ReasonBert: Pre-trained to Reason with Distant Supervision

Anonymous EMNLP submission

Abstract

We present ReasonBert, a pre-training method that augments language models with the ability to reason over long-range relations and multiple, possibly hybrid contexts. Unlike existing pre-training methods that only harvest learning signals from local contexts of naturally occurring texts, we propose a generalized notion of distant supervision to automatically connect multiple pieces of text and tables to create pre-training examples that require long-range reasoning. Different types of reasoning are simulated, including intersecting multiple pieces of evidence, bridging from one piece of evidence to another, and detecting unanswerable cases. We conduct a comprehensive evaluation on a variety of extractive question answering datasets ranging from single-hop to multi-hop and from text-only to table-only to hybrid that require various reasoning capabilities and show that ReasonBert achieves remarkable improvement over an array of strong baselines. Few-shot experiments further demonstrate that our pre-training method substantially improves sample efficiency.[1]

1 Introduction

Recent advances in pre-trained language models (LMs) have remarkably transformed the landscape of natural language processing. Pre-trained to reconstruct naturally occurring utterances sampled from massive text corpora with unsupervised objectives such as autoregressive language modeling (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020) and masked language modeling (MLM) (Devlin et al., 2019; Liu et al., 2019b; Joshi et al., 2020), PLMs encode a great deal of knowledge about language and significantly boost model performance on a wide range of downstream tasks (Liu et al., 2019a; Wang et al., 2018, 2019) ranging from spell checking (Awasthi et al., 2019) to sentiment analysis (Xu et al., 2019) and semantic parsing (Rongali et al., 2020).

Existing unsupervised objectives for LM pre-training primarily focus on consecutive, naturally occurring text. For example, MLM enables LMs to correctly predict the missing word "*daughters*" in the sentence "*Obama has two __ , Malia and Sasha.*" based on the local context and the knowledge stored in the parameters. However, many tasks require reasoning beyond local contexts: multi-hop question answering (QA) (Yang et al., 2018; Welbl et al., 2018) and fact verification (Jiang et al., 2020) require reasoning over multiple pieces of evidence, hybrid QA (Chen et al., 2020) requires simultaneously reasoning over unstructured text and structured tables, and dialogue systems require reasoning over the whole dialogue history to accurately understand the current user utterance (Semantic Machines et al., 2020).

To address this limitation in existing LM pre-training, we propose ReasonBert, a pre-training method to augment LMs to explicitly reason over long-range relations and multiple contexts. Unlike existing pre-training objectives that predict individual masked tokens or spans within a contiguous paragraph of text, ReasonBert pairs a query sentence with multiple relevant pieces of evidence drawn from possibly different places and defines a new LM pre-training objective, *span reasoning*, to recover entity spans that are masked out from the query sentence by jointly reasoning over the relevant evidence (Figure 1). In addition to text, we also include tables as evidence to further empower LMs to reason over hybrid contexts.

One major challenge in developing ReasonBert lies in how to create a large set of query-evidence pairs for pre-training. Unlike existing unsupervised pre-training methods, examples with complex reasoning cannot be easily harvested from naturally occurring texts. Instead, inspiration was drawn from *distant supervision* (Mintz et al., 2009a), which

[1] The pre-trained model is available in Huggingface https://huggingface.co/Anonymous

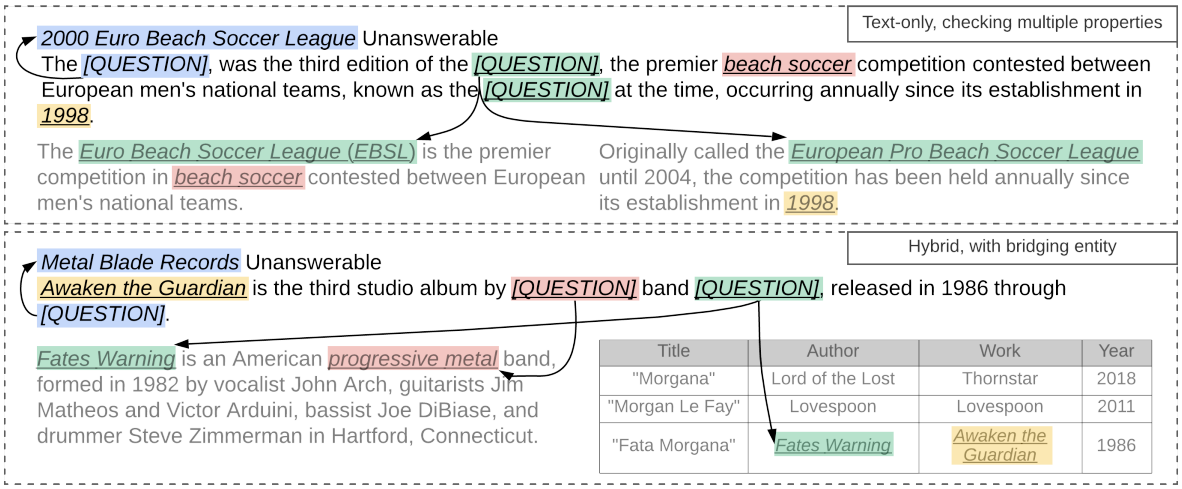

Figure 1: Examples of our pre-training data acquired via distant supervision, which covers a wide range of topics with both textual and tabular evidence. For each query sentence, we first select two pairs of entities (underlined) to find two pieces of evidence via distant supervision. We then randomly mask one entity from each selected pair and aim to recover it by reasoning over the evidence. Note that the two selected pairs may share a common entity; in case this entity is masked, we can mimic different types of multi-hop reasoning, e.g., intersection (Ex. 1) and bridging (Ex. 2). To simulate unanswerable cases, we additionally mask one entity (in blue) that does not exist in the evidence. Figure best viewed in color.

assumes that "*any sentence containing a pair of entities that are known to participate in a relation is likely to express that relation*," and generalize to our setting of multiple possibly hybrid pieces of evidence. Specifically, given a query sentence containing an entity pair, if we mask one of the entities, another sentence or table that contains the same pair of entities can likely be used as evidence to recover the masked entity. Moreover, to encourage deeper reasoning, we collect multiple examples of evidence that are jointly used to recover the masked entities in the query sentence, allowing us to scatter the masked entities among different pieces of evidence to mimic different types of reasoning. Figure 1 illustrates several examples using such distant supervision. In Ex. 1, a model needs to check multiple constraints (i.e., intersection reasoning type) and find "*the beach soccer competition that is established in 1998*." In Ex. 2, a model needs to find "*the type of the band that released Awaken the Guardian*," by first inferring the name of the band "*Fates Warning*" (i.e., bridging reasoning type).

We replace the masked entities in a query sentence with the `[QUESTION]` tokens, and the new pre-training objective, span reasoning, is then to extract the masked entities from the provided evidence. We augment existing LMs like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) by continuing to train them with the new objective, which leads to ReasonBert, a new LM with better reasoning capabilities. We use a transformer based encoder (Devlin et al., 2019) to encode the concatenated query sentence and textual evidence. When tabular evidence is present, we use the recent structure-aware transformer from TAPAS (Herzig et al., 2020) as the encoder to help capture the table structure.

We evaluate ReasonBert on the extractive QA task, which is arguably the most representative task requiring reasoning about world knowledge. We conduct a comprehensive evaluation using a variety of popular datasets: MRQA (Fisch et al., 2019), a single-hop QA benchmark including six datasets from different domains; HotpotQA (Yang et al., 2018), a multi-hop QA dataset; NQTables, a subset of the Natural Questions dataset (Kwiatkowski et al., 2019) where answers can be found in tables; and HybridQA (Chen et al., 2020), a hybrid multi-hop QA dataset that requires reasoning over both tables and text. Under the few-shot setting, ReasonBert substantially outperforms the baselines in almost all datasets, demonstrating that the reasoning ability learned from pre-training can easily transfer to downstream QA tasks and generalize well across domains. Under the full-data setting, ReasonBert obtains substantial gains in multi-hop and hybrid QA datasets. Despite its simple model architecture, ReasonBert achieves similar or better performance compared with more sophisticated state-of-the-art models for each dataset.

2 Background

Language model pre-training. Existing pre-training objectives such as MLMs (Devlin et al., 2019; Joshi et al., 2020) tend to implicitly memorize the learned knowledge in the parameters of the underlying neural network. In this work, we aim to augment pre-training by encouraging a model to *reason* about (instead of memorizing) world knowledge over the given contexts.

Extractive question answering. To measure a model's reasoning ability about world knowledge, we select extractive QA as a downstream task, which is perhaps one of the most representative tasks for this purpose. Given a question q and provided evidence E, an extractive QA model $p_\theta(a|q, E)$ aims to select a contiguous span a from E that answers the question, or output a special token if E is not sufficient to answer the question.

Our approach, ReasonBert, is inspired by this formulation and extends it to language model pre-training. The challenge in defining such a self-supervised task is in the creation of question-evidence pairs from unlabeled data. Moreover, we aim for a generic approach that works for a wide range of extractive QA settings including single-hop and multi-hop reasoning, hybrid contexts with both unstructured texts and structured tables, as well as few-shot settings. We discuss how to address the challenge and achieve this goal in the next two sections.

3 Distant Supervision (DS) for Pre-training

We use English Wikipedia as our data source for pre-training. We first extract sentences and tables from Wikipedia pages and then identify salient spans (such as named entities) from them. We apply the idea of distant supervision and match the sentences and tables to form query-evidence pairs, which are used to create pre-training examples.

3.1 Data Collection

Text. We first extract paragraphs from Wikipedia pages and split them into sentences. We consider named entities including both real-world entities (e.g., person, location) and temporal and numeric expressions (e.g., date and quantity) as potential answer entities for pre-training. We first identify real-world entities using existing hyperlinks. Since Wikipedia pages generally do not contain links to themselves, we additionally detect such self-mentions by searching the names and aliases of the topic entity for each page. Temporal and numeric expressions are identified using existing NER tool.

Setting	# queries	# sent.	# tab.	# ent. pairs
Text-only	7.6M	8.4M	-	5.5M
Hybrid	3.2M	4.3M	0.9M	6.0M

Table 1: Statistics about the pre-training data.

Table. We extract tables that are labeled as <wikitable> from Wikipedia, and only consider tables with no more than 500 cells. First, real-world entities are detected using existing hyperlinks. Unlike our method employed for textual sentences, we do not use traditional NER tools here as they are not tailored to work well on tables. Instead, for a cell that does not contain hyperlinks, we match the complete cell value with sentences that are closely related to the table, sourced either from the same page or a page containing a hyperlink pointing to the current page. If the matched span in the sentence contains a named entity, we consider the same entity as being linked to the cell as well. Otherwise we consider this cell as a unique entity in the table.

Please see Appendix A.1 for details about the tools and resources we use.

3.2 Query-Evidence Pairing via DS

As described in Section 2, a standard QA sample is composed of a question, an answer and evidence. The model infers the relationship between the answer and other entities in the question, and extract it from the evidence. In this work, we try to simulate such samples in pre-training. Given a sentence with entities, it can be viewed as a question by masking some entities as answers for prediction. The key issue is then how to find evidence that contains not only the answer entity, but also the relational information for inference. Here we borrow the idea of distant supervision (Mintz et al., 2009b).

Given a sentence as a query, we first extract pairs of entities in it. For each entity pair, we then find other sentences and tables that also contain the same pair as evidence. Since we do not have the known relation constraint in the original assumption of distant supervision, we use the following heuristics to collect evidence that has high quality relational knowledge about the entities and is relevant to the query. First, we only consider entity pairs that contain at least one real-world entity. For textual evidence, the entity pair needs to contain

the topic entity of the Wikipedia page, which is more likely to have relations to other entities. For tabular evidence, we consider only entity pairs that are in the same row of the table, but they do not need to contain the topic entity, as in many cases the topic entity is not present in the tables. In both cases, the query and evidence should come from the same page, or the query contains a hyperlink pointing to the evidence page. For tabular evidence, we also allow for the case where the table contains a hyperlink pointing to the query page.

3.3 Pre-training Data Generation

Given the query-evidence pairs, a naive way to construct pre-training examples is to sample a single piece of evidence for the query, and mask a shared entity as "answer", like Glass et al. (2020). However, this only simulates simple single-hop questions. In this work, we construct complex pre-training examples that require the model to conduct multi-hop reasoning. Here we draw inspiration from how people constructed multi-hop QA datasets. Take HotpotQA (Yang et al., 2018) as an example. It first collected candidate evidence pairs that contain two paragraphs (A, B), with a hyperlink from A to B so that the topic entity of B is a bridging entity that connects A and B. Crowd workers then wrote questions based on each evidence pair. Inspired by this process, we combine multiple pieces of evidence in each pre-training example and predict multiple masked entities simultaneously. The detailed process is described below. Figure 1 shows two examples.

We start by sampling up to two entity pairs from the query sentence and one evidence piece (sentence or table) for each entity pair. We then mask one entity in each pair as the "answer" to predict. The resulting pre-training examples fall into three categories: (1) Two disjoint entity pairs $\{(a, b), (c, d)\}$ are sampled from the query, and one entity from each pair, e.g., $\{a, c\}$, is masked. This is similar to a combination of two single-hop questions. (2) The two sampled entity pairs $\{(a, b), (b, c)\}$ share a common entity b, and b is masked. The model needs to find two sets of entities that respectively satisfy the relationship with a and c, and take an intersection (Type II in HotpotQA; see Ex. 1 in Figure 1). (3) The two sampled entity pairs $\{(a, b), (b, c)\}$ share a common entity b, and $\{b, c\}$ are masked. Here b is the bridging entity that connects a and c. The model needs to first identify b and then recover c based on its relationship with b (Type I and Type III in HotpotQA; see Ex. 2 in Figure 1). We also mask an entity from the query that is not shown in the evidence to simulate unanswerable cases. All sampling is done randomly during pre-training.

We prepare pre-training data for two settings: (1) one with only textual evidence (text-only) and (2) the other including at least one tabular evidence in each sample (hybrid). For the text-only setting, approximately 7.6M query sentences, each containing 2 entity pairs and paired with 3 different textual evidence on average are extracted. For the hybrid setting, we select approximately 3.2M query sentences, each containing 3.5 entity pairs that are paired with 5.8 different evidence on average.

4 Pre-training

4.1 Encoder

In this work, textual and tabular evidence is considered. For the text-only setting, we use the standard transformer encoder in BERT (Devlin et al., 2019). For settings where the input contains tables, we adopt the transformer variant recently introduced in TAPAS (Herzig et al., 2020), which uses extra token-type embeddings (indicating the row/column position of a token) to model the table structure.

4.2 Span Reasoning Objective

Now we describe our *span reasoning objective*, which can advance the reasoning capabilities of a pre-trained model.

Given a sample collected for pre-training as described in Section 3.3, we replace the masked entities $\mathcal{A} = \{a_1, \ldots, a_n\}$ ($n \leq 3$) in the query sentence q with special `[QUESTION]` tokens. The task then becomes recovering these masked entities from the given evidence E (concatenation of the sampled evidence). Specifically, we first concatenate q, E and add special tokens to form the input sequence as $[\texttt{[CLS]}, q, \texttt{[SEP]}, E]$, and get the contextualized representation $\mathbf{x}$ with the encoder. Since we have multiple entities in q masked with `[QUESTION]`, for each a_i, we use its associated `[QUESTION]` representation as a dynamic query vector $\mathbf{x}_{a_i}$ to extract its start and end position s, e of a_i in E (i.e., *question-aware* answer extraction).

$$P(s|q, E) = \frac{\exp\left(\mathbf{x}_s^\top \mathbf{S} \mathbf{x}_{a_i}\right)}{\sum_k \exp\left(\mathbf{x}_k^\top \mathbf{S} \mathbf{x}_{a_i}\right)}$$
$$P(e|q, E) = \frac{\exp\left(\mathbf{x}_e^\top \mathbf{E} \mathbf{x}_{a_i}\right)}{\sum_k \exp\left(\mathbf{x}_k^\top \mathbf{E} \mathbf{x}_{a_i}\right)} \quad (1)$$

Here $\mathbf{S}, \mathbf{E}$ are trainable parameters. $\mathbf{x}_{a_i}$ is the representation of special token `[QUESTION]` corresponding to a_i; $\mathbf{x}_k$ is the k-th token in E. If no answer can be found in the provided evidence, we set s, e to point to the `[CLS]` token.

The *span reasoning* loss is then calculated as follows:

$$L_{GSS} = -\sum_{a_i \in \mathcal{A}} \left(\log P\left(s_{a_i}|q, E\right) + \log P\left(e_{a_i}|q, E\right)\right) \quad (2)$$

We name this objective as *span reasoning*, as it differs from the *span prediction/selection* objectives in existing pre-training work such as SpanBert (Joshi et al., 2020), Splinter (Ram et al., 2021), and SSPT (Glass et al., 2020) in the following ways: (1) Unlike SpanBert and Splinter that use single contiguous paragraph as context, where the models may focus on local cues, we encourage the model to do long-range contextualization by including both query and evidence as input, which can come from different passages, and recovering the masked entities by grounding them on the evidence E. (2) Unlike SSPT, we improve the model's ability to reason across multiple pieces of evidence by including two disjoint pieces of evidence in a single sample and scattering the answer entities among them to mimic different types of reasoning chains. (3) Unlike all existing works, we mimic the scenario where a span cannot be inferred based on the given contexts, by masking entities in q that do not appear in E, in which case the model is trained to select the special `[CLS]` token.

4.3 Final Objective

We also include the *masked language modeling* (MLM) objective in pre-training to leverage other tokens in the input that are not entities. In particular, we randomly mask tokens that are not an entity or token in the header row for tables, and use an MLM objective to recover them. Following the default parameters from BERT, we use a masking probability of 15%.

The final loss is the sum of *grounded span selection* loss and *masked language modeling* loss. Following previous works (Glass et al., 2020; Herzig et al., 2020), we initialize with a pre-trained encoder, and extend the pre-training with our objectives. For the text part, we pre-train two models with BERT-Base (denoted as ReasonBert$_B$) and RoBERTa-Base (denoted as ReasonBert$_R$); for the table part, we use TAPAS-Base (denoted as ReasonBert$_T$). More implementation details of pre-training are included in Appendix A.2.

	MRQA	HotpotQA	NQTables	HybridQA
# train	86136.5	88881	17112	62686
# dev	-	1566	1901	3466
# test	9704	7405	1118	3463
# evidence	1	10	8.7	34.7
# tokens*	374.9	89.1	289.6	156.3
has text/table	✓/✗	✓/✗	✗/✓	✓/✓

Table 2: Dataset statistics. The statistics for MRQA are averaged over all 6 datasets. # tokens* is the average number of tokens per evidence.

5 Experiments

5.1 Datasets

We conduct thorough experiments with a wide range of extractive QA datasets. Statistics are summarized in Table 2.

MRQA (Fisch et al., 2019). A single-hop extractive QA benchmark that unifies various existing QA datasets into the same format. Here we use the in-domain subset that contains 6 datasets: 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) and Natural Questions (Kwiatkowski et al., 2019). Similar to Ram et al. (2021), we adapt these datasets to the few-shot setting by randomly sampling smaller subsets from the original training set for training, and use the original development set for testing.

HotpotQA (Yang et al., 2018). A multi-hop QA dataset that requires reasoning over multiple pieces of evidence. Here we follow the distractor setting, where 10 paragraphs are provided to answer a question while only two of them contain relevant information. We split 10% of the original train-hard split for development, and use the original development set for testing.

NQTables (Kwiatkowski et al., 2019). A subset of the Natural Questions dataset, where at least one answer to the question is present in a table. We extract 19,013 examples from the original training set (307,373 examples) and split them with a 9:1 ratio for training and development. The test set is then created from the original development split (7,830 examples) and contains 1,118 examples. Here we only keep tables from the original Wikipedia article as evidence. Similar subsets are also used in Herzig et al. (2021) and Zayats et al. (2021).

HybridQA (Chen et al., 2020). A multi-hop QA dataset with hybrid contexts. Each example contains a table and several linked paragraphs.

Train. Size	Model	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	Average
16	BERT	9.9±0.6	15.4±1.3	20.5±1.5	6.5±1.2	16.8±1.2	9.6±1.6	13.1
	RoBERTa	10.3±1.1	21.0±3.1	22.5±2.1	6.7±2.0	23.4±3.5	11.2±1.0	15.9
	SpanBERT	15.7±3.6	27.4±4.1	24.3±2.1	8.1±1.4	24.1±3.2	16.3±2.0	19.3
	SSPT	10.8±1.2	21.2±3.8	23.7±4.1	6.5±1.9	25.8±2.6	9.1±1.5	16.2
	Splinter	16.7±5.9	23.9±3.8	25.1±2.8	11.6±1.0	23.6±4.5	15.1±3.5	19.3
	Splinter*	**54.6**	18.9	27.4	**20.8**	26.3	24.0	28.7
	ReasonBert$_B$	33.2±4.0	37.2±2.6	33.1±2.7	11.8±2.3	**46.1**±5.2	22.4±2.8	30.6
	ReasonBert$_R$	41.3±5.5	**45.5**±5.8	**33.6**±3.9	16.2±3.2	45.8±4.5	**34.1**±2.9	**36.1**
128	BERT	21.5±1.4	23.9±0.8	31.7±0.8	11.3±1.3	32.6±2.3	14.0±0.8	22.5
	RoBERTa	48.8±4.2	36.0±2.9	36.4±2.0	22.8±2.4	41.3±2.0	35.2±1.4	36.7
	SpanBERT	61.2±4.7	48.8±6.6	38.8±2.6	31.0±5.3	50.0±3.7	44.0±2.3	45.7
	SSPT	41.5±5.0	30.3±3.7	35.0±2.4	14.0±3.6	42.8±3.5	23.7±3.4	31.2
	Splinter	55.0±10.3	45.7±4.1	41.1±2.7	33.9±2.8	48.8±3.7	46.9±7.1	45.2
	Splinter*	**72.7**	44.7	46.3	**43.5**	47.2	54.7	51.5
	ReasonBert$_B$	58.5±2.2	56.2±0.6	46.7±2.6	27.8±0.6	60.8±1.7	45.2±2.3	49.2
	ReasonBert$_R$	66.7±2.9	**62.1**±0.9	**49.8**±1.6	35.7±1.5	**62.3**±1.7	**57.2**±0.6	**55.6**
1024	BERT	64.1±0.9	41.6±2.6	50.1±0.6	43.0±0.3	53.1±1.0	46.5±1.9	49.7
	RoBERTa	77.9±0.5	62.2±1.3	60.3±0.6	55.0±0.5	67.5±0.8	63.4±0.8	64.4
	SpanBERT	81.1±0.7	67.0±1.0	63.2±0.9	56.4±0.4	70.0±0.8	67.6±1.1	67.5
	SSPT	77.6±1.4	60.1±2.0	58.7±0.7	52.8±1.1	65.9±0.8	63.3±1.6	63.1
	Splinter	79.8±3.5	67.3±1.5	63.8±0.5	54.6±1.4	68.9±0.3	68.4±1.2	67.1
	Splinter*	**82.8**	64.8	**65.5**	**57.3**	67.3	**70.3**	68.0
	ReasonBert$_B$	76.9±0.5	67.4±0.5	63.6±0.6	52.2±0.5	70.6±0.6	67.8±0.5	66.4
	ReasonBert$_R$	79.7±0.3	**70.1**±0.2	65.0±0.9	54.7±0.6	**72.8**±0.4	69.7±0.6	**68.7**

Table 3: Few-shot learning results on MRQA datasets. **Best** and Second Best results are highlighted. We report the average F1 score over five runs. Splinter* is the result reported in the original paper, where the authors use a deeper model with additional transformation layers on top of the encoder.

5.2 Baselines

We conduct a comprehensive comparison of ReasonBert with existing pre-training methods.

BERT (Devlin et al., 2019). A deep transformer model pre-trained with masked languge model (MLM) and next sentence prediction objectives.

RoBERTa (Liu et al., 2019b). An optimized version of BERT that is pre-trained with enlarged text corpus.

SpanBERT (Joshi et al., 2020). A pre-training method designed to better represent and predict spans of text. It extends BERT by masking contiguous random spans, and training the span boundary representation to predict the entire masked span.

SSPT (Glass et al., 2020). A pre-training method designed to improve question answering by training on cloze-like training instances. Unlike ReasonBert, SSPT only masks a single span in the query sentence and predicts it based on an evidence paragraph provided by a separated retriever.

Splinter (Ram et al., 2021). A pre-training method optimized for few-shot question answering, where the model is pre-trained by masking and predicting recurring spans in a passage.

TAPAS (Herzig et al., 2020). A pre-training method designed to learn representations for tables. The model is pre-trained with MLM on tables and surrounding texts extracted from Wikipedia.

For fair comparison, in each task, we use the same model architecture with different pre-trained encoders, which is similar to the one used for span reasoning in pre-training. We append the `[QUESTION]` token to a question and construct the input sequence the same way as in pre-training. We then score all the start, end locations and rank all spans (s, e) (See Eqn. 3 and 4 in Appendix). We use a pre-trained encoder and learn the answer extraction layers ($\mathbf{S}, \mathbf{E}$ in Eqn. 1) from scratch during fine-tuning.

Unless otherwise stated, we use the pre-trained base version so that all models have similar capacity (110M parameters for ReasonBert$_B$, 125M parameters for ReasonBert$_R$, and 111M parameters for ReasonBert$_T$).

5.3 Few-shot Single-hop Text QA

We first experiment with the easier, single-hop MRQA benchmark under the few-shot setting to show that our pre-training approach learns general knowledge that can be transferred to downstream QA tasks effectively. Results are shown in

Table 3. We can see that ReasonBert outperforms pre-trained language models like BERT, RoBERTa and SpanBERT by a large margin on all datasets, particularly with an average absolute gain of 20.3% and 14.5% over BERT and RoBERTa respectively. Compared with pre-training methods like SSPT and Splinter, ReasonBert also shows superior performance and obtains the best results on average. Under the full-data setting, ReasonBert performs competitively and all methods achieve similarly high accuracy. Please refer to Table 8 in Appendix for more details.

5.4 Multi-hop Text QA

To demonstrate that our approach is useful in conducting deep reasoning over multiple contexts, we experiment with the HotpotQA dataset. Here we design a simplified multi-hop QA model that first selects relevant paragraphs as evidence, and then extracts the answer from the top selected evidence. In addition to comparing ReasonBert with other pre-training methods using the same base model, we also show results for HGN (Fang et al., 2020), which is one of the top ranked models on the HotpotQA leaderboard that uses a more sophisticated model design.

Results are shown in Table 4. All models perform very well for evidence selection, with over 96% top 3 recall, but ReasonBert still maintains a slim lead over baselines. ReasonBert provides a 5.3% improvement for BERT and a 1.8% improvement for RoBERTa on overall F1 score, and outperforms all other pre-training methods. ReasonBert also outperforms the HGN model with BERT, but is lower than the one using RoBERTa-Large, which is probably due to simpler design and smaller size of the model. We further experiment under the few-shot setting. Here we focus on the QA performance, so we reuse the evidence selector trained with full data for each model, and train the QA module with different fractions of training data. We can see that the advantage of using ReasonBert is more obvious with limited training data. With 1% of training data, ReasonBert$_R$ obtains F1 score of 63.1%, a 7.1% absolute gain over RoBERTa. Please see Appendix A.3 and B.2 for implementation details and the full few-shot results.

5.5 Table QA

We demonstrate our approach also works with structured data like tables using the NQTables dataset. We first use a text based RoBERTa encoder as baseline, which linearizes a table as a text sequence, by concatenating tokens row by row and separating cells with the `[SEP]` token. We then experiment with the structure-aware encoder from TAPAS and compare the pre-trained TAPAS encoder with the one pre-trained using ReasonBert. Results are shown in Table 5. First, we can see that TAPAS outperforms RoBERTa by 2.3%, demonstrating the importance of modeling the table structure. ReasonBert$_R$ slightly outperforms TAPAS on test set, but ReasonBert$_T$ further boosts F1 to 72.5%, resulting in at least 6.6% absolute gains over existing methods.

Model	Recall		1%		Full	
	Top 2	Top 3	F1	EM	F1	EM
HGN$_{\text{RoBERTa-Large}}$	-	-	-	-	82.2	-
HGN$_{\text{BERT}}$	-	-	-	-	74.8	-
BERT	92.4	96.9	39.8	28.6	71.9	57.9
RoBERTa	93.1	97.5	56.0	43.1	76.3	62.9
SpanBERT	93.6	97.7	56.5	44.1	76.3	62.9
SSPT	93.9	97.9	54.7	41.8	75.4	61.5
Splinter	94.1	97.9	57.0	44.2	76.5	62.5
ReasonBert$_B$	93.8	97.8	57.6	45.3	77.2	63.4
ReasonBert$_R$	**94.0**	**98.0**	**63.1**	**50.2**	**78.1**	**64.8**

Table 4: Results on HotpotQA.

Model	Dev		Test	
	F1	EM	F1	EM
RoBERTa	58.9	52.8	63.6	58.1
ReasonBert$_R$	61.9	56.4	66.3	60.9
TAPAS	64.9	57.8	65.9	59.6
ReasonBert$_T$	**69.2**	**63.5**	**72.5**	**67.3**

Table 5: Results on NQTables.

5.6 Hybrid QA

We further evaluate our approach on HybridQA, a multi-hop question answering dataset using both text and tables as evidences. Chen et al. (2020) proposes a baseline model HYBRIDER that divides the problem into four tasks: linking, ranking, hopping and reading comprehension. We follow their design but simplify the model by merging ranking and hopping into a single cell selection task. We use the linking results from Chen et al. (2020), and then train a table based cell selector to select the cell which is the answer or is linked to the passage that contains the answer. Finally, we train a text based QA model to extract the final answer by taking the table snippet that contains the selected cell, and concatenating it with the hyperlinked passage as evidence. Results are shown in Table 6. First, we can see that our simplified architecture works surprisingly well, with TAPAS for cell se-

Model	Cell Selection		Dev		Test	
	Top 1	Top 2	F1	EM	F1	EM
HYBRIDER$_{\text{BERT-Base}}$	-	-	50.9	43.7	50.2	42.5
HYBRIDER$_{\text{BERT-Large}}$	68.5	-	50.7	44.0	50.6	43.8
TAPAS+RoBERTa	73.3	79.7	64.0	57.3	63.3	56.1
ReasonBert	**76.1**	**81.3**	**67.2**	**60.3**	**65.3**	**58.0**

Table 6: Results on HybridQA.

Model	1024		Full	
	F1	EM	F1	EM
ReasonBert$_{\text{R}}$	65.2	52.8	79.2	65.8
– MLM	63.7	51.3	77.7	64.0
– Unanswerable Ent.	64.4	51.8	78.4	65.0
– Multiple Evidences	60.8	48.6	77.8	64.5

Table 7: Ablation study on HotpotQA.

lection and RoBERTa for QA, we already outperforms HYBRIDER. The performance is further improved by replacing the encoders with ReasonBert$_{\text{T}}$ and ReasonBert$_{\text{R}}$, and substantially outperforms the best model on the leaderboard (52.04 EM) at the time of submission.

6 Ablation Study

We further conduct ablation studies on HotpotQA to verify our design choices in Table 7. Here we remove different components of ReasonBert$_{\text{R}}$ and test them under both the full-data and few-shot setting (with 1024 examples). To save computing resources, here all models are pre-trained with 5 epochs. We can see that combining multiple evidences and predicting multiple masked spans simultaneously brings the most gain, especially under the few-shot setting. This is probably because the setting allows us to simulate complex reasoning chains and encourage the model to do deep reasoning. Masking unanswerable entities and utilizing MLM also help to improve performance.

7 Related Work

Language Model pre-training. Contextualized word representations pre-trained on large-scale unlabeled text corpus have been widely used in NLP lately. Most prevalent approaches are variants of pre-trained language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b). More recently, generative language models like BART (Lewis et al., 2020) and GPT-3 (Brown et al., 2020) have also achieved great success in both generation and comprehension tasks. Meanwhile, there have also been works that use pre-training to accommodate specific needs of downstream NLP tasks, like REALM (Guu et al., 2020) for open-domain retrieval and SpanBERT (Joshi et al., 2020) for representing and predicting spans of text.

Machine Reading Comprehension. Machine reading comprehension (MRC) or extractive QA has become an important testbed for natural language understanding evaluation (Fisch et al., 2019). The conventional method to train an MRC model usually relies on large-scale supervised training data (Chen et al., 2017; Zhang et al., 2020). Recently, more and more works have focused on developing self-supervised methods that can reduce the need of labeled data for more efficient domain adaptation, while achieving the same or even better performance. One direction is question generation (Liangming Pan, 2021), which automatically generates questions and answers from unstructured and structured data sources using rules or neural generators. Recent works also try to directly simulate questions with cloze-like query sentences. Splinter (Ram et al., 2021) proposes to pre-train the model by masking and predicting recurring spans. However, this limits the query and context to come from the same passage. In contrast, SSPT (Glass et al., 2020) also pre-trains with a span selection objective, but uses a separate document retriever to get relevant paragraphs as context.

Our work is most related to SSPT, but uses distant supervision to collect query-evidence pairs and thus obviate the need for retriever. Meanwhile, to encourage the model to learn complex reasoning, we mimic different types of reasoning chains by masking multiple entities, including unanswerable ones, and simultaneously inferring them from disjoint pieces of evidence. Our method also works with heterogeneous sources including both text and tables, while most existing works consider only text-based question answering.

8 Conclusion and Future Work

We propose ReasonBert, a novel pre-training method to enhance the reasoning ability of language models. The resulting model obtains substantial improvements on multi-hop and hybrid QA tasks that require complex reasoning, and demonstrates superior few-shot performance. In the future, we plan to use our query-evidence pairs collected by distant supervision to improve the retrieval performance for open-domain QA, as well as empower ReasonBert to handle more types of reasoning, like comparison and numeric reasoning, in natural language understanding.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Dunn, Levent Sagun, Mike Higgins, V. U. Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv*, abs/1704.05179.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G P Shrivatsa Bhargav, Dinesh Garg, and Avi Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. *arXiv preprint arXiv:2103.12011*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Wenhan Xiong Min-Yen Kan William Yang Wang Liangming Pan, Wenhu Chen. 2021. Unsupervised multi-hop question answering by question generation. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Online.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

I. Loshchilov and F. Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009a. Distant supervision for relation extraction without labeled data. pages 1003–1011. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009b. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. In *Association for Computational Linguistics (ACL)*.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, pages 2962–2968.

Semantic Machines, Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

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, Vancouver, Canada. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Vicky Zayats, Kristina Toutanova, and Mari Ostendorf. 2021. Representations for question answering from documents with tables and text. *arXiv preprint arXiv:2101.10573*.

Zhuosheng Zhang, J. Yang, and Hai Zhao. 2020. Retrospective reader for machine reading comprehension. *ArXiv*, abs/2001.09694.

A Implementation Details

A.1 Pre-training Data Collection Details

We extract paragraphs from Wikipedia XML dump[2] use JWPL[3] and tables use wikitextparser[4]. The paragraphs are then processed with SparkNLP[5] for sentence boundary detection and named entity recognition.

A.2 Pre-training Details

We set the max length of query sentences to 100 tokens and the max length of each evidence sample to 200 if there are two evidence selections or 400 if there is only one. For textual evidence, we include the neighbouring sentences from the same paragraph as extra context for the selected evidence sentence and clip to the max evidence length. For tabular evidences, we take a snippet of the original table, and truncate the cells to 20 tokens. We always keep the first row and column in the table, as they often contain important information such as headers and subject entities. Based on the selected entity pair, we sample up to 5 columns and include as many rows as possible until reaching the budget.

We initialize our encoder with BERT-Base[6] and RoBERTa-Base[7] for the text part, and TAPAS-base[8] for the table part. We train ReasonBert using AdamW (Loshchilov and Hutter, 2019) for 10 epochs with batches of 256 sequences of length 512; this is approximately 290k steps with text-only data, and 120k steps with hybrid data. We base our implementation on Huggingface Transformers (Wolf et al., 2020), and train on a single eight-core TPU on the Google Cloud Platform.

A.3 Fine-tuning Details

To extract the answer span from given evidence, we score all the start, end locations and rank all spans (s, e) by $g(s, e|q, E)$ as follows:

$$f_{\textbf{start}} = \mathbf{x}_s^\top \mathbf{S}\mathbf{x}_q, \quad f_{\textbf{end}} = \mathbf{x}_e^\top \mathbf{E}\mathbf{x}_q \tag{3}$$

$$\begin{aligned} g(s, e|q, E) = & f_{\textbf{start}}(s|q, E) \\ & + f_{\textbf{end}}(e|q, E) \\ & - f_{\textbf{start}}(\texttt{[CLS]}|q, E) \\ & - f_{\textbf{end}}(\texttt{[CLS]}|q, E) \end{aligned} \tag{4}$$

For all fine-tuning experiments, we set the batch size to 20 and use a maximal learning rate of $5 \cdot 10^{-5}$, which warms up in the first 10% of the steps, and then decays linearly. We use the development set for model selection if it is present, otherwise we use the last model checkpoint.

Single-hop text QA. We split the text sequence to fit the max input length by sliding a window with a stride of 128 tokens.

For the few-shot setting, we fine-tune the model for either 10 epochs or 200 steps (whichever is larger). For the fully supervised setting, we fine-tune the model for 2 epochs.

Multi-hop text QA. We design a simplified multi-hop QA model that first selects relevant paragraphs as evidence, and then extracts the answer from the selected evidence samples. Specifically, we first generate all possible paragraphs by sliding a 200-token window over all articles with a stride of 128 tokens. We then train an evidence selector to pick the top 3 evidence samples. As the information for answering a question in HotpotQA is scattered in two articles, we list all possible combinations of paragraphs that come from two different articles and concatenate them together to form the final evidence. We then use the base QA model to extract the answer based on the question and the combined evidence.

We fine-tune the evidence selector model for 2 epochs, and the QA model for 5 epochs with full data. For the few-shot setting, we fine-tune the QA model for 10 epochs with 1&, 5% and 10% of the training data, and for 5 epochs with 25% and 50% of the training data.

Table QA. For the text based model, We split the text sequence to fit the max input length by sliding a window with a stride of 128 tokens. For the table based model, we truncate each cell to 50 tokens, and split the table into snippets horizontally. Same as pre-training, we include the first row and column in each table snippet.

We fine-tune the model for 5 epochs with full data. For the few-shot setting, we fine-tune the QA model for 10 epochs with 1&, 5% and 10% of the

[2] https://dumps.wikimedia.org/

[3] https://dkpro.github.io/dkpro-jwpl/

[4] https://github.com/5j9/wikitextparser

[5] https://nlp.johnsnowlabs.com/

[6] https://huggingface.co/bert-base-uncased/tree/main

[7] https://huggingface.co/roberta-base/tree/main

[8] https://huggingface.co/google/tapas-base/tree/no_reset

training data, and for 5 epochs with 25% and 50% of the training data.

Hybrid QA. Chen et al. (2020) proposes a baseline model that divides the problem into four tasks: 1) linking: link questions to their corresponding cells use heuristics. 2) ranking: rank the linked cells use a neural model. 3) hopping: based on the cell selected in the last step, decide which neighboring cell or itself contains the final answer. 4) reading comprehension: extract the answer from the predicted cell or its linked paragraph. We follow their design and simplify the model by merging ranking and hopping into a single cell selection task. We use the linking results from Chen et al. (2020). For each linked cell, we take a snippet out of the original table including the headers, the entire row of the linked cell, and concatenate the evidence sentence to the cell if it is linked through the hyperlinked passage. To select the cell, we train the model to select separately on the token, row and column level, and aggregate the final scores . More specifically, we calculate the probability of selecting on the token and row level as follows:

$$
\begin{aligned}
P(t|q,E) &= \frac{\exp\left(\mathbf{x}_t^\top \mathbf{S}\mathbf{x}_{a_i}\right)}{\sum_k \exp\left(\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_{a_i}\right)} \\
S_{cell} &= \text{mean}_{x_i \in cell}\left(\mathbf{x}_i^\top \mathbf{R}\mathbf{x}_a\right) \\
P(r_a = j \mid q, E) &= \frac{\exp\left(\max_{cell \in r_j} S_{cell}\right)}{\sum_k \exp\left(\max_{cell \in r_k} S_{cell}\right)}
\end{aligned}
\tag{5}
$$

Here $\mathbf{S}$ is the weight matrix of the token selection header, we only consider the first token in each cell, and t is the first token of the selected cell. $\mathbf{R}$ is the weight matrix of row selection header, and the column selection probability is calculated similarly with another column selection header. We first score each cell by averaging over all tokens in that cell. We then do a max pooling over all cells in the row or column so the model can focus on the strongest signal, for example the column header. The final probability of selecting a cell is the sum of token, row and column scores.

The input for the QA model then contains the header of the table, the row of the selected cell, and the hyperlinked passage.

We fine-tune the cell selection model for 2 epochs and the QA model for 3 epochs.

B More Results

B.1 Single-hop Text QA with Full Data

Results under the fully supervised setting is shown is Table 8. ReasonBert performs competitively and all methods achieve similarly high accuracy. We still bring improvements upon BERT and RoBERTa, and ReasonBertR get second best average score.

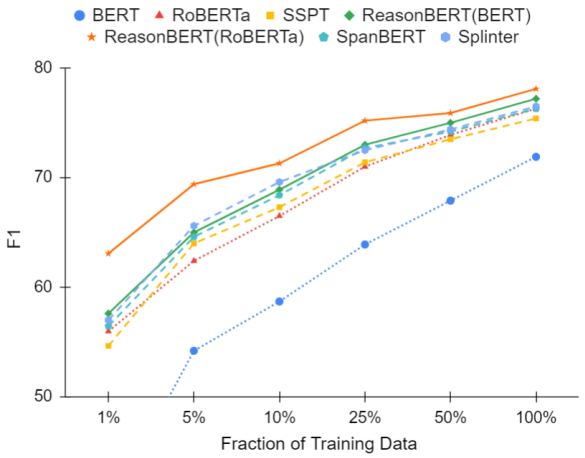

Figure 2: Few-shot learning results on HotpotQA.

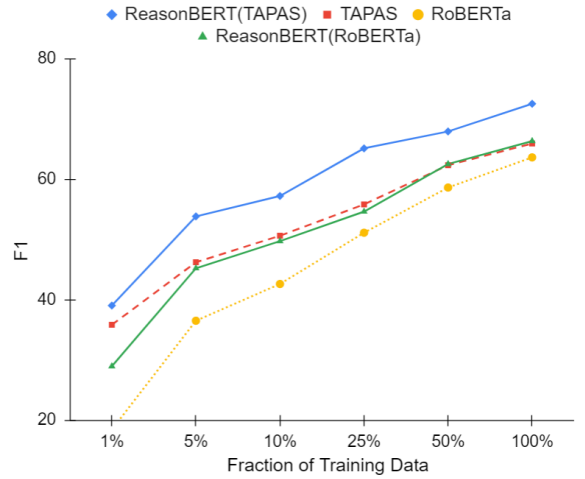

Figure 3: Few-shot learning results on NQTables.

B.2 Few-shot Multi-hop Text QA

Results for training the QA model with different fraction of training data is shown in Figure 2. We can see that ReasonBert obtains larger gain under the few-shot setting.

B.3 Few-shot Table QA

Results for training the Table QA model with different fractions of training data is shown in Figure 3. ReasonBertT consistently outperforms TAPAS while ReasonBertR gradually matches the performance of TAPAS with the increasing of training data.

#	Model	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	Average
	BERT	88.8	73.6	78.7	67.5	82.0	76.2	77.8
	RoBERTa	92.0	78.1	80.6	**71.9**	85.2	79.1	81.2
All	SpanBERT	**92.5**	**79.9**	80.7	71.1	84.8	**80.7**	**81.6**
	SSPT	91.1	77.0	80.0	69.7	83.3	79.7	80.1
	Splinter	92.4	79.7	80.3	70.8	84.0	80.6	81.3
	Splinter*	92.2	76.5	**81.0**	71.3	83.0	**80.7**	80.8
	ReasonBert$_B$	90.3	77.5	79.9	68.7	83.7	80.5	80.1
	ReasonBert$_R$	91.4	78.9	80.8	71.4	**85.3**	80.6	81.4

Table 8: F1 score on MRQA datasets with full data. Splinter* is the result reported in the original paper, where the authors use a deeper model with additional transformation layers on top of the encoder.