RobustQA: Benchmarking the Robustness of Domain Adaptation for Open-Domain Question Answering

Rujun Han  Peng Qi  Yuhao Zhang  Lan Liu  Juliette Burger
William Yang Wang  Zhiheng Huang  Bing Xiang  Dan Roth
AWS AI Labs
{rujunh, pengqi, yhzhang, liuall, burgerju}@amazon.com
{wyw, zhiheng, bxiang, drot}@amazon.com

Abstract

Open-domain question answering (ODQA) is a crucial task in natural language processing. A typical ODQA system relies on a retriever module to select relevant contexts from a large corpus for a downstream reading comprehension model. Existing ODQA datasets consist mainly of Wikipedia corpus, and are insufficient to study models’ generalizability across diverse domains, as models are trained and evaluated on the same genre of data. We propose RobustQA, a novel benchmark consisting of datasets from 8 different domains, which facilitates the evaluation of ODQA’s domain robustness. To build RobustQA, we annotate QA pairs in retrieval datasets with rigorous quality control. We further examine improving QA performances by incorporating unsupervised learning methods with target-domain corpus and adopting large generative language models. These methods can effectively improve model performances on RobustQA. However, experimental results demonstrate a significant gap from in-domain training, suggesting that RobustQA is a challenging benchmark to evaluate ODQA domain robustness.

1 Introduction

Open-domain question answering (ODQA) is a crucial and practical NLP task. Unlike traditional reading comprehension task where contexts are provided for a QA pair, in ODQA, a retriever needs to first extract relevant passages from a large amount of documents; then a QA model provides answers based on these passages. Due to the magnitude of the corpus, it is computationally prohibitive for a QA model to read through all documents. Therefore, ODQA becomes a popular research topic and is widely adopted in real-world applications.

Table 1: Examples in RobustQA. Highlighted text spans are the precise answers that our annotators need to identify. Different from NQ, our texts are diverse with more challenging question-answer pairs.

| FQA-Finance | Question: Why do investors buy stock that had appreciated?  
Document: Imagine how foolish the people that bought Apple at $100 must have felt. It was up tenfold for the $10 it traded at just years prior, how could it go any higher? Stocks have no memory. A stock’s earnings may grow and justify the new higher price people are willing to pay.  
When FB came public, I remarked how I’d analyze the price and felt it was overvalued until its earnings came up. Just because it’s gone down ever since, doesn’t make it a buy, yet. |
| LoTTE-Lifestyle | Question: What techniques, tricks or otherwise have you used to get upgrades on flights?  
Document: … but I think the best way to get upgraded is to fly a lot with the airline. Generally when the flight’s overbooked in one class, and they’re trying to pick which person to upgrade, frequent flyer status is the first metric they use. The higher your status, the higher up the list you go! Having a high status with a partner airline can work too, high tiers with a partner airline usually comes below the airline’s own frequent flyers, but above everyone else. Otherwise, if you’re flying on your own that’ll help in the event that there aren’t enough frequent flyer to upgrade to free the required number of seats! Offering to pay may be an option too - if they’re pretty full they may offer you a low price to upgrade. |

The practicality of ODQA necessitates the evaluation of systems’ out-of-domain (OOD) performances because a real-world system needs to be robust when confronting domain drift. Moreover, existing state-of-the-art (SOTA) ODQA systems (Karpukhin et al., 2020; Santhanam et al., 2022) are based on neural networks, which are known to overfit training data and suffer from degradation when domain changes. For example, Natural Questions (Kwiatkowski et al., 2019, NQ) is the most commonly used ODQA dataset, but a recent study shows that there is a significant amount of overlaps between its train and test sets. This partially explains why neural models trained only on NQ can struggle in unseen domains (Lewis et al., 2021).

However, evaluating OOD performances for ODQA is currently not feasible in the research community due to the lack of a public multi-domain benchmark. Existing popular ODQA datasets such as NQ and TriviaQA (Joshi et al., 2017) rely on Wikipedia or Web documents. Multi-domain evaluation datasets exist separately for each compo-
We briefly review ODQA in this section. Following the extractive QA set-up in the DPR paper (Karpukhin et al., 2020), we denote a collection of documents as $D$. We split each document $d^i \in D$ with a fixed length $N$ tokens and obtain a collection of $M$ ($\geq |D|$) passages denoted as $C = \{p_1, p_2, \ldots, p_m, \ldots, p_M\}$. Denoting a token as $w$, a passage can be defined as $p_m = \{w_{m,n} \in p_m, 0 \leq n < N \mid p_m \in C\}$. 

We denote a question as $q$ and a passage retriever as $R$. The task is to select $K$ most relevant passages for $q$ from $C$. Formally, $R(q, C) \rightarrow C_q$. Upon receiving $K$ passages $C_q$, a QA model predicts the most probable text span $A_q = \{w_{m,j1}, \ldots, w_{m,jk} \in p_m, p_m \in C_q\}$ that can answer the question. We also test generative models for our task, but it is crucial to note that we are still confined to the extractive QA setting as generated texts need to be evaluated against the ground-truth answers, which are contained in the contexts.

In real-world applications, a trained ODQA model may encounter domain changes including token distribution shifts in $C$, or type and length changes in questions and answers. Therefore, it is crucial to gauge domain robustness. However, a comprehensive evaluation benchmark does not exist, which we propose to address with RobustQA.

3 Data Creation

Existing public ODQA data mostly leverage Wikipedia (Kwiatkowski et al., 2019; Joshi et al., 2017) as a search corpus, and focus on factoid questions. To benchmark model robustness across a wider range of text genres and question types, we 1) annotate 6 datasets in finance, lifestyle, recreation, technology, science, and writing based on FiQA (Maia et al., 2018) and LoTTE (Santhanam et al., 2022); 2) adapt two public available ODQA datasets, SearchQA (Dunn et al., 2017) and BioASQ (Tsatsaronis et al., 2015). The newly annotated data consist of a significant portion of challenging reasoning type of questions that cannot be answered with entities or short phrases. Our data samples can be found in Table 1, Table 11-13, and Table 16 in the appendix.

3.1 Annotated Data

We describe the data annotation process for the new domains: finance, lifestyle, recreation, technology, science, and writing based on FiQA and LoTTE, both are IR-dataset with no precise answer spans annotated in the retrieved supporting documents.

As shown in Table 1, relevant documents are retrieved from corresponding IR systems. We present a question and its relevant documents to the annotators, and they need to identify up to 3 concise text.
<table>
<thead>
<tr>
<th>Domain</th>
<th>Label</th>
<th># Test Questions</th>
<th># Documents</th>
<th># Passages</th>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQ</td>
<td>[NQ]</td>
<td>3,610</td>
<td>-</td>
<td>21,015,324</td>
<td>NQ</td>
</tr>
<tr>
<td>RobustQA</td>
<td>Web-search</td>
<td>31,760</td>
<td>13,791,373</td>
<td>13,791,592</td>
<td>SearchQA</td>
</tr>
<tr>
<td></td>
<td>Biomedical</td>
<td>1,956</td>
<td>15,559,026</td>
<td>37,406,880</td>
<td>BioASQ</td>
</tr>
<tr>
<td></td>
<td>Finance</td>
<td>3,669</td>
<td>57,638</td>
<td>105,777</td>
<td>FiQA</td>
</tr>
<tr>
<td></td>
<td>Lifestyle</td>
<td>2,214</td>
<td>119,461</td>
<td>241,780</td>
<td>LoTTE</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>2,096</td>
<td>166,975</td>
<td>315,203</td>
<td>LoTTE</td>
</tr>
<tr>
<td></td>
<td>Technology</td>
<td>2,115</td>
<td>638,509</td>
<td>1,252,402</td>
<td>LoTTE</td>
</tr>
<tr>
<td></td>
<td>Science</td>
<td>1,426</td>
<td>1,694,164</td>
<td>3,063,916</td>
<td>LoTTE</td>
</tr>
<tr>
<td></td>
<td>Writing</td>
<td>2,696</td>
<td>199,994</td>
<td>347,322</td>
<td>LoTTE</td>
</tr>
</tbody>
</table>

Table 2: Data summary: NQ (top) v.s. RobustQA (bottom). # of Documents for NQ is missing because we directly use the passage split provided by Karpukhin et al. (2020). Passages consists of 100 continuous tokens at most from the original documents.

Figure 1: An illustration of our data quality control procedure, which starts by data experts sharing instructions and raw data to the annotators. Data Experts constantly audit the annotations and conduct a final validation to ensure data quality.

FiQA contains a task: “Opinion-based QA over financial data.” It aims at answering finance related questions from financial corpus such as microblogs, reports and news. However, the answers provided in the original dataset are documents instead of precise text spans. Besides, its test set does not include answer passages. Therefore, we use the original training set with all of their relevant passages to be examined by annotators. After filtering out samples with no precise answer spans, we obtain 3,669 questions.

LoTTE was proposed in the ColBERTv2 paper (Santhanam et al., 2022). Similar to FiQA, LoTTE consists of IR datasets across five domains: lifestyle, recreation, technology, writing and science, each can have potential answers coming from two resources: search and forum with dev and test splits using different text genres. We were able to annotate all data in the dev and test sets, but for our reported results, we only use the test split. After filtering out no-answer questions, we obtain 2,214, 2,096, 2,115, 1,426, 2,696 questions for lifestyle, recreation, technology, science and writing, respectively. We reserve the dev data for future model development purpose, and their data statistics can be found in Appendix A.2.

3.2 Adapted Data

We select BioASQ and SearchQA to represent the biomedical and web-search domains as they are both high quality and large-scale ODQA datasets that do not rely on Wikipedia corpus and follow the extractive QA format. Note that there are some existing abstractive ODQA datasets, such as TweetQA (Xiong et al., 2019) and AmazonQA (Gupta et al., 2019), but they do not fit into the extractive ODQA framework as we describe in Sec. 2.

For both datasets below, we use only the test split of the latest version for evaluation purpose.

SearchQA contains questions crawled from Jeopardy!, and leverages Google search engine to retrieve text snippets as relevant contexts. After filtering out questions without answers, we obtain 31,760 questions.

BioASQ contains questions written by biomedical experts based on PubMed documents. The
Table 3: Data statistics. All numbers are average over the evaluation data.

ODQA task (Task 1b) consists of four types of questions: 1) Yes/No, (2) factoid, (3) list, and (4) summary. We consider only (2) and (3) as they are suitable to our extractive QA task. After discarding no-answer questions, we acquire 1,956 questions.

3.3 Quality Control

Fig. 1 illustrates our data annotation and quality control procedure, which starts with the data experts (including co-authors) sending the annotation guidelines and raw data to the annotators. Annotation guidelines can be found in Appendix A.1. Upon receiving annotated data from the annotators, the data experts will randomly select 10% of the annotations to audit. If the selected samples fall below the validity requirements of 90%, they will be sent back to the annotators for re-annotation. The process repeats until randomly selected samples pass the 90% threshold.

To ensure annotator quality, we hire professional data providers. Based on the information shared, the data provider team consists of more than 20 data professionals, and each of them is paid >15 U.S. dollars per hour. The data expert team consists of co-authors and 10 additional internal data professionals. Note that due to the high costs of hiring data professionals, we were not able to provide multiple annotations per sample, and thus were not able to explicitly compute inter-annotator agreements. However, the 90% passing threshold we install in the process guarantees high annotation satisfaction rate, and thus ensures good data quality.

3.4 Data Statistics and Analysis

In this section, we compare data in different domains. Particularly, we want to highlight the drift of data distribution from NQ.

Passages. Following the DPR paper (Karpukhin et al., 2020) for passage pre-processing, we split documents into 100 maximum continuous tokens. Passage numbers for each domain can be found in Table 2. We observe that BioASQ (biomedical) and SearchQA (Web-search) contain the amount of passages that are in the same magnitude of NQ. The other newly annotated datasets all have relatively smaller collections of passages, which is common in many real-world applications.

Questions. Table 3 shows the length of questions, and Fig. 3 demonstrates the types of questions across different domains. We can see that BioASQ is relatively similar to NQ. They share similar question length and contain mostly factoid questions. However, other RobustQA data have longer questions compared with NQ, and they tend to ask reasoning type of questions such as “how to ...?” “how do ...?” and “why does ...?” (see details in Appendix A.4). Also, these questions ask long-tail topics, ones that might not be covered by an entity-centric knowledge base like Wikipedia (Santhanam et al., 2022).

\(^2\)Tokens are simply split by whitespace.
Answers. As mentioned above, except for SearchQA, RobustQA data consist of longer answers due to the nature of reasoning type of questions. Our questions also contain more individual answers, which is a consequence of compiling answers from multiple relevant passages during data creation. For example, if a question has two supporting passages, each containing three unique answers, the question will have six answers in total.

4 Passage Retrieval

In this section, we describe the retrievers adopted in this work. We benchmarked five leading retrievers on RobustQA.

**DPR** adopts a bi-encoder architecture to encode a pair of question and passage independently. The passage relevancy score is calculated using vector similarity measures such as dot product. We use the best checkpoint (trained on NQ) provided by the DPR paper (Karpukhin et al., 2020).

**BM25** is a widely used sparse retriever, which matches keywords efficiently with an inverted index and can be seen as representing the question and context as weighted, high-dimensional sparse vectors. We use the BM25 implementation provided by the BEIR paper (Thakur et al., 2021).

**BM25+CE** incorporates cross-encoder (CE) architecture as passage re-rankers after obtaining BM25 results. However, Reimers and Gurevych (2019) points out that the CE architecture is computationally expensive as it requires both the question and passage to be fed into a language model for encoding. For this reason, in the BEIR paper, the authors input only the top 100 ranked passages returned by BM25 into CE for re-ranking, and we follow this practice. The original CE model is trained on MS MARCO (Payal Bajaj, 2016), and we also re-train CE using NQ only from scratch, and denote this model as BM25+CE$_\text{NQ}$.

**ColBERTv2** (Santhanam et al., 2022). ColBERT was initially proposed by (Khattab and Zaharia, 2020) using late interaction to decompose relevance modeling into token-level computations, which improves the expressivity of the query-document matching, but increases the storage requirements drastically. ColBERTv2 alleviates this issue with a residual compression mechanism and improves the quality of the retriever by distilling from a cross-encoder with hard-negative mining. We use the best checkpoint (trained on MS MARCO) provided by the paper.

**Atlas** (Izacard et al., 2022b) jointly trains Contriever (Izacard et al., 2022a), a dense retriever with bi-encoder architecture and T5 (Raffel et al., 2019), a sequence-to-sequence language model as the reader. Since we adopt Atlas as one of the open-QA baseline models (Sec. 5), we report its retriever performances here.

5 Open-domain QA

To anchor our QA models against a widely tested baseline, we adopt the extractive QA model architecture used in the DPR paper. We further investigate whether we can improve extractive QA model’s OOD generalization by pretraining base models with unlabeled target domain corpus. As large language models (LLMs) are gaining research popularity nowadays, we also benchmark Flan-T5 (Chung et al., 2022) as a baseline. We also test the method that jointly trains LLMs with dense retriever (Izacard et al., 2022b), which is expected to have stronger performances than standalone QA.
models.

5.1 Extractive QA Model
For the baseline extractive QA model, we strictly follow the training objective of the DPR paper.\(^3\)

\[
P_{\text{start},i}(s) = \text{softmax}(P_i w_{\text{start}})_s
\]

\[
P_{\text{end},i}(t) = \text{softmax}(P_i w_{\text{end}})_t
\]

\[
P_{\text{selected}}(i) = \text{softmax}(P_i w_{\text{selected}})_i
\]

where \(P_i\) indicates the last encoded hidden layers of the i-th passage from BERT (Devlin et al., 2019). \(w_{\text{start}}, w_{\text{end}}, w_{\text{selected}}\) are learnable vectors. The training objectives consist of two scores: 1) span score of the s-th to t-th tokens are computed as \(P_{\text{start},i}(s) \times P_{\text{end},i}(s)\); 2) passage selection score is \(P_{\text{selected}}(i)\).

5.2 Pretraining with Target Corpus
Pre-training with target corpus/task has shown to help language models (LMs) adapt more effectively to unseen domains (Lee et al., 2020; Liu et al., 2020; Han et al., 2021; Garg et al., 2019; Zhou et al., 2021). Here, we are interested in the setting where only a small amount of passages is available for pre-training. Specifically, we conduct a second-step pre-training on BERT for the extractive QA model mentioned above. The target corpus has no QA annotations and consists of a small fraction of all available target domain text data. Finally, we fine-tune the target-pretrained LMs on in-domain data before testing it on the target-domain data in RobustQA.

Besides target-corpus pretraining, we also experimented with other unsupervised domain adaptation methods such as contrastive loss based on Long et al. (2022). Since its improvements are relatively marginal, we briefly describe it and report results in Appendix A.5.

5.3 Prompt Finetuning with LLMs
Recently, LLMs have shown incredible performances on a variety of NLP tasks. Here, we also test LLM’s ability on ODQA by finetuning one of the SOTA open-source LLMs, Flan-T5-xxl with 11B parameters, on the same open-domain NQ dataset used in training the extractive QA model. The prompt template used during finetuning and inference are shown in Table 4.

During finetuning, Instruction is fixed for all training samples. Question and Answer pairs are from NQ data. Passage 1-5 are retrieved by DPR.

Table 4: Prompt template for LLM finetuning and inference.

5.4 Joint Training LLM with Retriever
Joint training LLMs with dense retrievers has been shown to be an effective method for retrieval based tasks (Izacard and Grave, 2021; Lewis et al., 2020). One of the most recent efforts, Atlas (Izacard et al., 2022b) improves upon previous works by jointly pre-training T5-based models with Contriever on a large amount of corpus using various objective functions. Atlas achieves impressive zero/few-shot learning performances on open-domain QA. Therefore, we report its results to show a strong modeling baseline on RobustQA.

6 Experimental Setup

6.1 Passage Retrieval
As described in Sec. 4, there are 7 passage retrievers we compare. 1) DPR by (Karpukhin et al., 2020); 2) BM25 and 3) BM25 + CE by Thakur et al. (2021); 4) we also train CE on NQ data and denote this model as BM25 + CE\(_{NQ}\); 5) ColBERTv2 by Santhanam et al. (2022); 6) the contriever component of the Atlas-base finetuned on NQ; 7) the contriever component of Atlas-xxl finetuned on NQ (Izacard et al., 2022b).

As shown in Table 5, we use HIT@5 as the primary reporting metrics.\(^4\)

6.2 Open-domain QA
Following single-data setting of the QA model in the DPR paper, we use NQ as the in-domain training data with passages retrieved from the best DPR

\(^{3}\)https://github.com/facebookresearch/DPR

\(^{4}\)HIT@20 and HIT@100 results are shown in the appendix.
Table 5: Passage retrieval performance based on HIT@5. Atlas uses the Contriever (Izacard et al., 2022a) that has fixed model size for both Atlas-base and Atlas-xxl. “base” and “xxl” refer to the size of the reader model (T5). Neural retrievers’ model sizes and training data are summarized in Table 9 in the appendix.

For evaluation, we test the trained QA models on RobustQA with passages retrieved from ColBERTv2 as it provides the best retrieval results according to Table 5. Using DPR’s retrieved passages is a reasonable alternative as it may reduce the gap between the train and test time. As a benchmark paper, we leave more rigorous investigations of this option for future research efforts.

**Compared Models.** 1) FT: we finetune the extractive QA model on NQ data and test zero-shot on RobustQA. 2) PT → FT: before finetuning, we pretrain the language model, BERT-base (Devlin et al., 2019) on a small unlabeled OOD corpus. 3) Atlas-base and Atlas-xxl: jointly train the Contriever with T5-base (220M) and T5-xxl (11B) respectively as the reader. We use the provided Atlas checkpoints finetuned on NQ. 4) Flan-T5-xxl finetuned on NQ using the prompt specified in Sec. 5.

**Unsupervised Corpus.** Note that PT → FT method requires unsupervised source/target corpus. We construct them by randomly sample a subset of passages (Table 2) with no more than 20 million combined tokens (≈200K passages). We ensure positive contexts for the test questions are excluded.

### 7 Results and Analysis

In this section, we show our benchmark results on RobustQA for both passage retrieval and end-to-end question answering.

#### 7.1 Passage Retrieval

Table 5 shows the results for passage retrieval. Consistent with previous findings (Thakur et al., 2021), DPR achieves best in-domain performance on NQ, but generalizes poorly to RobustQA. On the contrary, BM25’s performances across all datasets are stable, and it outperforms DPR by 8.21% over HIT@5 on RobustQA.

We found that both BM25+CE and ColBERTv2 can work well for all domains. Their RobustQA average result outperforms BM25 by 13.56 and 14.73 percentage points, respectively per HIT@5. ColBERTv2 appears to be the most robust passage retriever according to our results. Table 14 and 15 in the appendix show ColBERTv2 gains wider margins against BM25+CE per the HIT@20 and HIT@100. Here, both CE and ColBERTv2 are trained on MS MARCO.

**Impact of Training Data.** As shown in Table 5, with the same base model, training the CE with NQ can improve its in-domain performances by 6.22 points, but hurt its results on RobustQA by 6.73 points. Since MS MARCO includes larger and potentially more diverse data than NQ, it suggests enhancing quantity and diversity of training data can be beneficial to improve domain robustness.

**Model Size.** As Table 9 in the appendix shows, all model sizes for neural models are comparable, except for ColBERTv2, which uses a much smaller model, but leverages distillation techniques to obtain knowledge from larger CE models. Nonetheless, its efficiency and remarkable performances make it an excellent retriever during inference.

**Joint Training.** Atlas’s retriever, Contriever uses the same base model as DPR. Atlas’ lower scores in NQ can be partially attributed to different training methods and negative sample selections. However,
<table>
<thead>
<tr>
<th>Method</th>
<th>NQ</th>
<th>RobustQA Average</th>
<th>[BI]</th>
<th>[SE]</th>
<th>[FI]</th>
<th>[L1]</th>
<th>[RE]</th>
<th>[TE]</th>
<th>[SC]</th>
<th>[WR]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>49.37</td>
<td>18.81</td>
<td>28.36</td>
<td>43.30</td>
<td>9.25</td>
<td>15.50</td>
<td>13.06</td>
<td>9.78</td>
<td>13.17</td>
<td>18.02</td>
</tr>
<tr>
<td>PT→FT</td>
<td>49.31</td>
<td>21.04</td>
<td>30.56</td>
<td>43.47</td>
<td>10.55</td>
<td>16.82</td>
<td>16.51</td>
<td>12.75</td>
<td>15.77</td>
<td>21.86</td>
</tr>
<tr>
<td>Atlas-base</td>
<td>51.87</td>
<td>28.29</td>
<td>39.55</td>
<td>59.20</td>
<td>15.60</td>
<td>23.84</td>
<td>22.76</td>
<td>19.81</td>
<td>18.25</td>
<td>27.34</td>
</tr>
<tr>
<td>Atlas-xxl</td>
<td>62.44*</td>
<td>37.31</td>
<td>45.31</td>
<td>78.51</td>
<td>21.34</td>
<td>32.24</td>
<td>29.08</td>
<td>25.94</td>
<td>27.28</td>
<td>38.79</td>
</tr>
<tr>
<td>Flan-T5-xxl</td>
<td>57.60</td>
<td>35.49</td>
<td>47.04</td>
<td>77.21</td>
<td>18.87</td>
<td>28.54</td>
<td>28.03</td>
<td>24.63</td>
<td>25.20</td>
<td>34.39</td>
</tr>
</tbody>
</table>

Table 6: End-to-end QA performance based on F\textsubscript{1} score. All readers are trained on NQ. Except for Atlas models, ColBERTv2 is used to retrieve up to 100 passages to be consumed by the reader during inference. *Atlas-xxl’s F\textsubscript{1} score on NQ is lower than the number reported in the original paper because we use DPR’s passage pool, which does not contain infobox data.

Atlas’ superior OOD performances show the effectiveness of joint pre-training of retriever and reader on large text corpus before fine-tuning on NQ. The signals from the QA models likely can help correct the errors in the retrieval stage, and the signals become stronger as we adopt larger language models, i.e., from “base” to “xxl.”

**NQ v.s. RobustQA.** We observe that the best RobustQA average is 9.45 percentage points below the best NQ HIT@5 (72.24%). Except for SearchQA and Writing and, which have the closest token distribution to NQ (Fig. 2), there are significant performance degradation on RobustQA datasets, suggesting our new benchmark provides a much more challenging contexts for passage retrieval compared with the commonly used Wikipedia corpus.

7.2 Open-domain QA

For end-to-end QA performances, Table 6 shows that simply applying an extractive QA model fine-tuned to NQ to RobustQA will result in a drastic performance drop of 30.56 percentage points per F\textsubscript{1} measure. Comparing and contrasting with the performance declines in the passage retrieval, it implies that domain drifts have strong impacts on both retriever and reader modules.

**Pretraining with Domain Corpus.** We observe that comparing with FT only, PT→FT can improve RobustQA by 2.23 percentage points. The best PT→FT F\textsubscript{1} score on RobustQA (21.04%) is still more than 28 points below its in-domain performance on NQ. These results again confirm that RobustQA is a challenging benchmark to work with, but point out a promising direction to leverage unlabeled corpus to help close the gap between in-domain and OOD data.

**Generative v.s. Extractive Approach.** Comparing to FT (on extractive readers), generative models Atlas-xxl and Flan-T5-xxl improve F\textsubscript{1} score on NQ by 13.07 and 8.23 percentage points, while gaining 18.5 and 16.68 points on RobustQA, respectively. These results demonstrate the superior performances of large language models. However, the in-domain and OOD gap is still wide, and LLMs may not be suitable for compute/latency-sensitive applications. Thus, we test a smaller generative model, Atlas-base whose reader has similar model size as the extractive QA model. We observe a lift of 9.48 points against FT, which suggests that generative approach can help extractive ODQA task.

**Analysis on challenging domains.** We observe in Table 6 that FiQA and Technology are the two most challenging domains, which can be particularly attributed to the statistical data differences (Sec. 3.4). Table 3 shows that FiQA has the longest answer spans (9.4 v.s. 2.3 for NQ). Since the QA model is trained on NQ to predict short spans, it is likely to have lowest token recall for FiQA, thus the lowest F\textsubscript{1} scores. On the other hand, Technology’s poor performances may be related to its largest token distribution drift from NQ as illustrated by Fig. 2. Moreover, Fig. 3 shows that Technology has the largest amount of reasoning type of questions, which indicates the largest question type drift. Both factors make it harder to adapt a model trained on NQ to Technology in a zero-shot manner.

**Error Analysis.** A common issue for ODQA is that when a retriever returns a mixture of relevant and irrelevant passages as QA inputs, the latter can mislead reader prediction by extracting answers from incorrect contexts. These issues can be potentially resolved by building stronger retrievers and leveraging retrieval results such as scores to help readers rank answer candidates. We leave this to
the future research efforts.

Here we focus on analyzing errors in reading comprehension only by selecting samples where a reader correctly picks relevant passages to extract answers. As Table 16 in the appendix shows, both PT→FT model and Atlas-xxl, when finetuned on NQ, tend to predict either entities or short phrases on FiQA, whereas the ground-truth answers tend to be longer and complete phrases that can fully answer the reasoning type of questions (Example 1 and 2 in Table 16) or complicated factoid questions (Example 3). This again suggests that training NQ alone may not be sufficient to solve RobustQA, and a more diverse ODQA dataset is crucial for training robust ODQA systems.

8 Related Work

Open-domain QA. ODQA has been studied by a large body of work (Chen et al., 2017; Kwiatkowski et al., 2019; Izacard and Grave, 2021). Among them, DPR (Karpukhin et al., 2020) proposes a competitive system by combining dense retrieval with a strong extractive reader (Devlin et al., 2019), which we adopt as the baseline. The key limitation of current ODQA work is that the evaluation is primarily done on datasets based with Wikipedia text (Rajpurkar et al., 2016; Kwiatkowski et al., 2019; Amouyal et al., 2019), which we address in this work.

Passage Retrieval. At the core of most ODQA systems is a passage retrieval system. While we only benchmark several strong baselines in this work, numerous other systems have been studied in the past (Thakur et al., 2021). These systems can be broadly divided into sparse retrievers (where the similarity between the query and a passage is calculated via inverted index), dense retrievers (where the similarity is calculated with dense vectors from neural encoders), or a combination of both. For sparse retrievers, apart from BM25, DeepCT (Dai and Callan, 2020), SPARTA (Zhao et al., 2021), docT5query (Nogueira and Lin) are good alternatives. For dense retrievers, ANCE (Xiong et al., 2021), TAS-B (Hofstätter et al., 2021) and Contriever (Izacard et al., 2022a) are some more recent developments and variations of DPR. In addition, it is a common practice to combine a retrieval module with a separate single-tower re-ranker model fine-tuned on retrieval datasets (Thakur et al., 2021).

Domain Robustness Benchmarks. Several benchmarks have been created for cross-domain evaluation of reading comprehension, MRQA (Fisch et al., 2019) or retrieval systems, BEIR (Thakur et al., 2021) and LoTTE (Santhanam et al., 2022). Asai et al. (2022) also provides various datasets for retrieval augmented NLP tasks, but most of them do not have a clean extractive ODQA format, which is the focus of this work. Outside of NLP, DomainBed (Gulrajani and Lopez-Paz, 2020), ESC (Gandhi et al., 2023) and DABS (Tamkin et al., 2021) are recent benchmarks for computer vision, speech recognition and self-supervised learning, respectively.

9 Conclusion

We propose RobustQA, a benchmark consisting of samples across 8 different domains that better evaluates ODQA systems’ robustness on domain adaptation. After adopting SOTA ODQA systems enhanced by unsupervised learning methods and LLMs, there still exists a significant performance gap between RobustQA and the commonly used NQ dataset, which suggests that RobustQA is a more reliable and challenging benchmark to evaluate ODQA systems’ cross-domain performances.

Limitations

We discuss some limitations of this work for future research efforts. The range of the domains could be more comprehensive to cover social media and law. The experiments can potentially cover more models. As we mention in Sec. 8, there are more comparable retrievers and QA readers. It would be useful in the future to benchmark more models on RobustQA. Finally, due to the complexity of the raw IR data, it is costly to collect our datasets. This is manifested by not only the monetary costs, but also the human efforts to create guidelines, to coach annotators, and to manually audit and validate annotations. In the future, it could be beneficial to leverage large language models with context learning to assist human labors.

Ethics Statement

The authors of this paper are committed to conducting research ethically. Data used in this work have been collected from public sources and used in accordance with all applicable laws and regulations. This work uses language models, for which the
risks and potential harms are discussed in numerous previous works (Bender et al., 2021; Weidinger et al., 2021). The authors strive to ensure that the research and its results do not cause harm.

References


Rodrigo Nogueira and Jimmy Lin. From doc2query to docTTTTTquery.


Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via


A Appendix

A.1 Annotation Guidelines

We summarize our annotation guidelines here. Before judging the answer validity and conciseness, annotators should first determine if a passage can be used to answer a given question. Though the passages associated with the question should have labels of "relevant" in the original IR data, we found they are not always appropriate for our annotations because 1) annotation errors in the original dataset, i.e., passages are in fact not relevant; 2) intent of a question is ambiguous, so it is hard to determine if an associated passage is relevant or not; 3) we couldn’t find precise answer spans in a passage to answer the question. In any of these cases, we instruct the annotators to discard the passage.

Answer Validity

- The answer span is exactly the same as in the passage. Typos/misspelling are acceptable as long as they are not preventing us from understanding the answer.
- The answer span does not combine two different excerpts/parts of the passage to create only one answer span.
- The answer span does not include leading or trailing punctuation marks, unless they are a part of the answer span. (e.g. Yahoo!)
- The answer span does not contain a URL and/or is not a URL link itself.
- The answer span does not correspond to the entire passage.

Answer Conciseness

- The answer span should be as short as possible, while still conveying the wanted meaning.
- The answer span does not contain more than 16 words, though adding 1-2 words to make the span complete is allowed (should be considered very carefully).
- The answer span does not contain unnecessary rhetorical expressions such as subject/object, time and location around a core concept.
Table 7: Data summary for the ODQA annotations in the dev split of LoTTE after filtering out no-answer questions.

<table>
<thead>
<tr>
<th>Label</th>
<th># Questions</th>
<th># Documents</th>
<th># Passages</th>
</tr>
</thead>
<tbody>
<tr>
<td>LI</td>
<td>2.151</td>
<td>268,893</td>
<td>597,729</td>
</tr>
<tr>
<td>RE</td>
<td>2.325</td>
<td>263,025</td>
<td>731,124</td>
</tr>
<tr>
<td>TE</td>
<td>2.223</td>
<td>1,000,000</td>
<td>1,707,346</td>
</tr>
<tr>
<td>SC</td>
<td>2.137</td>
<td>343,642</td>
<td>854,756</td>
</tr>
<tr>
<td>WR</td>
<td>1.972</td>
<td>277,072</td>
<td>713,692</td>
</tr>
</tbody>
</table>

The answer span does not contain unnecessary the explanation of a core concept.

A.2 Additional RobustQA Annotations

Table 7 shows the data statistics for our ODQA annotations in the dev split of LoTTE. We did not benchmark these data, but will release data for future model development purpose.

A.3 Annotation Examples

Table 11 - 13 show more examples of our annotated data. Additional FiQA examples can be found in the error analysis - Table 16.

A.4 Question Types

In Figure 3, we categorize questions into two types: factoid and reasoning. Here, we entail the string matching used for the categorization. Though these rules are not perfect, they largely capture question types based on our careful manual examination.

Factoid
- first word is “what”
- any match to the phrase in this list: [whats, what’s, when, who, how many, how much, how long, how old, how far, how often, list the, where, which]

Reasoning
- any match to the phrase in this list: [why, because, how is, how are, how’s, how am, how was, how were, how did, how does, how do, how will, how have, how has, how to, how can]

Questions do not have these patterns are either binary question, or questions that do not follow typical inquiry pattern. For example, SearchQA contains more than 90% of this type of questions. Here is an example: “magic-making mickey mouse movie of 1940” – the typical question should be “what is the magic-making mickey mouse movie of 1940.” We do not handle these questions separately and leave the impact on domain-robustness of this statement-type questions to the future research.

A.5 Domain Classification-based Contrastive Learning

Our contrastive learning method is based on Long et al. (2022). The core idea is that while training models to perform well on the key task (e.g. ODQA), we also encourage models to learn domain-invariant representations using unlabeled source/target corpus. In this way, the model can potentially be effective at adapting to the same task in a different domain.

This goal is accomplished by first introducing a domain classifier $f(x, l)$ where $x$ is the text representations and $l$ is the text’s domain label. Then we attempt to learn the optimal perturbation $\delta$ to $x$ that brings the representation close to domain-invariant. Adopting the virtual adversarial loss formulation (Miyato et al., 2017),

$$L_{\text{domain}} = \min_\theta \sum_{(x, l)} (\mathcal{L}(f(x), l; \theta)) + \alpha_{\text{adv}} \max_{||\delta|| \leq \epsilon} \mathcal{L}(f(x + \delta), l; \theta))$$

where $\theta$ is the classifier parameter to be learned and $\alpha_{\text{adv}}$ controls the balance between optimizing the classifier and learning the perturbation. $\epsilon$ is the $l_2$ norm boundary for $\delta$, which can be learned through Projected Gradient Descent (PGD) (Madry et al., 2018; Zhu et al., 2020) with an additional assumption that the loss function is locally linear.

To approximate the perturbation $\theta$, we can run one iteration of the following algorithm,

$$\theta_{t+1} = \prod_{||\delta|| \leq \epsilon} (\delta_t + \eta g_t^{\text{adv}}(\delta_t) ||g_t^{\text{adv}}(\delta_t)||)$$

where $\prod_{||\delta|| \leq \epsilon}$ performs a projection onto the $\epsilon$-ball, and $\eta$ is the step size. Finally, the contrastive loss is computed as,

$$L_{\text{contrastive}} = \log \frac{\exp(s(z, z')/\tau)}{\sum_k \mathbb{1}_{k\neq i} \exp(s(z, z_k)/\tau)}$$

where $z = g(f(x))$ and $z' = g(f(x + \delta))$. $g$ is a projection function and $s$ represents the cosine similarity between two vectors. $\tau$ is a constant temperature parameter. The indicator function $\mathbb{1}_{k\neq i}$ excludes the target sample $i$ from the normalization term. $N$ is the batch size. Intuitively, this
The contrastive loss function brings the original representation $z$ closer to its domain-invariant representation $z'$ while pushing it away from the representations from other negative samples in the batch. Combining all, the final training objective is,

$$\mathcal{L} = \mathcal{L}_{\text{task}} + w_d \mathcal{L}_{\text{domain}} + w_c \mathcal{L}_{\text{contrastive}}$$

where $w_d, w_c$ are weights for component losses.

### A.6 Additional Implementation Details

#### Computing Resources.
We run all experiments on 8 Nvidia A100 GPUs. A model is typically trained for 10-15 epochs, which takes 10 to 20 hours to complete depending on the model size and algorithm complexity. All neural models are implemented by PyTorch and Huggingface libraries. All benchmarked models are publicly available. Please refer to their code repo for software details.

#### Hyper-parameters.
All hyper-parameters for the QA models are selected based on the best NQ dev set performances. We use EM for top 50 passages to be consistent with the DPR paper. For pretraining method, we save checkpoints for every 8000 training steps, and report the best model per criteria mentioned above. Similarly, for FT + CL experiments, the best hyper-parameters are picked based on the same criteria.

We tuned 4 hyper-parameters as shown in Table 8. The search ranges are,

- $w_d$: {0.01, 0.05, 0.1}
- $w_c$: {0.01, 0.05, 0.1}
- $\alpha_{\text{adv}}$: {0.1, 1.0}
- $\epsilon$: {2.0, 5.0}

<table>
<thead>
<tr>
<th>Data</th>
<th>$w_d$</th>
<th>$w_c$</th>
<th>$\alpha_{\text{adv}}$</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchQA</td>
<td>0.1</td>
<td>0.01</td>
<td>1.0</td>
<td>5.0</td>
</tr>
<tr>
<td>BioASQ</td>
<td>0.1</td>
<td>0.05</td>
<td>0.1</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Table 8: Best hyper-parameters for each RobustQA dataset for the FT + CL experiments.

### A.7 Passage Retriever Models

Table 9 shows base models, number of parameters and training data for passage retrievers.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train Data</th>
<th>Retriever Model</th>
<th>$P$ Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>NQ</td>
<td>BERT-base</td>
<td>110M</td>
</tr>
<tr>
<td>BM25+CE</td>
<td>MS MARCO</td>
<td>ELECTRA-base</td>
<td>110M</td>
</tr>
<tr>
<td>BM25+CENQ</td>
<td>NQ</td>
<td>ELECTRA-base</td>
<td>110M</td>
</tr>
<tr>
<td>ColBERTv2</td>
<td>MS MARCO</td>
<td>MiniLM</td>
<td>22M</td>
</tr>
<tr>
<td>Atlas-base</td>
<td>NQ</td>
<td>BERT-base</td>
<td>110M</td>
</tr>
<tr>
<td>Atlas-large</td>
<td>NQ</td>
<td>BERT-base</td>
<td>110M</td>
</tr>
</tbody>
</table>

Table 9: Base model and training data used for the compared (neural) passage retrievers. Again, both Atlas models use Contriever as retrievers.

### A.8 Additional Passage Retrieval Results

Table 10 shows additional passage retrieval results per HIT@20 and HIT@100.

<table>
<thead>
<tr>
<th>Method</th>
<th>NQ</th>
<th>BI</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>49.37</td>
<td>28.36</td>
<td>43.30</td>
</tr>
<tr>
<td>FT + CL</td>
<td>49.23</td>
<td>29.58</td>
<td>44.88</td>
</tr>
</tbody>
</table>

Table 10: Comparison between finetuning directly on NQ (FT) v.s. finetuning with contrastive loss (FT + CL) on End-to-end QA performance. All numbers are $F_1$ scores.

### A.9 Contrastive Loss Results

As shown in Table 10, after finetuning using contrastive loss on NQ data, we observe noticeable gains against FT only method on BioASQ and SearchQA datasets with very little trade-off on in-domain NQ result. However, this method doesn’t work very well on other RobustQA datasets, and requires a lot of hyperparameter tuning. We leave more rigorous study on this method to future research efforts.

### A.10 Error Analysis

Table 16 shows error cases on FiQA.
Lifestyle: Example 1

**Question:** Why is international first class much more expensive than international economy class?

**Passage:** Your question is (I think): why does (US) domestic first class exceed the cost of economy by 50%, whereas international first class is many times the cost of economy. The answer to that is that the comparative services are vastly different. In (US) domestic first class you get a little more legroom, and a seat roughly 50% more wide. In international first class, you will often get far more than a fully reclining seat - a private bed is not unknown. Comparing the plane’s floor-plate take is a crude measure, but it would not surprise me if a first class seat took 10 times the amount of space as an economy seat on an international flight. Look at the floor plans here (image not copied for copyright reasons) and you can see the six seats in the back half of the first class cabin of a 747 take about the same space as 4 rows (40 seats) of economy.

Lifestyle: Example 2

**Question:** Why did the metro bus stop at each railway crossing, despite no warning indicating a train was coming?

**Passage:** Other people have cited the relevant laws. The laws exist because warning signals sometimes malfunction. It’s probably fine for a passenger car to take that risk, but a bus carrying passengers has a higher standard of care to adhere to.

Recreation: Example 1

**Question:** Why did you have to blow into an NES cartridge to make it work?

**Passage:** My brother and I did this all the time with our old nes. Beyond blowing out the dust there seemed to be some connection with moisture. If I blew it out and then breathed on it to make it damp it seemed to work better. The question about how everyone knew to do it? I agree with the instinct theory. My brother and I worked out this system by ourselves. You look at it, and for some reason the first thing you think of is either to stick your finger in it or to blow in it, LOL.

Recreation: Example 2

**Question:** How do you prevent Sims from aging?

**Passage:** Sort of. There is the Ambrosia, which you can make yourself, although it is a little tricky. Here is a quote from another helpful player There is the Ambrosia recipe that you can buy from the book store. It’s a level 10 cooking recipe and when you make it it resets your current life meter. So if you’re 20 days into the Adult stage of life, it sets it back to 0. Also, if you have a sim ghost eat it, they will be brought back to life. Of course, to be able to make it requires some items that are not easily come by. Noamely you need a life fruit which can only be grown from a special seed found somewhere on the ground and grown by a high level gardener, and you need a Deathfish, which I believe can only be found in the graveyard pond after midnight and probably only fished by a higher level fisherman. I hope that helps.

Technology: Example 1

**Question:** How can I disown a running process and associate it to a new screen shell?

**Passage:** Using GNU screen is your best bet. Start screen running when you first login - I run screen -D -R, run your command, and either disconnect or suspend it with CTRL-Z and then disconnect from screen by pressing CTRL-A then D. When you login to the machine again, reconnect by running screen -D -R. You will be in the same shell as before. You can run jobs to see the suspended process if you did so, and run %1 (or the respective job #) to foreground it again.

Technology: Example 2

**Question:** How can I reduce a videos size with ffmpeg?

**Passage:** This answer was written in 2009. Since 2013 a video format much better than H.264 is widely available, namely H.265 (better in that it compresses more for the same quality, or gives higher quality for the same size). To use it, replace the libx264 codec with libx265, and push the compression lever further by increasing the CRF value, Â¡ add, say, 4 or 6, since a reasonable range for H.265 may be 24 to 30. Note that lower CRF values correspond to higher bitrates, and hence produce higher quality videos. ffmpeg -i input.mp4 -vcodec libx265 -crf 28 output.mp4 To see this technique applied using the older H.264 format, see this answer, quoted below for convenience: Calculate the bitrate you need by dividing your target size (in bits) by the video length (in seconds). For example for a target size of 1 GB (one gigabyte, which is 8 gigabits) and 10 000 seconds of video (2 h 46 min 40 s), use a bitrate of 800 000 bit/s (800 kbit/s): ffmpeg -i input.mp4 -b 800k output.mp4 Additional options that might be worth considering is setting the Constant Rate Factor, which lowers the average bit rate, but retains better quality...
Table 14: Model Performance (HIT@20) for passage retrieval.

<table>
<thead>
<tr>
<th>Method</th>
<th>NQ</th>
<th>RobustQA Average</th>
<th>[BI]</th>
<th>[SE]</th>
<th>[FI]</th>
<th>[LI]</th>
<th>[RE]</th>
<th>[TE]</th>
<th>[SC]</th>
<th>[WR]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>81.33</td>
<td>54.75</td>
<td>44.30</td>
<td>86.60</td>
<td>42.85</td>
<td>65.00</td>
<td>53.91</td>
<td>36.41</td>
<td>41.09</td>
<td>67.87</td>
</tr>
<tr>
<td>BM25</td>
<td>62.83</td>
<td>65.72</td>
<td>66.10</td>
<td>91.27</td>
<td>45.74</td>
<td>65.46</td>
<td>53.24</td>
<td>51.96</td>
<td>78.85</td>
<td></td>
</tr>
<tr>
<td>BM25+CE</td>
<td>72.02</td>
<td>68.41</td>
<td>70.04</td>
<td>93.23</td>
<td>62.01</td>
<td>79.22</td>
<td>74.38</td>
<td>63.78</td>
<td>61.57</td>
<td>81.94</td>
</tr>
<tr>
<td>BM25+CE NQ</td>
<td>74.16</td>
<td>65.05</td>
<td>70.55</td>
<td>93.80</td>
<td>57.32</td>
<td>74.89</td>
<td>70.18</td>
<td>57.92</td>
<td>55.61</td>
<td>78.86</td>
</tr>
<tr>
<td>ColBERTv2</td>
<td>78.64</td>
<td>76.65</td>
<td>71.32</td>
<td>93.22</td>
<td>65.58</td>
<td>82.93</td>
<td>78.24</td>
<td>67.61</td>
<td>68.65</td>
<td>85.65</td>
</tr>
</tbody>
</table>

Table 15: Model Performance (HIT@100) for passage retrieval. Atlas results are omitted here because we used the officially recommended 40 as the number of retrieved contexts, and thus won’t have HIT@100 results.

<table>
<thead>
<tr>
<th>Method</th>
<th>NQ</th>
<th>RobustQA Average</th>
<th>[BI]</th>
<th>[SE]</th>
<th>[FI]</th>
<th>[LI]</th>
<th>[RE]</th>
<th>[TE]</th>
<th>[SC]</th>
<th>[WR]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>87.29</td>
<td>68.84</td>
<td>56.90</td>
<td>94.00</td>
<td>60.72</td>
<td>78.87</td>
<td>68.99</td>
<td>53.57</td>
<td>57.50</td>
<td>80.19</td>
</tr>
<tr>
<td>BM25</td>
<td>78.06</td>
<td>78.30</td>
<td>75.00</td>
<td>97.74</td>
<td>67.54</td>
<td>82.57</td>
<td>78.29</td>
<td>70.97</td>
<td>68.09</td>
<td>86.16</td>
</tr>
<tr>
<td>BM25+CE</td>
<td>78.06</td>
<td>78.30</td>
<td>75.00</td>
<td>97.74</td>
<td>67.54</td>
<td>82.57</td>
<td>78.29</td>
<td>70.97</td>
<td>68.09</td>
<td>86.16</td>
</tr>
<tr>
<td>BM25+CE NQ</td>
<td>78.06</td>
<td>78.30</td>
<td>75.00</td>
<td>97.74</td>
<td>67.54</td>
<td>82.57</td>
<td>78.29</td>
<td>70.97</td>
<td>68.09</td>
<td>86.16</td>
</tr>
<tr>
<td>ColBERTv2</td>
<td>85.87</td>
<td>85.69</td>
<td>77.51</td>
<td>98.43</td>
<td>82.98</td>
<td>91.28</td>
<td>87.36</td>
<td>80.99</td>
<td>79.38</td>
<td>92.32</td>
</tr>
</tbody>
</table>

Example 1

**Question:** explain the hsi - why do markets sometimes appear in sync and other times not?

**Passage:** why do markets sometimes appear in sync, but during other times, not so much By "markets" I’m assuming you mean equity indices such as the HSI. Financial products fluctuate with respect to the supply/demand of the traders. There’s been a large increase in the number of hedge funds, prop desks who trade relative values between financial products, that partially explains why these products seem to pick up "sync" when they get out of line for a while.

**Answer (PT→FT):** during other times

**Answer (Atlas-xxl):** supply/demand

Example 2

**Question:** how to determine how much to charge your business for rent (in your house)?

**Passage:** Your best approach is to assess rent levels in your local area for offices of a similar size. You need to take into account all the usuals - amenities, parking, etc, just as if your home-office was provided by a third-party. Get your $/sq ft and work out the monthly amount. With this figure, you need to then work out what % of it you can charge. If the space is used exclusively for the business, charge 100%. If it’s used about half the time, charge 50%, etc.

**Answer (PT→FT):** 100%

**Answer (Atlas-xxl):** assess rent levels

Example 3

**Question:** what is kirchstrasse on my statement bill?

**Passage:** POS stands for Point of Sale (like a specific store location) which indicates that the purchase occurred by using your debit card, but it can also be the on-line transaction done via 3-D Secure. Checking with bank, they said that Kirchstrasse transaction could be related to direct marketing subscription service ordered on-line. Investigating further what I’ve found these kind of transactions are performed by 2BuySafe company registered at Kirchstrasse in Liechtenstein with went through the MultiCards on-line cashier which can be used for paying different variety of services (e.g. in this case it was polish on-line storage service called Chomikuj)

**Answer (PT→FT):** pos stands for point of sale

**Answer (Atlas-xxl):** 2BuySafe

Table 16: Examples of errors in FiQA made by the QA models. Here we focus on reading comprehension component by examining examples with passages containing both gold answers and the predictions.