FewshotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models

Rakesh Chada
Amazon Alexa AI
rakchada@amazon.com

Pradeep Natarajan
Amazon Alexa AI
natarap@amazon.com

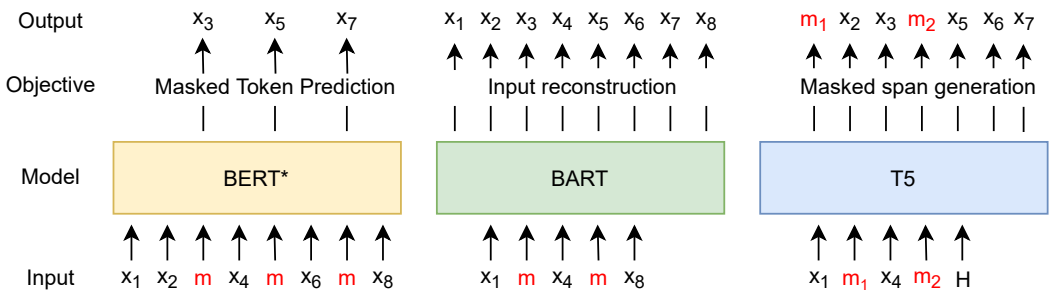
Abstract

The task of learning from only a few examples (called a few-shot setting) is of key importance and relevance to a real-world setting. For question answering (QA), the current state-of-the-art pre-trained models typically need fine-tuning on tens of thousands of examples to obtain good results. Their performance degrades significantly in a few-shot setting (< 100 examples). To address this, we propose a simple fine-tuning framework that leverages pre-trained text-to-text models and is directly aligned with their pre-training framework. Specifically, we construct the input as a concatenation of the question, a mask token representing the answer span and a context. Given this input, the model is fine-tuned using the same objective as that of its pre-training objective. Through experimental studies on various few-shot configurations, we show that this formulation leads to significant gains on multiple QA benchmarks (an absolute gain of 34.2 F1 points on average when there are only 16 training examples). The gains extend further when used with larger models (Eg:- 72.3 F1 on SQuAD using BART-large with only 32 examples) and translate well to a multilingual setting . On the multilingual TydiQA benchmark, our model outperforms the XLM-Roberta-large by an absolute margin of upto 40 F1 points and an average of 33 F1 points in a few-shot setting (<= 64 training examples). We conduct detailed ablation studies to analyze factors contributing to these gains.

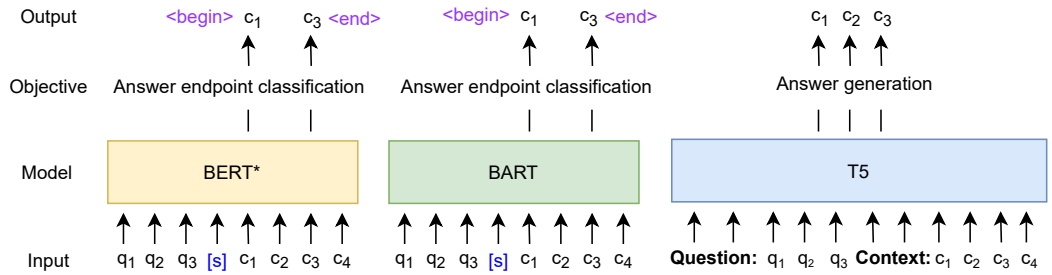
1 Introduction

The task of question answering (QA) in Natural Language Processing typically involves producing an answer for a given question using a context that contains evidence to support the answer. The latest advances in pre-trained language models resulted in performance close to (and sometimes exceeding) a human performance when fine-tuned on several QA benchmarks (Devlin et al., 2019), (Brown et al., 2020), (Bao et al., 2020), (Raffel et al., 2020). However, to achieve this result, these models need to be fine-tuned on tens of thousands of examples. In a more realistic and practical scenario, where only a handful of annotated training examples are available, their performance degrades significantly. For instance, (Ram et al., 2021) show that, when only 16 training examples are available, the Roberta-base (Liu et al., 2019) and SpanBERT-base (Joshi et al., 2020) obtain a F1 score of 7.7 and 18.2 respectively on SQuAD (Rajpurkar et al., 2016). This is far lower than the F1 score of 90.3 and 92.0 when using the full training set of >100000 examples. Through experimental analysis, we observe that this degradation is majorly attributed to the disparities between fine-tuning and pre-training frameworks (a combination of the input-output design and the training objective). And to address this, we propose a fine-tuning framework (referred to as FewshotQA hereby) that is directly aligned with the pre-training framework, in terms of both the input-output design and the training objective. Specifically, we construct the input as a concatenation of the question, a mask token and context (in that order) and fine-tune a text-to-text pre-trained model using the same objective used during its pre-training to recover the answer. These text-to-text pre-trained model(s) were originally trained to recover missing spans of text in a given input sequence. And since our proposed fine-tuning setup is very much identical to the pre-training setup, this enables the model to make the best use of the pre-training "knowledge" for the fine-tuning task of question answering.

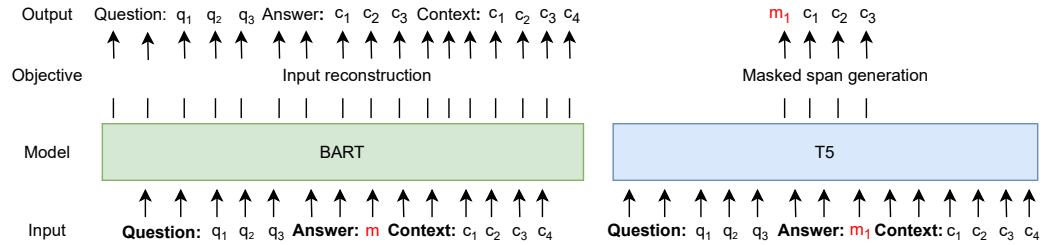
The effectiveness of our FewshotQA system is shown in its strong results (an absolute average gain of 34.2 F1 points) on multiple QA benchmarks in a few-shot setting. We show that the gains extend further when used with larger sized models. We also test FewshotQA on a multilingual benchmark by replacing the pre-trained model with its multi-



(a) Comparison of pre-training frameworks. x_i represent a single token at i^{th} position. m represents a special mask token. T5 uses a different mask token for each span masked (m_1, m_2).



(b) Comparison of existing fine-tuning frameworks. q_i represent the question tokens. c_i represents the context tokens. [s] is a special delimiter symbol. <begin>, <end> represent the begin and end tokens for the answer span.



(c) Our proposed FewshotQA fine-tuning framework. Note that the difference between this and the pre-training framework above are the inputs and outputs.

Figure 1: Comparison of different pre-training and fine-tuning systems for QA tasks with our proposed system. BERT* represents a class of BERT-like models that use the same pre-training objective.

lingual counterpart and observe significant gains in comparison to a strong XLM-Roberta baseline (an absolute gain of 40 F1 points when there are only 16 training examples).

2 Few-shot fine-tuning framework design

Our proposed few-shot fine-tuning framework design involves a different choice of input-output design and the training objective than the current standard for QA fine-tuning frameworks. We provide a motivation for this design by comparison with the existing frameworks. Figure 1 illustrates this in detail. The pre-training framework is also pictured for comparison. Note that we focus on the bi-directional masked language models (MLMs) instead of the auto-regressive language models (such as GPT-2 (Radford et al., 2019)) as the MLMs typically are deemed superior for QA tasks (Devlin et al., 2019), (Lewis et al., 2020).

2.1 Pre-training

Figure 1a illustrates the comparison between pre-training setups for three types of models. Firstly, there are BERT-style encoder-only models (referred to as BERT$*$) (Devlin et al., 2019) that are pre-trained with the standard masked language modeling objective (also called a denoising objective) of predicting the masked tokens in an input sequence I. The masked tokens here typically correspond to a single word or a sub-word. Then, BART (Lewis et al., 2020) uses a corrupted input reconstruction objective to recover the original input. The corruption involves replacing a span of multiple tokens with a mask token and sentence shuffling. Finally, T5 (Raffel et al., 2020) uses a masked span generation objective to predict the masked spans in an input. The input here is similar to that of BART

where multiple spans are replaced with masked tokens. However, instead of generating the full input, only the masked spans are generated.

2.2 Fine-tuning

Figure 1b illustrates the fine-tuning setups for each of these models for the task of question answering. The input to both the BERT-style encoder-only models and BART is a concatenation of the question and the context. And both of them use a similar objective that encourages the model to predict the correct start and end positions of the answer in a given input. This is referred to as a span-selection objective. The input to T5 is also the concatenation of the question and the context. However, T5 uses an answer span generation objective to let the model directly generate the answer from scratch.

2.3 Aligning the fine-tuning with pre-training

The intuition behind aligning the fine-tuning and pre-training frameworks is that the model can make the best use of the "knowledge" obtained during pre-training phase in the fine-tuning phase. For question answering, the fine-tuning task involves predicting an answer span that could contain multiple tokens. This makes it non-trivial to align BERT* models for QA task during fine-tuning as their pre-training objectives let the model predict only a single word (or a sub-word) for a mask token. Similarly, it would require knowing the answer length in advance to have SpanBERT predict multi-masked tokens.
Given that their pre-training objectives naturally involve multi-token span generation, BART and T5 make good candidates for this alignment. We further enhance the alignment by constructing the inputs (and outputs) to be similar to that of what the model sees during pre-training. This is done by appending a mask token (that should correspond to the answer in the target) as part of the input. This framework is illustrated in Figure 1c. We test the effectiveness of our formulation by putting it to test in various few-shot scenarios and observe significant gains.

Overall, we establish that the combination of text-to-text models and fine-tuning framework that is aligned with its pre-training counterpart makes a strong few-shot QA system. We now describe our experimental setup in Section 3.

3 Modeling details

3.1 Architecture

Our model follows the standard pre-trained text-to-text model architecture. It consists of a Transformer-based encoder and decoder models. We default to the "base" versions of the BART and T5 models as they contain a modest number of parameters (140M and 220M respectively) in comparison to the larger sized ones. For T5, we use the T5-V1.1 as the publicly released T5-V1.0 is fine-tuned on downstream tasks including question answering thereby contaminating it for our experimentation. BART-base consists of 6 encoder, 6 decoder layers with a hidden dimension of 768 and T5-base consists of 12 encoder, 12 decoder layers with a hidden dimension of 768 and a feed-forward hidden dimesion of 3072. We call the variants of BART, T5 used with our fine-tuning framework as FewshotBART and FewshotT5 respectively.

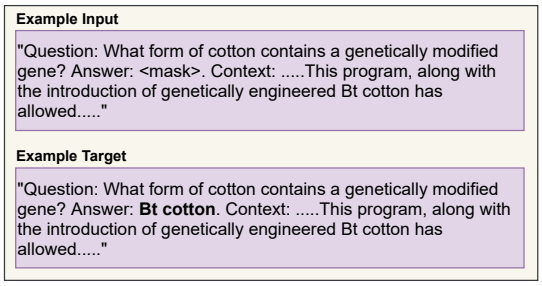


Figure 2: An example showing the input and target design for the FewshotQA model for BART.

3.2 Input-output design & fine-tuning objective

The input (x_M) to the model consists of a concatenation of three text sequences. The first (x_q) is a set of question tokens (q) prefixed with the phrase "Question:", the second (x_a) is a mask token (m) prefixed with the phrase "Answer:" and the third (x_c) is a set of context tokens (c) prefixed with the phrase "Context:".

x_q = Question: q
x_a = Answer: m
x_c = Context: c

$$x_M = [x_q; x_a; x_c]$$

The target for the BART model (y_{BART}) is a concatenation of two text sequences, x_q and y_a where y_a is the set of answer tokens prefixed with the phrase "Answer:"

y_a = Answer: a

$$y_{BART} = [x_q; y_a]$$

The target for our T5 variant is a concatenation of a mask token m and y_a

$$y_{T5} = [m; y_a]$$

An example of the input-target pairs from a dataset is shown in Figure 2.

The choice of text-to-text models in our system allows us to use to the standard encoder-decoder objective that maximizes the log likelihood of the text in the ground truth target from the output of the model. Formally, given the input x_M and the target y (one of y_{BART} and y_{T5}), the loss function L would now be:

$$L(\theta) = \sum_{(x_M, y) \in (X_M, Y)} log(\prod_{i=1}^{n} P(y_i | y_{<i}, x_M; \theta))$$

Here, X_M is the set of inputs, Y is the set of targets, n is the number of tokens in the target sequence. y_i is the target token at timestep i. $y_{<i}$ represents all the target tokens preceding the timestep i. $P(y_i|y_{<i})$ represents the probability of the generating token y_i given all the preceding ground truth tokens $y_{<i}$. And θ represents the parameters of the model.

We chose the order of concatenation in the input as question followed by a mask token followed by context as it enables us to run the generation process for a far fewer number of steps before an answer is generated. (see **Generation Strategy** section below). The context can be quite long so generating the entire context auto-regressively before generating an answer would be inefficient and cause a degradation in performance.

3.3 Generation strategy

During both validation and testing, the model is provided the special start token as input and asked to generate tokens in an auto-regressive manner for a fixed number of steps. For BART, since the question and answer tokens are at the beginning of the sequence in the input and the model is trained to reconstruct the input, we just need to generate until the answer is generated. In practice, for the datasets experimented, the generation length of 50 is sufficient to generate the answer. We stop the generation once these 50 tokens are generated. For T5, the generation length is set to 25 as only the answer is generated. For both, we use greedy decoding with a beam size = 1.

3.4 Answer extraction

Once the outputs are generated, the answers are then extracted via a simple post-processing rule that extracts the answer part of the generation. The use of a fixed input pattern (Question: a Answer: a Context: c) makes this step a simple deterministic one without the need for additional heuristics.

3.5 Multilingual extension

Our fine-tuning framework can be extended to a multilingual question answering setting by switching the pretrained model with its multilingual counterpart. We experiment by replacing BART-base model with mBART-50 model that was pre-trained with the same objective as that of BART on a multilingual corpus. The rest of the components, fine-tuning objective and the answer extraction, remain the same as that of the FewshotBART. We call this model FewshotmBART.

3.6 Hyperparameters

We use Adam optimizer with a learning rate of 2e-5. We use a training batch size of 4. We don't use learning rate scheduling. For evaluation on the test set, we pick the best model based off the development set performance. The maximum sequence length is set to the 99th percentile length of all sequence lengths in the development set. We train for a total of 35 epochs or 1000 steps (whichever is the maximum).

4 Experiments

4.1 Datasets

We follow (Ram et al., 2021) and choose the datasets sampled from the MRQA shared task (Fisch et al., 2019) for our few-shot experiments. We also use the same train and test splits provided in (Ram et al., 2021) for fine-tuning and evaluating our models. However, instead of fine-tuning for a fixed number of iterations, we use a development set to determine the best checkpoint to use for testing. These datasets contain 5000 to 17000 test examples.

Development data split: To cater to a realistic and a practical setting for a few-shot scenario, we pick the development set to be the same size

Model	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	BioASQ	TbQA
16 Examples								
BART	10.44±5.9	2.5±2.1	13.3±2.4	2.9±1.4	5.3±2.4	5.6±2.4	10.3±2.9	1.7±1.2
FewshotBART	**55.5**±2.0	**50.5**±1.0	**46.7**±2.3	**18.9**±1.8	**39.8**±0.04	**45.1**±1.5	**49.4**±0.02	**19.9**±1.25
128 Examples								
BART	42.2±3.4	11.1±0.9	24.6±1.9	18.1±1.3	17.8±2.3	23.0±2.5	39.1±2.8	9.2±0.7
FewshotBART	**68.0**±0.3	**50.1**±1.8	**53.9**±0.9	**47.9**±1.2	**58.1**±1.4	**54.8**±0.8	**68.5**±1.0	**29.7**±2.4

Table 1: Comparison of F1 scores across all datasets for the standard QA fine-tuning objective (BART) vs the proposed aligned fine-tuning objective (FewshotBART). The value after ± indicates the standard deviation across 5 runs with different seeds. NQ stands for Natural Questions. TbQA stands for TextbookQA.

as of the training set. As reported in (Gao et al., 2021), having access to the full development set during training would create an unrealistic few-shot setting. We also make sure there is no overlap between training and development sets.

Below, we describe results for several experiments conducted on MRQA few-shot datasets. We run each experiment five times using five different random seeds. And we report the mean and standard deviation of the results for each run.

4.2 Comparing the standard vs aligned fine-tuning framework

First, we present results comparing the standard QA span-selection fine-tuning framework (BART) and our proposed fine-tuning framework (FewshotBART) that uses an input-output and an objective that is aligned with pre-training framework. We choose BART, two training set sizes (16, 128 examples) to illustrate this and present elaborate results across all configurations in a further section. Both BART and FewshotBART use the base version which contains 140M parameters. As seen in Table 1, our proposed fine-tuning framework improves the F1 score significantly across all datasets in both the 16 example (an absolute gain of upto 48 F1 points and an average of 34.2 F1 points) and 128 example scenarios (an absolute gain of upto 39 F1 points and an average of 30.8 F1 points).

4.3 Few-shot results

Next, we present detailed experimental results (Table 2) obtained with our FewshotBART, FewshotBARTL, FewshotT5 models on several few-shot configurations across multiple datasets. For FewshotBARTL, the base model in FewshotBART is replaced with the larger 406M parameter model. We compare our models with RoBERTa, SpanBERT baselines and the recently proposed Splinter (Ram et al., 2021). The RoBERTa and SpanBERT are fine-tuned with the span-selection objective and we use the results for these models from (Ram et al., 2021).

We can see that FewshotBART, FewshotBARTL and FewshotT5 outperform the baselines by a big margin on almost all datasets. A few highlights are listed below:

- (a) Our best large model (FewshotBARTL) outperforms all other models by a big margin. Specifically, in a 16 example setting, it provides gains upto 61.2 F1 points in comparison to a similar-sized RoBERTa model that is fine-tuned with a span-selection objective.
- (b) Our best comparable model to Splinter (in terms of model size) - FewshotBART outperforms it by upto 31.6 F1 points in a 16 example setting and upto 10.9 F1 points in a 128 example setting. TextbookQA dataset is one exception where Splinter is stronger.
- (c) FewshotBART is stronger in a 16 example setting in comparison to FewshotT5. This difference starts fading in 32, 64 and 128 example settings. However, FewshotT5 still performs better on most of the datasets in comparison to Splinter in 16, 32 and 64 example settings.

4.4 Choice of input-outputs and fine-tuning objectives

In this section, we investigate the impact of changing the input-output design and the fine-tuning objective on the model performance (see Figure 3). Given a question q, answer a and context c, we evaluate the following input-output choices and objectives:

Span-selection: This is the standard extractive question answering objective where the model

Model	SQuAD	TriviaQA	NQ	NewsQA	SearchQA	HotpotQA	BioASQ	TbQA
16 Examples								
FewshotBART	**55.5**±2.0	50.5±1.0	**46.7**±2.3	**38.9**±0.7	**39.8**±0.04	**45.1**±1.5	**49.4**±0.02	19.9±1.25
FewshotBARTL	**68.9**±2.7	**65.2**±1.8	**60.4**±2.0	**48.4**±2.2	**47.8**±5.4	**58.0**±1.8	**63.0**±1.1	**37.7**±3.7
FewshotT5	47.8±6.9	**50.6**±4.9	28.5±14.5	26.8±2.7	37.0±3.3	44.9±3.5	46.3±5.9	**25.9**±5.0
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
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
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
32 Examples								
FewshotBART	56.8±2.1	**52.5**±0.7	**50.1**±1.1	**40.4**±1.5	41.8±0.02	47.9±1.4	52.3±0.02	22.7±2.3
FewshotBARTL	**72.3**±1.0	**65.1**±1.2	**61.5**±1.7	**51.7**±1.7	**58.3**±1.5	**60.4**±0.2	**67.8**±1.0	**37.7**±9.8
FewshotT5	56.6±1.5	50.2±9.0	37.5±12.5	33.2±4.6	**48.4**±5.6	**53.6**±1.4	**57.7**±4.2	**29.8**±2.6
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
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
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
64 Examples								
FewshotBART	61.5±2.3	50.8±2.2	**53.0**±0.5	**42.7**±2.2	46.1±2.9	51.2±1.0	61.8±2.8	27.6±1.8
FewshotBARTL	**73.6**±1.9	**64.6**±1.4	**63.0**±2.1	**53.5**±0.9	**65.5**±2.4	**62.9**±1.6	**73.9**±0.8	**45.0**±1.7
FewshotT5	57.2±5.6	**52.4**±5.9	48.6±2.1	40.2±4.1	**54.4**±3.0	**56.3**±2.9	**63.8**±2.5	32.1±2.7
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
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
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
128 Examples								
FewshotBART	68.0±0.3	50.1±1.8	**53.9**±0.9	**47.9**±1.2	**58.1**±1.4	54.8±0.8	**68.5**±1.0	29.7±2.4
FewshotBARTL	**79.4**±1.5	**65.8**±0.9	**64.3**±1.3	**57.0**±0.9	**67.7**±1.0	**75.1**±1.5	**75.0**±1.5	**48.4**±2.7
FewshotT5	64.6±6.1	**51.7**±3.1	47.0±4.6	40.0±1.9	57.0±4.5	**56.1**±3.7	68.2±3.6	33.6±2.1
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
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
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

Table 2: F1 scores across all datasets and training set sizes. Bold indicates the best result. FewshotBARTL's results are colored blue as it has many more parameters (406M) than Splinter (110M). FewshotBART has a comparable number of parameters (130M). FewshotT5 has 220M parameters.

is made to predict begin and end tokens corresponding to the answer in a given input I: Question: q [S] Context: c

Full input generation: This is the objective where the model is made to predict the entire input including the masked answer span.

Input: Question: q Answer: <mask>. Context: c

Target: Question: q Answer: a. Context: c

Question->Answer generation: In this setting, the model is asked to generate only the question and the (masked) answer part of the input. The question tokens are followed by the answer tokens. The context tokens are not included in the fine-tuning objective.

Input: Question: q Answer: <mask>. Context: c

Target: Question: q Answer: a.

Answer->Question generation: This is similar to the **Question->Answer generation** objective with the difference being that the answer tokens are followed by the question tokens.

Input: Question: q Answer: <mask>. Context: c

Target: Question: q Answer: a.

Answer generation: This is the generation-based objective equivalent of the standard span-selection objective. The model is asked to generate the answer given an input question and context.

Input: Question: q Context: c

Target T: a

The results are plotted in Figure 3. We choose SQuAD and NewsQA datasets for illustration. There are several key findings here, in the context of a few-shot QA setting (upto 128 examples):

- (a) The fine-tuning objectives that are aligned with their pre-training objective (red, green, violet lines) show large gains over the standard span-selection fine-tuning objective (blue line). The answer generation objective (orange line) is superior to a span-selection based objective (blue line) on both the datasets.
- (b) The sequencing of question and answer tokens in the input-output has an impact on

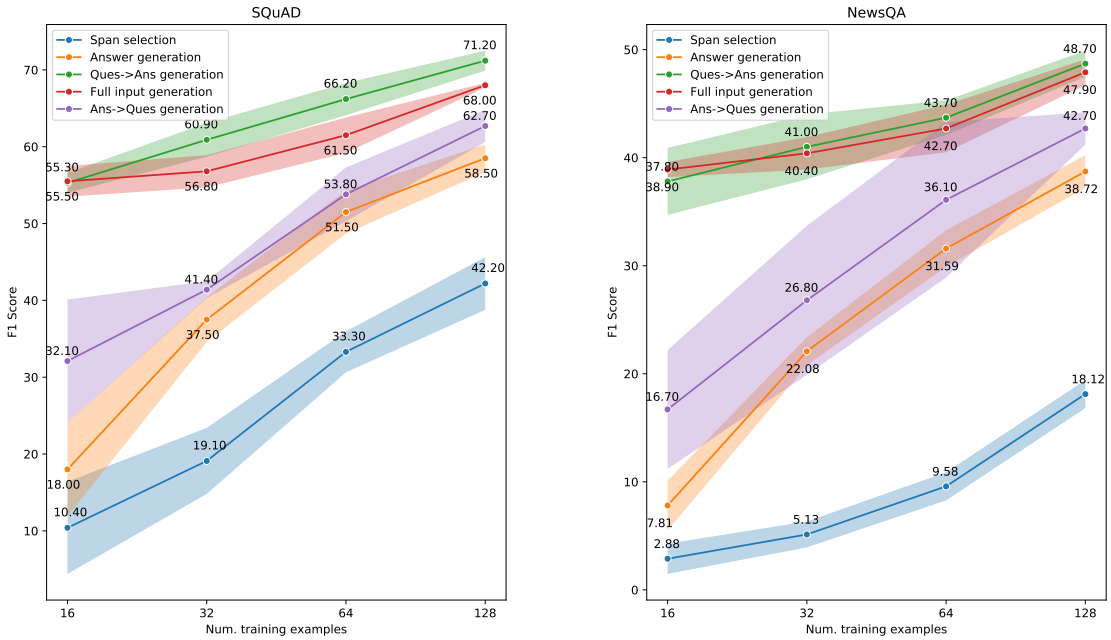


Figure 3: Comparison of fine-tuning objectives. The value on the markers indicates the mean. The beam width indicates the standard deviation.

the performance with the specific sequencing of question followed by the answer being superior.

- (c) The Question->Answer generation and Full input generation objectives show strong performance even when there are 16 examples. The gap between the span-selection and other objectives continue to be large even when there are 128 training examples.

4.5 Multilingual results

Here, we describe the results of applying FewshotmBART described in Section to a multilingual corpus TydiQA (Clark et al., 2020a). TydiQA consists of question answering datasets from 9 languages (Arabic, Bengali, English, Finnish, Indonesian, Korean, Russian, Swahili, Telugu). We compare this to the results from applying XLM-Roberta-large (Conneau et al., 2020) to the same dataset. We fine-tune XLM-Roberta-large using the standard span-selection objective used in extractive question answering tasks. The results are shown in Figure 4.

FewshotmBART outperforms XLM-Roberta-large by an average of 32.96 absolute F1 points for training data sizes spanning from 2 to 64 examples.

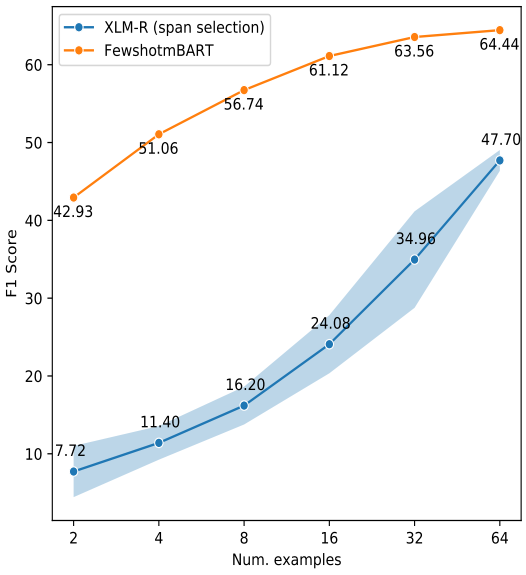


Figure 4: F1 comparison: XLM-Roberta vs FewshotmBART

5 Related Work

Question Answering (QA) is an active area of research in Natural Language Processing and the recent advances in pre-trained language models enabled lots of rapid progress in the field ((Devlin et al., 2019), (Brown et al., 2020), (Bao et al., 2020), (Raffel et al., 2020)). QA is also used as a format to cast several NLP problems (McCann et al., 2018), (Chada, 2019). A common way to build a high performing question answering model is to fine-tune these pre-trained models on the entire training dataset - either via a span-extraction objective (Lan et al., 2020), (Clark et al., 2020b), (Bao et al., 2020) or a span-generation objective (Raffel et al., 2020). However, in this work, we explore a more challenging and a practical setting where only a handful of annotated training and development samples are available. Related to this, (Ram et al., 2021) develop a new pretrained model that uses a recurring span selection objective suitable for QA tasks. They then fine-tune this customized pre-trained model on downstream QA tasks using the standard span selection objective. They argue that the existing strategy of fine-tuning large language models fail in a few-shot QA setting. In contrast, we take existing pre-trained text-to-text models BART (Lewis et al., 2020), T5 (Raffel et al., 2020) and simply modify their fine-tuning objective to build a stronger few-shot QA model. As our solution only relies on fine-tuning modifications, we are able to easily extend the framework to larger sized models and multilingual settings without having to build a new pre-trained model each time.

An alternative line of work that caters to building question answering models in low-data settings involves dataset synthesis (Lewis et al., 2019), (Alberti et al., 2019), (Puri et al., 2020). (Lewis et al., 2019) generate pairs of synthetic context, question and answer triples by sampling context paragraphs from a large corpus of documents. Using these, they generate answer spans, mask the answer and use this cloze-style text to generate natural questions. To do this, they assume access to NLP tools such as named entity recognizer and part of speech tagger. Puri et al. (2020) use a mix of BERT-based answer generation, GPT-2 (Radford et al., 2019) based question generation and and roundtrip filtration to train an extractive QA model. They show promising results with larger scale models. However, the QA model is still fine-tuned on the entire dataset and this entire process including synthetic data generation is computationally expensive. Our work deviates from these by not relying on additional synthetic data, not assuming access to external NLP tools and using only a few training examples for fine-tuning.

There is also a connection of our work to some of the recent developments in few-shot learning for classification tasks that cast the problem as a mask-filling problem (Schick and Schütze, 2021), (Gao et al., 2021), (Schick and Schutze, 2020). However, these solutions are geared towards classification tasks with a fixed set of classes. That assumption doesn't hold true for QA tasks.

6 Conclusion

We present an effective few-shot question answering (QA) system that combines the use of pre-trained text-to-text models and a fine-tuning framework aligned with their pre-training counterpart. Through experimental studies on various QA benchmarks and few-shot configurations, we show that this system can produce significant gains including in scenarios where the training data is extremely scarce (an absolute gain of 34 F1 points on average in comparison to the current standard of the fine-tuning framework). We also present extensions to multilingual and larger model settings and show that the gains translate well to these settings (eg:- up to an absolute 40 F1 point gain in comparison to XLM-Roberta + a span-selection objective). Through ablation studies, we study the impact of model size, fine-tuning objectives, input-output design and illustrate the factors leading to such strong gains. For future, as our framework doesn't explicitly enforce the answer to be a span in the input text, it'd be interesting to consider its applications to generative QA tasks.

Acknowledgements

We thank Siddhant Garg and Karthik Gopalakrishnan for their valuable feedback and discussions. We thank Ori Ram for providing additional data on their Splinter experiments.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–

6173, Florence, Italy. Association for Computational Linguistics.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, X. Liu, Yu Wang, Songhao Piao, Jianfeng Gao, M. Zhou, and H. Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*.

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.

Rakesh Chada. 2019. Gendered pronoun resolution using BERT and an extractive question answering formulation. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 126–133, Florence, Italy. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020a. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020b. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 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.

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.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, 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.

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 *International Conference on Learning Representations*.

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.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. 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. *CoRR*, abs/1907.11692.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

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

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

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3066–3079. Association for Computational Linguistics.

Timo Schick and H. Schutze. 2020. It's not just size that matters: Small language models are also few-shot learners. *ArXiv*, abs/2009.07118.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.