MinPrompt: Graph-based Minimal Prompt Data Augmentation for Few-shot Question Answering

Anonymous ACL submission

Abstract

Recent advances in few-shot question answering (QA) mostly rely on the power of pre-trained large language models (LLMs) and fine-tuning in specific settings. Although the pre-training stage has already equipped LLMs with powerful reasoning capabilities, LLMs still need to be fine-tuned to adapt to specific domains to achieve the best results. In this paper, we propose to select the most informative data for fine-tuning, thereby improving the efficiency of the fine-tuning process with comparative or even better accuracy on the open-domain QA task. We present MinPrompt, a minimal data augmentation framework for open-domain QA based on an approximate graph algorithm and unsupervised question generation. We transform the raw text into a graph structure to build connections between different factual sentences, then apply graph algorithms to identify the minimal set of sentences needed to cover the most information in the raw text. We then generate QA pairs based on the identified sentence subset and train the model on the selected sentences to obtain the final model. Empirical results on several benchmark datasets and theoretical analysis show that MinPrompt is able to achieve comparable or better results than baselines with a high degree of efficiency, bringing consistent improvements in F-1 scores.

1 Introduction

Question answering (QA) provides accurate responses to a series of questions based on given narrative contexts. Its diverse applications extend to areas such as chatbots (Yang et al., 2019), dialogue systems (Burtsev et al., 2018), and instant information retrieval (Esteva et al., 2021), making it a key pursuit in the field of natural language processing (NLP). Supervised learning has traditionally been the approach for developing efficient QA systems that deliver commendable results. However, this method is intrinsically restricted by its reliance on a large set of annotated QA training examples, which becomes problematic due to the substantial cost associated with acquiring expert-level annotations.

Our research focuses on the few-shot QA task, an effort to address the QA challenge with the presence of only a limited number of training examples. The prevalent approaches under the few-shot setting either introduce a new task and pre-train an extensive language model from scratch (Ram et al., 2021), or they fine-tune an already pre-trained model on the given training examples (Chada and Natarajan, 2021). The fine-tuning stage is crucial in the sense that it stimulates the power of the LLMs obtained during the pre-training stage and makes the model align with the input/output distribution of a certain domain or dataset. However, with an increasing data size for fine-tuning, the training duration increases accordingly, which is undesirable, especially when the model size is also large (OpenAI, 2023). As such, the importance of minimal data augmentation cannot be understated. The fine-tuning data, often a limited resource in our consideration (up to 128 shots), is directly used to adjust the parameters of a pre-trained model to enhance performance on the downstream task. The data is usually labeled by domain experts and thus could be time-consuming to obtain in large quantities. On the other hand, augmented data represents a broader dataset, generated in an unsupervised manner by converting statements into question-answer pairs. In QA tasks, it is vital for a model to be exposed to a diverse range of questions, answers, and contexts to develop a robust understanding of the language and the task at hand. However, not all parts of the training data hold equal relevance or significance for the model's learning process. Some parts may contain more valuable information or more complex language structures that the model needs to understand to improve its performance. Consequently, identifying and augmenting

these critical portions of the training data could substantially enhance the model's capacity to answer questions accurately and comprehensively.

To address the above challenges, we present MinPrompt, which consists of the following three modules: (1) A **sentence graph construction module** that leverages sentence graph representation to structurize the raw text. Each node in the graph symbolizes a sentence, while edges illustrate the shared entities between sentences. This sentence graph effectively encapsulates the complex interconnections between various textual elements; (2) A **data selection module** that features an approximate minimal dominating set algorithm. The algorithm is applied to the sentence graph to identify the smallest set of sentences to cover all shared entities. This module ensures efficient use of computational resources, reduces the risk of overfitting, and enhances the model's generalization ability, resulting in an overall improvement in QA performance; and (3) A **question generation module** that transforms the selected plain factual sentences into QA pairs. The synthesized QA pairs are further turned into prompts, providing a condensed, yet comprehensive representation of the text. The generated prompts serve as high-quality, information-rich training instances for the QA model. This model trained on the compact and meaningful prompts is then capable of generating accurate answers to the posed questions, all without requiring any additional explicit supervision.

In summary, our contributions are as follows:

- We propose to study minimal data augmentation for effective and efficient few-shot QA
- We introduce MinPrompt, a minimal data augmentation framework that uses a graph-based algorithm and unsupervised question generation to synthesize the most informative QA training samples out of the raw text.
- We conduct extensive experiments on publicly accessible benchmarks to validate the effectiveness of MinPrompt, and observe a solid improvement over competitive compared methods. Beyond that, we also study the necessity of different parts of the model.

2 Related Work

Question generation. Chen et al. (2019) presented an answer-aware question generation (QG) model that employs reinforcement learning for improved question quality. The model incorporates a coverage mechanism to alleviate the common issue of answer-related content being left out from the generated questions. Ma et al. (2020) developed a more sophisticated approach to answer-aware question generation. Their model uses sentence-level semantic matching and answer position inferring within a sequence-to-sequence framework, resulting in higher-quality questions. Do et al. (2023) proposed a two-stage framework for Conversational Question Generation (CQG). It selects sentences from a semantic graph to pick up coherent topics and then uses a classifier to determine the answer type of the question. Their approach produces more natural dialogues, as real-life interlocutors often discuss relevant content that is non-sequential. Mohammadshahi et al. (2022) introduces RQUGE, a novel metric for assessing the quality of automatically generated questions. Traditional methods may unfairly penalize valid questions that don't mirror reference questions closely. RQUGE overcomes these issues by evaluating on the basis of the answerability of a question given the context. Utilizing pre-trained models for its QA scorer modules, RQUGE does not require additional training. The paper presents evidence of RQUGE's high correlation with human judgment and robustness against adversarial corruption.

Few-shot QA. Previous research in QA has mainly focused on either reusing pre-trained language models (PLMs) (Lan et al., 2020; Joshi et al., 2020) or training a model from scratch using synthetic QA data (Puri et al., 2020; Lewis et al., 2019a; Alberti et al., 2019). However, both approaches require a large amount of annotated data from the downstream QA task to fine-tune the models, which can be impractical in real-world scenarios. To address this problem, several recent approaches have been developed that allow the model to adapt to the downstream task with only a small amount of annotated data (Ram et al., 2021; Chada and Natarajan, 2021). For example, Ram et al. (2021) proposed a pretraining scheme tailored for QA tasks by designing a recurring span selection objective that aligns with the common objective in extractive QA tasks. Chada and Natarajan (2021) proposed a framework called FewshotQA, which leverages the capacity of existing PLMs by constructing a QA-style prompt that casts the QA problem as a text generation problem, specifically by concatenating the question and a mask token representing the answer span. This approach aims to save pretraining the model on a

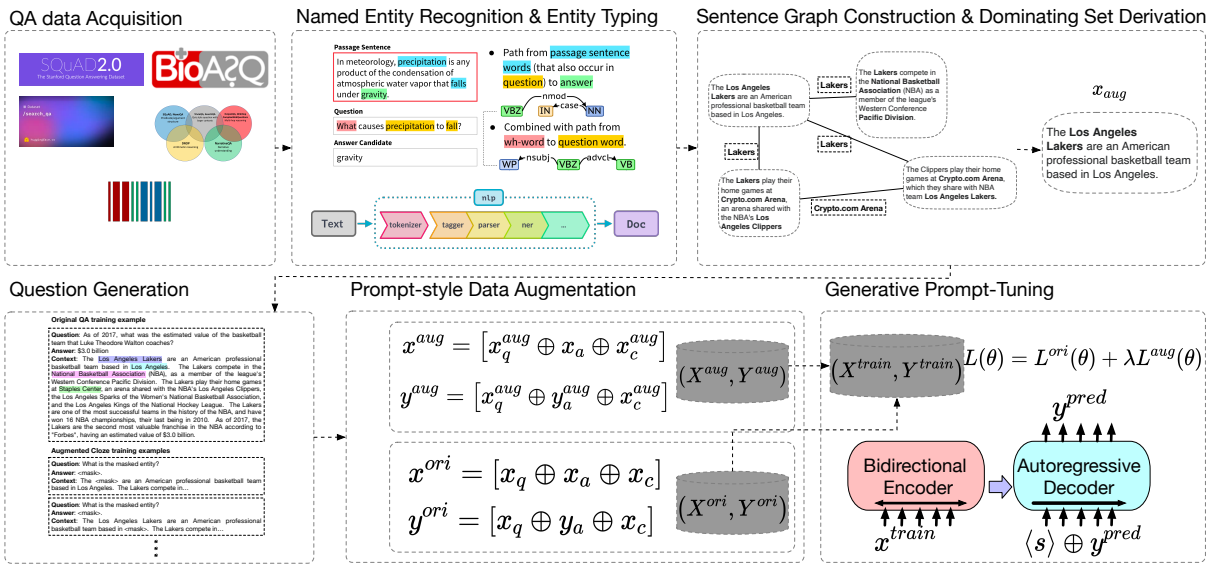

Figure 1: **Framework overview for MINPROMPT.**

large-scale corpus. In contrast to these previous studies, this paper proposes to focus on identifying and leveraging more relevant information from the context data in addition to the annotated QA pairs to fine-tune the model in a few-shot setting.

3 MINPROMPT: Graph-based Prompt Data Augmentation for Few-shot QA

As shown in Figure 1, our overall framework, MINPROMPT, is designed to extract the most semantically rich and factually dense sentences to serve as candidates for conversion into a prompt tuning QA dataset. This process is guided by the principal intuition that the most informative sentences are those that encompass facts or declarations concerning a greater number of entities. Hence, these high-impact sentences should ideally cite more entities within their purview. To implement this, we start by extracting the co-reference of entities across sentences. Essentially, it allows us to map the discourse in a way that allows us to understand which sentences are speaking about the same entities. Next, we construct a graph to depict the higher-order coreference relationships. In this graph, the sentences serve as nodes, and sentences are connected if they mention the same entity. This representation allows us to establish and understand the intricate network of relationships between sentences and the entities they mention. Employing graph-based algorithms, we are then able to identify and extract the most informative sentences. These are typically sentences that have a high degree of connectivity in the graph, indicating that they mention or discuss a larger number of entities. We then transform these selected sentences into a fine-tuning dataset. The transformation process entails restructuring the sentences to meet the format requirements of a QA dataset, which generally involves turning declarative sentences into question-and-answer pairs. This method thus combines insights from computational linguistics and graph theory to achieve its goal of creating a high-quality fine-tuning dataset for QA tasks. The approach ensures that the dataset is not only rich in informative sentences, but also maps intricate entity relationships, thus providing a comprehensive context for each question and answer pair. This context helps in the training of more robust and nuanced QA systems.

3.1 Named Entity Recognition & Entity Typing

We use the entities as the bridge to build connections between all the factual sentences. We first conduct named entity recognition (NER) on the raw text to extract all the entity mentions along with their types. For the purpose of unsupervised QA data generation in our setting, the key lies in generating the questions given the raw text and the extracted entities (as answers). The most straightforward way to generate questions is to convert factual sentences into cloze questions (Chen et al., 2023). Creating a conventional cloze question involves extracting the original sentence containing the answer from the context and replacing the answer with a chosen token. However, training a model on these data primarily imparts text-matching and fill-in-the-blank skills, while offering minimal gen-

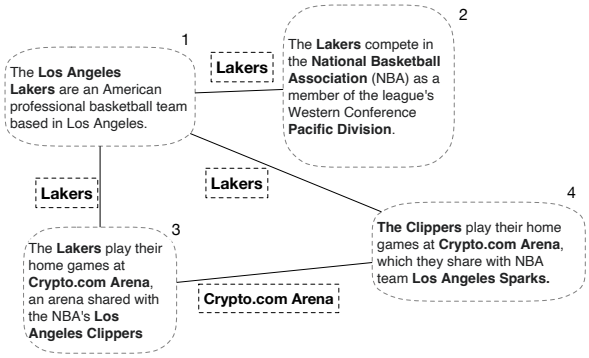

Figure 2: **Illustration of the Sentence graph.** In the sentence graph, nodes correspond to sentences and edges represent the coreference of entities across sentences. Sentences 1, 2 and 3 shares the entity *Lakers* while sentence 4 shares the entity *Crypto.com Arena* with sentence 3.

eralizability. As a result, we opt for a retrieval-based method to procure a sentence akin to the one containing the answer and subsequently use this to formulate a question. This has been evidenced in the work by (Lewis et al., 2019b) and further affirmed by our preliminary experiments. Our initial step involves indexing all sentences from a Wikipedia dump using the ElasticSearch search engine. Named entities were extracted from each sentence within the Wikipedia corpus as well as from the sentences utilized as queries. We presupposed access to a named-entity recognition system and leveraged the spaCy[1] NER pipeline for this work, which is proven effective in NER and entity typing. Subsequently, for a given context-answer pair, we queried the index. This query involved using the original context sentence to return a sentence that either (1) includes the answer, or (2) does not originate from the *context*, thus discarding sentences with high similarity. Aside from guaranteeing that the retrieved sentence and the query sentence share the answer entity, we require that at least one additional matching entity be present in both the query sentence and the entire context. Finally, these retrieved sentences were introduced into our sentence graph construction module.

3.2 Sentence Graph Construction

As aforementioned, we construct the sentence graph to capture the semantic overlap of the factual sentences in the raw text. A proportion of the sentence graph is visualized in Figure 2. Upon building the sentence graph, we aim at extracting the minimal sentence set that covers the most semantics in the whole graph. Now the question becomes how we can leverage the high-order co-reference relationship to reduce the size of the training data. To dive deep into this question, we start by making the following assumption:

Assumption 1. *Suppose two sentences in a sentence set S, $\{s_e, s'_e\} \subset S$, mention the same entity e. The quality of a QA model M_S trained by a sentence set S will be similar to the quality of the other model $M_{S'}$ trained by the set $S' = S - \{s_e\}$ because $s'_e \in S'$ still cover the similar topics and knowledge in s_e.*

Based on Assumption 1, an intuitive idea of leveraging the sentence graph to effectively reduce the size of the training data is to find a minimal set of sentence nodes that can cover the whole sentence graph without losing the quality of the model. In other words, the challenge can be reduced to finding the *minimal dominating set* (Allan and Laskar, 1978) of the sentence graph.

3.3 Minimal Dominating Set Approximation

Unfortunately, finding the minimal dominating set is an NP-Complete problem (Hedetniemi and Laskar, 1991), so it is extremely time-consuming to obtain the optimal minimal dominating set as training data. Hence, an efficient approximation approach to derive a decent dominating set with few enough sentences is essential. To address this challenge, we leverage a greedy algorithm as shown in Algorithm 1 by iteratively choosing the node that can cover the most uncovered nodes.

Algorithm 1 ApproximateDominantingSet

```
S ← ∅
Let H be a priority queue
Add all nodes in H with their node degrees
while H is not empty do
    v ← H.pop_max()
    S ← S ⋃ {v}
    Remove v and its neighbors in E from H
    Update degrees of the remaining nodes in H
end while
return S
```

Complexity Analysis. Here we analyze the complexity of Algorithm 1. Suppose V and E are the numbers of nodes and edges. For time complexity, the algorithm first spends $O(V \log V)$ time to establish the max heap. For each iteration, taking the node with the highest degree costs $O(1)$ with the priority queue. In total, we need to update the priority queue $O(E)$ times, where each update costs

[1] https://spacy.io

$O(\log V)$ time. Hence, the total time complexity is $O(E \log V)$. For space complexity, the additional space complexity is only $O(V)$ to record the current set of uncovered nodes and the max heap.

Theoretical Analysis. We also conduct some theoretical analysis on Algorithm 1. According to Theorem 1, the quality of dominating set derived by Algorithm 1 is guaranteed. The proof is shown in Appendix A.

Theorem 1. *Algorithm 1 computes an* $(\ln \Delta + 2)$*-approximation of the optimal dominanting set. In other words, for the computed dominating set* S *and an optimal dominating set* S^**, we have*

$$\frac{|S|}{|S^*|} \leq \ln \Delta + 2,$$

where $\Delta = \max_v d(v)$ *is the maximal degree of* G.

3.4 Question Generation

Our approach considers two question styles, including (1) generic cloze-style questions, wherein the answer is substituted by the token "[MASK]", and (2) a templated question format termed "Wh+B+A+?" as well as its diverse ordering variations, as depicted in Figure 3. Given a retrieved sentence structured as `[Fragment A]` `[Answer]` `[Fragment B]`, the template "Wh + B + A +?" replaces the *Answer* with a component Wh (for instance, what, who or where). This component is determined by the entity type of the *Answer* and is placed at the beginning of the question. It is then followed by `Fragment B` and `Fragment A`. The selection of the wh-component involves sampling a bi-gram based on the likelihood of that particular bi-gram being connected with the named entity type of the answer. This likelihood is calculated from the named entity and questions bigram starters found in the SQuAD dataset. This information, while not leveraging the complete context-question-answer framework, can be considered as prior knowledge that does not disrupt the wholeness of our unsupervised methodology. It is also important to note that the choice of wh-component does not have a substantial impact on the results. Although we experimented with clause-based templates for this template-driven approach, we did not observe any significant differences in performance.

3.5 Prompt-style Data Augmentation

We extend the recent progress in prompt tuning to create augmented data for MinPrompt. Specifically, we have formulated a template to enable

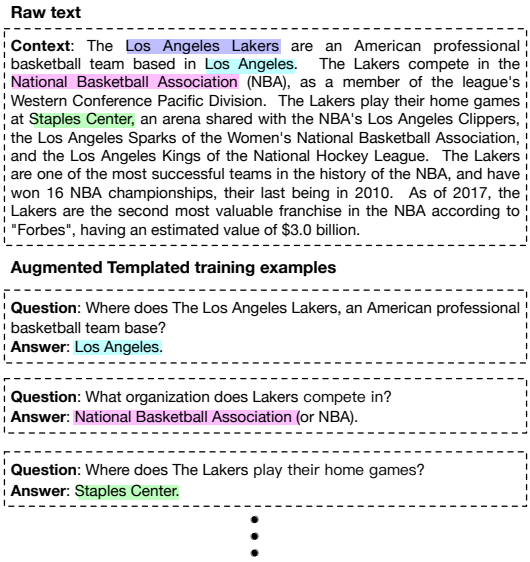

Figure 3: **Examples of generated questions.** When MinPrompt runs into an $entity$ in the raw text during the question generation phase, it turns the factual sentence into a QA pair of $(question, entity)$, with the question type depending on the entity type.

QA input, designated as x^{ori}. The template is constructed as follows:

$$x_q = \textit{Question} : \mathbf{q}$$
$$x_a = \textit{Answer} : \textit{<mask>}$$
$$x_c = \textit{Context} : \mathbf{c}$$
$$x^{ori} = [x_q \oplus x_a \oplus x_c]$$

Here, we formulate the labels y as:

$$y_a = \textit{Answer} : \mathbf{a},$$
$$y = [x_q \oplus y_a \oplus x_c],$$

where $\mathbf{q}$, $\mathbf{a}$, and $\mathbf{c}$ represent the query text, response text, and background context respectively, and $\oplus$ symbolizes string concatenation.

In the augmented QA data samples, we apply the masking to the chosen entity in x_c to construct the context text for the augmented data x_c^{aug}, along with the mask token in x_a. The specifics of an augmented data sample (x^{aug}, y^{aug}) are depicted in Figure 3. Let the set of all training samples from original QA datasets and augmented QA pairs be denoted by (X^{ori}, Y^{ori}) and (X^{aug}, Y^{aug}) respectively. Thus, our entire training set (X^{train}, Y^{train}) comprises of both (X^{ori}, Y^{ori}) and (X^{aug}, Y^{aug}).

3.6 Training

One of the key benefits of harmonizing the augmented and original data lies in the ability of the

model to effectively process both data types without any significant loss. Concisely, MINPROMPT derives a prediction utilizing an encoder-decoder model as

$$y^{pred} = decoder_{\theta_D}(encoder_{\theta_E}(x)), \quad (1)$$

where θ_E and θ_D represent learnable parameters, and $x \in X^{train}$ can be either an original or an augmented training sample.

The training objective of our system aims to maximize the log-likelihood of the text in the reference answer, denoted by $y \in Y^{train}$. The loss functions concerning the original samples and the augmented samples are expressed in the following equations:

$$L^{ori}(\theta) = \sum_{(x,y)\in(X^{ori},Y^{ori})} \log\left(\prod_{i=1}^{n} P\left(y_i \mid y_{<i}, x; \theta\right)\right)$$

$$L^{aug}(\theta) = \sum_{(x,y)\in(X^{aug},Y^{aug})} \log\left(\prod_{i=1}^{n} P\left(y_i \mid y_{<i}, x; \theta\right)\right)$$

where $\theta = \{\theta_D, \theta_E\}$. The overall loss function is the weighted average of two losses:

$$L(\theta) = L^{ori}(\theta) + \lambda L^{aug}(\theta). \quad (2)$$

We consider $\lambda > 0$ to be a hyperparameter that establishes a balance between the few-shot QA training samples and the augmented QA samples.

4 Experiments

4.1 Experimental Setup

Datasets. Following Splinter (Ram et al., 2021) and FewshotQA (Chada and Natarajan, 2021), we sample subsets from the MRQA 2019 shared task (Fisch et al., 2019) for our few-shot experiments. Taking a closer look, there are in total eight widely used benchmark QA datasets in MRQA: SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019), BioASQ (Tsatsaronis et al., 2015), and TextbookQA (Kembhavi et al., 2017). Following Splinter (Ram et al., 2021), smaller training datasets are sampled in a logarithmic manner from the original full datasets, resulting in few-shot datasets with 16, 32, 64, and 128 training examples.

Comparative Baselines. We evaluate the performance of MINPROMPT against four competitive few-shot QA methods, including **RoBERTa** (Liu et al., 2019), **SpanBERT** (Joshi et al., 2020), **Splinter** (Ram et al., 2021), **FewshotQA** (Chada and Natarajan, 2021), and **PMR** (Xu et al., 2023). Details of these baselines, raw text data source, implementation details, and evaluation metric are in Appendix B, C, D, and E, correspondingly.

4.2 Performance Comparison

Table 2 presents the few-shot QA performance comparison of various models across all benchmarks when provided with 16, 32, 64, and 128 training examples. BART-large serves as the backbone pretrained language model (PLM) for FewshotQA.

The experiment was repeated five times, each with a different random seed, and we report the average and standard deviation of the results for each method. As a general observation, PMR, Splinter and FewshotQA with MINPROMPT excel over other compared methods by a respectable margin in most cases. On average, models with MINPROMPT yield better results with consistently lower variances (the rightmost column). The only exception is the 128 examples, where MINPROMPT and PMR ended in a draw. Note that FewshotQA with MINPROMPT performs better in fewer-shot cases because BART is pretrained on general domain plain texts, so MINPROMPT can apply its broad knowledge and rapidly adapt to the specifics of the QA task with just a few examples. PMR gradually catches up with more few-shot examples because its specialized training allows it to learn more efficiently from and utilize the additional examples, scaling its performance in a way that is directly relevant to the task. There are several cases in which performance degrades when using MINPROMPT. This is probably because the augmented data samples outweigh the original fine-tuning data samples for these datasets, directing the pretrained model towards the distribution of the augmented data which is slightly shifted from the distributions of the fine-tuning and test data after all. More notably, MINPROMPT exhibits less variance in results compared to FewshotQA in most cases, particularly when there are fewer training examples available.

In digging deeper into specific models, both Splinter and FewshotQA enhanced by MINPROMPT consistently outperform their original model in terms of higher F1 scores with generally lower variances. On SQuAD, NQ, BioASQ, and TextbookQA, the performance improvements over the top baseline are relatively more substantial. Our

# examples	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	BioASQ	TextbookQA
# nodes	104,160	123,183	418,049	356,408	25,413	417,895	60,080	30,723
# edges	20,310,486	36,716,957	408,935,741	339,619,544	13,425,062	766,206,565	6,821,645	3,150,557
# dominating set	8,260	11,099	30,452	24,015	1,518	34,830	4,480	1,116
# training samples	**17,409**	**24,091**	**48,213**	**32,391**	**4,509**	**116,385**	**6,884**	**1,505**

Table 1: **Number of augmented training examples per dataset.** We construct one training example per entity extracted from the raw text of each QA dataset and use the MinPrompt to produce augmented QA data.

Model	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	BioASQ	TextbookQA	Average
16 Examples									
RoBERTa	7.7±4.3	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	9.0±3.4
SpanBERT	18.2±6.7	11.6±2.1	19.6±3.0	7.6±4.1	13.3±6.0	12.5±5.5	15.9±4.4	7.5±2.9	13.3±4.3
PMR	60.3±4.0	**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	47.5±3.4
Splinter	54.6±6.4	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	27.4±4.5
Splinter w/ MinPrompt	**58.9±3.6**	**35.7±1.9**	**37.6±2.8**	**31.9±1.8**	**35.2±1.6**	**34.0±6.3**	**38.7±3.6**	**37.0±5.1**	**36.1±3.3**
FewshotQA	72.5±3.7	47.1±7.6	57.3±3.2	44.9±4.5	54.3±5.9	**59.7±2.2**	**62.7±4.4**	33.1±3.2	53.9±4.3
FewshotQA w/ MinPrompt	**73.6±3.3**	50.9±4.6	**58.5±1.9**	**46.5±1.8**	55.4±2.7	57.1±2.9	57.2±2.3	**42.2±4.1**	**55.2±2.9**
32 Examples									
RoBERTa	18.2±5.1	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	13.3±2.6
SpanBERT	25.8±7.7	15.1±6.4	25.1±1.6	7.2±4.6	14.6±8.5	13.2±3.5	25.1±3.3	7.6±2.3	16.7±4.7
PMR	70.0±3.2	**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	54.8±2.7
Splinter	59.2±2.1	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	35.3±3.0
Splinter w/ MinPrompt	**64.6±1.5**	**35.6±2.1**	**42.8±1.3**	**33.0±1.2**	**39.2±3.4**	**41.4±3.1**	**49.2±3.2**	**38.2±2.5**	**43.0±2.3**
FewshotQA	73.8±2.2	56.7±5.9	**60.6±2.4**	50.0±2.8	61.4±3.6	61.6±1.5	**66.9±4.7**	41.7±4.2	59.1±3.4
FewshotQA w/ MinPrompt	**78.0±1.1**	53.5±4.0	59.3±1.0	**51.8±1.8**	60.3±2.6	**61.6±3.1**	63.6±2.9	**46.5±2.0**	**59.3±2.3**
64 Examples									
RoBERTa	28.4±1.7	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	18.0±2.0
SpanBERT	45.8±3.3	15.9±6.4	29.7±1.5	12.5±4.3	18.0±4.6	23.3±1.1	35.3±3.1	13.0±6.9	24.2±3.9
PMR	71.2±2.8	**67.1±1.8**	51.2±3.1	43.2±1.8	66.2±1.8	56.3±2.0	68.2±1.6	41.8±2.3	58.1±2.2
Splinter	65.2±1.4	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	43.4±2.7
Splinter w/ MinPrompt	**68.6±1.8**	**35.4±2.9**	**45.9±1.3**	36.1±1.7	**44.3±3.1**	**48.6±2.3**	**59.4±2.4**	**42.6±1.6**	**47.6±2.1**
FewshotQA	77.9±2.1	57.9±4.4	**60.9±2.5**	53.7±1.1	65.4±2.4	**63.1±2.2**	**73.2±3.1**	44.8±1.8	62.1±2.5
FewshotQA w/ MinPrompt	**79.2±1.0**	55.3±3.2	59.7±1.3	**54.2±1.0**	**67.1±1.0**	61.1±3.0	72.4±2.5	**48.7±2.4**	**62.5±1.9**
128 Examples									
RoBERTa	43.0±7.1	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	27.3±3.1
SpanBERT	55.8±3.7	26.3±2.1	36.0±1.9	29.5±7.3	26.3±4.3	36.6±3.4	52.2±3.2	20.9±5.1	35.4±3.9
PMR	79.8±1.8	**68.6±1.4**	57.4±2.6	52.3±1.4	**68.5±1.8**	**65.9±1.0**	76.8±2.1	45.1±1.2	**64.3±1.7**
Splinter	**72.7±1.0**	**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	51.9±2.3
Splinter w/ MinPrompt	70.2±2.8	45.4±1.3	**51.2±1.3**	40.2±1.6	**48.5±2.1**	54.5±2.2	**67.8±1.6**	**44.2±2.1**	**52.8±1.9**
FewshotQA	78.8±2.7	55.2±1.8	63.3±1.6	56.8±1.1	67.0±1.8	64.9±1.8	77.2±1.5	46.2±5.9	63.7±2.3
FewshotQA w/ MinPrompt	**80.5±1.4**	52.9±3.9	**64.2±1.4**	**56.9±1.0**	68.1±1.9	61.7±1.4	**77.8±1.2**	**52.5±3.7**	**64.3±2.0**

Table 2: **Overall performance** in F1 scores across all datasets when the numbers of training examples are 16, 32, 64, and 128. NQ stands for Natural Questions. RoBERTa, SpanBERT, Splinter and Splinter w/ MinPrompt have 110M parameters. PMR, FewshotQA and FewshotQA w/ MinPrompt have parameters of size 406M. Comparisons with more baselines are in Appendix F and G.

hypothesis is that the factual statements are more concentrated in a small number of sentences, thus MinPrompt can more effectively extract the most informative data for fine-tuning. Consequently, the influence from the is adequate to impact the primary QA task. We also observe that with the decrease in the number of few-shot QA training examples, MinPrompt demonstrate more improvement. This is also expected since MinPrompt essentially introduces external prior knowledge that is not present in the few-shot training examples. When the models see more actual training examples that are with the same distribution as the test set, the external knowledge helps less and even becomes noise in the extreme case. Finally, we also observe a greater improvement brought about by MinPrompt to Splinter than to FewshotQA. This is because Splinter has a smaller model size; therefore, it naturally acquires less knowledge during the pre-train stage. Adding external knowledge to it in the form of QA benefits even more than bigger models, such as FewshotQA.

4.3 Effect of Deriving the Dominating Set

To validate the necessity of deriving the dominating set of the sentence graph to keep the most informative factual sentences in the raw text, we further

Context: *"…In species with sexual reproduction, each cell of the body has two copies of each chromosome. For example, human beings have 23 different chromosomes. Each body cell contains two of each chromosome, for a total of 46 chromosomes. The number of different types of chromosomes is called the haploid number. In humans, the haploid number is 23. The number of chromosomes in normal body cells is called the diploid number. The diploid number is twice the haploid number. The two members of a given pair of chromosomes are called homologous chromosomes …"* **Question**: What is the *number of chromosomes in a gamete called?*		**Context**: *"…For example, cystic fibrosis gene therapy is targeted at the respiratory system, so a solution with the vector can be sprayed into the patients nose. Recently, in vivo gene therapy was also used to partially restore the vision of three young adults with a rare type of eye disease. In ex vivo gene therapy, done outside the body, cells are removed from the patient and the proper gene is inserted using a virus as a vector. The modified cells are placed back into the patient. One of the first uses of this type of gene therapy was in the treatment of a young girl with a rare genetic disease, adenosine deaminase deficiency, or ADA deficiency…"* **Question**: *Which disorder has been treated by ex vivo gene therapy?*	
Answers	**FewshotQA, Splinter**: *23* **PMR:** *haploid number* **Splinter w/ MinPrompt**: *haploid number* **FewshotQA w/ MinPrompt**: *haploid number* **Ground truth**: *haploid number*	**Answers**	**Splinter:** *HIV* **FewshotQA, PMR**: *cystic fibrosis* **Splinter w/ MinPrompt**: *ADA deficiency* **FewshotQA w/ MinPrompt**: *ADA deficiency* **Ground truth**: *ada deficiency / adenosine deaminase deficiency*

Figure 4: **Case study.** In both cases, MinPrompt successfully generates the correct answer, whereas baselines without entity masking can not accurately recover the entity-level details.

Model	SQuAD	TextbookQA
16 Examples		
FewshotQA w/ MinPrompt-random	72.0±3.5	39.2±4.8
FewshotQA w/ MinPrompt	**73.6±3.3**	**42.2±4.1**
32 Examples		
FewshotQA w/ MinPrompt-random	75.9±1.8	43.3±2.2
FewshotQA w/ MinPrompt	**78.0±1.1**	**46.5±2.0**
64 Examples		
FewshotQA w/ MinPrompt-random	78.6±1.3	46.2±2.2
FewshotQA w/ MinPrompt	**79.2±1.0**	**48.7±2.4**
128 Examples		
FewshotQA w/ MinPrompt-random	79.9±1.4	49.5±3.5
FewshotQA w/ MinPrompt	**80.5±1.4**	**52.5±3.7**

Table 3: **Ablation study.** Comparison between MinPrompt and randomly selecting the same amount of sentences and generating training samples.

conduct an ablation study. We construct a variant of MinPrompt called MinPrompt-random where we randomly sample the same number of sentences as shown in Table 1 for each dataset, and then generate training samples out of these randomly sampled factual sentences. We run MinPrompt-random and report the results on SQuAD and TextbookQA in Table 3. When comparing the two models, we can observe that MinPrompt consistently perform better than MinPrompt-random. We also observe this pattern on all the other datasets. This observation empirically validates that the dominating set derivation process indeed provides factual sentences that preserve as much information as possible about the crucial entities in the raw text.

4.4 Case Study

Further exploration of two specific test cases from the TextbookQA test set provides insightful results, as depicted in Figure 4. In the left case, both FewshotQA and Splinter without MinPrompt yield the incorrect response, *23*. Despite its semantic relevance to the accurate answer, *haploid number*, the response goes overly detailed, since the value *23* is specific only to human beings. This case underlines the advantage of MinPrompt's full model, equipped with a sentence construction module anchored by entities, in deriving detailed answer text at the entity level, over FewshotQA and Splinter. In the right case, both FewshotQA and Splinter with MinPrompt successfully identify the correct answer, whereas Splinter supplies an incorrect answer, *HIV*, not even present in the context. Meanwhile, FewshotQA and PMR produced another treatment instead of what the question asks (a disorder), indicating that the question generation module of MinPrompt improved the models' ability to deal with various kinds of questions. This comparison effectively highlights the utility of the sentence graph in forging higher-order entity interconnections within the same context. Although the baselines provide a contextually relevant response, they do not adequately address the question. The two cases substantiate the indispensable role of the sentence graph construction module and the question generation module in MinPrompt, fortifying its capacity to delve into complex question and context semantics.

5 Conclusion

In this paper, we present MinPrompt, a robust data augmentation framework that leverages a graph-based algorithm and unsupervised question generation to extract minimally meaningful QA training samples from raw text. Our contributions reside in the application of minimal data augmentation, enhancing computational efficiency and model performance while mitigating overfitting. Through extensive experiments, our model consistently outperformed competitive methods in public benchmarks, demonstrating its effectiveness.

6 Limitations

While MinPrompt is capable of achieving comparative or better performance over existing studies, it still has some limitations as follows: First, MinPrompt integrates the trained NER model as part of the pipeline, so the performance of the SpaCy NER model greatly affects the overall performance of MinPrompt. Second, MinPrompt uses all shared entities to construct the sentence graph. However, some entities might be more crucial than others for the downstream QA task. As a result, treating the entities differently might lead to a different result. Lastly, the template utilized for prompt-tuning in this study still relies on manual design. Our approach is influenced by previous research that has been shown to be effective. Nevertheless, it would be intriguing to explore the development of automated methods for constructing superior prompt-tuning templates.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *ACL'19*, pages 6168–6173.

Robert B Allan and Renu Laskar. 1978. On domination and independent domination numbers of a graph. *Discrete mathematics*, 23(2):73–76.

Haytham Assem, Rajdeep Sarkar, and Sourav Dutta. 2021. Qasar: self-supervised learning framework for extractive question answering. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1797–1808. IEEE.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yurii Kuratov, Denis Kuznetsov, et al. 2018. Deeppavlov: Open-source library for dialogue systems. In *ACL'18, System Demonstrations*, pages 122–127.

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. In *EMNLP'21*, pages 6081–6090.

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.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement learning based graph-to-sequence model for natural question generation. *arXiv preprint arXiv:1908.04942*.

Xuan Long Do, Bowei Zou, Shafiq Joty, Anh Tai Tran, Liangming Pan, Nancy F Chen, and Ai Ti Aw. 2023. Modeling what-to-ask and how-to-ask for answer-unaware conversational question generation. *arXiv preprint arXiv:2305.03088*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. 2021. Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. *NPJ digital medicine*, 4(1):1–9.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13.

Stephen T Hedetniemi and Renu C Laskar. 1991. Bibliography on domination in graphs and some basic definitions of domination parameters. In *Annals of discrete mathematics*, volume 48, pages 257–277. Elsevier.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL'17*, pages 1601–1611.

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 *CVPR'17*, pages 4999–5007.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *TACL*, 7:453–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR'20*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart:

Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL'20*, pages 7871–7880.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019a. Unsupervised question answering by cloze translation. In *ACL'19*, pages 4896–4910.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019b. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

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

Xiyao Ma, Qile Zhu, Yanlin Zhou, and Xiaolin Li. 2020. Improving question generation with sentence-level semantic matching and answer position inferring. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8464–8471.

Alireza Mohammadshahi, Thomas Scialom, Majid Yazdani, Pouya Yanki, Angela Fan, James Henderson, and Marzieh Saeidi. 2022. Rquge: Reference-free metric for evaluating question generation by answering the question. *arXiv preprint arXiv:2211.01482*.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Unsupervised multi-hop question answering by question generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5866–5880.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *EMNLP'20*, pages 5811–5826.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP'16*, pages 2383–2392.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. In *ACL'21*, pages 3066–3079.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

Weiwen Xu, Xin Li, Wenxuan Zhang, Meng Zhou, Wai Lam, Luo Si, and Lidong Bing. 2023. From cloze to comprehension: Retrofitting pre-trained masked language models to pre-trained machine reader. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *ACL'19, System Demonstrations*, pages 72–77.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP'18*, pages 2369–2380.

A Proof of Algorithm 1

Proof. Here we prove the theorem in an amortized way. Suppose each iteration costs 1 (i.e., contributing to the cardinality of the final dominating set). Instead of letting the selected node takes all the cost, we amortize and distribute the cost among all newly covered nodes.

Assume S' is an optimal dominating set. By the definition of dominating set, we can assign each node in V to exactly one neighboring node in S' so that the graph can be decomposed into several stars, where the center is a dominating node and non-dominating nodes are leaves.

Consider a certain star with a center $v' \in S'$ while choosing a node u in Algorithm 1. By the greedy condition and the optimality of v', after cost distribution, the charged cost of u would be at most $d(v')$. Also, after removing u, the degree of v' will be reduced by 1. Following this process to iteratively select dominating nodes, the total amortized cost would be at most:

$$\begin{aligned}\frac{1}{d(v')+1}+\frac{1}{d(v')}+\cdots+\frac{1}{1} &= H(d(v^*)+1)\\ &\leq H(\Delta+1)\\ &< \ln\Delta+2,\end{aligned}$$

where Δ is the maximal degree of the graph; $H(n)=\sum_{i-1}^{n}1/i$.

□

B Baseline Details

- **RoBERTa (Liu et al., 2019)** is a robustly optimized BERT-based PLM. It improves BERT by techniques such as training the model for a longer time, with larger batches and getting rid of the next sentence prediction task. It is known to demonstrate substantially better performance on a variety of natural language understanding tasks over BERT, including QA.

- **SpanBERT (Joshi et al., 2020)** is another variant of BERT that emphasizes the encoding of spans instead of tokens. It is pretrained on two tasks: (1) masked language modeling, which is the same as BERT, and (2) span boundary prediction, which pulls the representations of the span boundary into a direction where the entire content of the masked span can be predicted correctly. SpanBERT achieves substantially better performance on span selection tasks in particular.

- **Splinter (Ram et al., 2021)** is a pretraining framework dedicated to the extractive QA task based on SpanBERT. It is pretrained by the recurring span selection task, which masks all but one instance of each recurring span and asks the model to select the correct span for each masked position.

- **FewshotQA (Chada and Natarajan, 2021)** is the first QA-dedicated fine-tuning framework that takes advantage of pre-trained encoder-decoder models such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). In FewshotQA, the input is constructed as a concatenation of the question, a mask token as the placeholder for the answer span, and a context. Given this input, the model is fine-tuned using the same objective as its pretraining objective.

- **PMR (Xu et al., 2023)** constructs general-purpose machine reading comprehension training data by using Wikipedia hyperlinks and designed a Wiki Anchor Extraction task to guide the MRC-style pretraining.

C QA data acquisition

The first step in our framework is to retrieve the raw text corpus as the super set from which all our prompt dataset comes. For pretraining, text corpus from general domains such as Wikipedia is commonly used. On the contrary, since we focus on the fine-tuning stage, we use domain-specific text as a starting point. Following Splinter (Ram et al., 2021) and FewshotQA (Chada and Natarajan, 2021), we take MRQA (Fisch et al., 2019) as a benchmark to test the performance of all the comparative methods.

D Implementation Details

For all the models, we use the same hyperparameters during training for a fair comparison. Specifically, the models are optimized by Adam (Kingma and Ba, 2014) with bias corrections. The learning rate is 2×10^{-5} without learning rate scheduling. The training batch size is set to 2. The maximum sequence length of sequence generation is 100 for FewshotQA and MinPrompt. We train all the models compared for 25 epochs. The reported results are given by the best-performing checkpoint in the development sets. For MinPrompt, we perform a grid search for the loss weight λ in the

space $\{0.01, 0.05, 0.1, 0.5, 1.0, 10.0\}$. All experiments are run on NVIDIA Tesla A100-SXM4 Tensor Core GPUs with 40GB memory.

E Evaluation Metrics

Following previous studies (Ram et al., 2021; Chada and Natarajan, 2021), we use the F1 score as our evaluation metric. Specifically, for each sample in the test set, the predicted span and the ground truth answer are treated as bags of words, and F1 scores are applied to compute the overlap between these two sets. If there are multiple ground-truth answers to a particular question, we take the maximum of the corresponding F1 scores.

F Comparisons against MQA-QG

Here we compare with the other few-shot data augmentation approach, MQA-QG (Pan et al., 2021). For a fair comparison, we first run the released implementation of MQA-QG, apply their approach on Splinter, and then compare it with our method. The results of 16-shot experiments are as shown in Table 4. We see consistent improvements derived by MinPrompt over MQA-QG, and a similar pattern is also observed in 32, 64, and 128-shot scenarios.

G Comparisons against Unsupervised Domain Adaption

In addition to the few-shot approach, some studies apply unsupervised domain adpation to tackle the limitation of training data (Assem et al., 2021). As an additional study, we compare with Qasar (Assem et al., 2021) Qasar, we focus on four overlapping datasets (i.e., NQ, NewsQA, BioASQ, and TextbookQA) between their paper and our studies as shown in Table 5. We can observe that FewshotQA w/ MinPrompt outperforms Qasar across four datasets from 0.4% to 22.2%. We also would like to emphasize that Qasar uses fine-tuning training samples ranging from 142 to 4,185 while MinPrompt using only 16 to 128 fine-tuning examples surpasses Qasar with certain disadvantages in the limited amount of fine-tuning data.

H Additional Discussions

Here we list some additional discussions on our approach.

H.1 Generalization Ability to Different Answer Types

To different types of answers (e.g., why v.s. how and longanswers), we would like to mention that MinPrompt raises different types of questions based on the results of the entity typing. During this process, why / how questions would be raised once a conjunction (e.g., because) or an adverb (e.g., by) is recognized from the raw text. We agree that the why / how questions with longer answers might be less than some other types of questions like what / who / when ones in the augmented training samples, and it might cause generalization issues. An intuitive fix is to assign larger sample weights to the augmented samples with why / how questions or to repeat these samples multiple times to make different types of questions roughly be of the same number. However, the main focus of this paper is to demonstrate the idea that graph-based data selection can help the overall downstream performance, so we leave the detailed analysis for certain types of answers for future work.

H.2 Potential Solution to Overfitting with Prompt-style Augmentation

It could introduce an ovefit with prompt-style agumentation to the distribution of different quesetion formats as we observed in the experiments, especially for the cases with only few shot training samples. The distribution of different types of questions in the augmented data might be skewed, for example, the what / who / when questions might be more than the why / how questions. In this way, the what / who / when questions in the test set might get more precise answers than the why / how questions. The intuitive fix is to put larger sample weights to the augmented samples with why / how questions or to repeat these samples multiple times to make different types of questions roughly be of the same number.

Model	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	BioASQ	TextbookQA
MQA-QG	54.38	32.28	37.36	25.12	31.35	33.89	36.39	29.71
MinPrompt	58.91	35.67	37.64	31.88	35.17	34.03	38.68	36.98

Table 4: Performance comparisons against MQA-QG.

Model	NQ	NewsQA	BioASQ	TextbookQA
Qasar	59.76	56.63	63.70	47.02
Splinter w/ MinPrompt	51.17	40.22	67.80	44.24
FewshotQA w/ MinPrompt	**64.17**	**56.84**	**77.84**	**52.53**

Table 5: Performance of MinPrompt with 128 examples against the unsupervised domain adation method.