# MoQA☕: Benchmarking Multi-Type Open-Domain Question Answering

**Anonymous ACL submission**


## Abstract

Previous research on open-domain question answering (QA) mainly focuses on questions with short answers. However, information-seeking QA often requires various formats of answers depending on the nature of the questions, e.g., why/how questions typically require a long answer. In this paper, we present MoQA[1], a benchmark for open-domain QA that requires *building one system* that can provide short, medium, long and yes/no answers to different questions accordingly. MoQA builds upon Natural Questions (Kwiatkowski et al., 2019) with multiple types of questions and additional crowdsourcing efforts to ensure high query quality. We adapt state-of-the-art models, and reveal unique findings in multi-type open-domain QA: (1) For retriever-reader models, training one retriever on all types achieves the overall best performance, but it is challenging to train one reader model to output answers of different formats, or to train a question classifier to distinguish between types; (2) An end-to-end closed-book QA model trained on multiple types struggles with the task across the board; (3) State-of-the-art large language models such as GPT-3 and InstructGPT also lag behind open-book QA models. Our benchmark and analysis call for more effort into building versatile open-domain QA models in the future.


## 1 Introduction

Open-domain question answering (QA) leverages a large knowledge source such as Wikipedia to answer open-domain questions (Voorhees and Tice, 2000; Chen et al., 2017). Such a task mimics humans' information-seeking process—finding relevant documents and composing answers based on them—and has potential to become a fundamental component of next generation chatbots and search engines. An ideal open-domain QA system should be able to answer all types of questions human may ask, and provide answers in proper formats depending on the nature of questions.

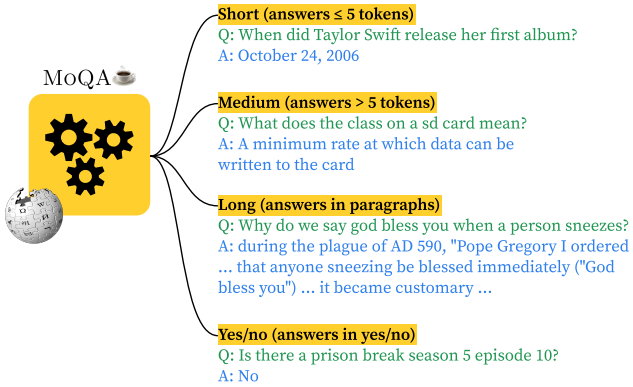



Figure 1: Examples of question-answer pairs. MoQA is designed to build one single open-domain QA system to answer various types of questions.

However, for the ease of evaluation, most existing research on open-domain QA focuses on questions with short answers (Karpukhin et al., 2020; Lee et al., 2021; Izacard and Grave, 2020), originating from earlier work by Lee et al. (2019)[2]. This trend leaves a large portion of the open-domain QA task unattended: for example, *Why* questions often require sentence-level answers as you cannot explain "*why is the sky blue*" in a few words; yes-no questions like "*is there a Prison Break 5 episode 10?*" are also prevalent but largely neglected.

There has been recent efforts to study other forms of answers in open-domain QA (Fan et al., 2019; Stelmakh et al., 2022). In particular, GooAQ (Khashabi et al., 2021) is a benchmark consisting of different types of questions mined from Google autocomplete system and Google's answers boxes without human annotation. Nonetheless, all of these works consider each type of questions *separately*. The closest work to ours is KILT (Petroni et al., 2021), which also considers different types

[1] MoQA = Multi-type Open-domain Question Answering. It is pronounced as *mocha*.

[2] Lee et al. (2019) only considered answers $\leq 5$ tokens in Natural Questions and exact match as the metric. This setting has been adopted by numerous follow-up work.

of open-domain questions and builds one system to answer them. However, KILT is a collection of different tasks such as QA, fact checking, entity linking, etc., each with its own unique collection pipeline, which can introduce superficial cues that help the model determine the format of the answer.

Building one system for different types of questions has been studied in the reading comprehension setting (Kwiatkowski et al., 2019; Khashabi et al., 2020), where the supporting evidence is given as input. On the other hand, open-domain QA requires retrieving the evidence from a large text corpus or recalling specific facts from a model's parametric knowledge, both of which poses significantly different challenges.

Different from previous work, we bring different types of questions drawn from the same distribution together, and aim to build a single open-domain system (either open-book or closed-book QA) that can handle various types of questions with fine-grained annotations, based on a single knowledge source i.e., English Wikipedia. We further discuss the differences with past works in Section 6.

In this paper, we propose MoQA, a benchmark that requires an open-domain QA system to answer multiple types of questions (Figure 1). We build our benchmark by extending Natural Questions (NQ) (Kwiatkowski et al., 2019) into the open-domain setup, aligning each answer to a fixed-length Wikipedia passage, while keeping questions of four types: short, medium, long, and yes/no. We also use additional human annotations on the test set to filter out ill-defined long-answer questions, ensuring the high quality of MoQA evaluation.

We conduct a comprehensive set of experiments on MoQA with state-of-the-art QA models (Karpukhin et al., 2020; Izacard and Grave, 2021; Roberts et al., 2020; Brown et al., 2020; Ouyang et al., 2022), and reveal multiple interesting findings:

- Retrieving text passages for different types of questions exhibits a similar difficulty, and simply training a dense retriever (Karpukhin et al., 2020) on all types achieves overall best performance.

- However, building a single reader remains challenging. Joint training of a reader model on all types performs poorly, and it is also difficult to train a classifier to identify question types before applying individual reader models of each type.

- Finally, closed-book QA models trained on all questions as well as few-shot large language models (without explicit retrieval) significantly lag behind their open-book counterparts.

Our findings suggest building such a versatile open-domain system still has a series of unsolved challenges and call for more effort in the direction.

| Type | Train | Dev | Test | Description (avg. ans. length) |
|---|---|---|---|---|
| Short | 79,168 | 8,757 | 3,610 | $\leq 5$ tokens (3.3) |
| Medium | 16,668 | 1,853 | 565 | $> 5$ tokens (15.5) |
| Long | 19,649 | 2,169 | 201 | Long answers (102.9) |
| Yes/No | 3,154 | 351 | 99 | YES/NO (1.0) |
| Total | 118,639 | 13,130 | 4,475 | - |

Table 1: Dataset statistics of MoQA. Avg. ans. length is the average number of words in the answers.

## 2 The MoQA Benchmark

In open-domain QA, models are given a set of $K$ passages $\mathcal{D} = \{d_1, \ldots, d_K\}$ for a question $q$. In MoQA, the correct answer $a$ to the question $q$ could be a span ranging from a couple words to a paragraph, , YES, or NO. Unlike existing open-domain benchmarks that assume an answer span to be short (usually $\leq 5$ tokens), MoQA does not assume a length limit and requires models to output different formats of answers considering the nature of each question.

### 2.1 Dataset Collection

We first build our dataset by adapting the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019) into the open-domain setting while keeping all types of questions presented below[3]. We use the 2018-12-20 Wikipedia snapshot as the knowledge source and slice it to 100-word passages following Karpukhin et al. (2020). We then align each question-answer pair to a supporting passage and slightly modify the gold answers so that they could be found in the passage. Table 1 shows statistics of our dataset, and about 33% of the dataset is not short-answer questions. For more details of the dataset, and preprocessing steps, see §A and §C.

**Short.** The short-answer questions are defined as those having at least one short answer up to 5 tokens. It is exactly the same as Lee et al. (2019), followed by most open-domain QA research. We keep our short-type questions consistent with the train/dev/test sets from Min et al. (2019).

[3]We do not include unanswerable questions in the original NQ dataset, since many of them become answerable in the open-domain setting.

**Medium.** The medium-answer questions can still be answered by span of text but contain more than 5 tokens. Most of them are noun or verb phrases or text around the length of one to two sentences, other than proper nouns, dates or numbers. For example, the question "what does the class on a sd card mean?" can be answered by "a minimum rate at which data can be written to the card", which cannot be reduced to just a couple words.

**Long.** The long-answer questions have paragraph-level answers and cannot be answered by a short string. We focus on long-answers that originates from paragraph as these questions tend to be inherently complex and therefore require long natural language explanations, and filter out those originating from tables and lists.

**Yes/No.** The yes/no-type questions can only be answered by either `YES` or `NO`. The original NQ annotations consider them as short-type questions, but we separate them since yes/no-type questions often require a different reasoning ability (Clark et al., 2019) and model design.

### 2.2 Mechanical Turk for Data Filtering

From preliminary qualitative analysis, we found that the original NQ long-answer annotations are noisy (many questions are marked as 'long answer' even though they can be answered by a few words), we carry out a manual filtering for the long-answer questions in the test set to ensure that these questions truly require long explanations to answer. To this end, we use Mechanical Turk (MTurk) for further data filtering, where each question is annotated by at least five MTurk workers. Each worker is shown both the question and the original long answers, and decides if the question can only be answered by the long answer, and if not, what is the shortest answer. Then, we filter out any question that is marked as short by at least one worker.

### 2.3 Evaluation

We use the original NQ development set as the test set and split the original NQ training sets into training and development sets with a $9:1$ ratio.

For retrieval, we report A@$k$ (accuracy at top-$k$ passages) and MRR@$k$ (mean reciprocal rank). We consider a passage to be correct if it contains the long answer of the question. For answer prediction, we report exact match (EM) and F1 for short and medium answers, EM and ROUGE-L for long answer, and accuracy for yes/no answers. We also report the macro-averaged EM as the overall result.

Furthermore, we report the BERTScore (Zhang et al., 2019) for short, medium, and long-type answers in Table 8, and found that these metrics correlates with the F1 score and the ROUGE-L score. We also conduct human evaluation for selected models to study the difficulty of our dataset in Section 5. For more details about the metric, please refer to Appendix D.

## 3 Adapting QA models for MoQA

We establish multiple baselines for MoQA, by adapting state-of-the-art open-domain QA systems, including both supervised open-book QA models (Chen et al., 2017; Lee et al., 2019), supervised closed-book QA models (Roberts et al., 2020), and large language models (Brown et al., 2020; Chen et al., 2021; Ouyang et al., 2022).

### 3.1 Open-book QA Models

Open-book QA models first retrieve supporting passages from a large corpus such as Wikipedia, and then apply a more expensive reader to predict answers using the passages. We carefully consider both components in MoQA by training both the retriever and the reader on different portions of the dataset and examine their trade-offs.

**Retrievers.** Retrievers return the most relevant passages from a large text corpus for a given query. We investigate the trade-offs of training one retriever on all questions together versus training a separate retriever for each question types.

We use DPR (Karpukhin et al., 2020) for the retriever component. (1) For the jointly trained model, we include all questions, and use all annotated gold passages as positive passages and passages retrieved by BM25 that are not the gold passages as hard negatives. The resulting model is $\text{DPR}_{\text{all}}$. (2) We also add a reference baseline, $\text{DPR}^{\dagger}$, which assumes we know the question type, and we train a DPR model on only the corresponding type of questions. See §E for more details and baselines.

**Readers.** Readers leverages attention across both the query and the retrieved passages to predict an answer. Similar to retriever, we are interested in the trade-offs between training a single reader model for all question types versus training separate readers for each question type.

Furthermore, we use two common types of readers: (1) An extractive reader based on RoBERTa (Liu et al., 2019), which predicts the

rank of a passage and the start and the end positions of the answer span given the concatenation of the question and the retrieved passage[4]. (2) the generative reader model Fusion-in-Decoder (FiD; Izacard and Grave, 2021) based on T5-base(Raffel et al., 2019), which achieves the state-of-the-art performance on short-form answers since it is able to attend to multiple passages during decoding. We then analyze the performance of readers with different architectures on different types of questions.

For each reader model, we consider training them on all questions combined, as well as training them on each individual question type (there will be four readers in this case). For the latter, we need to train an extra **question classifier** in the pipeline to decide which individual reader to be applied. The question classifier is implemented by feeding the question into a RoBERTa model (Liu et al., 2019) and training a classifier on top of the `[CLS]` token (see §G for details).

### 3.2 Closed-book QA Models

Unlike open-book QA models, closed-book QA models entirely rely on their parameters to generate the answers. We consider both fine-tuning approach and few-shot in-context learning approach. Furthermore, we examine how the more powerful LLMs behave under the settings where the question type is either given or not given.

We follow Roberts et al. (2020) and fine-tune a T5-large model (Raffel et al., 2019) on all the questions in the training set, since the fine-tuned T5 model achieve impressive results on short-form QA and closed the gap to open-book models. See Appendix H for training details.

We evaluate three large language models (LLMs): GPT-3 (`davinci`; Brown et al., 2020), Codex (`code-davinci-002`; Chen et al., 2021), and InstructGPT (`text-davinci-003`; Ouyang et al., 2022) due to their abilities to recall factual knowledge and performance on open-domain QA benchmarks. For each LLM, we use in-context learning with eight-shot ICL consisting of two demonstrations from each question type. We randomly sample one question from each type to construct sets of four demonstration, and balance the answer labels for yes/no-type questions. We also consider an oracle setting where the question type is given, and we only include two demonstrations from that question type for comparison.

Additionally, we use GENREAD (Yu et al., 2023) and prompt InstructGPT to first generate a supporting passage before outputting the final answer. Previous works showed that generating a supporting evidence improves the LLM's performance on short-form QA, and we investigate the method's robustness to different question types. For details on the prompting and examples, see Appendix H.2.

[4]To support yes/no answers, we prepend every passage with two special tokens: `[YES]` and `[NO]`.

| Retriever | #Train | A@1 | A@5 | A@20 | MRR@20 |
|---|---|---|---|---|---|
| **Short** | | | | | |
| $\text{DPR}_{\text{all}}$ | 89k | **36.0** | **62.8** | **77.0** | **47.6** |
| $\text{DPR}^{\dagger}_{\text{short}}$ | 56k | 33.6 | 59.7 | 75.1 | 45.0 |
| **Medium** | | | | | |
| $\text{DPR}_{\text{all}}$ | 89k | 41.4 | 69.5 | 83.3 | 53.7 |
| $\text{DPR}^{\dagger}_{\text{medium}}$ | 11k | **45.8** | **71.9** | **86.7** | **57.9** |
| **Long** | | | | | |
| $\text{DPR}_{\text{all}}$ | 89k | **37.8** | 64.7 | **80.6** | **49.2** |
| $\text{DPR}^{\dagger}_{\text{long}}$ | 20k | 34.8 | **65.2** | 78.1 | 47.4 |
| **Yes/No** | | | | | |
| $\text{DPR}_{\text{all}}$ | 89k | **37.0** | **64.4** | **79.5** | **49.7** |
| $\text{DPR}^{\dagger}_{\text{yes/no}}$ | 2k | 35.6 | 57.5 | 72.6 | 45.9 |

Table 2: Passage retrieval results on the MoQA test set (A@$k$: top-$k$ retrieval accuracy; MRR@$k$: mean reciprocal rank at $k$).

## 4 Results

**Retriever: different types of questions have similar difficulty.** We first demonstrate the retrieval results in Table 2. We can see that overall using all question type training data outperforms using the corresponding training data for each type, and the performance across different types does not differ much. We assume this comes from the nature of retrieval – because retrieval is mostly about topic, answer types do not matter much and training data can generalize to all types of questions.

**Reader: challenging to train one versatile reader that handles all questions.** Table 3 shows the main QA results. Different from the retrieval component, we see that a single reader trained on all types of questions shows very imbalanced performance on different subsets. The single reader also significantly lags behind the oracle (marked by $^{\dagger}$) that assumes knowing the question type and applies corresponding readers. Between the extractive RoBERTa and the generative FiD,

| Retriever | Reader | Short | | Medium | | Long | | | Yes/No | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | ROUGE-L | BScore | Acc. | |
| **Oracle Question Types** | | | | | | | | | | |
| $\text{DPR}_{\text{all}}$ | RoBERTa$^{\dagger}_{*}$ | 45.1 | 53.3 | 25.1 | 47.3 | 23.9 | 49.8 | 71.7 | 59.6 | 38.4 |
| | FiD$^{\dagger}_{*}$ | 47.6 | 55.7 | 25.3 | 50.7 | 18.4 | 48.3 | 71.0 | 63.6 | 38.7 |
| None | GPT-3$^{\dagger}_{*}$ | 20.6 | 30.1 | 0.8 | 21.1 | 0.0 | 20.2 | 51.9 | 54.2 | 18.9 |
| None | Codex$^{\dagger}_{*}$ | 35.3 | 46.2 | 1.9 | 27.2 | 0.2 | 23.3 | 54.5 | 67.7 | 26.3 |
| None | InstructGPT$^{\dagger}_{*}$ | 20.8 | 35.9 | 0.5 | 26.3 | 0.0 | 24.2 | 59.1 | 60.3 | 20.4 |
| GENREAD | InstructGPT$^{\dagger}_{*}$ | 31.6 | 46.9 | 2.8 | 29.7 | 0.0 | 23.3 | 59.0 | 67.3 | 25.4 |
| **Open-book QA Models** | | | | | | | | | | |
| $\text{DPR}_{\text{all}}$ | $\text{RoBERTa}_{\text{all}}$ | 43.8 | 52.1 | **18.6** | **40.5** | **9.0** | **31.7** | **59.1** | 54.6 | 31.5 |
| | Cls. + $\text{RoBERTa}_{*}$ $\times$ 4 | 43.4 | 51.7 | 8.5 | 29.4 | 6.0 | 21.3 | 52.4 | 57.6 | 28.9 |
| | $\text{FiD}_{\text{all}}$ | **46.5** | **54.8** | 15.0 | 36.0 | 0.0 | 9.5 | 44.3 | **69.7** | **32.8** |
| | Cls. + $\text{FiD}_{*}$ $\times$ 4 | 46.1 | 54.5 | 6.9 | 29.3 | 4.5 | 21.8 | 52.2 | 61.6 | 29.8 |
| **Closed-book QA Models** | | | | | | | | | | |
| None | $\text{T5-large}_{\text{all}}$ | 16.4 | 23.0 | 6.0 | 19.2 | 1.5 | 12.7 | 46.5 | 59.6 | 20.9 |
| None | $\text{GPT-3}_{\text{all}}$ | 17.7 | 27.1 | 0.5 | 17.9 | 0.0 | 16.3 | 51.4 | 29.0 | 11.8 |
| None | $\text{Codex}_{\text{all}}$ | 28.7 | 38.4 | 1.5 | 22.1 | 0.2 | 21.0 | 54.6 | 51.5 | 20.5 |
| None | $\text{InstructGPT}_{\text{all}}$ | 10.9 | 25.7 | 0.3 | 23.6 | 0.0 | 20.5 | 57.5 | 14.8 | 6.5 |
| GENREAD | $\text{InstructGPT}_{\text{all}}$ | 28.5 | 43.8 | 4.1 | 31.1 | 0.0 | 14.8 | 51.6 | 67.0 | 24.9 |

Table 3: QA results on the MoQA test set. BScore: BERTScore(Zhang et al., 2019). $^{\dagger}$: An oracle that assumes that the question type is known and uses a reader trained only on each individual type. Cls.: a question classifier is used, and there are four readers trained independently, either for RoBERTa or FiD. Otherwise, all questions are jointly trained. $\text{T5-large}_{\text{all}}$: a closed-book QA model trained on all questions. $\text{GPT-3}_{\text{all}}$(`davinci`), $\text{Codex}_{\text{all}}$ (`code-davinci-002`), $\text{InstructGPT}_{\text{all}}$ (`text-davinci-003`) are prompted with two demonstrations from each question type, and averaged across 3 randomly seeded runs. Avg. takes the macro-averaged EM of the four types.

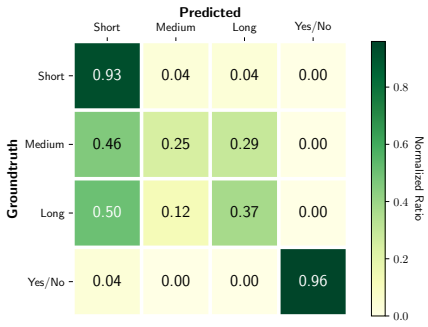



Figure 2: Normalized confusion matrix for question type predictions. For each ground truth type (row), we show the normalized prediction ratios over all types.

we find that FiD performs better on short-answer and yes/no questions but RoBERTa is better on medium and long-answer questions, even though in previous work FiD always prevails.

Though the oracle model is much better than the single reader, we see that training question classifier and 4 readers underperforms training jointly. This comes down to the difficulty to train the question classifier – Figure 2 shows the confusion matrix for question type prediction. It is noticeable that medium and long-answer questions are the most difficult to predict. For example, it is not trivial to understand that the question "who is covered under payment of gratuity act 1972" requires a long answer. Improving the classifier will be a straightforward to advance MoQA performance, as the current systems still have a huge gap to the corresponding type oracles. Please refer to §J for qualitative examples.

**Closed-book QA models significantly lag behind.** Though competitive on short-answer questions as shown in previous work, closed-book QA models perform poorly on other types, especially on long-answer questions. Both the trained T5-large model as well as the LLMs lag behind open-book QA models. Our result suggests that it is still challenging to solve MoQA relying purely on parameters.

One possible explanation is the fact that more complex questions and extensive explanations require more memorization than short-form answers from the closed-book QA models by definition, so it is harder to rely on knowledge stored in the model's parameters for this task. Furthermore,

these explanations are less likely to appear in the pre-training corpus than popular entities such as people's names, famous places, and significant dates. Even for models such as GPT-3 that has been trained on an extremely large corpus, thse answers still appear to be harder to recall than shorter answers.

**Generating supporting passages improve performance.** By first generating a supporting passage, InstructGPT can improve its performance on all question types except for long questions. Our findings are consistent with previous works that LLM can improve downstream performance with intermediate generation steps(Yu et al., 2023, Wei et al., 2022, Kojima et al., 2022). However, InstructGPT performs worse with GENREAD as its outputs becomes shorter. We will further analyze this behavior in Section 5.

## 5 Human Evaluation

### 5.1 Annotation Process

To further analyze the difficulty of MOQAand gain insights into how and why current models fail, we randomly sample and annotate 25 outputs from each question type on the test set for selected models: We assign two scores to each output: a correctness score and a sufficiency score, and show the results in Figure 3.

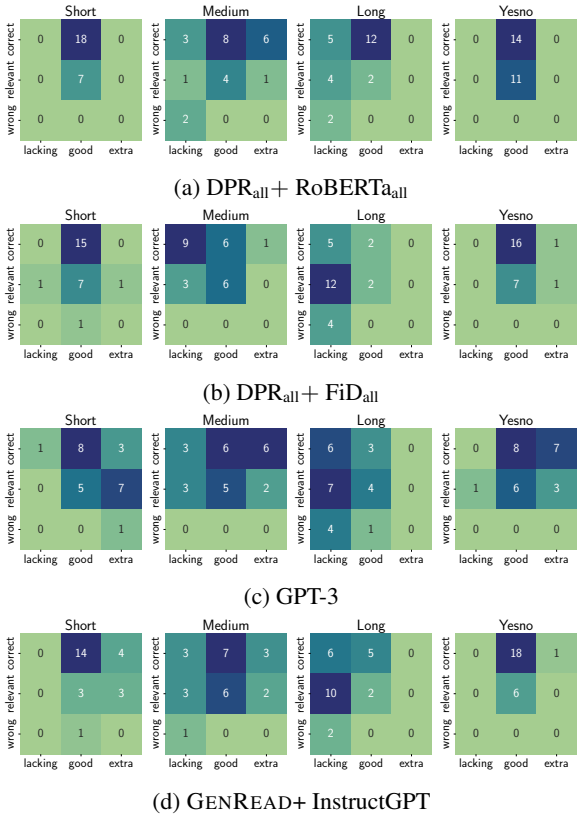
**Correctness score.** It measures the factuality and relevance of the output. `correct` means the output is factually correct with respect to the question; `relevant` means the output is relevant to the query but not factually correct; `wrong` means the output is factually incorrect and irrelevant to the query.

**Sufficiency score.** It reflects if the output provides the appropriate amount of context: `lacking` means the output lacks the amount of explanation and context expected for the query; `good` means the output contains the perfect amount of explanation; `extra` means the output supply excessive explanation.

For more details on the human evaluation and examples of the anntations, please see §I.

### 5.2 Analysis

**Extractive reader can adapt to short, long, and yes/no type questions.** Outputs from the RoBERTa model are often both correct and contain the right amount of information. We also noticed




(a) $DPR_{all}$+ $RoBERTa_{all}$

(b) $DPR_{all}$+ $FiD_{all}$

(c) GPT-3

(d) GENREAD+ InstructGPT

Figure 3: Human evaluation results of 25 randomly sampled questions from the test set on select models. We plot correctness scores and sufficiency scores. Correctness score rates the output on its factually correctness and if it answers the question. Sufficiency score reflects if the output contains the appropriate amount of information required by the question.

that the model will always return either `YES` or `NO` for yes/no type questions, which supports our finding that boolean questions are relatively easy to identify. Since the outputs are always a span from the top 24 retrieved passages, they are almost always relevant to the query.

**Generative models tend to generate shorter responses.** Consider the examples shown in Table 4, where the question is inherently complex as it is asking about the historical origin of the phrase "go commando". In this example, the RoBERTa reader outputs the ground truth answer, while the other models fail in various ways. $DPR_{all}$ + $FiD_{all}$ simply generates "unknown", which is irrelevant to the question and does not answer it at all. GPT-3 elaborates on the word "commando" but does not consider the phrase "go commando". GENREAD+ InstructGPT only offers one explanation, while the complexity of the query necessitates more context.

| |
|---|
| **Q**: Where does the expression go commando come from?<br>**A**: The origins of the phrase "go commando" are uncertain, with some speculating that it may refer to being "out in the open" or "ready for action". "Slate" magazineś Daniel Engber dates the modern usage to United States college campuses circa 1974, where it was perhaps associated with soldiers in the Vietnam War, who were reputed to go without underwear to "increase ventilation and reduce moisture". However, more recently, Graeme Donald has pointed out that the US forces are "Rangers" rather than "Commandos" |
| **DPR$_{\text{all}}$+ FiD$_{\text{all}}$**: unknown |
| **GPT-3**: The term commando is derived from the Spanish word comando, meaning a "detachment" or "body of troops". In English, "commando" was originally a term for a type of soldier, but in the United States military it has come to mean a member of certain special forces units. |
| **GENREAD+ InstructGPT**: The phrase "go commando" is believed to have originated in the British military during World War II. |

Table 4: Prediction samples on a long-type question from the test set. While DPR$_{\text{all}}$ + RoBERTa$_{\text{all}}$ outputs the ground truth answer, other models' outputs lack the expected historical context.

**All models typically output relevant information.** Even when the model does not output the correct answer, they often return relevant information. For instance, for the question "*who plays the woodsman in Over the Garden Wall*", the correct answer is "*Christopher Lloyd*". GPT-3$_{\text{all}}$ returned "*Tom Kenny*", who is a voice actor involved in many animated series similar to *Over the Garden Wall*.

Furthermore, one of the main challenges of long-type questions is the complexity of the query. Generative models output incorrect answers more often on long-type questions, which suggests that **composing an answer with detailed explanations and historical contexts means more room for the model to make mistakes**.

**All models struggle with the boundaries of medium-type questions.** All model sometimes output insufficient or excessive explanations for medium-type questions, which suggests that the boundary between medium-type questions and the other question types are not quite clear. Indeed, this can be the case since medium-type answers may contain characteristics of both short and long-type answers. For example, the answer "Harry Potter and the Deathly Hallows" is more than 5 tokens long, but it is also a proper noun typically found in short-type answers. On the other hand, some answers such as "a transformative change of heart; especially: a spiritual conversion" is more descriptive and similar to long-type questions.

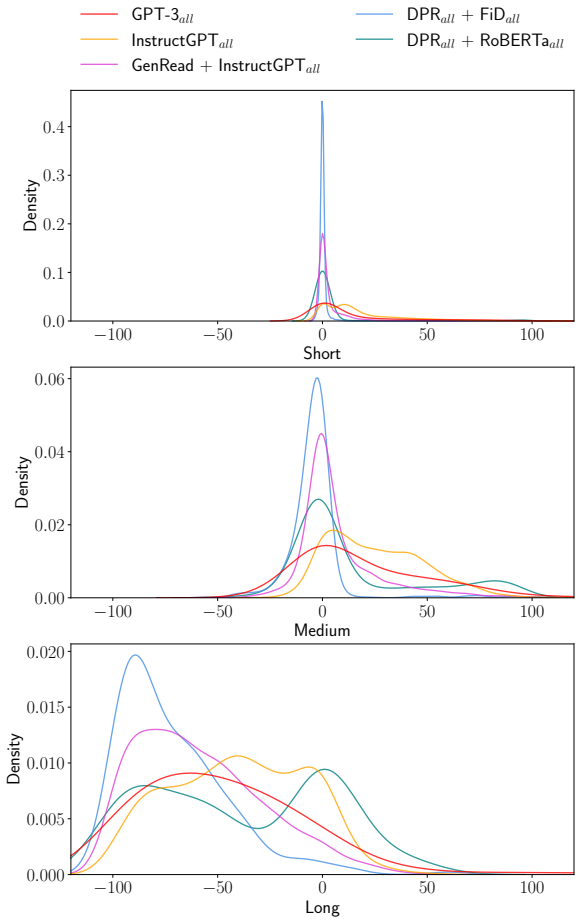



Figure 4: The distribution of $\Delta l_i$ over the test set for DPR$_{\text{all}}$+RoBERTa$_{\text{all}}$, DPR$_{\text{all}}$+FiD$_{\text{all}}$, GPT-3$_{\text{all}}$, InstrctuGPT$_{\text{all}}$, and GENREAD+ InstructGPT$_{\text{all}}$. We omit yes/no-type questions since every model almost always return either YES or NO.

### 5.3 Output Length Distributions

To illustrate the issue of providing insufficient and excessive information, we show the difference between the length of the output generation and the length of the ground truth answer in Figure 4.

Specifically, let $A_i = \{a_1, a_2, \ldots, a_n\}$ be the set of $n$ possible answers to the question $q_i$ and $\texttt{len}(s)$ be the function that returns the number of tokens in the string $s$ after whitespace tokenization. We define the difference in length $\Delta l_i$ between output $o_i$ with $A_i$:

$$\Delta l_i = \begin{cases} \texttt{len}(o_i) - \min L_i & \\ \qquad \text{if } \texttt{len}(o_i) < \min L_i & \\ \texttt{len}(o_i) - \max L_i & \\ \qquad \text{if } \texttt{len}(o_i) > \max L_i & \\ 0 \qquad \text{otherwise} & \end{cases} \tag{1}$$

where $L_i = \{\texttt{len}(a_1), \texttt{len}(a_n), \ldots, \texttt{len}(a_n)\}$.

**Extractive model is better at identifying the question type.** Even though the RoBERTa model often returns outputs shorter than the ground truth of long-type question, its $\Delta l_i$ distribution is centered around 0 for all question types. In contrast, all generative models' outputs often lack explanation for long-type questions.

Interestingly, GENREAD causes InstructGPT to generate shorter outputs on average as the density of the $\Delta l_i$ distribution shifts left. One possible explanation for this is that InstructGPT treats the final output as a summary of the generated passage. As a result, it ends up generating sequences shorter than the passage.

## 6 Related Work

There have been significant prior works that focus on expanding question answering beyond the short-form questions in popular QA datasets such as NQ-Open(Lee et al., 2019), SQuAD(Rajpurkar et al., 2016), and TriviaQA(Joshi et al., 2017).

**Long-form QA** has attracted interests in the community due to its inherent complexity. For example, ELI5 (Fan et al., 2019) mines questions and answers from the subreddit Explain Like I'm Five[5], the answers are abstractive and are not grounded in a knowledge source. This dataset has since been included in KILT (Petroni et al., 2021), which grounds the answers in the development and test set in Wikipedia pages. However, KILT is composed of questions from different distributions, which can introduce superficial cues that help the model determine the format of the answer. Similar to our work, ASQA (Stelmakh et al., 2022) also considers long-form answers in open-domain QA and its questions originates from NQ (Kwiatkowski et al., 2019). However, it only focuses on the questions that have long answers due to ambiguity. GooAQ (Khashabi et al., 2021) propose a dataset consisted of different question types, but these questions are mined from Google's answer boxes and therefore noisy. Furthermore, ELI5, ASQA, and GooAQ all study each question type independent of each other.

**Other forms of QA** Recent works leverage the original NQ beyond those with only short answers. For example, AquaMUSE (Kulkarni et al., 2020) makes use of the long answer only questions from the original NQ for the task of query-based multi-document summarization. They consider NQ questions that only have long answers as these question "result in open-ended and complex topic answers". BoolQ (Clark et al., 2019) focuses on boolean questions that can be answered by true or false, and expands on the yes/no questions from the original NQ by collecting additional questions using the same pipeline. The authors found the task challenging and require robust reasoning ability. The unanswerable questions in NQ has also been explored in Asai and Choi (2020), but such questions are difficult to study in the open-domain setting and require extensive manually annotations. In contrast to these works, we study draw a diverse set of questions with different answers from the same query distribution and study these types together.

**Modeling different types of questions** has also been studied in past works. UnifiedQA (Khashabi et al., 2020) builds a single QA system to answer different types of questions. However, their setting is limited to the closed-domain setup, and only analyzes fine-tuned generative models. Large language models(LLMs) achieves impressive performance on knowledge-intensive tasks such as open-domain QA due to the knowledge packed in its parameters (Brown et al., 2020). These LLMs do not require any additional fine-tuning and only rely on in-context examples to adapt to different tasks. A recent line of work proposes to further improve LLMs capabilities by prompting them to generate a series of intermediate reasoning steps (Wei et al., 2022). GENREAD (Yu et al., 2023) applies a similar idea to open-domain QA, where the LLM first generates a supporting passage instead of relying on external retriever models, and then output the answer. In this work, we use GENREADto further improve the performance of our InstructGPT baseline, and study how this intermediate study affect its ability to generalize to different question types.

## 7 Conclusion

We propose MOQA, an open-domain QA benchmark with multi-type questions, and evaluate a range of baseline models. Our findings suggest that the main difficulty lies in the reader's task, and building a versatile reader or an accurate classifier is challenging. Competitive closed-book QA models also degrade on the new benchmark. Improving the classifier or mining more data to build a generalizable reader might be promising directions towards truly all-round systems.

[5] https://www.reddit.com/r/explainlikeimfive/

## Limitations

Though our ultimate goal is to build a versatile QA system that can handle all types of questions, our benchmark mainly focuses on extractive questions – those can be explicitly answered by copying from a document in the knowledge source. We start from extractive QA because they cover a wide range of real-world questions and are easier to be automatically evaluated. We also omit the evaluation of large language models on MoQA due to limited resource and leave it for future work.

Furthermore, all questions are in English and possibly collected from English-speaking users. We also use the English Wikipedia as our knowledge source. Thus, our models and dataset may under-represent the non-English speakers.

## Ethical Statement

Training language models can use significant amount of energy as the process is very computationally expensive, this can come at an environmental cost. In our work, we attempt to minimize this effects by using pre-trained models like RoBERTa (Liu et al., 2019) and only doing the necessary fine-tuning to minimize the computation cost. Furthermore, to promote reproducible and accessible academic research, we will publicly release all of our dataset and code. Natural language datasets can contain biases like gender and racial stereotypes. Although this issue is not as prominent in QA datasets compared to large copra used for pre-training language models, we encourage the community to build robust QA models that are more resistant to these biases.

## References


Akari Asai and Eunsol Choi. 2020. Challenges in information-seeking qa: Unanswerable questions and paragraph retrieval. In *Annual Meeting of the Association for Computational Linguistics*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and

Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. GooAQ: Open question answering with diverse answer types. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 421–433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *ArXiv*, abs/2010.12694.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *TACL*.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard em approach for weakly supervised question answering. In *EMNLP*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research (JMLR)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *ACL*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

## A Dataset Statistics

We show detailed statistics about the MoQA dataset in Table 5. We also show the majority label for our yes/no questions in Table 6.

## B Mechanical Turk

The annotation results of our MTurk campaign on the test set long-type questions is shown in Table 7.

## C Pre-processing Steps

### C.1 Original Natural Questions

Natural Question is a question answering dataset collected from the Google search engine. Real Google queries are first inputted into the Google search engine to obtain the top ranked Wikipedia page. Then, both the question and the Wikipedia article are given to annotators. The annotators first decide if a question is of good quality, in which case the annotator will attempt to find an answer to the question within the Wikipedia page. The annotators may not find a valid answer or select an HTML bounding box–typically a natural paragraph or a table–as the long answer. If a long answer is selected, then the annotator can choose to select a substring as the short answer or annotate the answer as simply "YES" or "NO". For more details about the original Natural Questions dataset, we refer readers to Kwiatkowski et al. (2019). If the annotator does not find an answer or deem the question of bad quality, then the annotator would denote the answer as "NULL".

We first obtain all the questions that are found in NQ-Open (Lee et al., 2019) and denote them as short-type questions. Specifically, these questions are all the questions that have at least one annotated answer that is 5 words or shorter.

For all other questions, we first filter out all questions where the top Wikipedia page title contains "(disambiguation)", "(disambiguation page)", "(List of .+)", "(Index of .+)", or "(Outline of .+)". Then, we use the Python library BeautifulSoup[6] to clean the HTML tags and obtain the text of each long answer(s) and short answer(s) associated with the question (if they exist).

We denote all questions with at least one short answer as medium-type questions. We denote all remaining questions with at least one YES/NO answers as yes/no-type questions. We denote all remaining questions with at least one long answer as long-type questions except for those questions where the long answer is originally from the HTML boxes "<table>", "<td>", "<tr>", or "<ol>" or if the length is shorter than 10 tokens. We do this because answers from tables and lists are often low quality after being converted to paragraph-like texts, when their structure is lost, or too short to fit the criteria of a long answer, and we discard these questions. We do not use any questions that were not annotated with a long answer.

Finally, we use the original development set as our test set, as the original test is hidden. Following the original Natural Questions, we filter out any questions in the test set that do not have at least 2 non-NULL annotations.

### C.2 Long Answers

Since Natural Questions were collected at different timestamps, each Wikipedia page and therefore the long answers and short answers are from different times. This typically don't affect the short answers, because short spans of text can be easily found in different snapshots of Wikipedia.

However, long answers can be affected by minor grammar changes and the exact paragraph often cannot be found in different versions of Wikipedia. This poses a challenge for extractive models because some long answers might be unanswerable purely due to the difference in Wikipedia snapshots. Therefore, we replace every long answer with the corresponding string from the December 20, 2018, dump of Wikipedia, which is a popular version of Wikipedia dump used by many retrievers (Karpukhin et al., 2020, Lee et al., 2021). Specifically, we use the version where paragraphs are split into 100 token passages from Karpukhin et al. (2020) to minimize the variance in passage length.

To do this, we follow the matching strategy from KILT (Petroni et al., 2021). For every long answer, we first find the same Wikipedia article in our snapshot using the page title. Then, for each passage in the page, we find the span with the highest BLEU score to the long answer. We denote the passage containing the span with the highest BLEU as the gold passage, and the span as the long answer.

For long-type questions, we do an additional filtering where the best span must have a BLEU score of at least 0.5 to ensure that the matching long answer does not deviate significantly from the

[6] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

| Type | # Examples | % of Total | Avg. Question Length | Avg. Answer Length | Min Answer Length | Max Answer Length | Matched Long Answer |
|---|---|---|---|---|---|---|---|
| **Train Set** | | | | | | | |
| Short | 79168 | 66.73 | 9.1 | 2.2 | 1 | 66 | 56346 |
| Medium | 16668 | 14.05 | 9.0 | 11.2 | 1 | 215 | 10891 |
| Long | 19649 | 16.56 | 9.1 | 76.5 | 4 | 100 | 19649 |
| Yes/No | 3154 | 2.66 | 9.1 | 1.0 | 1 | 1 | 2192 |
| Total | 118639 | 100 | 9.1 | 13.8 | 1 | 215 | 89078 |
| **Development Set** | | | | | | | |
| Short | 8757 | 66.69 | 9.0 | 2.2 | 1 | 37 | 6228 |
| Medium | 1853 | 14.11 | 9.1 | 11.2 | 2 | 140 | 1196 |
| Long | 2169 | 16.52 | 9.1 | 76.8 | 7 | 100 | 2169 |
| Yes/No | 351 | 2.67 | 8.9 | 1.0 | 1 | 1 | 234 |
| Total | 13130 | 100 | 9.0 | 13.8 | 1 | 140 | 9827 |
| **Test Set** | | | | | | | |
| Short | 3610 | 80.67 | 9.1 | 3.2 | 1 | 48 | 2645 |
| Medium | 565 | 12.63 | 9.0 | 13.0 | 1 | 62 | 406 |
| Long | 201 | 4.49 | 9.0 | 83.0 | 9 | 100 | 201 |
| Yes/No | 99 | 2.21 | 9.2 | 1.0 | 1 | 1 | 73 |
| Total | 4475 | 100 | 9.1 | 9.1 | 1 | 100 | 3325 |

Table 5: Comprehensive statistics of the MoQA dataset. Text length is calculated using whitespace tokenization. "Has Length" is the number of examples where a matching long answer was found in the 2018-12-20 Wikipedia snapshot. Recall that every example were originally annotated with at least one long answer, and the long answer matching process is descried in C.

| Dataset | %Yes |
|---|---|
| Train | 61.41 |
| Dev | 62.11 |
| Test | 69.70 |

Table 6: Percentage of YES in yes/no questions.

| Annotation | Count |
|---|---|
| Long | 201 |
| Not Long | 145 |
| Total | 346 |

Table 7: Resulting annotation of our Mechanical Turk (MTurk) campaign.

original long answer, and we discard any long-type questions where we fail to find such span.

For yes/no-type questions, we only add the gold passage and matching long answer annotation if the best span's BLEU score is 0.5, but we do not discard the questions where no matching long answer is found.

For short-type and medium-type question, we restrict the gold passage selection to only the passages where the short or medium answer is found as an exact substring, but use no BLEU score threshold when matching the long answer, because if the short or medium answer is found within the same Wikipedia page, then the long answer is likely to be found in one of the paragraphs. We also do not discard the questions where no matching long answer is found.

Therefore, every long-type question is guaranteed to have at least one matching long answer, while the other types are not guaranteed to have any matching long answer.

### C.3 Unanswerable Questions

Although the original Natural Questions dataset (Kwiatkowski et al., 2019) includes questions that were not annotated with any answers, we chose not to include them in MoQA. This is because the nature of the dataset collection process does not guarantee the question to be unanswerable or truly have NULL as an answer.

Specifically, annotators are only shown the top Wikipedia result returned by the used search engine and they can only choose an extractive answer based on that Wikipedia article. However, it is not always true that the gold answer in the in top Wikipedia page returned by the search engine: they can exist in other pages.

Therefore, even though unanswerable questions are important to study (Rajpurkar et al., 2018), the questions not annotated with any answers in the original Natural Question cannot be taken as true

| Retriever | Reader | Short | Medium | Long |
|---|---|---|---|---|
| $\text{DPR}_{\text{all}}$ | $\text{RoBERTa}_*^{\dagger}$ | 78.8 | 72.4 | 71.7 |
| | $\text{FiD}_*^{\dagger}$ | 80.3 | 74.5 | 71.0 |
| None | GPT-3$^{\dagger}$ | 67.9(1.8) | 60.2(0.2) | 51.9(1.7) |
| None | Codex $^{\dagger}$ | 74.3(1.9) | 62.6(0.9) | 54.5(2.9) |
| None | InstructGPT$^{\dagger}$ | 68.0(0.3) | 63.7(0.2) | 59.1(0.2) |
| GENREAD | InstructGPT$^{\dagger}$ | 75.9(0.3) | 65.2(0.6) | 59.0(0.4) |
| $\text{DPR}_{\text{all}}$ | $\text{RoBERTa}_{\text{all}}$ | 77.9 | **67.2** | **59.1** |
| | Cls. + $\text{RoBERTa}_* \times 4$ | 77.7 | 61.5 | 52.4 |
| | $\text{FiD}_{\text{all}}$ | **80.0** | 65.0 | 44.3 |
| | Cls. + $\text{FiD}_* \times 4$ | 79.4 | 61.5 | 52.2 |
| None | $\text{T5-large}_{\text{all}}$ | 64.9 | 56.4 | 46.5 |
| None | GPT-3 | 63.2(2.8) | 56.4(1.3) | 51.4(1.3) |
| None | Codex | 67.4(3.9) | 58.7(1.5) | 54.6(0.8) |
| None | InstructGPT | 60.5(1.4) | 61.5(0.8) | 57.5(0.4) |
| GENREAD | InstructGPT | 73.1(0.7) | 65.6(0.3) | 51.6(0.4) |

Table 8: BERTScore(Zhang et al., 2019) results on the MoQA test set. For the LLMs, we show the mean and the standard deviation in parentheses across three randomly seeded runs. $^{\dagger}$: An oracle that assumes that the question type is known and uses a reader trained only on each individual type. Cls.: a question classifier is used, and there are four readers trained independently, either for RoBERTa or FiD. Otherwise, all questions are jointly trained. $\text{T5-large}_{\text{all}}$: a closed-book QA model trained on all questions. GPT-3(Brown et al., 2020): `davinci` model with two-shot ICL. InstructGPT(Ouyang et al., 2022): `text-davinci-003` model with two-shot ICL. Codex(Chen et al., 2021): `code-davinci-002` model with two-shot ICL.

unanswerable questions without significantly more annotations.

## D Evaluation

For retrieval, we regard a retrieved passage is correct if the passage contains the matching long answer of the question (we exclude the questions without matching long answers in retrieval evaluation). We report accuracy@$k$ that checks if the top $k$ retrieved passages contains a correct passage. We also report mean reciprocal rank (MRR)@$k$, which averages the reciprocal rank of the top correct passage.

For the final answer, we report the exact match (EM) and the F1 score for short-type and medium-type questions, EM and ROUGE-L for long answers, and accuracy for yes/no-type questions. Although EM is often the primary metric used in open-domain QA research, we believe that the F1 and ROUGE-L scores are more suitable for medium and long answers following past work on long-form QA (Fan et al., 2019; Stelmakh et al., 2022; Khashabi et al., 2021). We also report the macro-averaged EM, which averages the EM scores across the 4 question types.

Furthermore, we also report the BERTScore(Zhang et al., 2019) on short, medium, and long-type questions. BERTScore is a model-based metrics that is better at capturing semantic similarities between long-form texts than traditional metrics. This is especially important for the medium and long type questions, since there are many possible ways of answering the question without using the exact same words. We use the authors' implementation[7] and the set-up with the best human correlation score at the time of writing: model is `microsoft/deberta-xlarge-mnli`[8] and the layer is 40[9]. We found that the BERTScore correlates with F1 scores and ROUGE-L scores on short, medium, and long type answers. This suggests that the best performing models RoBERTaall and FiDall also output the most semantically similar texts to the ground truth answers.

[7] https://github.com/Tiiiger/bert_score

[8] https://huggingface.co/microsoft/deberta-xlarge-mnli

[9] hash is `microsoft/deberta-xlarge-mnli_L40_no-idf_version=0.3.12(hug_trans=4.23.0)`

## E Retrieval

We use the code repo[10] from Karpukhin et al. (2020) for our DPR experiments. We also follow all of their hyperparameter settings; specifically, we use a learning rate of $2 \times 10^{-5}$, and a per GPU batch size of 16 and a gradient accumulation of 2 on 4 80GB A100 GPUs to achieve a total batch size of 128. We train for 40 epochs, except for $\text{DPR}_{\text{yesno-gold}}$ and $\text{DPR}_{\text{yesno-DS}}$ where we train for 400 epochs due to its small train set. We then use the model with the highest validation rank loss as our final model.

Similar to the original DPR, we also explore how the final retrieval model perform when using the annotated gold passage and when using a distant-supervision annotation approach that selects the positive passage based on the answer alone. These two strategies are particularly interesting to investigate in Natural Questions because it's unique from other open-domain QA dataset such that all answers are grounded in one gold passage.

In this section, we describe our two approaches to training DPR, even though we only present DPR trained using gold passages in the main results.

### E.1 Distantly-Supervised DPR

In distant supervision, we first retrieve the top 100 passages from the corpus using an BM25 index. We use the Pyserini[11] implementation of BM25. The input to the BM25 search is the question in each example, and we use the default BM25 hyperparameters also used by DPR (Karpukhin et al., 2020). Specifically, the parameters we use are: $b = 0.4$ (document length normalization) and $k1 = 0.9$ (term frequency scaling). Then, for each question-answer pair, we choose the positive passages by checking if the answer exists within the retrieved passage.

Another interesting aspect to consider is which answer we use for selecting the positive passage. Namely, we can use either short or long answers for short questions, and medium or long answers for medium questions. From preliminary experiments, we found that using the long answers for choosing the positive passages for short questions performed better on the development set while using the medium answers for choosing the positive passages for medium questions. This is likely due to long answer filtering reduces the number of false positives, and it is often likely to get the short answer string in irrelevant passages, and even though the training set size is reduced significantly, the higher quality of data makes up for the smaller dataset. However, using medium answers for filtering already obtain high enough quality of positive passages such that the harsher filtering with long answers will cut down on the training set size enough that it hurts retrieval performance.

When using long answer for filtering, we use a fuzzy matching where the positive passages are those that obtain a BLEU score of at least 0.5 with the long answer. When using short/medium answers for filtering, we check for exact match within the passage. All passages that were filtered out are used as hard negative passages.

For $\text{DPR}_{\text{all-DS}}$, we use the strategy that worked the best for each individual types. Specifically, we use long answer filtering for short, long, and yes/no

| Retriever | | A@1 | A@5 | A@20 | MRR@20 |
|---|---|---|---|---|---|
| **Short** | | | | | |
| $\text{DPR}_{\text{short-DS}}$ | 20k | **35.5** | 58.4 | **73.1** | 45.6 |
| $\text{DPR}_{\text{medium-DS}}$ | 8k | 23.6 | 42.6 | 58.0 | 32.5 |
| $\text{DPR}_{\text{long-DS}}$ | 8k | 27.3 | 46.9 | 62.5 | 36.4 |
| $\text{DPR}_{\text{yesno-DS}}$ | 1k | 16.7 | 31.4 | 43.6 | 23.4 |
| $\text{DPR}_{\text{all-DS}}$ | 36k | 35.4 | **58.9** | **73.1** | **45.8** |
| **Medium** | | | | | |
| $\text{DPR}_{\text{short-DS}}$ | 20k | 42.4 | 67.7 | **85.0** | 53.7 |
| $\text{DPR}_{\text{medium-DS}}$ | 8k | 38.7 | 64.5 | 75.6 | 49.5 |
| $\text{DPR}_{\text{long-DS}}$ | 8k | 37.9 | 66.0 | 79.1 | 50.2 |
| $\text{DPR}_{\text{yesno-DS}}$ | 1k | 28.1 | 43.1 | 56.2 | 34.7 |
| $\text{DPR}_{\text{all-DS}}$ | 36k | **44.3** | **69.7** | 83.5 | **55.7** |
| **Long** | | | | | |
| $\text{DPR}_{\text{short-DS}}$ | 20k | 30.4 | 52.2 | 67.2 | 40.0 |
| $\text{DPR}_{\text{medium-DS}}$ | 8k | 26.9 | 45.3 | 57.2 | 35.1 |
| $\text{DPR}_{\text{long-DS}}$ | 8k | **34.8** | **59.2** | 70.7 | **45.6** |
| $\text{DPR}_{\text{yesno-DS}}$ | 1k | 22.9 | 33.8 | 45.8 | 28.5 |
| $\text{DPR}_{\text{all-DS}}$ | 36k | 33.8 | 55.7 | **71.6** | 43.8 |
| **Yes/No** | | | | | |
| $\text{DPR}_{\text{short-DS}}$ | 20k | **31.5** | 54.8 | 71.2 | 43.1 |
| $\text{DPR}_{\text{medium-DS}}$ | 8k | 26.0 | 52.1 | 65.8 | 38.2 |
| $\text{DPR}_{\text{long-DS}}$ | 8k | **31.5** | 57.5 | 76.7 | 44.4 |
| $\text{DPR}_{\text{yesno-DS}}$ | 1k | 27.4 | 52.1 | 65.8 | 37.8 |
| $\text{DPR}_{\text{all-DS}}$ | 36k | 28.8 | **65.8** | **82.2** | **45.0** |

Table 9: Passage retrieval results on the MoQA test set using distantly-supervised DPR models. We retrieve top $k$ passages from each model and evaluate them based on the presence of the gold long answer in the passages. We report top-$k$ retrieval accuracy (A@$k$), and mean reciprocal rank at $k$ (MRR@$k$).

[10] https://github.com/facebookresearch/DPR
[11] https://github.com/castorini/pyserini

| Retriever | | A@1 | A@5 | A@20 | MRR@20 |
|---|---|---|---|---|---|
| **Short** | | | | | |
| $\text{DPR}_{\text{short-gold}}$ | 56k | 33.6 | 59.7 | 75.1 | 45.0 |
| $\text{DPR}_{\text{medium-gold}}$ | 11k | 27.0 | 50.4 | 67.0 | 37.4 |
| $\text{DPR}_{\text{long-gold}}$ | 20k | 23.8 | 48.7 | 65.1 | 34.7 |
| $\text{DPR}_{\text{yesno-gold}}$ | 2k | 17.7 | 34.4 | 48.5 | 25.2 |
| $\text{DPR}_{\text{all-gold}}$ | 89k | **36.0** | **62.8** | **77.0** | **47.6** |
| **Medium** | | | | | |
| $\text{DPR}_{\text{short-gold}}$ | 56k | 30.3 | 62.8 | 81.0 | 44.5 |
| $\text{DPR}_{\text{medium-gold}}$ | 11k | **45.8** | **71.9** | **86.7** | **57.9** |
| $\text{DPR}_{\text{long-gold}}$ | 20k | 32.3 | 64.5 | 81.0 | 46.1 |
| $\text{DPR}_{\text{yesno-gold}}$ | 2k | 27.8 | 50.2 | 63.3 | 37.3 |
| $\text{DPR}_{\text{all-gold}}$ | 89k | 41.4 | 69.5 | 83.3 | 53.7 |
| **Long** | | | | | |
| $\text{DPR}_{\text{short-gold}}$ | 56k | 24.4 | 51.7 | 69.7 | 36.9 |
| $\text{DPR}_{\text{medium-gold}}$ | 11k | 28.9 | 52.2 | 65.7 | 39.0 |
| $\text{DPR}_{\text{long-gold}}$ | 20k | 34.8 | **65.2** | 78.1 | 47.4 |
| $\text{DPR}_{\text{yesno-gold}}$ | 2k | 20.4 | 42.3 | 54.2 | 29.7 |
| $\text{DPR}_{\text{all-gold}}$ | 89k | **37.8** | 64.7 | **80.6** | **49.2** |
| **Yes/No** | | | | | |
| $\text{DPR}_{\text{short-gold}}$ | 56k | 24.7 | 43.8 | 75.3 | 35.2 |
| $\text{DPR}_{\text{medium-gold}}$ | 11k | 24.7 | 53.4 | 69.9 | 36.6 |
| $\text{DPR}_{\text{long-gold}}$ | 20k | 21.9 | 54.8 | **79.5** | 38.3 |
| $\text{DPR}_{\text{yesno-gold}}$ | 2k | 35.6 | 57.5 | 72.6 | 45.9 |
| $\text{DPR}_{\text{all-gold}}$ | 89k | **37.0** | **64.4** | **79.5** | **49.7** |

Table 10: Passage retrieval results on the MoQA test set using gold passage supervised DPR models. We retrieve top $k$ passages from each model and evaluate them based on the presence of the gold long answer in the passages. We report top-$k$ retrieval accuracy (A@$k$), and mean reciprocal rank at $k$ (MRR@$k$).

questions and medium answer filtering for medium questions.

### E.2 Gold Passage Supervised DPR

Our main results use the gold passage supervised DPR. We follow the BM25 retrieval steps previously described, but we choose our positive passage differently. That is, we denote the gold passage as the positive passage for every question, and we choose all other retrieved passages as the hard negative passages, except for the gold passage if it were retrieved. Our final model $\text{DPR}_{\text{all}}$ uses all questions and their gold passages.

## F Reader

### F.1 Extractive Reader

For the reader model, we employ a similar architecture to the reader model from Karpukhin et al. (2020). However, one additional change we make is prepending every passage with two special tokens: [YES] and [NO]. This is a necessary addition in order to always give the reader model the choice of outputting YES/NO. We separate these two special tokens from the rest of the passage with a separator token </s>.

Before training the reader, we first run inference on the retriever and obtain the top 100 passages for each question. We train the reader model on 8 RTX-3090 GPU. We train $\text{RoBERTa}_{\text{all}}$ for 5 epochs, $\text{RoBERTa}_{\text{short}}$ and $\text{RoBERTa}_{\text{medium}}$ for 10 epochs, $\text{RoBERTa}_{\text{long}}$ for 20 epochs, $\text{RoBERTa}_{\text{yesno}}$ for 40 epochs. We use a learning rate of $2 \times 10^{-5}$ and a batch size of 32 questions with 24 passages each; specifically, we use a batch size of 1 question with 24 passages per GPU and a gradient accumulation of 4. However, we use a total batch size of 16 for $\text{RoBERTa}_{\text{yesno}}$ due to its small training set by reducing the gradient accumulation to 2.

We first did a hyperparameter sweep over batch size $= \{16, 32\}$ and learning rate $= \{1 \times 10^{-5}, 2 \times 10^{-5}\}$ for each question type, and select our final model based on the development set performance.

For each question, we use 1 positive passage and 23 negative passages. The positive passages for short-type, medium-type, and long-type questions are passages that contain the gold answer, and the negative passages are any passages that do not contain the gold answer. We annotate the positive passage with the starting and ending position of all answers in the passage. The positive passages for yes/no-type questions are the retrieved passages with the highest F1 score with the long answer, and all other passages can serve as negative passages. In practice, we use the passages with the lowest F1 score with the long answer as negative passages. We annotate these positive passages with the correct answer span as the special tokens [YES] or [NO] accordingly to the answer.

For each question, we maximize the marginal log-likelihood of all correct spans in the positive passage combined with the log-likelihood of the correct passage being selected. We refer reader to Karpukhin et al. (2020) for more details.

During evaluation, we input the top 50 retrieved passages and select the span with the highest combined span score and passage rank score. We follow the computation in Karpukhin et al. (2020). We chose to include the top 50 passages by first sweeping over the hyperparameter $k = \{10, 25, 50, 100\}$, and chose the best $k$ on the development set.

### F.2 Fusion-in-Decoder

We follow (Izacard and Grave, 2021) and use their code repo[12] for training our FiD models, which uses T5-base. We follow the hyperparameters described in the original paper: the top 100 retrieved passages and their title are included in the input during training and evaluation, learning rate is $1 \times 10^{-4}$, 10k total steps with 1k warm up steps, and a batch size of 64. We train on 8 RTX-3090 GPUs and a gradient accumulation of 8. We evaluate the model every 500 steps and chose the model with the best development set performance for our final model.

## G Classifier

We train our classifier similar to sentence classification tasks. Specifically, we use the pre-trained RoBERTa-base (Liu et al., 2019) as our base model, and finetune it for 10 epochs over our entire dataset. The input is the question, and we use the last hidden state of the `[CLS]` token followed by a linear layer and softmax to predict the probabilities of four question types. We use Huggingface[13]'s implementation of RoBERTaForSequenceClassification to train our classifier; we use Cross Entropy Loss to maximum the log probability of the correct question type.

For our hyperparameters, we use a batch size of 8 and a learning rate of $1 \times 10^{-5}$. We performed hyperparameter search over batch size $= \{8, 16, 32\}$ and learning rate $= \{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$, and chose the model with the highest accuracy on the development set. We also evaluate the model after every epoch and choose the model with the best development accuracy.

## H Closed-Book QA

### H.1 Supervised models

For supervised closed-book QA, we use T5-large (Raffel et al., 2019) as our base model. T5-large$_{\text{all}}$ is trained on all question types together. We follow (Roberts et al., 2020) for training with a constant learning rate of 0.001. We use a batch size per GPU of 8 and 8 32GB GPUs for a total batch size of 64. We train for 20k steps, and evaluate using beam search with a beam size of 4. We evaluate our model on the development set every 1k steps and choose the model with the highest performance on the development set for our final model.

### H.2 Unsupervised Large Language Models

We use the following large language models in our experiments: GPT-3 (`davinci`; (Brown et al., 2020)), Codex (`code-davinci-002`; (Chen et al., 2021)), and InstructGPT (`text-davinci-003`; Ouyang et al., 2022).

The prompt used for the two-shot experiments with only two demonstration from one question type is shown in 5. We follow the same prompting strategy for the experiments with two demonstrations from each question type. Furthermore, we balance the in-context examples such that the first four demonstrations contain one example from each question type and that the last four demonstrations also contain one example from each question type. We also use one yes/no question with `YES` as the answer and another question with `NO` as the answer with the yes/no-type questions are used in the demonstrations.

For the experiments with GENREAD, we use the two-step prompting strategy shown in 6, but other experimental setups are identical to the experiments without GENREAD.

## I Human Evaluation

We describe our procedure for assigning the correctness score and the explanation in further detail here. We also show examples of the annotation in 11.

We use the following guidelines when assigning the correctness score:

1. We assign an output $o_i$ with $\texttt{correct}(o_i) =$ correct if it is factually correct and supported by $A_i$ even if the word/phrasing is not exact. For instance, the question "Who is the 44th President of the United States" has multiple valid outputs: "Obama", "Barack Obama", and "Barack Hussein Obama" even though only "Barack Obama" is annotated as the old answer.

2. We assign an output $o_i$ with $\texttt{correct}(o_i) =$ relevant if it does not correctly answer the question $q_i$ according to $A_i$, but include some relevant information. For instance, an example of such output to the question "Who is the 44th President of the United States" include "Donald Trump" (who is the 45th President

[12] https://github.com/facebookresearch/FiD
[13] https://huggingface.co

| | | |
|---|---|---|
| Context | → | Q: What two products are produced in the dehydration synthesis of two amino acids?<br>A: a molecule of water (H2O) and two amino acids joined by a peptide bond<br>Q: When does it change from bc to ad?<br>A: conception or birth of Jesus of Nazareth<br>Q: What purpose did seasonal monsoon winds have on trade? |
| Completion | → | A: |

Figure 5: Example of two-shot in-context learning used for large pretrained language models. We prompt the model with Context + A: and take the generation as the final output.

| | | |
|---|---|---|
| Context | → | Q: What purpose did seasonal monsoon winds have on trade? |
| Completion 1 | → | Passage: The seasonal monsoon winds were an important factor in the development of trade in the Indian Ocean region. The monsoons were used to transport goods and people from the Arabian Peninsula to India, and from India to East Africa and Southeast Asia. The monsoons also allowed traders to travel from the Persian Gulf to the Bay of Bengal and the Malay Peninsula. The monsoons were also used to transport goods from India to China, and from China to the Middle East. The monsoons allowed traders to travel from the Persian Gulf to the Bay of Bengal and the Malay Peninsula. |
| Completion 2 | → | A: The seasonal monsoon winds were an important factor in the development of trade in the Indian Ocean region, allowing traders to travel from the Persian Gulf to the Bay of Bengal and the Malay Peninsula. |

Figure 6: Example of zero-shot in-context learning with GENREAD used for large pretrained language models. We first prompt the model with Context + Passage: to generate Completion 1. Then, we prompt the model again with the Context + Completion 1 + A: to generate the answer. We take the generation from the second step as the final output.

of the United States) and "Joe Biden" (who was Obama's Vice President as well as the 46th President of the United States). In practice, this is the most common mistake that models make, where if the ground truth answer is a politician, artist, or date, the model would typically output something in the same category. Furthermore, outputs are also put into this category if only parts of the output is factually correct but also contain unsupported information (e.g. "Obama and Biden" in this example).

3. We assign an output $o_i$ with $\texttt{correct}(o_i) =$ wrong if it does not correctly answer the question $q_i$ according to $A_i$ and does not include any relevant information. For instance, an example of such output to the question "Who is the 44th President of the United States" could be "turtle eggs". In practice, this mistake rarely occurs, and we only assign this score when there is no obvious connection between the output and the question topic.

In general, the procedure for determining the explanation score involves first reading the question and the ground truth answers to decide the minimal amount of information that is needed to answer the entire question. For instance, if the question asks "what are Poseidon's symbols and what do they mean?", then an appropriate answer should include all of his symbols in Greek mythology (e.g. the trident, the horse, the bull, etc.) as well as the meaning of each symbol. If the output only include one symbol or multiple symbols without their meaning, then we would consider the output to be not sufficient, yielding a score of lacking. If the output includes things beyond the symbols but also the stories from Greek mythology about Poseidon, then we will consider the output to be too long, yielding a score of extra.

## J Prediction Samples

We show example output for selected models for each question type in Figures 12, 13, 15, and 14. It is interesting to note that while both models are capable of outputting the correct answers and the appropriate granularity for some question, they still have much room for improvement. For example, both models can generate a short answer where a long answer would be more appropriate, while

| Question Type | Question | Model | Output | Correctness Score | Sufficency Score |
|---|---|---|---|---|---|
| Short | who plays the woodsman in over the garden wall | $FiD_{all}$ | Christopher Lloyd | Correct | Good |
| | | $GPT\text{-}3_{all}$ | Tom Kenny | Relevant | Good |
| Medium | what is the meaning of the greek word metanoia | $RoBERTa_{all}$ | a transformative change of heart; especially: a spiritual conversion | Correct | Good |
| | | $GPT\text{-}3_{all}$ | metanoia is a Greek word meaning "change of mind" or "change of heart". It is used in the New Testament to describe a change of mind or heart in regards to one's relationship with God. | Correct | Extra |
| Long | where does the expression go commando come from | $RoBERTa_{all}$ | The origins of the phrase "go commando" are uncertain, with some speculating that it may refer to being "out in the open" or "ready for action". "Slate" magazineś Daniel Engber dates the modern usage to United States college campuses circa 1974, where it was perhaps associated with soldiers in the Vietnam War, who were reputed to go without underwear to "increase ventilation and reduce moisture". However, more recently, Graeme Donald has pointed out that the US forces are "Rangers" rather than "Commandos" | Correct | Good |
| | | $GPT\text{-}3_{all}$ | The term commando is derived from the Spanish word comando, meaning a "detachment" or "body of troops". In English, "commando" was originally a term for a type of soldier, but in the United States military it has come to mean a member of certain special forces units. | Relevant | Good |
| | | $InstructGPT_{all}$ | The phrase "go commando" is believed to have originated in the British military during World War II. | Correct | Lacking |
| | | $FiD\text{-}all_{all}$ | unknown | Wrong | Lacking |
| Yes/No | do irish passport holders need visa for thailand | $GPT\text{-}3_{all}$ | No, Irish citizens do not require a visa to enter Thailand. | Correct | Extra |
| | | $InstructGPT_{all}$ | No | Correct | Good |

Table 11: Prediction samples from the test set with the correctness and sufficency scores.

they can also generate some short answer where a yes/no answer would be correct.

It is also worth noting that questions that may appear to require short answers may actually require a much more length explanation. In other words, even though "who", "when", and "where" questions may seem to only require short answers, they can actually sometimes require much longer explanation. For example, both "Who is responsible for redrawing congressional and legislative districts?" and "Who is covered under payment of gratuity act 1972?" may appear to be simple and straight forward at first, but there is actually greater complexity behind the question that cannot be answered by just a few words, and it only becomes clear that they require length explanations given extensive knowledge on the topic. It is worth noting that our classifier model classifies both of those questions as short questions, which motivates future work into question classification.

Other questions that require long answers even though they appear to be short questions include:

- "When is a system said to be reproducible?"
- "Where does the word christ come from and what does it mean?"
- "What makes miranda a central character in the tempest?"
- "Where does the term spelling bee come from?"

In summary, question type classification is not as simple and straight forward as it may appear, and it actually requires extensive knowledge on the related background.

| **Short** - *answers up to 5 tokens* |
|---|
| **Q**: Who plays the woodsman in over the garden wall?<br>**A**: Christopher Lloyd<br>**DPR$_{all}$+ RoBERTa$_{all}$**: Christopher Lloyd<br>**DPR$_{all}$+ FiD$_{all}$**: Christopher Lloyd<br>**GPT-3**: Tom Kenny<br>**GENREAD+ InstructGPT**: Christopher Lloyd |
| **Q**: Who has a ring of power in lotr?<br>**A**: Sauron<br>**DPR$_{all}$+ RoBERTa$_{all}$**: Frodo<br>**DPR$_{all}$+ FiD$_{all}$**: Sauron<br>**GPT-3**: Sauron<br>**GENREAD+ InstructGPT**: Sauron |
| **Q**: the witch and the hundred knight 2 us release<br>**A**: 'March 2018' *or* 'March 27, 2018'<br>**DPR$_{all}$+ RoBERTa$_{all}$**: March 2018<br>**DPR$_{all}$+ FiD$_{all}$**: March 2018<br>**GPT-3**: The Witch and the Hundred Knight 2 is a Japanese action role-playing game developed by Nippon Ichi Software and published by NIS America. It is the sequel to The Witch and the Hundred Knight, and was released in Japan on March 24, 2017, and in North America and Europe on August 29, 2017.<br>**GENREAD+ InstructGPT**: March 27, 2018 |

Table 12: Prediction samples on a short-type question from the test set.

| **Medium** - *answers more than 5 tokens* |
|---|
| **Q**: What is the meaning of the greek word metanoia?<br>**A**: a transformative change of heart; especially: a spiritual conversion<br>**DPR$_{all}$+ RoBERTa$_{all}$**: a transformative change of heart; especially: a spiritual conversion<br>**DPR$_{all}$+ FiD$_{all}$**: a transformative change of heart<br>**GPT-3**: metanoia is a Greek word meaning "change of mind" or "change of heart". It is used in the New Testament to describe a change of mind or heart in regards to one's relationship with God.<br>**GENREAD+ InstructGPT**: change of heart, mind, and outlook |
| **Q**: Where do you get male pattern baldness from?<br>**A**: combination of genetics and the male hormone dihydrotestosterone<br>**DPR$_{all}$+ RoBERTa$_{all}$**: genetics<br>**DPR$_{all}$+ FiD$_{all}$**: genetics<br>**GPT-3**: The most common cause of male pattern baldness is a genetic predisposition. Male pattern baldness is an inherited trait that is passed down from your parents.<br>**GENREAD+ InstructGPT**: a combination of genetics and hormones |
| **Q**: the stonewall riot in new york city in 1969 involved a protest by<br>**A**: 'members of the gay (LGBT) community' *or* 'the gay (LGBT) community'<br>**DPR$_{all}$+ RoBERTa$_{all}$**: members of the gay (LGBT) community<br>**DPR$_{all}$+ FiD$_{all}$**: members of the gay (LGBT) community<br>**GPT-3: gay men and lesbians**<br>**GENREAD+ InstructGPT**: members of the gay community |

Table 13: Prediction samples on a medium-type question from the test set.

| **Yes/no** - *answers yes/no* |
|---|
| **Q**: Do Irish passport holders need visa for Thailand?<br>**A**: NO<br>**DPR$_{all}$+ RoBERTa$_{all}$**: YES<br>**DPR$_{all}$+ FiD$_{all}$**: YES<br>**GPT-3**: No, Irish citizens do not a visa to enter Thailand.<br>**GENREAD+ InstructGPT**: NO |
| **Q**: Does any country still fly the f4 phantom?<br>**A**: YES<br>**DPR$_{all}$+ RoBERTa$_{all}$**: YES<br>**DPR$_{all}$+ FiD$_{all}$**: Iran<br>GPT-3: Yes, the United States Air Force still flies the F-4 Phantom II. The F-4 was retired from the U.S. Navy in the 1990s, but the U.S. Marine Corps continues to operate the F-4. The F-4 is also operated by the militaries of Egypt, Greece, Iran, Israel, Japan, South Korea, Turkey, and Taiwan.<br>**GENREAD+ InstructGPT**: YES, Iran |
| **Q**: Is there going to be another season of degrassi next class?<br>**A**: YES<br>**DPR$_{all}$+ RoBERTa$_{all}$**: YES<br>**DPR$_{all}$+ FiD$_{all}$**: YES<br>**GPT-3**: Yes, there will be a fifth season of Degrassi: Next Class.<br>**GENREAD+ InstructGPT**: NO |

Table 14: Prediction samples on a yes/no-type question from the test set.

| **Long** - *answers in paragraph* |
|---|
| **Q**: Where does the expression go commando come from?<br>**A**: The origins of the phrase “go commando" are uncertain, with some speculating that it may refer to being “out in the open" or “ready for action". “Slate" magazineś Daniel Engber dates the modern usage to United States college campuses circa 1974, where it was perhaps associated with soldiers in the Vietnam War, who were reputed to go without underwear to “increase ventilation and reduce moisture". However, more recently, Graeme Donald has pointed out that the US forces are “Rangers" rather than “Commandos"<br>**DPR$_{all}$+ RoBERTa$_{all}$**: The origins of the phrase “go commando" are uncertain, with some speculating that it may refer to being “out in the open" or “ready for action". “Slate" magazine’s Daniel Engber dates the modern usage to United States college campuses circa 1974, where it was perhaps associated with soldiers in the Vietnam War, who were reputed to go without underwear to “increase ventilation and reduce moisture". However, more recently, Graeme Donald has pointed out that the US forces are “Rangers" rather than “Commandos", and<br>**DPR$_{all}$+ FiD$_{all}$**: unknown<br>**GPT-3**: The term commando is derived from the Spanish word comando, meaning a “detachment" or “body of troops". In English, “commando" was originally a term for a type of soldier, but in the United States military it has come to mean a member of certain special forces units.<br>**GENREAD+ InstructGPT**: The phrase “go commando" is believed to have originated in the British military during World War II. |
| **Q**: bankruptcy abuse prevention and consumer protection act of 2005 summary<br>**A**: Bankruptcy Abuse Prevention and Consumer Protection Act of 2005 (BAPCPA) (), is a legislative act that made several significant changes to the United States Bankruptcy Code. Referred to colloquially as the "New Bankruptcy Law", the Act of Congress attempts to, among other things, make it more difficult for some consumers to file bankruptcy under Chapter 7; some of these consumers may instead utilize Chapter 13. Voting record of S. 256.<br>**DPR$_{all}$+ RoBERTa$_{all}$**: The Bankruptcy Abuse Prevention and Consumer Protection Act of 2005 (BAPCPA) (), is a legislative act that made several significant changes to the United States Bankruptcy Code. Referred to colloquially as the "New Bankruptcy Law", the Act of Congress attempts to, among other things, make it more difficult for some consumers to file bankruptcy under Chapter 7; some of these consumers may instead utilize Chapter 13<br>**DPR$_{all}$+ FiD$_{all}$**: YES<br>**GPT-3**: The Bankruptcy Abuse Prevention and Consumer Protection Act of 2005 (BAPCPA) is a United States federal law that made major changes in the United States Bankruptcy Code. The Act was passed by the 109th United States Congress and signed into law by President George W. Bush on April 20, 2005. The Act was the first major overhaul of the United States Bankruptcy Code since 1978.<br>**GENREAD+ InstructGPT**: The Bankruptcy Abuse Prevention and Consumer Protection Act of 2005 (BAPCPA) is a federal law that made it more difficult for individuals to file for bankruptcy under Chapter 7. The law requires debtors to pass a means test to determine if they are eligible to file for Chapter 7 bankruptcy. It also requires debtors to complete a credit counseling course before filing for bankruptcy. Additionally, the law makes it more difficult for debtors to discharge certain types of debts, such as student loans and domestic support obligations. |
| **Q**: What are poseidon’s symbols and what do they mean?<br>**A**: In Greek myth, Poseidon used his trident to create water sources in Greece and the horse. Poseidon, as well as being god of the sea, was also known as the “Earth Shaker" because when he struck the earth in anger he caused mighty earthquakes and he used his trident to stir up tidal waves, tsunamis and sea storms. In relation to its fishing origins, the trident is associated with Poseidon, the god of the sea in Greek mythology<br>**DPR$_{all}$+ RoBERTa$_{all}$**:love interest of the Cyclops Polyphemus.They symbolized everything that is beautiful and kind about the sea<br>**DPR$_{all}$+ FiD$_{all}$**: the god of the sea<br>**GPT-3**: Poseidon’s symbols are a trident and a dolphin. The trident is a three-pronged spear, and the dolphin is a creature that is half fish and half mammal.<br>**GENREAD+ InstructGPT**: The trident symbolizes his power over the sea, the horse symbolizes his power over horses, the bull symbolizes his power over earthquakes, and the dolphin symbolizes his power over the sea creatures. He was also associated with the colors blue and green, which symbolize the sea and the land, respectively. |

Table 15: Prediction samples on a long-type question from the test set.