrics over increasing values of k.

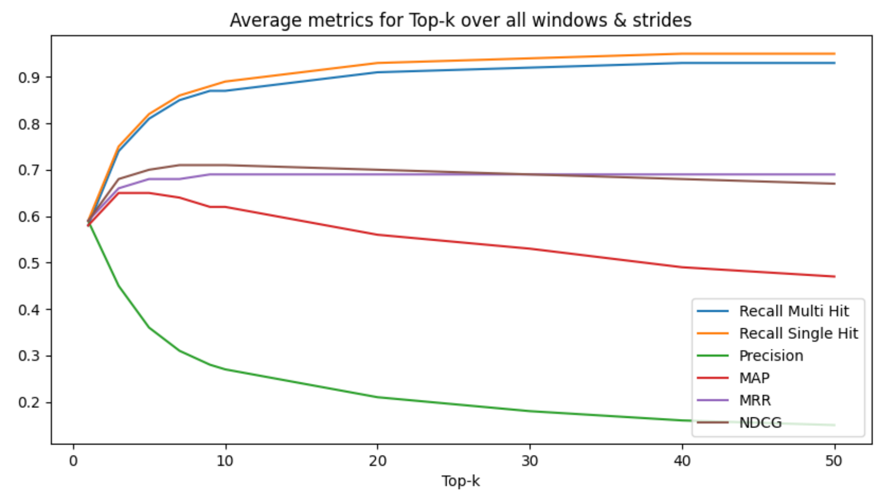

Figure 4.1: Information Retrieval metrics for k, averaged over all window sizes w and stride s

We have selected a sparse retriever (Okapi BM25) and an extractive reader (RoBERTa base fine-tuned on SQuAD2.0) as they are relatively efficient and we are evaluating a large number of combinations. Note that the values for w and s are in number of words and not tokens.

When analyzing the performance metrics of Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG) across different values of k, we observe an interesting trend. The MAP metric reaches its maximum value at $k = 5$, indicating that the retrieval system achieves the highest precision in returning relevant results within the top 5 ranks. However, beyond $k = 5$, the MAP metric starts to degrade, suggesting a decrease in the average precision of the retrieval system. Similarly, both the MRR and NDCG exhibit stationarity and a slight decline at $k = 5$ respectively, suggesting that the reciprocal rank and cumulative gain of relevant results do not significantly improve beyond the top 5 ranks, a conclusion that is in line with the fact that there is only 1 document regarded as correct.

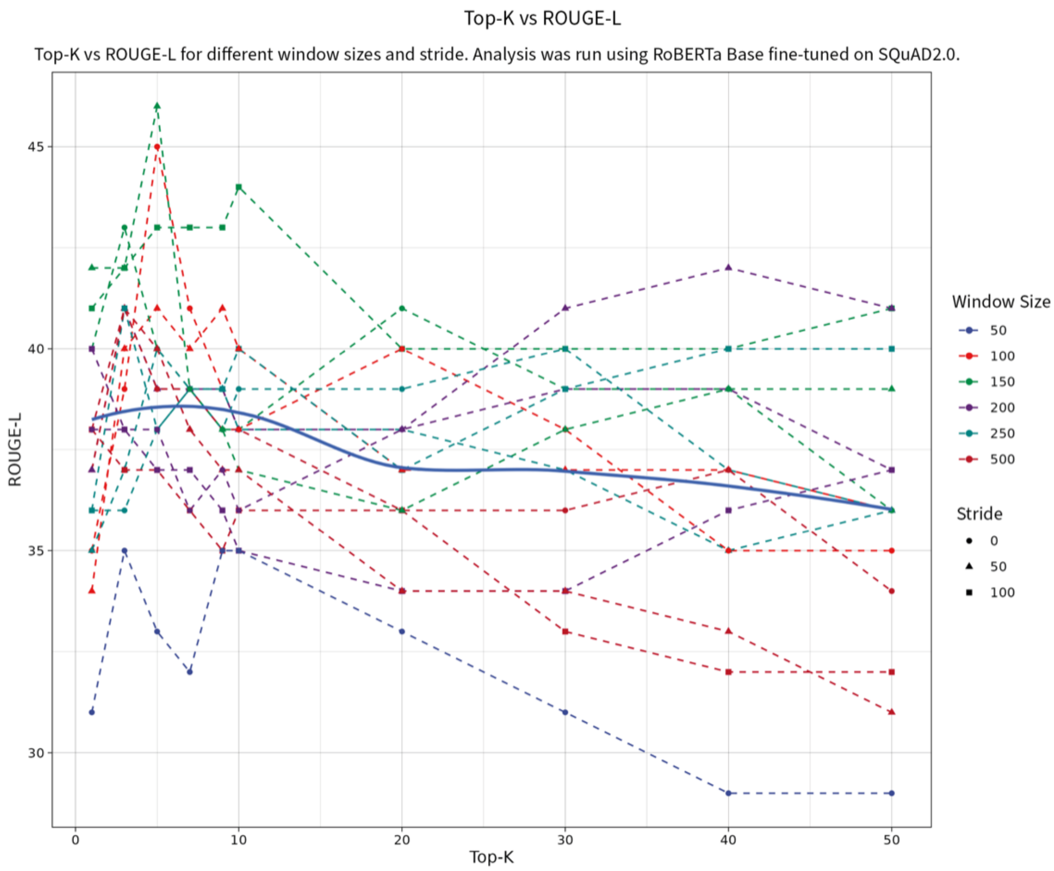

Figure 4.2: ROUGE-L vs. Top-k using Okapi BM25 + RoBERTa-base

Model performance contains some variability as we can see in figure 4.2. Overall the optimal top-k seems to be between $k = 5$ and $k = 10$ and performance seems to depreciate after this value, although some settings show increasing performance when $k > 20$. The maximum ROUGE-L is reached at ($w = 150$, $s = 50$, $k = 5$) while ($w = 100$, $s = 0$, $k = 5$)

and ($w = 150$, $s = 100$, $k = 10$) also perform quite well.

4.3 Model comparison

We ran our final analysis using the optimal hyperparameters ($w = 150$, $s = 50$, $k = 5$). The top-10 performing models according to F1 score have been visualized in figure 4.3, while the full results are shown in table 4.3. RoBERTa-large achieves a new State-Of-The-Art (SOTA) F1 score of 59.19, gaining a respectful 18.66 compared to the baseline and 16.99 compared to the best results in the original study [1].

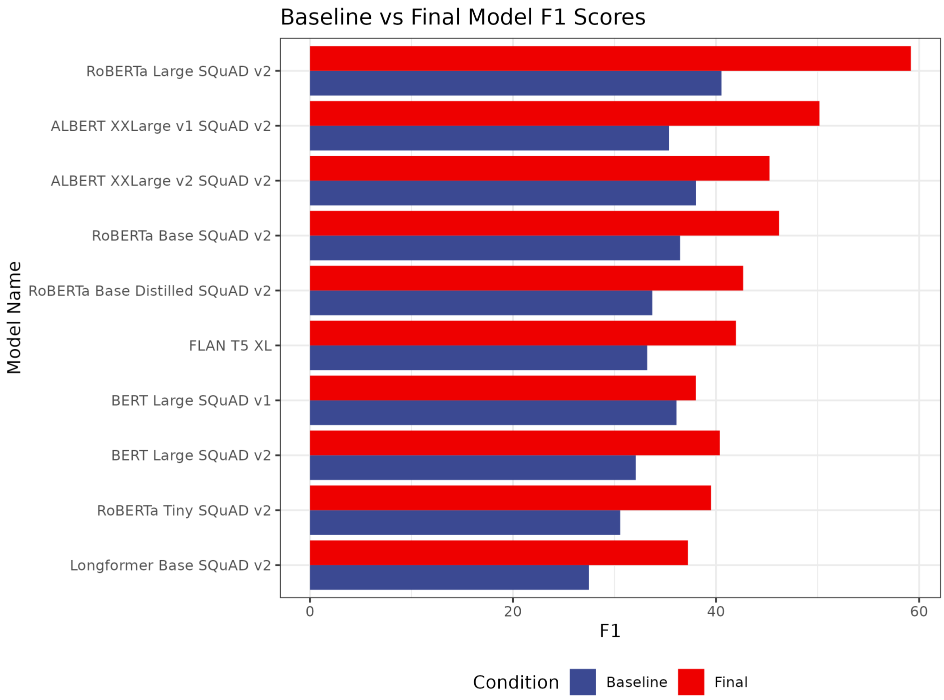

Figure 4.3: Top-10 performing final models according to F1

An interesting observation is that 9/10 of the best performing models according to F1 are bi-directional ones, while FLAN-T5 XL is the only encoder-decoder model. Once again highlighting that generative multi-task models can match or even surpass the performance in an extractive question-answering setting [7]. This demonstrates the effectiveness of generative models in producing accurate answers, while allowing for phrases that consist comprehensive natural language. This in line with what BLEU is telling us, it being the highest for RoBERTa Large, closely followed by FLAN T5 XL with a score of 12.19 and 11.55 respectively. This might indicate that the FLAN series is susceptible to performance increases with document splitting, as the low token input size requires efficient information input. This does not seem to apply to the FLAN Alpaca GPT series, we can observe the BLEU score decreasing when compared to the baseline performance. These results slightly contradict other metrics like F1, METEOR and SAS as these are slightly higher in the final analysis. Just like in our baseline analysis, the experiments show that the BART ELI5, LFQA and ChatGPT model are not able to give meaningful responses, as is the GPT-2 model. This is

Model Name	ROUGE-L	BLEU	F1	EM	METEOR	SAS
BERT Tiny SQuAD v2	12.87	0.85	13.3	3.0	9.39	22.16
BERT Base SQuAD v2	34.93	1.55	35.83	20.0	24.74	45.01
BERT Large SQuAD v1	37.31	3.36	38.01	25.0	26.31	44.88
BERT Large SQuAD v2	38.82	4.64	40.37	22.0	29.12	49.98
DistilBERT Base SQuAD v1	27.18	3.05	27.75	13.0	19.67	35.9
DistilBERT Base SQuAD v2	25.95	1.22	27.2	11.0	18.55	33.22
ALBERT Base v2 SQuAD v2	29.88	3.89	30.67	11.0	22.28	42.2
ALBERT XXLarge v1 SQuAD v2	48.95	7.35	50.18	29.0	36.44	56.33
ALBERT XXLarge v2 SQuAD v2	44.39	4.19	45.26	27.0	32.64	52.43
RoBERTa Tiny SQuAD v2	38.69	6.52	39.51	18.0	29.25	49.18
RoBERTa Base SQuAD v2	45.53	9.19	46.21	26.0	32.83	52.79
RoBERTa Base Distilled SQuAD v2	41.71	7.69	42.67	24.0	31.55	49.65
RoBERTa Large SQuAD v2	58.37	12.19	59.19	38.0	43.32	63.4
Longformer Base SQuAD v1	32.35	2.17	32.88	17.0	22.48	40.79
Longformer Base SQuAD v2	36.82	6.66	37.23	23.0	27.09	45.12
FLAN T5 Base	31.96	5.33	32.71	19.0	27.41	45.26
FLAN T5 Large	34.66	11.55	36.43	15.0	36.98	55.34
FLAN T5 XL	40.66	8.42	41.96	24.0	36.63	59.21
FLAN Alpaca Base	20.72	6.56	21.72	1.0	28.06	49.27
FLAN Alpaca GPT4 XL	23.61	7.0	25.47	1.0	36.49	58.76
FLAN Alpaca ShareGPT XL	14.74	4.59	15.54	0.0	23.17	47.31
GPT-2	4.36	0.17	4.99	0.0	8.83	67.23
GPT-3.5 Turbo	16.96	3.78	18.5	0.0	32.06	61.31
BART ChatGPT	6.68	0.71	7.94	0.0	13.47	55.36
BART ELI5	4.12	0.64	4.59	0.0	6.76	26.1
BART LFQA	2.94	0.31	3.09	0.0	6.19	61.07

Table 4.3: All model results from $w = 150$, $s = 50$, $k = 5$.

confirmed by the low lexical scores, showing the models do not comprehend the question.

Overall, our results show improvement in F1 score over the original study for all models evaluated, as can be seen in figure 4.4. All bi-directional models show stronger performance when pre-processing the documents using a sliding window of $w = 150$ and stride $s = 50$.

In comparing the current results to the original paper, a notable difference lies in the approach to retriever models. We exclusively employ the BM25 retriever for all experiments, while the original study utilized two different retriever models (Whoosh and AWS Kendra) [1]. Additionally, the model checkpoints used for comparison were different. The original study fine-tuned BERT-based models on various datasets, such as SQuAD [21] [22], and Natural Questions [5], whereas this study utilized publicly available model checkpoints from Huggingface [52]. This difference in fine-tuning methods may affect the performance and results between the two studies. Furthermore, the results might differ due to the evaluation procedure, as we have evaluated using a multi-label procedure.

While the the RoBERTa Base model performs on par with the ALBERT-XXL V1 and V2 models in the baseline, it outperforms these models after document splitting. There seems to be somewhat of a trend break for this model, as all the other results are ranked in the same manner as the original study. It is not entirely clear why there is such a significant difference between the studies for this model specifically.

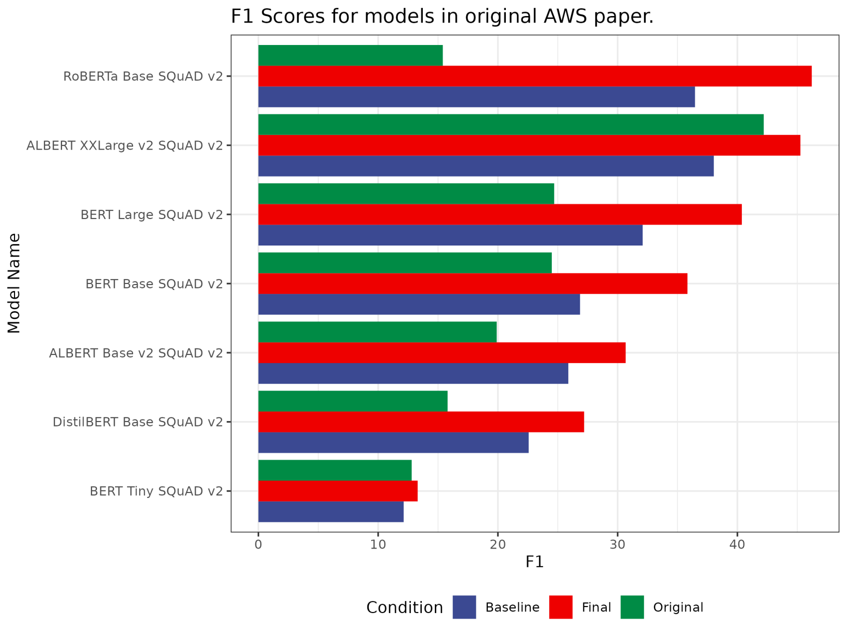

Figure 4.4: Comparison between the original study with both baseline and final F1.

The evaluation of METEOR and SAS sheds light on the performance of the generative models, as can be seen in figure 4.5. The FLAN-Alpaca GPT4 XL model shows strong performance, particularly in METEOR, outperforming the GPT-3.5 model across all metrics except SAS. This finding suggests that paid services are not necessarily superior to open-source models when comparing them through NLP metrics, also hinting at the limits of measuring a 'good' answer. The GPT models exhibit relatively poor performance on lexical metrics, but they demonstrate high SAS and METEOR scores. This contrast suggests that the usefulness of cross encoders for judging the similarity of text, particularly for answers that contain relatively long sequences of text as in LFQA. Traditional metrics, such as Rouge, may not capture the full extent of the model's performance in this context. Additionally, using the AWS dataset for a direct comparison between GPT models (GPT-3.5 Turbo and Alpaca GPT) and other models is not be fair due to potential data contamination, as discussed in the methodology section.

4.4 Samples

Question: **Does AMI supports tags?**
Gold Answers: *['AMI supports tags', 'Yes', 'Yes, AMI supports tags']*
RoBERTa Large: *Yes, Supports tagging on creation: No (F1: 0.4 Exact Match: 0.0, Rouge: 0.4, SAS: 0.596)*
FLAN Alpaca GPT XL: *Yes, AMI supports tags. However, it does not support tagging on creation like AFI and DHCP option. (F1: 0.38 Exact Match: 0.0, Rouge: 0.381, SAS: 0.618)*
GPT-3.5-Turbo: *AMI does support tags. These tags can be used to organize resources and track costs. Cost allocation tags can be activated to track AMI usage and AWS costs. Tags can also be created for EC2 instances or Systems Manager managed instances. However, tagging on creation is*

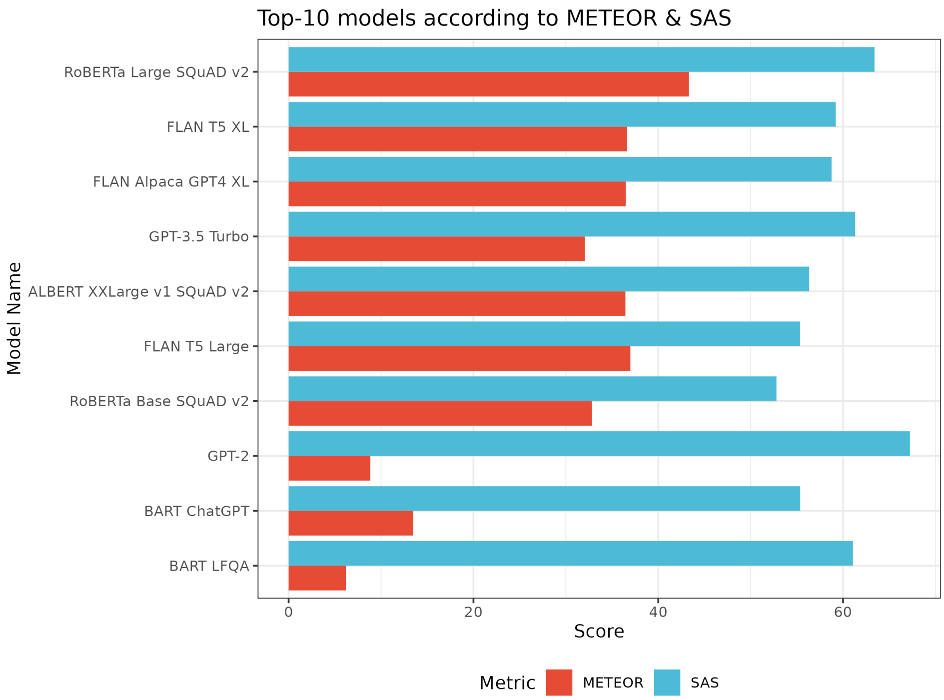

Figure 4.5: Top-10 performing final models according to METEOR and SAS

not supported for AMI. (F1: 0.08 Exact Match: 0.0, Rouge: 0.075, SAS: 0.542)

Question: **Does AMI supports tagging on creation?**
Gold Answers: *['AMI support tagging on creation', 'No', 'No, AMI support tagging on creation']*
RoBERTa Large: *No (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.948)*
FLAN Alpaca GPT XL: *No, AMI does not support tagging on creation. (F1: 0.86 Exact Match: 0.0, Rouge: 0.857, SAS: 0.88)*
GPT-3.5-Turbo: *AMI does not support tagging on creation. (F1: 0.83 Exact Match: 0.0, Rouge: 0.833, SAS: 0.966)*

Question: **What are the factors that affect storage performance in Amazon RDS?**
Gold Answers: *['system activities database workload DB instance class']*
RoBERTa Large: *system activities and database workload (F1: 0.67 Exact Match: 0.0, Rouge: 0.667, SAS: 0.814)*
FLAN Alpaca GPT XL: *Factors that affect storage performance in Amazon RDS include:*
1. Storage capacity: The size of the storage used by the instance.
2. I/O requests: The number of I/O requests that are waiting in the queue.
3. Queue depth: The average depth of the queue.
4. Latency: The time it takes for the application to process I/O requests.
5. Service time: The time it takes for the device to process I/O requests. (F1: 0.03 Exact Match: 0.0, Rouge: 0.03, SAS: 0.463)
GPT-3.5-Turbo: *Factors that affect storage performance in Amazon RDS include system activities such as multi-AZ standby creation, read replica creation, and changing storage types. Database*

workload also plays a role, with Provisioned IOPS storage recommended for applications that require fast and consistent I/O performance. Other factors like the number of nodes, node types, and provisioned IOPS can also impact query performance. It is important to consider these factors when optimizing storage performance in Amazon RDS. (F1: 0.1 Exact Match: 0.0, Rouge: 0.101, SAS: 0.719)

Question: **Why should I use Amazon Elastic Inference?** Gold Answers: *['speed up the throughput and decrease the latency at a fraction of the cost of using a GPU instance']* RoBERTa Large: *IAM permission policies (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.009)* FLAN Alpaca GPT XL: *Amazon Elastic Inference is a powerful tool that helps you automate the process of generating and analyzing code. It provides a fast and efficient way to analyze and improve code quality, reducing the time and effort required to develop and maintain software. It can be used to analyze and improve code quality, identify and fix issues, and automate the process of generating and analyzing code. Additionally, it can be used to automate the process of generating and analyzing code, reducing the time and effort required to develop and maintain software. (F1: 0.06 Exact Match: 0.0, Rouge: 0.062, SAS: 0.71)* GPT-3.5-Turbo: *Amazon Elastic Inference should be used for troubleshooting common problems encountered when working with Amazon CodeGuru Reviewer. It provides information on repository association status, code review status, third-party source provider connection status, and recommendations. It also addresses issues such as association and code review failures, disagreement with recommendations, and repository access during ownership changes. Additionally, it explains the permissions needed for various services, such as SageMaker, AWS AI Marketplace, CodeCommit, and CloudWatch. (F1: 0.02 Exact Match: 0.0, Rouge: 0.024, SAS: 0.167)*

More examples can be found in Appendix C.

5. Conclusion

5.1 Discussion

The goal of this thesis was to answer the question: *"How can we utilize Large Language Models in a Zero-Shot Open Domain Question Answering setting in an attempt to answer complex technical questions?"*

We can do so by deciding on a system architecture (e.g. retriever-reader) and using multi-label evaluation. We have showed that splitting the document in to passages significantly increases model performance on the AWS dataset with RoBERTa large reaching a new SOTA F1-score of 59.19 compared to the previous 42.20 [1]. Splitting documents into passages offers performance benefits, although it is important to note that document splitting may not necessarily be superior for all use cases and the optimal hyper parameter values are expected to vary depending on the specific application.

In this investigation, we explored the utility of Long Form Question Answering (LFQA) compared to traditional Machine Reading Comprehension (MRC) approaches. While bidirectional MRC models excel in extractive reasoning, they have limitations. LFQA is particularly valuable for generating natural language answers that may not have a single correct response. However, it is crucial to exercise caution when incorporating these tools into professional products. Fine-tuning and alignment are necessary to ensure responsible usage. Proper implementation of LFQA appears to alleviate some of the issues associated with standalone generative language models.

Generative models and LFQA demonstrate great potential, but when the complexity of the model far exceeds the evaluation metrics, the relevance and meaning of the metrics become questionable. Evaluating a Question Answering system using lexical metrics such as F1, EM, or ROUGE may not capture the notion of a "useful" or "meaningful" answer generated by the system. In this context, SAS and METEOR prove useful for analyzing diverse model outputs, as they are not dependent on lexical stride. These metrics attempt to address the ambiguous nature of text and seem to work better for LFQA, although they should not be relied upon exclusively.

In conclusion, this investigation highlights the possible benefits and risk of LFQA over traditional MRC approaches and emphasizes the need for careful implementation and evaluation using appropriate metrics that capture the desired qualities of the system's responses.

5.2 Future Research

Future research could focus on several areas to further enhance the technology. Firstly, improvements to the retriever component could be explored, aiming to enhance document retrieval accuracy and efficiency using neural retrievers. Additionally, considering changes in pre-processing techniques or leveraging larger and more powerful language models could potentially lead to better performance. Fine-tuning models specifically for LFQA tasks may also yield improved results. Furthermore, incorporating multi-modal approaches might expand the capabilities of a system, while multi-hop reasoning agents could enhance the system's ability to handle complex questions by 'thinking' iteratively. As for the the Haystack PromptNode, integrating a broader default set of NLP metrics within the framework could provide valuable insights into the effectiveness of prompt engineering techniques and guide the optimization of prompts for improved performance in LFQA systems. Future research in these directions can contribute to advancing the capabilities and accuracy of ODQA technology.

Appendices

A. Retriever Performance

Metric	@1	@3	@5	@7	@9	@10	@20	@30	@40	@50
R_s	0.59	0.75	0.82	0.86	0.88	0.89	0.93	0.94	0.95	0.95
R_m	0.59	0.74	0.81	0.85	0.87	0.87	0.91	0.92	0.93	0.93
P	0.59	0.45	0.36	0.31	0.28	0.27	0.21	0.18	0.16	0.15
MAP	0.59	0.65	0.65	0.64	0.62	0.62	0.56	0.53	0.49	0.47
MRR	0.59	0.66	0.68	0.68	0.69	0.69	0.69	0.69	0.69	0.69
$NDCG$	0.59	0.68	0.7	0.71	0.71	0.71	0.7	0.69	0.68	0.67

Table A.1: Information Retrieval Metrics over k, averaged for all window sizes w and stride s

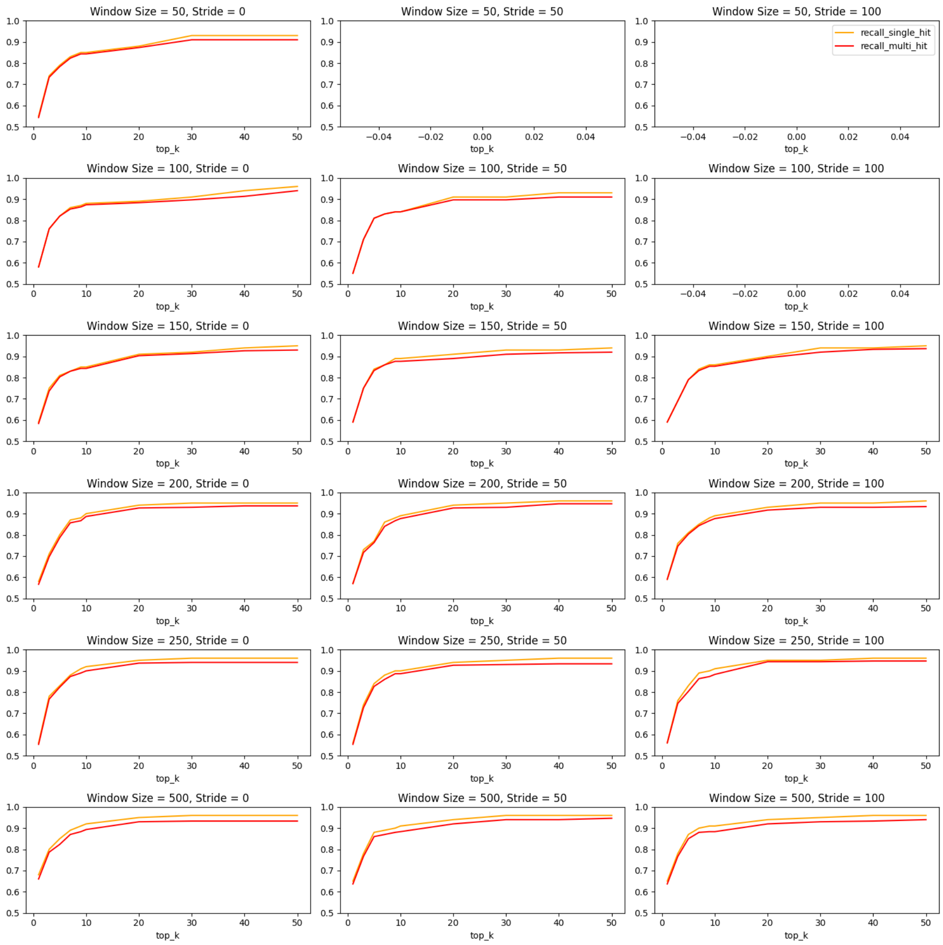

Figure A.1: Information Retrieval metrics over k for all valid combination of window size w and stride s

B. Model Checkpoints

Name	SQuAD Version	Huggingface URL
BERT Tiny	v2	https://huggingface.co/mrm8488/bert-tiny-finetuned-squadv2
BERT Base	v2	https://huggingface.co/deepset/bert-base-cased-squad2
BERT Large	v1	https://huggingface.co/bert-large-uncased-whole-word-masking-finetuned-squad
BERT Large	v2	https://huggingface.co/deepset/bert-large-uncased-whole-word-masking-squad2
DistilBERT Base	v1	https://huggingface.co/distilbert-base-uncased-distilled-squad
DistilBERT Base	v2	https://huggingface.co/twmkn9/distilbert-base-uncased-squad2
RoBERTa Tiny	v2	https://huggingface.co/deepset/tinyroberta-squad2
RoBERTa Base	v2	https://huggingface.co/deepset/roberta-base-squad2
RoBERTa Base Distilled	v2	https://huggingface.co/deepset/roberta-base-squad2-distilled
RoBERTa Large	v2	https://huggingface.co/deepset/roberta-large-squad2
ALBERT Base v2	v2	https://huggingface.co/twmkn9/albert-base-v2-squad2
ALBERT XXLarge v1	v2	https://huggingface.co/ahotrod/albert_xxlargev1_squad2_512
ALBERT XXLarge v2	v2	https://huggingface.co/mfeb/albert-xxlarge-v2-squad2
Longformer Base	v1	https://huggingface.co/valhalla/longformer-base-4096-finetuned-squadv1
Longformer Base	v2	https://huggingface.co/mrm8488/longformer-base-4096-finetuned-squadv2
FLAN T5 Base	-	https://huggingface.co/google/flan-t5-base
FLAN T5 Large	-	https://huggingface.co/google/flan-t5-large
FLAN T5 XL	-	https://huggingface.co/google/flan-t5-xl
FLAN Alpaca Base	-	https://huggingface.co/declare-lab/flan-alpaca-base
FLAN Alpaca GPT4 XL	-	https://huggingface.co/declare-lab/flan-alpaca-gpt4-xl
FLAN Alpaca ShareGPT XL	-	https://huggingface.co/declare-lab/flan-sharegpt-xl
BART LFQA	-	https://huggingface.co/vblagoje/bart_lfqa
BART ELI5	-	https://huggingface.co/yjernite/bart_eli5
BART ChatGPT	-	https://huggingface.co/Qiliang/bart-large-cnn-samsum-ChatGPT_v3
GPT-2	-	https://huggingface.co/gpt2
GPT-3.5 Turbo	-	https://platform.openai.com/docs/models/gpt-3-5

Table B.1: All models

C. Inference Samples

C.1 Extractive Samples

Question: **What is the instance store volumes for c5d.12xlarge?**
Gold Answers: *['2 x 900 GB']*
bert-tiny-squad-v2: 18xlarge (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.121)
bert-base-squad-v2: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
bert-large-squad-v1: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
bert-large-squad-v2: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
distilbert-base-squad-v1: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
distilbert-base-squad-v2: i3en.12xlarge (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.135)
roberta-tiny-squad-v2: c5.18xlarge | c5.24 (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.032)
roberta-base-squad-v2: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
roberta-base-squad-v2-distilled: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
roberta-large-squad-v2: 2 x 900 GB (1.8 TB), (F1: 0.8 Exact Match: 0.0, Rouge: 0.8, SAS: 0.791)
albert-base-v2-squad-v2: 2 x 900 GB (1.8 TB), (F1: 0.8 Exact Match: 0.0, Rouge: 0.8, SAS: 0.791)
albert-xxlarge-v1-squad-v2: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
albert-xxlarge-v2-squad-v2: 2 x 900 GB (1.8 TB), (F1: 0.8 Exact Match: 0.0, Rouge: 0.8, SAS: 0.791)
longformer-base-squad-v1: 2 x 900 GB (1.8 TB), (F1: 0.8 Exact Match: 0.0, Rouge: 0.8, SAS: 0.791)
longformer-base-squad-v2: 2 x 900 GB (F1: 1.0 Exact Match: 1.0, Rouge: 1.0, SAS: 0.971)
flan-t5-base: 2 x 900 GB (1.8 TB). (F1: 0.8 Exact Match: 0.0, Rouge: 0.8, SAS: 0.791)
flan-t5-large: c5d.12xlarge, Instance store volumes: 2 x 900 GB (1.8 TB), Type: NVMe SSD, Needs initialization*: nan, TRIM support**: . Instance type: c5d.18xlarge, Instance store volumes: 2 x 900 GB (1.8 TB), Type: NVMe SSD, Needs initialization*: nan, TRIM support**: . (F1: 0.19 Exact Match: 0.0, Rouge: 0.19, SAS: 0.873)
flan-t5-xl: c5d.12xlarge, Instance store volumes: 2 x 900 GB (1.8 TB), Type: NVMe SSD, Needs initialization*: nan, TRIM support**: . (F1: 0.36 Exact Match: 0.0, Rouge: 0.364, SAS: 0.606) flan-alpaca-base: The c5d.12xlarge has a total of 900 GB of storage. (F1: 0.33 Exact Match: 0.0, Rouge: 0.333, SAS: 0.63)
flan-alpaca-gpt4-xl: The instance store volumes for the c5d.12xlarge instance type are 2 x 900 GB (1.8 TB). (F1: 0.44 Exact Match: 0.0, Rouge: 0.444, SAS: 0.728)
flan-alpaca-sharegpt-xl: The instance store volumes for c5d.12xlarge are 2 x 900 GB (1.8 TB)

of NVMe SSD. (F1: 0.42 Exact Match: 0.0, Rouge: 0.421, SAS: 0.694)
bart-lfqa: I'm not sure if this is a good question, but I'm curious about the storage capacity of the SSD. Is it possible to store a lot of data on a single SSD? (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.291)
bart-eli5: I'm not sure what you mean by "sketch" but I think you're doing a good job. (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.01)
bart-chatgpt: The text describes a question and a brief response to it, including the answer to the question and the description of the instance store volumes for c5d.12xlarge. The text also mentions the instance storage volumes for other instances of the same device, such as c5ad.12Xlarge, c5D.2xlarge, and c5.4xlarge are instances of a single instance of an NVMe SSD. The answer should be in the form of a statement or a text with a few sentences, and should not be longer than 50 words. (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.461)
gpt-2: 12xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18xlarge | c5.18x (F1: 0.0 Exact Match: 0.0, Rouge: 0.0, SAS: 0.553) gpt-3.5-turbo-16k: The instance store volumes for the c5d.12xlarge instance type are 2 x 900 GB (1.8 TB) NVMe SSD volumes. They do not need initialization and support TRIM. (F1: 0.28 Exact Match: 0.0, Rouge: 0.276, SAS: 0.611)

C.2 Generative Samples

Question: **Can I use SparkML to serve my model with Amazon SageMaker?**
Gold Answers: *['the Amazon SageMaker open-source SparkML Serving container support deploying Apache Spark ML pipelines serialized with MLeap in SageMaker to get inferences', 'Yes', 'Yes, the Amazon SageMaker open-source SparkML Serving container support deploying Apache Spark ML pipelines serialized with MLeap in SageMaker to get inferences']*
BART LFQA: *I'm not sure if this is the right subreddit for this, but I'll try to answer it anyway. The first thing to know is that you can't use SparkML to serve your model with Amazon SageMaker. You can use SparkML to serve your model with Amazon SageMaker, but you can't use SparkML to serve your model with Amazon SageMaker. The second thing to know is that you can't use SparkML to serve your model with Amazon SageMaker. You can use SparkML to serve your model with Amazon SageMaker, but you can't use SparkML to serve your model with Amazon SageMaker.*

Question: **Can you delete an option group that is associated with a stopped DB instance?**
Gold Answers: *["You can't delete an option group that is associated with a stopped DB instance", 'No', "No, You can't delete an option group that is associated with a stopped DB instance"]*
FLAN-Alpaca GPT4 XL: *No, you cannot delete an option group that is associated with a stopped DB instance. You can, however, change the option group or DB parameter group that is associated with a stopped DB instance.*
GPT-3.5-Turbo: *No, you cannot delete an option group that is associated with a stopped DB instance. Additionally, you cannot modify the stopped DB instance or delete the DB parameter group*

associated with it.

Question: **How to download an entire amazon S3 bucket?**
Gold Answers: *['aws s3 sync']*
FLAN-Alpaca GPT4 XL: *To download an entire Amazon S3 bucket, you can use the following steps: 1. Open the Amazon S3 console. 2. Navigate to the bucket you want to download from. 3. Click on the "Download" button. 4. Select the "Enter the key for the bucket" option. 5. Enter the key for the bucket. 6. Click on the "Download" button. 7. The entire Amazon S3 bucket will be downloaded to your computer.*
GPT-3.5-Turbo: *To download an entire Amazon S3 bucket, you can use the aws s3 sync command to synchronize the bucket with a local directory. This command will download the data set and keep the local copy up-to-date with the remote set. If you have the necessary permissions, you can also use the AWS CLI to copy objects from the S3 bucket to your instance or vice versa. Additionally, if you are a developer, you can use the Amazon S3 API to access data in the bucket. It's important to note that data transfer fees may apply when downloading objects.*

Question: **What are the recommended and supported storage engines for MariaDB DB instances on Amazon RDS?**
Gold Answers: *['InnoDB (for version 10.2 and higher) and XtraDB (for version 10.0 and 10.1)']*
FLAN-Alpaca GPT4 XL: *The recommended and supported storage engines for MariaDB DB instances on Amazon RDS are InnoDB and XtraDB. These engines are optimized for recovery and data durability and are supported by Amazon RDS features such as Point-In-Time Restore and snapshot restore. Amazon RDS also supports Aria, although using Aria might have a negative impact on recovery in the event of an instance failure.*
GPT-3.5-Turbo: *The recommended and supported storage engines for MariaDB DB instances on Amazon RDS are InnoDB for version 10.2 and higher, and XtraDB for version 10.0 and 10.1. Aria is also supported, but it may negatively impact recovery in the event of an instance failure. InnoDB supports spatial indexes on MariaDB 10.2 and higher. Other storage engines are not currently supported by Amazon RDS for MariaDB.*

Question: **What is the autovacuum feature for PostgreSQL databases?**
Gold Answers: *['Autovacuum automates the execution of the VACUUM and ANALYZE command']*
FLAN-Alpaca GPT4 XL: *The autovacuum feature for PostgreSQL databases automates the execution of the VACUUM and ANALYZE command. It is a feature that is strongly recommended for maintaining the health of your PostgreSQL DB instance. It is required by PostgreSQL and its use is critical to good performance. The feature is enabled by default for all new Amazon RDS instances.*
GPT-3.5-Turbo: *The autovacuum feature for PostgreSQL databases automates the execution of the VACUUM and ANALYZE commands, which is required for maintaining the health and performance of the database. It is enabled by default for new Amazon RDS PostgreSQL DB instances. While it is not a "resource free" operation, it works in the background and prioritizes user operations. It also protects against data loss due to transaction ID wraparound. It is highly recommended*

to keep autovacuum enabled and to use adaptive autovacuum parameter tuning for optimal performance.

D. AWS Dataset

Question	Answer True	Yes / No / None	Document True
Is Amazon EBS encryption available on M3 instances?	Amazon EBS encryption is available on all current generation instance types and the following previous generation instance types: C3 cr1.8xlarge G2 I2 M3 and R3	Yes	amazon-ec2-user-guide/EBSEncryption.md
Does AMI supports tagging on creation?	AMI support tagging on creation	No	amazon-ec2-user-guide/Using$_{T}ags.md$
Does AMI supports tags?	AMI supports tags	Yes	amazon-ec2-user-guide/Using$_{T}ags.md$
Can I use my own keys in AWS KMS to encrypt data in Amazon Forecast?	Use one of your own AWS Key Management Service (AWS KMS) symmetric keys to encrypt Forecast data in your Amazon S3 buckets	Yes	amazon-forecast-developer-guide/data-protection.md
Is Alias an Amazon Forecast reserved field name?	Amazon Forecast reserves the following names Alias	Yes	amazon-forecast-developer-guide/reserved-field-names.md
Is Admin an Amazon Forecast reserved field name?	Amazon Forecast reserves the following names Admin	Yes	amazon-forecast-developer-guide/reserved-field-names.md
Is it recommended that I enable Offline mode in Microsoft SQL Server?	We recommend that you do not enable the following modes because they turn off transaction logging which is required for Multi-AZ: offline mode	No	amazon-rds-user-guide/CHAP$_{B}estPractices.md$
Does Amazon RDS Magnetic Storage support elastic volumes?	Magnetic storage Doesn't support elastic volumes	No	amazon-rds-user-guide/CHAP$_{S}torage.md$
Can I stop a DB instance that has a read replica?	You can't stop a DB instance that has a read replica	No	amazon-rds-user-guide/USER$_{S}topInstance.md$

Can you delete an option group that is associated with a stopped DB instance?	You can't delete an option group that is associated with a stopped DB instance	No	amazon-rds-user-guide/USER$_{S}topInstance.md$
What is the size of a null attribute in DynamoDB?	length of attribute name + 1 byte	None	amazon-dynamodb-developer-guide/CapacityUnitCalculations.m
How to download an entire amazon S3 bucket?	aws s3 sync	None	amazon-ec2-user-guide/AmazonS3.md
What is the instance store volumes for c5d.12xlarge?	2 x 900 GB	None	amazon-ec2-user-guide/InstanceStorage.md
What is the instance store type for d2.xlarge?	HDD	None	amazon-ec2-user-guide/InstanceStorage.md
What are the built-in algorithms in Amazon Forecast?	CNN-QR DeepAR+ Prophet NPTS ARIMA ETS	None	amazon-forecast-developer-guide/aws-forecast-choosing-recipes.md
Which TLS version is used in Amazon Forecast?	1.2	None	amazon-forecast-developer-guide/data-protection.md
What is the maximum number of rows in a dataset in Amazon Forecast?	1 billion	None	amazon-forecast-developer-guide/limits.md
What is Amazon Forecast's maximum number of columns in an item metadata dataset?	10	None	amazon-forecast-developer-guide/limits.md
What is maximum number of dataset groups in Amazon Forecast?	500	None	amazon-forecast-developer-guide/limits.md
What is maximum number of datasets in Amazon Forecast?	1500	None	amazon-forecast-developer-guide/limits.md
What is an Amazon Forecast reserved field name?	Names you can't use for your schema fields or dataset headers	None	amazon-forecast-developer-guide/reserved-field-names.md

what is the maximum number of saved filters per AWS account per Region in Amazon GuardDuty?	100	None	amazon-guardduty-user-guide/guardduty$_{l}imits.md$
what is the maximum number of Threat intel sets that you can add per AWS account per Region in Amazon GuardDuty?	6	None	amazon-guardduty-user-guide/guardduty$_{l}imits.md$
What is a source repository in Amazon Kendra?	A source repository contains the documents to index	None	amazon-kendra-developer-guide/how-it-works.md
What is the recommended and supported storage engine for MySQL DB instances on Amazon RDS?	InnoDB	None	amazon-rds-user-guide/CHAP$_{B}estPractices.md$
What are the recommended and supported storage engines for MariaDB DB instances on Amazon RDS?	InnoDB (for version 10.2 and higher) and XtraDB (for version 10.0 and 10.1)	None	amazon-rds-user-guide/CHAP$_{B}estPractices.md$
What is the autovacuum feature for PostgreSQL databases?	Autovacuum automates the execution of the VACUUM and ANALYZE command	None	amazon-rds-user-guide/CHAP$_{B}estPractices.md$
What are the results of not running autovacuum in PostgreSQL databases?	an eventual required outage to perform a much more intrusive vacuum operation.	None	amazon-rds-user-guide/CHAP$_{B}estPractices.md$
What can I do to shorten the failover time in SQL Server?	Make sure you have enough Provisioned IOPS allocated for your workload and Use smaller transactions.	None	amazon-rds-user-guide/CHAP$_{B}estPractices.md$
What is the upper limit of Amazon RDS DB instances?	40	None	amazon-rds-user-guide/CHAP$_{L}imits.md$
What is the maximum number of SQL Server instances I can have per account?	10	None	amazon-rds-user-guide/CHAP$_{L}imits.md$

What are the Amazon RDS storage types?	General Purpose SSD Provisioned IOPS Magnetic	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What is the latency of an Amazon RDS General Purpose SSD storage?	single-digit millisecond	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What is the size limit of PostgreSQL database general purpose SSD storage?	64 TiB	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What is the maximum storage for MySQL on db.m5.12xlarge?	64	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What is the maximum storage for MariaDB on db.t2.medium?	32	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What is the maximum storage for Oracle on db.t2.small?	16	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What are the factors that affect storage performance in Amazon RDS?	system activities database workload DB instance class	None	amazon-rds-user-guide/CHAP$_{S}torage.md$
What is the max number of defined per hyperparameter tuning job?	20	None	amazon-sagemaker-developer-guide/automatic-model-tuning.md
What is the number of concurrent hyperparameter tuning jobs limit in Amazon SageMaker?	100	None	amazon-sagemaker-developer-guide/automatic-model-tuning.md
What is the limit on number of training jobs per hyperparameter tuning job?	500	None	amazon-sagemaker-developer-guide/automatic-model-tuning.md
What are the Amazon SageMaker Autopilot problem types?	Regression Binary classification Multiclass classification	None	amazon-sagemaker-developer-guide/autopilot-automate-model-development-problem-types.md

Why should I use Amazon Elastic Inference?	speed up the throughput and decrease the latency at a fraction of the cost of using a GPU instance	None	amazon-sagemaker-developer-guide/ei.md
What is F32 Throughput in TFLOPS of ml.eia2.large?	2	None	amazon-sagemaker-developer-guide/ei.md
What is F16 Throughput in TFLOPS of ml.eia1.medium?	8	None	amazon-sagemaker-developer-guide/ei.md
How much memory does ml.eia1.xlarge have?	4 GB	None	amazon-sagemaker-developer-guide/ei.md
What is Amazon SageMaker Model Monitor?	Amazon SageMaker Model Monitor continuously monitors the quality of Amazon SageMaker machine learning models in production	None	amazon-sagemaker-developer-guide/model-monitor.md
Which TensorFlow version can I use with Amazon SageMaker Python SDK script mode?	1.11 and later	None	amazon-sagemaker-developer-guide/tf.md
What are the benefits of using Amazon SageMaker Debugger?	you can use the supported features and frameworks to inspect training job issues and use a visual interface analyze your tensor data	None	amazon-sagemaker-developer-guide/train-debugger.md
Where does AWS CloudTrail store the log files?	S3 bucket	None	aws-cloudtrail-user-guide/how-cloudtrail-works.md
How does AWS IoT Greengrass encrypt data in transit?	data is sent over an TLS connection using MQTT or HTTPS protocols	None	aws-greengrass-developer-guide/encryption-in-transit.md
What is the messaging protocol in AWS IoT Greengrass core?	MQTT	None	aws-greengrass-developer-guide/gg-core.md

What is the maximum number of channels I can have in AWS IoT?	50 per account	None	aws-greengrass-developer-guide/iot-analytics-connector.md
How can I list my current channels in aws iot?	aws iotanalytics list-channel	None	aws-iotanalytics-developer-guide/create-channel.md
What is the minimum SQL data set refresh interval in AWS IoT?	1 minute	None	aws-iotanalytics-developer-guide/limits.md
What is the maximum function timeout in AWS Lambda?	900 seconds	None	aws-lambda-developer-guide/gettingstarted-limits.md
What is the maximum unzipped deployment package size in AWS Lambda?	250 MB	None	aws-lambda-developer-guide/gettingstarted-limits.md
what is the maximum bandwidth per VPN tunnel in Transit gateway?	1.25 Gbps	None	aws-transit-gateway-guide/transit-gateway-quotas.md
what is the maximum number of peering attachments per transit gateway?	50	None	aws-transit-gateway-guide/transit-gateway-quotas.md
What are the two types of subnets for Application Load Balancers?	Availability Zone Local Zone	None	elb-application-load-balancers-user-guide/application-load-balancers.md
What are the three states of Application Load Balancers?	provisioning active failed	None	elb-application-load-balancers-user-guide/application-load-balancers.md
What is the maximum number of load balancers per region?	50	None	elb-application-load-balancers-user-guide/load-balancer-limits.md

What is the maximum number of security groups associated with a load balancer?	5	None	elb-application-load-balancers-user-guide/load-balancer-limits.md
What is the default HealthCheckTimeoutSeconds for lambda target type in load balancers?	30 seconds	None	elb-application-load-balancers-user-guide/target-group-health-checks.md
When you stop a DB instance does it retains its DNS endpoint?	When you stop a DB instance it retains its DNS endpoint	Yes	amazon-rds-user-guide/USER$_{S}topInstance.md$
Can I deploy multiple variants of a model to the same SageMaker HTTPS endpoint?	You can deploy multiple variants of a model to the same SageMaker HTTPS endpoint	Yes	amazon-sagemaker-developer-guide/how-it-works-deployment.md
Can I set an Amazon CloudWatch model monitor automated alert for deviations in the model quality?	Amazon CloudWatch model monitor enables you to set up an automated alert triggering system when there are deviations in the model quality	Yes	amazon-sagemaker-developer-guide/how-it-works-model-monitor.md
Is the library provided by Amazon SageMaker similar to using Apache Spark MLLib?	Using the library provided by SageMaker is similar to using Apache Spark MLLib	Yes	amazon-sagemaker-developer-guide/how-it-works-training.md
Can I use my own custom algorithm in an Amazon SageMaker training job?	Use your own custom algorithms	Yes	amazon-sagemaker-developer-guide/how-it-works-training.md
Can I connect a SageMaker Notebook Instance to a resource in a VPC?	Instances can be connected to Customer VPC	Yes	amazon-sagemaker-developer-guide/inter-network-privacy.md
Can I use Amazon EC2 Spot instances to train my machine learning models?	Amazon SageMaker makes it easy to train machine learning models using managed Amazon EC2 Spot instances	Yes	amazon-sagemaker-developer-guide/model-managed-spot-training.md

Are multi-model endpoints supported on GPU instance types in Amazon SageMaker?	Multi-model endpoints are not supported on GPU instance types	No	amazon-sagemaker-developer-guide/multi-model-endpoints.md
Can I access Amazon SageMaker Studio with AWS SSO?	you use your SSO credentials through a unique URL to directly access SageMaker Studio	Yes	amazon-sagemaker-developer-guide/notebooks-comparison.md
Is Amazon SageMaker available in us-east-2?	Region Tables us-east-2	Yes	amazon-sagemaker-developer-guide/regions-quotas.md
Does CloudTrail monitor calls to runtime$_{I}nvokeEndpoint$?	CloudTrail does not monitor calls to runtime$_{I}nvokeEndpoint$	No	amazon-sagemaker-developer-guide/sagemaker-incident-response.md
Can I use SparkML to serve my model with Amazon SageMaker?	the Amazon SageMaker open-source SparkML Serving container support deploying Apache Spark ML pipelines serialized with MLeap in SageMaker to get inferences	Yes	amazon-sagemaker-developer-guide/sparkml-serving.md
Can I associate a Git Repository with my SageMaker Notebook Instance?	You can associate one default repository and up to three additional repositories with a notebook instance	Yes	amazon-sagemaker-developer-guide/nbi-git-repo.md
Can I run codePipeline in a VPC?	AWS CodePipeline now supports Amazon Virtual Private Cloud (Amazon VPC) endpoints powered by AWS PrivateLink	Yes	aws-codepipeline-user-guide/vpc-support.md
Is AWS IoT Greengrass HIPAA compliant?	Third-party auditors assess the security and compliance of AWS IoT Greengrass as part of multiple AWS compliance programs. These include SOC PCI FedRAMP HIPAA and others	Yes	aws-greengrass-developer-guide/compliance-validation.md

Can I adjust concurrent data set content generation limit in AWS IoT?	Concurrent data set content generation adjustable? No	No	aws-iotanalytics-developer-guide/limits.md
Does AWS Lambda supports versioning?	You can use versions to manage the deployment of your functions	Yes	aws-lambda-developer-guide/configuration-versions.md
Can I run my AWS Lambda in a VPC?	You can configure a Lambda function to connect to private subnets in a virtual private cloud (VPC) in your AWS account	Yes	aws-lambda-developer-guide/configuration-vpc.md
Can I use AWS Lambda as a target for messages sent to Amazon Simple Notification Service notifications?	You can use a Lambda function to process Amazon Simple Notification Service notifications	Yes	aws-lambda-developer-guide/with-sns.md
Can I use AWS Lambda as a target group for Application Load Balancers in local zones?	Local Zones You cannot use a Lambda function as a target	No	elb-application-load-balancers-user-guide/application-load-balancers.md

Table D.1: The original 100 Questions with gold answers and corresponding documents

Bibliography

[1] S. Gholami and M. Noori, *Zero-shot open-book question answering*, 2021. arXiv: 2111.11520 [cs.CL].

[2] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 [cs.CL].

[3] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, *Gpts are gpts: An early look at the labor market impact potential of large language models*, 2023. arXiv: 2303.10130 [econ.GN].

[4] C. Dong, Y. Li, H. Gong, *et al.*, "A survey of natural language generation," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, Dec. 2022. DOI: 10.1145/3554727. [Online]. Available: https://doi.org/10.1145/3554727.

[5] T. Kwiatkowski, J. Palomaki, O. Redfield, *et al.*, "Natural questions: A benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019. DOI: 10.1162/tacl_a_00276. [Online]. Available: https://aclanthology.org/Q19-1026.

[6] C. Raffel, N. Shazeer, A. Roberts, *et al.*, *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2020. arXiv: 1910.10683 [cs.LG].

[7] H. W. Chung, L. Hou, S. Longpre, *et al.*, *Scaling instruction-finetuned language models*, 2022. arXiv: 2210.11416 [cs.LG].

[8] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].

[9] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[10] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, 2022, ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.04.024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809922006324.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[12] Y. Liu, M. Ott, N. Goyal, *et al.*, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL].

[13] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *Albert: A lite bert for self-supervised learning of language representations*, 2020. arXiv: 1909.11942 [cs.CL].

[14] M. Lewis, Y. Liu, N. Goyal, *et al.*, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, 2019. arXiv: 1910.13461 [cs.CL].

[15] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, *Xlnet: Generalized autoregressive pretraining for language understanding*, 2020. arXiv: 1906.08237 [cs.CL].

[16] R. Thoppilan, D. D. Freitas, J. Hall, *et al.*, *Lamda: Language models for dialog applications*, 2022. arXiv: 2201.08239 [cs.CL].

[17] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[19] L. Ouyang, J. Wu, X. Jiang, *et al.*, *Training language models to follow instructions with human feedback*, 2022. arXiv: 2203.02155 [cs.CL].

[20] D. Su, *Generative long-form question answering: Relevance, faithfulness and succinctness*, 2022. arXiv: 2211.08386 [cs.CL].

[21] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, *Squad: 100,000+ questions for machine comprehension of text*, 2016. arXiv: 1606.05250 [cs.CL].

[22] P. Rajpurkar, R. Jia, and P. Liang, *Know what you don't know: Unanswerable questions for squad*, 2018. arXiv: 1806.03822 [cs.CL].

[23] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, *Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension*, 2017. arXiv: 1705.03551 [cs.CL].

[24] A. Chowdhery, S. Narang, J. Devlin, *et al.*, *Palm: Scaling language modeling with pathways*, 2022. arXiv: 2204.02311 [cs.CL].

[25] J. Wei, Y. Tay, R. Bommasani, *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022, Survey Certification, ISSN: 2835-8856. [Online]. Available: https://openreview.net/forum?id=yzkSU5zdwD.

[26] S. Bubeck, V. Chandrasekaran, R. Eldan, *et al.*, *Sparks of artificial general intelligence: Early experiments with gpt-4*, 2023. arXiv: 2303.12712 [cs.CL].

[27] I. Magar and R. Schwartz, *Data contamination: From memorization to exploitation*, 2022. arXiv: 2203.08242 [cs.CL].

[28] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, no. 3, pp. 291–302, 2020, Overview paper. DOI: 10.31449/inf.v44i3.2828.

[29] B. Min, H. Ross, E. Sulem, *et al.*, *Recent advances in natural language processing via large pre-trained language models: A survey*, 2021. arXiv: 2111.01243 [cs.CL].

[30] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[31] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975, ISSN: 0001-0782. DOI: 10.1145/361219.361220. [Online]. Available: https://doi.org/10.1145/361219.361220.

[32] S. P and A. Shaji, "A survey on semantic similarity," Dec. 2019, pp. 1–8. DOI: 10.1109/ICAC347590.2019.9036843.

[33] D. Hiemstra, "A probabilistic justification for using tf×idf term weighting in information retrieval," *International Journal on Digital Libraries*, vol. 3, pp. 131–139, 2000. DOI: 10.1007/s007999900025.

[34] S. E. Robertson and K. Sparck Jones, "Relevance weighting of search terms," *Journal of the American Society for Information Science*, vol. 27, no. 3, pp. 129–146, Jun. 1976. DOI: 10.1002/asi.4630270302.

[35] N. Reimers and I. Gurevych, *Sentence-bert: Sentence embeddings using siamese bert-networks*, 2019. arXiv: 1908.10084 [cs.CL].

[36] V. Karpukhin, B. Oğuz, S. Min, *et al.*, *Dense passage retrieval for open-domain question answering*, 2020. arXiv: 2004.04906 [cs.CL].

[37] C. Zeng, S. Li, Q. Li, J. Hu, and J. Hu, *A survey on machine reading comprehension: Tasks, evaluation metrics and benchmark datasets*, 2020. arXiv: 2006.11880 [cs.CL].

[38] K. Li, X. Xian, J. Wang, and N. Yu, "First-principle study on honeycomb fluorated-InTe monolayer with large rashba spin splitting and direct bandgap," *Applied Surface Science*, vol. 471, pp. 18–22, Mar. 2019. DOI: 10.1016/j.apsusc.2018.11.214. [Online]. Available: https://doi.org/10.1016%5C%2Fj.apsusc.2018.11.214.

[39] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, and T.-S. Chua, *Retrieving and reading: A comprehensive survey on open-domain question answering*, 2021. arXiv: 2101.00774 [cs.AI].

[40] Z. Ji, N. Lee, R. Frieske, *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, Mar. 2023. DOI: 10.1145/3571730. [Online]. Available: https://doi.org/10.114/3571730.

[41] B. Kostić, J. Risch, and T. Möller, *Multi-modal retrieval of tables and texts using tri-encoder models*, 2021. arXiv: 2108.04049 [cs.CL].

[42] P. Banerjee, K. K. Pal, A. Mitra, and C. Baral, *Careful selection of knowledge to solve open book question answering*, 2019. arXiv: 1907.10738 [cs.CL].

[43] A. Roberts, C. Raffel, and N. Shazeer, *How much knowledge can you pack into the parameters of a language model?* 2020. arXiv: 2002.08910 [cs.CL].

[44] Q. Zhang, S. Chen, D. Xu, *et al.*, *A survey for efficient open domain question answering*, 2022. arXiv: 2211.07886 [cs.CL].

[45] D. Chen, A. Fisch, J. Weston, and A. Bordes, *Reading wikipedia to answer open-domain questions*, 2017. arXiv: 1704.00051 [cs.CL].

[46] L. Weng, "How to build an open-domain question answering system?" *lilianweng.github.io*, Oct. 2020. [Online]. Available: https://lilianweng.github.io/posts/2020-10-29-odqa/.

[47] F. Petroni, T. Rocktäschel, P. Lewis, *et al.*, *Language models as knowledge bases?* 2019. arXiv: 1909.01066 [cs.CL].

[48] P. Lewis, E. Perez, A. Piktus, *et al.*, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021. arXiv: 2005.11401 [cs.CL].

[49] D. Su, X. Li, J. Zhang, *et al.*, *Read before generate! faithful long form question answering with machine reading*, 2022. arXiv: 2203.00343 [cs.CL].

[50] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, *Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface*, 2023. arXiv: 2303.17580 [cs.CL].

[51] S. Zhang, C. Gong, L. Wu, X. Liu, and M. Zhou, *Automl-gpt: Automatic machine learning with gpt*, 2023. arXiv: 2305.02499 [cs.CL].

[52] T. Wolf, L. Debut, V. Sanh, *et al.*, *Huggingface's transformers: State-of-the-art natural language processing*, 2020. arXiv: 1910.03771 [cs.CL].

[53] Q. Lhoest, A. Villanova del Moral, Y. Jernite, *et al.*, "Datasets: A community library for natural language processing," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 175–184. arXiv: 2109.02846 [cs.CL]. [Online]. Available: https://aclanthology.org/2021.emnlp-demo.21.

[54] Z. Jia, B. Tillman, M. Maggioni, and D. P. Scarpazza, *Dissecting the graphcore ipu architecture via microbenchmarking*, 2019. arXiv: 1912.03413 [cs.DC].

[55] Deepset AI. "Haystack." (2021), [Online]. Available: https://github.com/deepset-ai/haystack.

[56] J. Carlsmith, *Is power-seeking ai an existential risk?* 2022. arXiv: 2206.13353 [cs.CY].

[57] E. Yudkowsky, "Artificial Intelligence as a positive and negative factor in global risk," in *Global Catastrophic Risks*, Oxford University Press, Jul. 2008, ISBN: 9780198570509. DOI: 10.1093/oso/9780198570509.003.0021. eprint: https://academic.oup.com/book/0/chapter/348239228/chapter-pdf/43261407/isbn-9780198570509-book-part-21.pdf. [Online]. Available: https://doi.org/10.1093/oso/9780198570509.003.0021.

[58] R. Institute, "Policymaking in the pause," *Policy Commons*, 2023, CID: 20.500.12592/cm9pws. [Online]. Available: https://policycommons.net/artifacts/3846414/policymaking-in-the-pause/4652369/ (visited on 06/28/2023).

[59] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: The long-document transformer*, 2020. arXiv: 2004.05150 [cs.CL].

[60] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. [Online]. Available: https://aclanthology.org/P02-1040.

[61] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013.

[62] J. Risch, T. Möller, J. Gutsch, and M. Pietsch, *Semantic answer similarity for evaluating question answering models*, 2021. arXiv: 2108.06130 [cs.CL].

[63] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*, 2020. arXiv: 1910.01108 [cs.CL].

[64] Y. K. Chia, P. Hong, L. Bing, and S. Poria, *Instructeval: Towards holistic evaluation of instruction-tuned large language models*, 2023. arXiv: 2306.04757 [cs.CL].