Prompting-based Synthetic Data Generation for Few-Shot Question Answering

Maximilian Schmidt[1], Andrea Bartezzaghi[2], Ngoc Thang Vu[1]
[1] University of Stuttgart, [2]IBM Research Zurich
[1]{maximilian.schmidt, thang.vu}@ims.uni-stuttgart.de, [2]abt@zurich.ibm.com

Abstract

Although language models (LMs) have boosted the performance of Question Answering, they still need plenty of data. Data annotation, in contrast, is a time-consuming process. This especially applies to Question Answering, where possibly large documents have to be parsed and annotated with questions and their corresponding answers. Furthermore, Question Answering models often only work well for the domain they were trained on. Since annotation is costly, we argue that domain-agnostic knowledge from LMs, such as linguistic understanding, is sufficient to create a well-curated dataset. With this motivation, we show that using large language models can improve Question Answering performance on various datasets in the few-shot setting compared to state-of-the-art approaches. For this, we perform data generation leveraging the Prompting framework, suggesting that language models contain valuable task-agnostic knowledge that can be used beyond the common pre-training/fine-tuning scheme. As a result, we consistently outperform previous approaches on few-shot Question Answering.

Keywords: Question Answering, Natural Language Generation, Named Entity Recognition

1. Introduction

Machine Reading Question Answering (MRQA) is an important task in Natural Language Processing and allows to easily access information by providing answers to specific questions. While there are several subtasks related to MRQA such as open-domain, binary/multiple choice, conversational or generative QA, we focus on extractive QA in this work. In extractive QA, the goal is to find the answer to a question by extracting it from a given context. MRQA has also raised attention in the community as a surrogate where other tasks are cast as question answering problems, thereby enabling a broad range of applications. This includes, for example, Named Entity Recognition (NER, Li et al., 2020; Arora and Park, 2023), entity relation extraction (Levy et al., 2017; Li et al., 2019; Zhang et al., 2022) and slot filling (Gao et al., 2019).

Pre-training language models (LMs) on Natural Language Understanding (NLU) objectives such as Masked Language Modeling (MLM, Devlin et al., 2019) led to strong MRQA models (Rajpurkar et al., 2016) and even surpasses human level[1]. Since the downstream task uses a different objective function, fine-tuning pre-trained LMs (PLMs) is necessary to adapt to the task. Arguably, this misalignment leads to poor results if labeled data for the downstream task is scarce. However, annotating data for MRQA is time-consuming and expensive. Additionally, few-shot MRQA poses an interesting challenge, especially for specific domains, where high effort is needed to annotate data or domain experts are missing. Also, there is still a gap in the performance of few-shot models when compared to the high-resource setting. For example, the best model on TextbookQA (Kembhavi et al., 2017) using 16 labeled samples is currently reported in literature to reach at most 49.9% F1 (Castel et al., 2022).

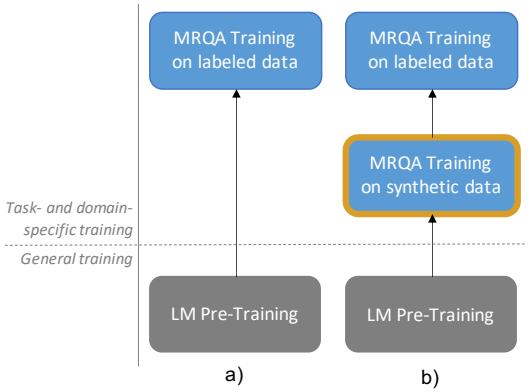

Figure 1: Comparison of a) common approaches, e.g., Prompting, for MRQA and b) our approach adding synthetic task- and domain-specific data without the need of additional labeled data.

To deal with the low-resource MRQA setting, previous work has proposed to generate synthetic data to augment the training set (e.g., Alberti et al., 2019; Puri et al., 2020; Shakeri et al., 2020, 2021). With a similar objective, PLMs have been employed for other tasks in Natural Language Processing (e.g., Anaby-Tavor et al., 2020; Schick and Schütze, 2021). However, MRQA is more challenging when generating synthetic data: We cannot simply gen-

[1]e.g., on the SQuAD (Rajpurkar et al., 2016) benchmark: https://rajpurkar.github.io/SQuAD-explorer/

LREC-COLING 2024, pages 13168–13178
20-25 May, 2024.

erate text given a label, but have to come up with a sample's input as well as its label in the form of a question and its answer. Additionally, both the answer and the question are mutually dependent. In this work, we explore an approach that enables this; a high-level overview is given in figure 1.

More precisely, we aim to answer the following research questions:

RQ1: How can we use LMs to generate synthetic data for improving the few-shot MRQA task?

RQ1.1: To what extend can synthetic data improve performance?

RQ1.2: How does the answer selection affect the performance?

RQ1.3: Do we need labeled data at all or can LMs generate helpful data out of the box?

RQ2: How does the proposed approach generalize to other domains?

To achieve this, we believe that there is a more effective way to employ LMs: We propose to use the linguistic knowledge encoded in these models to generate synthetic data for the target domain to counteract the effect of data scarcity. For this, we use the LM's ability to generate questions conditioned on the input, and we argue that this can easily be carried out in any target domain since we build on unsupervised PLMs.

In summary, our contributions are as follows: 1) We propose an approach generating valuable labeled data for the target domain by using the linguistic knowledge encoded in LMs[2]; 2) we improve the performance of few-shot QA for many dataset sizes across various domains to further bridge the performance gap between the few-shot and the full data setting; 3) we demonstrate the high quality of questions generated by our approach in a user study.

While introducing a new, strong approach outperforming many state-of-the-art approaches in few-shot MRQA, our model even outperforms the full data setting of TextbookQA with 64% F1 with only 64 labeled samples.

2. Related Work

In this section, we review existing work related to our setting, i.e., few-shot, as well as applications for Prompting.

Low-Resource MRQA

Although there is no agreement among research on how much data the few-shot setting may comprise (Hedderich et al., 2021), the objective when dealing with few-shot settings is to reduce the cost- and time-expensive annotation effort which, depending on the domain, may require domain expert knowledge. Furthermore, for some domains, it is a challenge by itself to find experts or other resources (Otegi et al., 2020).

Several approaches deal with settings where the amount of data is constrained. Many of them adopt the unsupervised pre-training technique. While Ram et al. (2021) come up with a QA-specific pre-training objective, many others adapt the LM to the target domain using its pre-training objective (Zhang et al., 2020; Nishida et al., 2020; Pergola et al., 2021; Chen et al., 2023).

Although self-supervised LM pre-training is also considered data augmentation, there also exist several approaches that deal with task- and domain-specific data augmentation. For example, training instances can be manipulated by performing operations on the input keeping the labels the same (Zhang et al., 2020), or new labeled data can by synthesized (Alberti et al., 2019; Shakeri et al., 2020, 2021).

When humans are actively engaged in model development, Active Learning becomes possible (Settles, 2012; Schmidt et al., 2022).

(L)LMs, Prompting

As mentioned in the beginning, Prompting (Liu et al., 2021) aims to improve downstream tasks by aligning the pre-training objective with the downstream objective. There are also several works that employ Prompting for the few-shot setting (Liu et al., 2021; Schick and Schütze, 2021). Empirically, Prompting alleviates the need for labeled data (Radford et al., 2019; Brown et al., 2020) and also boosts QA performance in the few-shot setting (Chada and Natarajan, 2021; Castel et al., 2022). For example, Chada and Natarajan (2021) and Wang et al. (2022a) align the MRQA task with the pre-training objective by casting the context-question-answer tuples as answer reconstruction, where the answer is decoded using an LM from the context and the question. Castel et al. (2022) adapt this method for extractive MRQA by only decoding from the given context, i.e., computing probabilities on all possible spans from the context.

Several approaches aim to improve Prompting: For example, soft tokens (Liu et al., 2022; Li and Liang, 2021; Zhong et al., 2021) can allow the model to better adapt to downstream tasks and input data, and demonstration learning (Gao et al., 2021) can help the model perform well in few-shot settings.

Prompting can also further be used for data augmentation (Anaby-Tavor et al., 2020).

[2]Our source code is available publicly here: https://github.com/mxschmdt/mrqa-prompting-gen

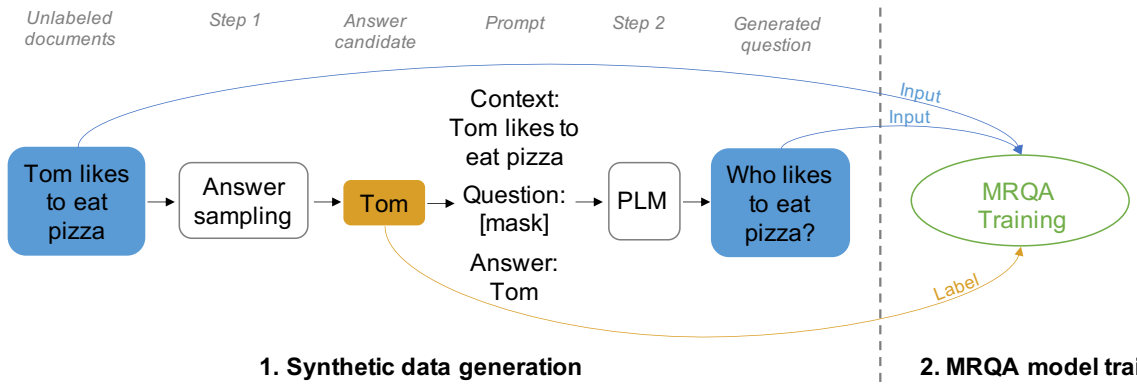

Figure 2: An example of our data generation pipeline: We first sample answer candidates (using NER) and then prompt a PLM to generate a question conditioned on context and answer (1). The generated question-answer pair is then used with the initial context to train an MRQA model (2). We afterwards perform additional training on labeled data if available.

3. Method

Here, we give an overview of the problem and describe our approach in detail.

Formally, MRQA is defined as given context c and question q, the goal is to predict the answer $a = f(c, q)$. We further focus on extractive MRQA, that is, a is a single contiguous span within c.

Next, we introduce our approach leveraging the Prompting framework. The high-level idea of our approach is composed of two steps: First, we sample answer candidates from a document. In a second step, we then query a pre-trained LM for generating questions using the document and the previously sampled answers. An overview of our data generation pipeline is given in figure 2. In the following we describe each step of o ur method in detail.

3.1. Answer Sampling

For sampling answer candidates, we apply NER[3] to the context c and the resulting entities are used as textual answers a_c with spans s (a tuple of character start and end indices). We chose this technique because it is a simple resource-sparse approach and does not need to have knowledge on the domain's topic (i.e., any English NER model is sufficient for our domains). Furthermore, NER is feasible in many languages and the datasets on which we evaluate our method, the few-shot MRQA benchmark, work with such style. As a result, it can be applied to any domain for which a NER model exists in the given language. Note that NER does not necessarily rely on many labeled samples or labeled data at all (e.g., rule-based approaches, using weak supervision (Lison et al., 2020) or Prompting (Liu et al., 2022; Ma et al., 2022)).

[3] In fact, we also select values similar in style to names, e.g., periods.

3.2. Question Generation

In Prompting, a LM takes a text input and, depending on the training objective, predicts the next token (as in the case of language modeling) or one or multiple masked tokens. For example, T5 (Raffel et al., 2020) is an encoder-decoder model which is fed a text input possibly containing multiple masked tokens. For each masked token, one or more tokens can appear in the output prefixed by a *sentinel* token, marking the masked token in the input to which the following tokens belong. For our purpose we only use a single mask in the input.

For generating questions, we transform the sample inputs into prompts for the LM. For this purpose we apply a template, thus replacing placeholders (marked starting with < and ending with >) with the actual values from the sample. Since we aim at generating a question given a context and a sampled answer candidate, the template is formulated to include the context and the answer, and the expected output is the question. Therefore the question is formally defined as

$$p(q|c, a_c) = \sum_{t=1}^{T} \log p(q_t|q_{<t}, c, a_c). \quad (1)$$

At training time, we use the original objective used for pre-training the underlying language model in order to model the question's probability by only computing the loss on the question q in the output. In our preliminary experiments, we have found that it is crucial to rely on sequence-to-sequence models, as these allow to condition the output not only on previous tokens but on the whole sequence. We believe this is due to the more natural formulation where a question occurs before its answer in a sentence. In contrast, using a left-to-right decoding

model only, the input would have to be formulated such that the generated question can be answered by the previously (i.e., left to the question) given answer. Obviously, this not only yields a longer prompt but also increases its complexity. Furthermore, we make use of soft tokens in the input. That is, all textual tokens from the template are trained in addition to the remaining model weights and we initialize them with the corresponding weights from the pre-trained word embedding.

For generation, we decode the question q token-by-token and filtering is applied. We do this for the following reasons: 1) the generated question could be noisy, e.g., not a valid question, and 2) the generated question may not be helpful for the Question Answering downstream task, possibly being underspecified. For example, the generated question may have – in addition to the provided answer – several other correct answers in a given context. As filtering technique we apply a two-step process. First, we discard generated samples based on rule-based filtering. We then apply consistency filtering (Alberti et al., 2019; Anaby-Tavor et al., 2020) for which a model similar to the final MRQA model is employed. Generated samples are discarded depending on the F1 score of the predicted answer (where the reference is the generated answer) using the MRQA model. We do not use iterative consistency filtering (Wang et al., 2022b) as this involves MRQA model re-training in each iteration, resulting in increased hyperparameter tuning complexity and resource drain.

4. Experimental Setup

Here, we first introduce the few-shot SQuAD dataset followed by implementational details of our approach and the description of baselines we consider.

4.1. Few-shot Setting

We perform experiments on several datasets in order to compare with existing approaches. For this, we rely on the subsampled train and test splits from the Few-Shot MRQA benchmark[4] from Ram et al. (2021) which are based on the preprocessed versions from the MRQA Shared Task 2019[5] (Fisch et al., 2019). More specific, this includes as domains SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017), NaturalQuestionsShort (NQ) (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), HotpotQA (Yang et al., 2018), BioASQ (Tsatsaronis et al., 2015) and TextbookQA (Kembhavi et al., 2017) and we evaluate our approach using the splits with 16, 32, 64 and 128 training samples. We perform our experiments for RQ1 on SQuAD while the remaining datasets are used to test the generalization capability of our approach (RQ2).

4.2. Data Generation

For generating questions, we have tested various models, templates and pre-processing strategies in preliminary experiments. We found T5 (Raffel et al., 2020) to perform best. Instead of the original v1 model, which makes use of labeled data during pre-training thus violating our few-shot setting, we employ the v1.1 model in its large variant (∼800M parameters). Decoder-only models like GPT-2 (Radford et al., 2019) performed worse as mentioned in section 3.2. In the templates, we considered case sensitivity as well as different wordings. As a result of manual investigation, *context:* ⟨*context*⟩ *question:* ⟨*mask*⟩ *answer:* ⟨*answer*⟩.[6] turned out to work well for our purpose and is similar to the findings of Castel et al. (2022).

4.2.1. Training of Question Generation Model

For training the data generation model, similar to Castel et al. (2022) we create an academic development dataset to cater to the few-shot setting where having a separate development split leads to bad generalizability due to its small size. Therefore, we tune the learning rate and the number of training steps on a validation set of 2048 samples from SQuAD's training data, and choose the set of hyperparameters that has the best normalized performance across all few-shot sizes as described in Castel et al. (2022). As a result, the question generation model was trained for 130 training steps with a batch size of 32 using a linear learning rate of 1e-4 with the Adafactor optimizer. Furthermore, the soft tokens add 8192 weights for the question generation model. We additionally chunk the provided contexts using a stride of 100 tokens so that at most 450 tokens of the context are included in a single input to allow for sufficient space for the question. Since chunking can create instances where the answer is not part of the context, we drop these as we cannot expect them to yield a semantically correct question.

4.2.2. Synthetic Data Generation

For generating synthetic data, in case of SQuAD, TriviaQA, NQ, NewsQA, SearchQA and HotpotQA, we use the documents from the training corpus. Since BioASQ and TextbookQA both comprise rather few documents, we collect abstracts from

[4] https://github.com/oriram/splinter#downloading-few-shot-mrqa-splits

[5] https://github.com/mrqa/MRQA-Shared-Task-2019

[6] ⟨mask⟩is replaced by the model-specific mask token.

PubMed[7] and lessons from CK-12[8], respectively, for the purpose of generating data. We then apply stanza's NER[9] using all its entity types[10] in order to sample answers from these documents. Afterwards, similar to training time, we apply chunking with a stride of 100 tokens to feed the documents into the model (by realizing the template) whereby we again only keep instances where the answer is contained in the context for the same reason as above. In a subsequent step, the generated question is greedily decoded with a beam size of 5, top-k sampling with k equal to 20 and nucleus sampling (Holtzman et al., 2020) keeping tokens comprising 95% of probability mass in each step.

The language models' special tokens are stripped in a subsequent step. Since we only allow one mask in the input (for the question), we make sure that only the output tokens corresponding to this mask are used.

For the rule-based filtering in a subsequent step, we randomly select 1,000,000 samples and discard generated samples where the answer is contained in the question, or the question is only containing meaningless words or is empty. Afterwards, we apply consistency filtering discarding generated samples with an F1 score of less than 80% using a Prompting-based MRQA model which is trained similar to the one described in the next subsection.

4.3. MRQA Model

For the final step of our approach, we train an MRQA model using synthetic and available labeled data as shown in figure 2. Since a Prompting-based approach turned out to perform better than a span extraction head on top of a Transformer-based (Vaswani et al., 2017) encoder model, we also use T5 v1.1 (large) with the Prompting framework as our MRQA model. Additionally, we compared T5 v1.1 pre-trained with and without recurring span selection (RSS)[11] (Castel et al., 2022) on the few-shot MRQA benchmark (see below in section §4.4) and found that MRQA performance is generally improved if RSS is used. Therefore we use this model as basis for our MRQA models. As template, we use *context:* ⟨*context*⟩ *question:* ⟨*question*⟩ *answer:* ⟨*mask*⟩., and again use soft tokens (accounting for 9216 weights) which are optimized in addition to the full model during training. The MRQA model is first trained on synthetic data for 1 epoch or at least 500 steps. In a subsequent step, we further train on the annotated data from the few-shot splits. For this, we use the hyperparameters reported in Castel et al. (2022), that is a constant learning rate of 5e-5 for 512 training steps using the Adafactor optimizer (Shazeer and Stern, 2018) with a batch size of 32 and a dropout of 0.1.

We report the mean and standard deviation over 5 MRQA model runs while training the data generation model only once to save computation resources.

4.4. Comparison Models

We compare our approach to several recent and well performing MRQA models which we describe in the following.

Splinter This is the model proposed by Ram et al. (2021) using an RSS pre-training phase which is fine-tuned on the few-shot MRQA datasets. We show the results as reported by the authors for the base model.

FewshotBARTL FewshotBARTL is the best performing model reported in FewshotQA (Chada and Natarajan, 2021). This is a Prompting-based MRQA model using BART (Lewis et al., 2019).

Prompting Castel et al. (2022) reported results of a Prompting-based MRQA model similar to FewshotBARTL but using T5 v1.1.

Prompting+RSS This model is similar to the Prompting model above, but with additional pre-training using RSS. Since Castel et al. (2022) only report results on SQuAD, we consider this model only for RQ1. For RQ2, we evaluate a re-implemented version of this model (see next model).

Prompting+RSS Re-Impl To account for implementational differences and to be able to evaluate the Prompting model with RSS on the full few-shot MRQA benchmark, we also consider a reimplementation of the Prompting+RSS model where we also directly perform MRQA via Prompting. Additionally, we fine-tune soft tokens in the input initialized with weights from the embedding using the template. To this end, we transform the sample into a prompt such that the pre-trained model answers the question using the given context. This model is equal to the MRQA model we use in our approach, i.e. using the same template and the same hyperparameters.

Gotta This model proposed by Chen et al. (2023) is similar to Prompting with additional pre-training on entity-aware masks.

[7] https://pubmed.ncbi.nlm.nih.gov/

[8] https://www.ck12.org

[9] https://stanfordnlp.github.io/stanza/ner.html

[10] https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf p.21f

[11] https://huggingface.co/tau/t5-v1_1-large-rss

Model	0	16	32	64	128
Our approach	**85.5**	**86.4±0.6**	**88.3±0.4**	87.7±0.4	89.3±0.6
Prompting+RSS Re-Impl	71.5	84.0	86.8	86.8	88.8
Prompting+RSS (Castel et al., 2022)	71.4	85.6	86.7	**87.9**	**89.4**
Prompting (Castel et al., 2022)	60.0	82.6	85.2	86.7	89.0
Gotta (Chen et al., 2023)	-	74.6±1.9	76.0±2.0	78.9±10.5	80.8±1.7
PMR (large) (Xu et al., 2022)	17.2	60.3±4.0	70.0±3.2	76.6±1.9	81.7±1.2
FewshotBARTL (Chada and Natarajan, 2021)	-	68.9±2.7	72.3±1.0	73.6±1.9	79.4±1.5
Splinter (base) (Ram et al., 2021)	-	54.6±6.4	59.2±2.1	65.2±1.4	72.7±1.0
Roberta (base) (Ram et al., 2021)	-	7.7±4.3	18.2±5.1	28.4±1.7	43.0±7.1

Table 1: F1 score on SQuAD in the zero- and few-shot setting (16, 32, 64 & 128 samples) for several existing models as well as our proposed data generation technique: Our approach yields a competitive result across all dataset sizes, but the settings with 16 and 32 labeled samples profit the most. Compared to our re-implementation of the Prompting+RSS model, our approach always performs better. Therefore, differences in implementation might result in slightly different numbers. Best model per dataset size marked **bold**.

Model	0	16	32	64	128
sampled answer	85.5	86.4	88.3	87.7	89.3
gold answers	87.3	87.6	89.5	88.3	90.3
synthetic data only	87.3	90.0	91.0	90.7	91.3

Table 2: F1 score of an MRQA model using generated data from our approach on SQuAD in the zero- and few-shot setting (16, 32, 64 & 128 samples) as well as comparing gold answers with sampled answers.

PMR PMR (Xu et al., 2022) employs pre-training on automatically generated data in MRQA style and has a dedicated MRQA fine-tuning stage where the structure of inputs and outputs are similar.

Roberta We also show results reported by Ram et al. (2021) for a model following the standard pre-training/fine-tuning paradigm using Roberta (base) (Liu et al., 2019) with a span extraction head.

5. Results and Discussion

In order to judge the performance of our approach, we report the F1 score on the tested datasets for our approach as well as for the models we compare with. We now examine our research questions using the reported results.

5.1. RQ1: Synthetic Data Generation using LMs

First, we answer the nether research questions in order to answer RQ1. For this, we only evaluate on SQuAD.

RQ1.1: Synthetic Data for MRQA In general, as reported in table 1, our proposed method outperforms many existing approaches on SQuAD on all sizes although it does not perform best with 64 and 128 samples but is very close. Also, there is a trend that more data improves our data generation approach although there is a fluctuation which we trace back to difficulties in training LMs for the MRQA task on little data in the final step as observed.

RQ1.2: Answer Selection For this research question, we compare the performance of generated data by our approach when using answers sampled using NER and when using the gold answers for SQuAD in table 2. We can observe that generated data with sampled answers by NER performs in terms of F1 score on the MRQA model on average only 1.3% worse than data generated using the gold answers. Therefore the chosen answer sampling strategy can be a good replacement for answers as in the case of SQuAD. Since we observed suboptimal training performance of the MRQA model on labeled data in the final step, we additionally report MRQA performance on synthetic data only (i.e., before fine-tuning on labeled data in a final step). This shows that generated data can even perform better if care is taken when integrating the labeled samples into the eventual MRQA model.

RQ1.3: Zero-shot Performance As reported in table 1, without any labeled data our approach gets an F1 score of 85.5%. Although the performance increases if labeled data is added, this clearly outlines that there are strong zero-shot capabilities by employing data generation for MRQA. Therefore, we can conclude that the LM has learned during pre-training the relationship between questions and answers to an extent that is useful for SQuAD.

In summary, as an answer to RQ1, our

Model	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	BioASQ	TextbookQA	Mean
16 Samples								
Our approach	**76.4±0.5**	**68.5±0.6**	**51.4±0.8**	71.1±1.3	**72.4±0.6**	72.3±1.5	60.1±2.4	**67.5**
Prompting+RSS Re-Impl	76.1	67.0	48.3	**71.9**	71.3	73.0	**60.4**	66.9
Prompting (Castel et al., 2022)	74.8	64.4	44.7	64.1	66.3	**74.7**	49.9	62.7
Gotta (Chen et al., 2023)	63.3±8.0	58.9±1.9	47.3±2.5	56.8±3.9	59.8±2.1	66.1±3.1	38.5±5.3	55.8
PMR (large) (Xu et al., 2022)	56.2±3.1	43.6±1.7	30.1±3.7	58.2±5.0	46.1±4.7	54.2±3.4	31.0±1.8	45.6
FewshotBARTL (Chada and Natarajan, 2021)	65.2±1.8	60.4±2.0	48.4±2.2	47.8±5.4	58.0±1.8	63.0±1.1	37.7±3.7	54.4
Splinter (base) (Ram et al., 2021)	18.9±4.1	27.4±4.6	20.8±2.7	26.3±3.9	24.0±5.0	28.2±4.9	19.4±4.6	23.6
Roberta (base) (Ram et al., 2021)	7.5±4.4	17.3±3.3	1.4±0.8	6.9±2.7	10.5±2.5	16.7±7.1	3.3±2.1	9.1
32 Samples								
Our approach	**76.8±0.5**	**68.5±0.8**	50.6±0.8	**72.9±0.8**	**73.4±0.6**	74.5±0.8	**61.0±1.8**	**68.2**
Prompting+RSS Re-Impl	75.6	64.0	49.0	71.1	71.7	73.1	**61.0**	66.5
Prompting (Castel et al., 2022)	74.8	66.7	48.8	66.2	70.3	**76.8**	51.2	65.0
Gotta (Chen et al., 2023)	61.9±4.8	59.8±2.4	**51.2±1.5**	63.1±3.1	62.7±1.2	69.5±1.0	46.3±3.7	59.2
PMR (large) (Xu et al., 2022)	66.3±2.5	48.5±3.5	36.6±2.1	64.8±2.2	52.9±2.5	62.9±2.4	36.4±3.2	52.6
FewshotBARTL (Chada and Natarajan, 2021)	65.1±1.2	61.5±1.7	51.7±1.7	58.3±1.5	60.4±0.2	67.8±1.0	37.7±9.8	57.5
Splinter (base) (Ram et al., 2021)	28.9±3.1	33.6±2.4	27.5±3.2	34.8±1.8	34.7±3.9	36.5±3.2	27.6±4.3	31.9
Roberta (base) (Ram et al., 2021)	10.5±1.8	22.9±0.7	3.2±1.7	13.5±1.8	10.4±1.9	23.3±6.6	4.3±0.9	12.6
64 Samples								
Our approach	76.2±0.5	**70.4±0.4**	**56.4±0.7**	**75.0±1.4**	**74.7±0.2**	76.8±0.5	**64.0±1.0**	**70.5**
Prompting+RSS Re-Impl	**76.3**	68.7	52.9	72.5	73.4	78.5	61.8	69.2
Prompting (Castel et al., 2022)	75.3	68.5	49.9	71.7	73.1	**80.4**	55.6	67.8
Gotta (Chen et al., 2023)	59.6±11.9	63.6±11.0	54.3±13.0	66.3±12.5	64.3±11.7	73.2±11.5	51.2±12.8	61.8
PMR (large) (Xu et al., 2022)	67.5±1.7	53.4±2.3	46.8±2.6	69.3±2.4	61.7±2.1	71.5±1.8	43.4±3.6	59.1
FewshotBARTL (Chada and Natarajan, 2021)	64.6±1.4	63.0±2.1	53.5±0.9	65.5±2.4	62.9±1.6	73.9±0.8	45.0±1.7	61.2
Splinter (base) (Ram et al., 2021)	35.5±3.7	38.2±2.3	37.4±1.2	39.8±3.6	45.4±2.3	49.5±3.6	35.9±3.1	40.2
Roberta (base) (Ram et al., 2021)	12.5±1.4	24.2±1.0	4.6±2.8	19.8±2.4	15.0±3.9	34.0±1.8	5.4±1.1	16.5
128 Samples								
Our approach	**77.4±0.3**	**73.0±0.4**	**57.4±0.7**	**78.6±0.5**	**76.5±0.2**	82.4±0.3	**63.5±1.6**	**72.7**
Prompting+RSS Re-Impl	77.1	71.3	55.1	75.9	75.9	81.8	62.4	71.4
Prompting (Castel et al., 2022)	76.7	69.9	51.8	73.4	74.6	**85.2**	58.0	69.9
Gotta (Chen et al., 2023)	60.0±3.6	64.9±1.2	**57.4±1.2**	69.8±1.5	66.7±1.8	78.6±2.1	53.3±1.7	64.4
PMR (large) (Xu et al., 2022)	70.3±0.5	57.4±2.6	52.3±1.4	70.0±1.1	65.9±1.0	78.8±0.5	45.1±1.2	62.8
FewshotBARTL (Chada and Natarajan, 2021)	65.8±0.9	64.3±1.3	57.0±0.9	67.7±1.0	75.1±1.5	75.0±1.5	48.4±2.7	64.8
Splinter (base) (Ram et al., 2021)	44.7±3.9	46.3±0.8	43.5±1.3	47.2±3.5	54.7±1.4	63.2±4.1	42.6±2.5	48.9
Roberta (base) (Ram et al., 2021)	19.1±2.9	30.1±1.9	16.7±3.8	27.8±2.5	27.3±3.9	46.1±1.4	8.2±1.1	25.0
Full Dataset								
Splinter (base) (Ram et al., 2021)	76.5	81.0	71.3	83.0	80.7	91.0[1]	54.5[1]	76.9
Roberta (base) (Ram et al., 2021)	74.0	79.6	69.8	81.5	78.7	84.1[1]	35.8[1]	71.9

Table 3: Results of several models as evaluated by means of the F1 score for the few-shot MRQA benchmark excluding SQuAD. Our approach performs best across various dataset domains and sizes, and even outperforms the full data setting for TriviaQA and TextbookQA. Best model per dataset domain and size marked **bold**. [1] 1024 training samples

Prompting-based data generation approach efficiently employs LMs for MRQA increasing performance compared to existing work. The proposed method proves to be competitive on SQuAD and performs well especially without training data, establishing a new state of the art.

5.2. RQ2: Domain Generalization

To answer the second research question, we report results on the few-shot MRQA benchmark excluding SQuAD in table 3 (we additionally show the results of the best performing approaches for the mean of all datasets including SQuAD and all dataset sizes in figure 3). For NQ, HotpotQA and TextbookQA we consistently rank first across all dataset sizes. Largest absolute increases of F1 score can be observed on NQ with 32 samples (1.8%), SearchQA with 32, 64 and 128 samples (1.8%, 2.5% and 2.7%, respectively), HotpotQA with 32 samples (1.7%), and TextbookQA with 64 samples (2.2%). Also, we see that in general the performance increases with more labeled data, although this behavior is not consistent in the case of TriviaQA, NewsQA and TextbookQA.

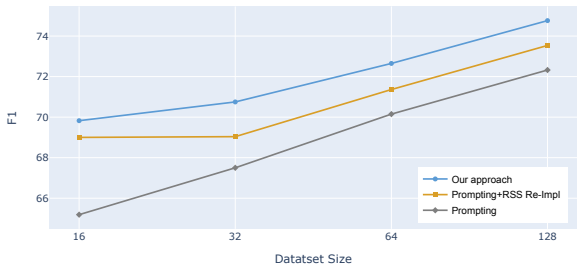

Figure 3: MRQA performance (F1) as a function of dataset sizes for the best performing approaches on the mean of all datasets in the few-shot MRQA benchmark.

Interestingly, on all sizes of BioASQ, our approach performs worse than directly using a Prompting model. Since the same applies to the Prompting approach using RSS, we assume that the MRQA model in our approach also suffers from the RSS pre-training although we cannot find rea-

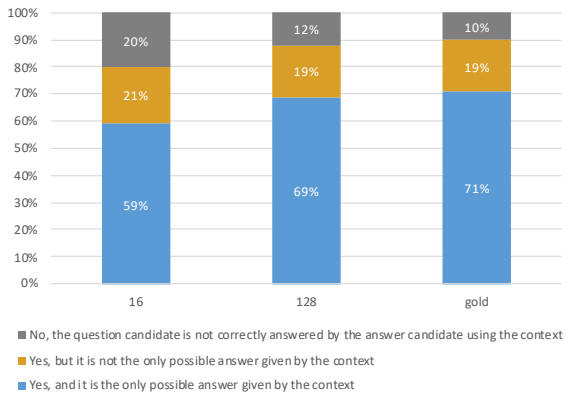

Figure 4: For the NewsQA dataset, 100 question-answer pairs were quality-assessed by humans (question: "Is the question candidate correctly answered by the answer candidate?") in each setting (generated data taking 16 and 128 samples into account as well as labeled (gold) data).

sons for RSS performing worse in this domain.

Finally, we note that we saw quite high fluctuations between model training runs in the few-shot setting. We assume this is owed to suboptimal hyperparameters which do not generalize well across domains and to too few samples. Training on 128 samples or less can easily lead to overfitting resulting in loss of generalization. Therefore, with a better incorporation of few labeled data into the models, we believe that the MRQA performance of a Prompting-based data generation approach can further be increased.

For answering research question RQ2, we can conclude that our data generation approach also generalizes to other domains as demonstrated by the few-shot MRQA benchmark. To analyze the benefit of our approach, we further investigate the quality of the generated question-answer pairs.

6. Analysis

In order to assess the data quality of the generated questions and answers a user study was taken. A total of 30 people, which were recruited via the Prolific platform, took part in the study. In order to achieve a high significance, the selection of participants was restricted by the following screening: The participants must have a bachelor's degree or higher, speak English as their primary language, and have a 100% approval rate with Prolific. The aim of the study is to find out whether the generated data from our approach provide a comparable data quality as labeled data, are correct question-answer pairs with respect to the context, and can be improved quality-wise with more labeled samples.

In order to carry out the analysis, the study participants were randomly given 10 samples per participant. In these, the context, the question candidate as well as the answer candidate was provided. In total 300 question-answer pairs were individually assessed by humans with regard to their quality. For this purpose, one question was asked per shown sample about the data quality[12], which made it possible to distinguish between correct answers, partially correct answers (in the case of several answers possible from the context) as well as incorrect answers. In the introduction it was explicitly pointed out that only the given context may be considered for answering the question candidate.

The dataset used was NewsQA, which has the lowest F1 compared to the other datasets used (cf. table 3) and the quality of the generated question-answer pairs is thus tested on a comparatively difficult domain. To achieve the above study objectives, 3 different settings were chosen, each given to 10 participants: generated data using 16 and 128 samples as well as gold data.

The results of the study are shown in figure 4. First, we can observe that with 128 labeled samples, generated data is comparable to the quality of labeled data. Although with only 16 samples the majority of generated question-answer pairs is of high quality (59%) despite the extremely small effort needed by humans, this lacks behind 10 points in absolute percentage when compared to the data quality generated using 128 samples. Therefore, labeling 128 samples can be sufficient for our approach for the NewsQA dataset to get a similar quality of question-answer pairs compared to human annotated data that is more complex and costly.

7. Conclusion

In summary, we introduced a new approach for MRQA that makes use of the linguistic knowledge encoded in LMs. To this end, we proposed to generate synthetic question-answer pairs for MRQA and run several experiments to test the performance of our approach in the zero- and few-shot setting, thereby also showing its generalizability. As a result, we have shown that LMs can be more effectively used, and find that our approach outperforms many state-of-the-art approaches for the MRQA task. Furthermore, in some settings, synthetic data is even on par with human annotated data. However, the performance heavily depends on the domain under consideration, with the highest absolute increase of performance for the most difficult domain. Finally, we demonstrated in a user study that it is possible with only taking into account 128

[12]The participants were asked to answer the following question: "Is the question candidate correctly answered by the answer candidate?"

human annotated samples to generate question-answer pairs which are comparable to human annotated data in terms of quality. We believe that there are many more ways to effectively use LMs and hope that our work will be an incentive to explore other possibilities.

8. Future Work

Although we have shown that our approach using NER sampled answers performs comparably well, other methods are worth to be explored too. For example, leveraging LLMs to also generate the answer is interesting, but poses additional challenges for extractive MRQA. They struggle in providing the start and end indices of answers if they are also used for selecting the answers, which renders some model architectures invalid. We therefore believe that more investigation is necessary to enable more effective and more efficient use of LLMs. For the generation, feedback (for example provided by humans) could be included to continuously improve the quality of synthetic data.

Regarding the MRQA model, other methods to incorporate synthetic data should be taken into account too. For example, adopting in-context learning for extractive MRQA is a highly interesting direction but out of scope of this work.

Furthermore, since our approach performs well for SQuAD in the zero-shot setting as well, it should further be investigated, also for other domains.

9. Ethical Considerations

Regarding our user study, participants were acquired using the Prolific platform. Prior to obtaining consent, we provided detailed instructions and descriptions of how their answers are processed, and that their participation is voluntary. We also ensured that payment was not below Prolific's recommendation for the participants. Regarding privacy, we did not collect any personal or identifying data, or data other than for the question mentioned in this work.

Acknowledgements

This work is supported by IBM Research AI through the IBM AI Horizons Network.

10. Bibliographical References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do Not Have Enough Data? Deep Learning to the Rescue! *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7383–7390.

Jatin Arora and Youngja Park. 2023. Split-NER: Named Entity Recognition via Two Question-Answering-based Classifications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 416–426, Toronto, Canada. 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.

Or Castel, Ori Ram, Avia Efrat, and Omer Levy. 2022. How Optimal is Greedy Decoding for Extractive Question Answering?

Rakesh Chada and Pradeep Natarajan. 2021. FewshotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models.

Xiusi Chen, Yu Zhang, Jinliang Deng, Jyun-Yu Jiang, and Wei Wang. 2023. Gotta: Generative Few-shot Question Answering by Prompt-based Cloze Data Augmentation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog State Tracking: A Neural Reading Comprehension Approach.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are You Smarter Than a Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-Shot Relation Extraction via Reading Comprehension.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A Unified MRC Framework for Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Pierre Lison, Aliaksandr Hubin, Jeremy Barnes, and Samia Touileb. 2020. Named Entity Recognition without Labelled Data: A Weak Supervision Approach.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022. Low-Resource NER by Data Augmentation With Prompting. volume 5, pages 4252–4258.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.

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.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free Prompt Tuning for Few-shot NER. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Kosuke Nishida, Kyosuke Nishida, Itsumi Saito, Hisako Asano, and Junji Tomita. 2020. Unsupervised Domain Adaptation of Language Models for Reading Comprehension.

Arantxa Otegi, Aitor Agirre, Jon Ander Campos, Aitor Soroa, and Eneko Agirre. 2020. Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 436–442, Marseille, France. European Language Resources Association.

Gabriele Pergola, Elena Kochkina, Lin Gui, Maria Liakata, and Yulan He. 2021. Boosting Low-Resource Biomedical QA via Entity-Aware Masking Strategies.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training Question Answering Models From Synthetic Data.

Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *undefined*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena,

Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-Shot Question Answering by Pretraining Span Selection.

Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference.

Maximilian Schmidt, Andrea Bartezzaghi, Jasmina Bogojeska, A. Cristiano I. Malossi, and Thang Vu. 2022. Improving Low-Resource Question Answering using Active Learning in Multiple Stages.

B. Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning Series. Morgan & Claypool.

Siamak Shakeri, Noah Constant, Mihir Sanjay Kale, and Linting Xue. 2021. Towards Zero-Shot Multilingual Synthetic Question and Answer Generation for Cross-Lingual Reading Comprehension.

Siamak Shakeri, Cicero Nogueira dos Santos, Henry Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artiéres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need.

Jianing Wang, Chengyu Wang, Minghui Qiu, Qiuhui Shi, Hongbin Wang, Jun Huang, and Ming Gao. 2022a. KECP: Knowledge Enhanced Contrastive Prompting for Few-shot Extractive Question Answering.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022b. PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks.

Weiwen Xu, Xin Li, Wenxuan Zhang, Meng Zhou, Lidong Bing, Wai Lam, and Luo Si. 2022. From Clozing to Comprehending: Retrofitting Pre-trained Language Model to Pre-trained Machine Reader.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering.

Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avirup Sil, and Todd Ward. 2020. Multi-Stage Pre-training for Low-Resource Domain Adaptation.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022. EntQA: Entity Linking as Question Answering.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual Probing Is [MASK]: Learning vs. Learning to Recall.