

UNIFIEDQA: Crossing Format Boundaries with a Single QA System

Daniel Khashabi¹ Sewon Min² Tushar Khot¹ Ashish Sabharwal¹
Oyvind Tafjord¹ Peter Clark¹ Hannaneh Hajishirzi^{1,2}

¹Allen Institute for AI, Seattle, U.S.A.
²University of Washington, Seattle, U.S.A.

Abstract

Question answering (QA) tasks have been posed using a variety of formats, such as extractive span selection, multiple choice, etc. This has led to format-specialized models, and even to an implicit division in the QA community. We argue that such boundaries are artificial and perhaps unnecessary, given the reasoning abilities we seek to teach are not governed by the format. As evidence, we use the latest advances in language modeling to build a *single pre-trained QA model*, UNIFIEDQA, that performs well across 20 QA datasets spanning 4 diverse formats. UNIFIEDQA performs on par with 8 different models that were trained on individual datasets themselves. Even when faced with 12 unseen datasets of observed formats, UNIFIEDQA performs surprisingly well, showing strong generalization from its out-of-format training data. Finally, fine-tuning this pre-trained QA model into specialized models results in a new state of the art on 10 factoid and commonsense QA datasets, establishing UNIFIEDQA as a strong starting point for building QA systems.¹

1 Introduction

Question answering is a common tool for assessing how well can computers understand language and reason with it. To this end, the NLP community has introduced several distinct datasets, with four popular *QA formats* illustrated in Fig. 1. For instance, some datasets expect the answer to be “yes” or “no”, or a unique answer span in the associated paragraph (as opposed to multiple or no spans). These differences have motivated their study in silos, often encoding QA format into the model architecture itself. Efforts to exploit multiple datasets remain largely restricted to a single format. For example, Clark et al. (2019c) limit consideration to

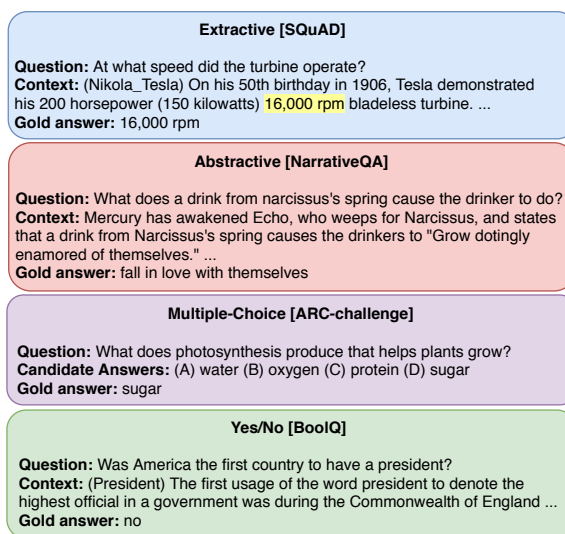


Figure 1: Four formats (color-coded throughout the paper) commonly used for posing questions and answering them: **Extractive (EX)**, **Abstractive (AB)**, **Multiple-Choice (MC)**, and **Yes/No (YN)**. Sample dataset names are shown in square brackets. We study generalization and transfer across these formats.

multiple-choice datasets, while Talmor and Berant (2019) focus their generalization study on extractive span prediction models. To the best of our knowledge, no single QA system targets, not to mention excels at, all of these formats.

This raises the question: *Can QA models learn linguistic reasoning abilities that generalize across formats?* Our intuition is simple: while question format and relevant knowledge may vary across QA datasets, the underlying linguistic understanding and reasoning abilities are largely common. A multiple-choice model may, therefore, benefit from training on an extractive answers dataset. Building upon this intuition, we present a single pre-trained QA system, named UNIFIEDQA, that exploits information across 4 different QA formats to achieve strong performance across 20 different factoid and

¹ <https://github.com/allenai/unifiedqa>

Datasets	SQuAD11	SQuAD2	NewsQA	Quoref	ROPES	NarQA	DROP	NatQA	RACE	MCTest	OBQA	ARC	QASC	CQA	WG	PIQA	SIQA	BoolQ	NP-BoolQ	MultiRC
Format	Extractive QA (EX)					Abstractive QA (AB)			Multiple-choice QA (MC)									Yes/NO QA (YN)		
Has paragraphs?	✓	✓	✓	✓	✓	✓	✓		✓	✓								✓	✓	✓
Has explicit candidate ans?									✓	✓	✓	✓	✓	✓	✓	✓	✓			
# of explicit candidates									4	4	4	4	8	5	2	2	3			
Para contains ans as substring?	✓	✓	✓	✓																
Has idk questions?		✓																		

Figure 2: Properties of various QA datasets included in this study: 5 extractive (EX), 3 abstractive (AB), 9 multiple-choice (MC), and 3 yes/no (YN). ‘idk’ denotes ‘I don’t know’ or unanswerable questions. BoolQ represents both the original dataset and its *contrast-sets* extension BoolQ-CS; similarly for ROPES, Quoref, and DROP.

commonsense QA datasets listed in Fig. 2.

In this work, we advocate for a unifying view of QA formats by building a format-agnostic QA system. Our work leverages recent progress in text-to-text pre-trained neural models, specifically T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), but with a strong focus on differing QA formats. This paradigm allows unifying many NLP models, which formerly had task-specific designs, into a single text-to-text framework. Previous work uses textual prefixes to explicitly define the task associated with each input instance (Raffel et al., 2020; Radford et al., 2019b); often such attempts to build a single model for multiple NLP tasks underperform the standard pre-training plus fine-tuning setup (a model per task) (Raffel et al., 2020).

Our work narrows down the scope to tasks that stay within the boundaries of QA, demonstrating that a unified text-to-text paradigm can, in fact, be successful across different QA tasks and formats. We develop a single pre-trained QA model by training text-to-text models on a set of seed QA datasets of multiple formats, taking natural text as input, without using format-specific prefixes. Our experiments show that UNIFIEDQA can be applied as-is to different QA tasks, generalizes well to other unseen datasets (zero-shot), and with further fine-tuning achieves state-of-the-art results on many QA tasks including commonsense and factual datasets.

Contributions. This work advocates for a unified view of different QA formats, and for building format-agnostic QA systems. To support this view, we present UNIFIEDQA, a single pre-trained QA system that works well on and generalizes to datasets with different formats (§6.2), while performing on par with state-of-the-art dedicated systems tailored to each dataset (§6.1). Additionally, fine-tuning UNIFIEDQA into specialized systems sets a new state of the art for 10 datasets (§6.3), establishing it as a powerful starting point for QA research. Our findings demonstrate that crossing QA format boundaries is not only qualitatively de-

sirable but also quantitatively beneficial.

2 Related Work

Several QA efforts have studied generalization across datasets of a *single* format. For instance, in MultiQA, Talmor and Berant (2019) study generalization and transfer, but only across extractive span selection datasets. Further, while they show strong leave-one-out style results, they find a single system performs substantially worse than one tuned to each dataset. In ORB, Dua et al. (2019a) propose a multi-dataset evaluation benchmark spanning extractive and abstractive formats. However, that study is limited to an *evaluation* of systems, falling short of addressing how to build such generalized models. The MRQA shared task (Fisch et al., 2019) focuses on span-prediction datasets. Unlike all these efforts, our goal is to investigate transfer and generalization across different QA formats, as well as to build a single system that does this well.

Exploiting commonality across machine learning tasks has a rich history studied under transfer learning (Caruana, 1997; Clark et al., 2019b). McCann et al. (2018) and Keskar et al. (2019) study transfer among various NLP tasks by casting them into a single QA format—an elegant transfer learning approach but orthogonal to the goal of this work. As noted earlier, Raffel et al. (2020) investigate the transfer between several diverse NLP tasks (machine translation, summarization, etc). Their key contribution is a text-to-text framework, and a powerful model called T5, that makes it easier to mix multiple tasks by encoding both inputs and outputs as text. They rely on textual prefixes to explicitly define the task corresponding to each input instance. While we build upon their framework, we narrow our focus to variations of QA. This allows us to achieve strong results while avoiding reliance on any format-specific prefixes. Our models *learn to infer* the format of each input question based on its content (e.g., whether the phrasing of the question demands a yes/no answer). Moreover, we are able to demonstrate generalization across QA tasks,

which prior work failed to achieve presumably due to its focus on too broad a set of NLP tasks.

3 UNIFIEDQA: Multi-format Training

Suppose we would like to train a unified QA model that can operate over k formats F_1, F_2, \dots, F_k . For each format F_i , suppose we have ℓ_i datasets sets $D_1^i, D_2^i, \dots, D_{\ell_i}^i$ where $D_j^i = (T_j^i, E_j^i)$ includes both training and evaluation examples. In some cases, the training set T_j^i may be empty or we may want to ignore it in order to treat D_j^i as an ‘unseen’, *evaluation-only* dataset and assess a model’s generalization to it.

We use the text-to-text paradigm to convert each training question q in format F_i into a *plain-text* input representation $enc_i(q)$. This conversion uses a natural encoding process that will be described shortly (§3.1) for four common QA formats, and is easily extensible to other formats as well. We follow a simple approach of creating a mixed training pool consisting of all available training instances:

$$\tilde{T} = \bigcup_{i=1}^k \bigcup_{j=1}^{\ell_i} \left\{ enc_i(q) \mid q \in T_j^i \right\}$$

Training batches are drawn from this pooled data, \tilde{T} , by including each $q \in T_j^i$ with a probability proportional $1/|T_j^i|$. Each batch thus, on average, contains the same number of instances from each training set, regardless of its size. Similar treatments of task mixing have also been adopted by Arivazhagan et al. (2019) and Raffel et al. (2020). As our experiments will show, our multi-format mixing approach works well. It clearly highlights the value of training on out-of-format data and confirms our intuition that there are strong ties across QA formats in terms of the underlying reasoning abilities.²

Our unified question-answering system is based on the recent text-to-text frameworks, particularly, T5 (Raffel et al., 2020) and BART (Lewis et al., 2020). We first define a unifying encoding of the instances across various formats (§3.1). We then introduce UNIFIEDQA (§3.2) that is a QA system trained on datasets in multiple formats, indicating new state-of-the-art results on 10 datasets and generalization to unseen datasets.

²A more sophisticated teaching curriculum (Sachan and Xing, 2016) or approaches such as model distillation and teacher annealing (Clark et al., 2019b) are likely to further improve the performance of the resulting unified model, bolstering the strength of our advocacy for a unified view of all QA formats. We leave their exploration to future work.

3.1 Text-to-Text Encoding

We convert each of our target datasets into a text-in/text-out format (Raffel et al., 2020; Lewis et al., 2020; Radford et al., 2019b). The question always comes first, followed by some additional information (context paragraph or candidate answers, or both). We use “\n” separators between different parts of the input. This ensures having a human-like encoding while not making it overly-specific to a certain format.

Our unified model incorporates the following four common question-answering formats. Specific datasets within them are deferred to Section 4.1.

Extractive (EX) questions Q include a context paragraph C (typically a paragraph) and require models to extract the answer as a substring from the context. In some datasets, ‘unanswerable’ can sometimes be the correct response.

Abstractive (AB) questions Q require models to produce answers that are often not mere substrings of the provided context paragraph C .

Multiple-choice (MC) questions Q come with a set of candidate answers $\{A_i\}$, of which generally exactly one is correct. In some cases, they also include a context paragraph C .

Yes/No (YN) questions Q expect a ‘yes’ or ‘no’ answer as the response and may include a context paragraph C .

Table 1 provides examples of the natural input and output encoding for each of these formats, where both input and output representations are raw text. There is no explicit information regarding a question being an MC question or having exactly four candidate answers. Specifically, MC questions without any context paragraph are encoded as `question \n (A) c1 (B) c2 ...` where `c1, c1, ...` are the set of candidate answers (see the example from ARC dataset). If the question includes a context paragraph, it is appended after the candidate answers: `question \n (A) c1 (B) c2 ... \n paragraph`, as shown in the example from the MCTest dataset. Questions in the other three formats (EX, AB, and YN) are encoded simply as `question \n paragraph`.

To re-emphasize, unlike prior work (Raffel et al., 2020), we do not specify any task-, dataset-, or format-specific prefixes in the input representation. Whether the answer should be extracted or abstracted, and whether from the provided context paragraph or candidate answers (or the fact that

EX	Dataset	SQuAD 1.1
	Input	At what speed did the turbine operate? \n (Nikola_Tesla) On his 50th birthday in 1906, Tesla demonstrated his 200 horsepower (150 kilowatts) 16,000 rpm bladeless turbine. ...
	Output	16,000 rpm
AB	Dataset	NarrativeQA
	Input	What does a drink from narcissus's spring cause the drinker to do? \n Mercury has awakened Echo, who weeps for Narcissus, and states that a drink from Narcissus's spring causes the drinkers to "Grow dotingly enamored of themselves." ...
	Output	fall in love with themselves
MC	Dataset	ARC-challenge
	Input	What does photosynthesis produce that helps plants grow? \n (A) water (B) oxygen (C) protein (D) sugar
	Output	sugar
	Dataset	MCTest
	Input	Who was Billy? \n (A) The skinny kid (B) A teacher (C) A little kid (D) The big kid \n Billy was like a king on the school yard. A king without a queen. He was the biggest kid in our grade, so he made all the rules during recess. ...
	Output	The big kid
YN	Dataset	BoolQ
	Input	Was America the first country to have a president? \n (President) The first usage of the word president to denote the highest official in a government was during the Commonwealth of England ...
	Output	no

Table 1: Example text-to-text encoding of instances.

these even are candidate answers) is expected to be inferred by the system.

3.2 UNIFIEDQA: The Pre-Trained Model

The specific pre-trained QA model we provide and use in all our experiments is trained on representative datasets for each of the 4 formats discussed earlier. We empirically chose the following 8 *seed datasets* for training UNIFIEDQA,³ based on their effectiveness in our pilot study (details deferred to Section 5) assessing which datasets are most valuable for out-of-format training:

- EX: SQuAD 1.1, SQuAD 2.0
- AB: NarrativeQA
- MC: RACE, ARC, OBQA, MCTest
- YN: BoolQ

One can easily use other combinations of formats and datasets to create variants of our UNIFIEDQA model, or extend it as future datasets become available or new formats are introduced.

Unless otherwise noted, we use the largest available T5 model (11B parameters) as the starting point for training our model and call the system UNIFIEDQA. We also report results of training our system with BART_{large}, referred to as UNIFIEDQA_{BART} (see §6.3). Details on the parameters of the models used are deferred to Appendix A.2.

³Future references to ‘seed dataset’ point to the QA datasets used in this section.

Similar to pre-trained language models, the resulting pre-trained QA model can be used as a starting point for fine-tuning on other QA datasets.

4 Formats and Datasets

4.1 Datasets

We evaluate UNIFIEDQA on 20 existing datasets that target different formats as well as various complex linguistic phenomena. Fig. 2 summarizes key properties of our datasets (whether it comes with a paragraph or answer candidates, whether the paragraph explicitly contains the answer, etc). Most importantly, they are grouped into several formats/categories as described below. Table 2 gives certain statistics of these datasets. We next provide a summary enumerating these datasets, with additional details deferred to Appendix A.1.

Extractive QA (EX). Among the datasets in this popular format, we adopt SQuAD 1.1 (Rajpurkar et al., 2016), SQuAD 2 (Rajpurkar et al., 2018), NewsQA (Trischler et al., 2017), Quoref (Dasigi et al., 2019), ROPES (Lin et al., 2019).

Abstractive QA (AB). The datasets used from this format are: NarrativeQA/NarQA (Kociský et al., 2018), the open-domain version of NaturalQuestions/NatQA (Kwiatkowski et al., 2019), and DROP (Dua et al., 2019b).

Multiple-choice QA (MC). We use the following MC datasets: MCTest (Richardson et al., 2013), RACE (Lai et al., 2017), OpenBookQA/OBQA (Mihaylov et al., 2018), ARC (Clark et al., 2018, 2016), QASC (Khot et al., 2019), CommonsenseQA/CQA (Talmor et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), and Winogrande (Sakaguchi et al., 2020). Several of the MC datasets do not come with accompanying paragraphs (such as ARC, QASC, OBQA). For most of this the work, we keep the questions as is with no additional retrieval (unless otherwise mentioned). One other variability among these datasets is their number of candidate answers. While many datasets have four candidates (see Fig. 2), others have more. Later (in §6.2) we will see that our approach generalizes to datasets with different numbers of candidates, even if such questions have not been seen during training.

Yes/No QA (YN). The YN datasets we use are BoolQ (Clark et al., 2019a) and a

Dataset	Train set size	Eval. set size	Best published	95% CI (%)	Input length	Output length
SQuAD 1.1	87k	10k	95.6	0.4	136.2	3.0
SQuAD 2.0	130k	11k	91.2	0.5	139.9	2.6
NewsQA	76k	4k	66.8	1.4	606.6	4.0
Quoref	22k	2k	86.1	1.5	352.7	1.7
Quoref-CS	-	700	55.4	3.6	324.1	2.2
ROPES	10k	1.4k	61.1	2.5	169.1	1.4
ROPES-CS	-	974	32.5	3.0	182.7	1.3
<hr/>						
NarQA	65k	21k	58.9	0.7	563.6	6.2
NatQA	79k	3.6k	42.2	1.6	607.0	2.2
DROP	77k	9k	89.1	0.6	189.1	1.6
DROP-CS	-	947	54.2	3.2	206.0	2.1
<hr/>						
RACE	87k	4k	89.5	0.9	317.9	6.9
OBQA	4k	501	80.0	3.3	28.7	3.6
MCTest	1.4k	320	86.5	3.4	245.4	4.0
ARC (easy)	2k	2k	80.0	1.7	39.4	3.7
ARC (chal.)	1k	1k	67.8	2.9	47.4	5.0
CQA	9.7k	1.2k	79.1	2.2	26.8	1.5
WG	40.3k	1.7k	67.5	2.2	25.2	3.0
PIQA	16.1k	3k	79.4	1.4	49.6	20.2
SIQA	33.4k	2.2k	78.0	1.7	37.3	4.7
<hr/>						
BoolQ	9k	3k	91.0	1.0	105.1	1.0
BoolQ-CS	-	461	71.1	4.0	108.9	1.0
NP-BoolQ	10k	3k	78.4	1.4	106.2	1.0
MultiRC	-	312	91.7	2.6	293.3	1.0

Table 2: Dataset Statistics. CQA, OBQA, WG, and NarQA refer to CommonsenseQA, OpenBookQA, Winogrande, and NarrativeQA, respectively. The CI column shows the upper 95% confidence interval for the evaluation set as a percentage, based on the Wilcoxon test around the mean score listed as a percentage in the best known performance column. Input and output representation lengths are measured in the number of tokens and averaged across the dataset.

naturally-perturbed version of this dataset, BoolQ-NP (Khashabi et al., 2020), and the binary (yes/no) subset of MultiRC (Khashabi et al., 2018).

Contrast-sets. Additionally, we use *contrast-sets* (Gardner et al., 2020) for several of our datasets (denoted with “CS”): BoolQ-CS, ROPES-CS, Quoref-CS, DROP-CS. These evaluation sets are expert-generated perturbations that deviate from the patterns common in the original dataset.

4.2 Evaluation Metrics for Textual Output

We evaluate each dataset using the metric used most often for it in prior work. For the EX format, it’s the F1 score of the extracted span relative to the gold label. For the AB format, we use ROUGE-L metric (Lin et al., 2006; Min et al., 2019; Nishida et al., 2019). For NatQA we use the exact-match metric, following Min et al. (2020). For the MC format, we match the generated text with the closest answer candidate based token overlap and compute the accuracy. For the YN format, we follow Clark et al. (2019a) to measure if the generated output matches the correct ‘yes’ or ‘no’ label. In rare cases where the output is longer than one word (e.g., ‘yes it is’), we check if it contains the correct label but

not the incorrect one.⁴

5 Pilot Study: Can Out-of-Format Training Help?

We first answer the question: *Is the broad idea of benefiting from out-of-format training even viable?* For instance, is our intuition correct that an MC dataset can, in practice, benefit from training on an EX dataset? Before discussing our main experimental results, we briefly report on a pilot study that assesses the following basic question: Given a training set T_1^i (the *anchor* dataset) of QA format F_i , is there an out-of-format training set T_1^j of format F_j such that training jointly on $T_1^i \cup T_1^j$ improves performance relative to training only on T_1^i ? To this end, we evaluate both on the matching evaluation set E_1^i as well as on ‘unseen’ data E_2^i, E_3^i, \dots of the same format.

The results are summarized in Table 3. The two rows in each individual table correspond to training on T_1^i (the *anchor* dataset) and on $T_1^i \cup X$, where X is an out-of-format dataset corresponding to T_1^j above. The columns represent various evaluation sets of format F_i . For each column, ‘ $X = \dots$ ’ at the very bottom indicates the out-of-format dataset X that was the most helpful in improving performance on the evaluation set in that column.⁵

Consider the case of the anchor set T_1^i being BoolQ and the evaluation set being NP-BoolQ, both of format YN. Here, including out-of-format training data X =SQuAD2 boosts performance from 51% to as much as 59%. The gain may be less in other cases, but across all anchor and evaluation datasets, we generally observe that there is at least one out-of-format training set whose inclusion improves performance.

This pilot study thus provides a proof of concept that out-of-format training can indeed help a QA model in nearly every case. Of course, this study only shows the existence of such an out-of-format dataset, rather than provide a single unified model. Nevertheless, it helps identify *representative training sets* from each format that were most helpful. As alluded to earlier, we used this empirical data to guide which training sets to include when building UNIFIEDQA in Section 3.2.

The experimental results from this case study are summarized in the aggregated plot shown in

⁴The evaluation code is available at the URL in Footnote 1.

⁵Appendix A.5 reports extended results, including the performance with various choices of X .

Trained on ↓ - Evaluated on →	SQuAD11	SQuAD2	NewsQA	Quoref	Quoref-CS
SQuAD11	85.9	42.8	51.7	28.2	28.11
SQuAD11 + X	85.8	42.8	52.1	29.4	29.84
Best X	BoolQ	OBQA	OBQA	NarQA	OBQA

Trained on ↓ - Evaluated on →	RACE	OBQA	ARC-chal	MCTest
RACE	55.8	26.6	28.0	62.5
RACE + X	59.1	32.2	28.4	69.4
Best X	SQuAD11	NarQA	NewsQA	SQuAD11

Trained on ↓ - Evaluated on →	BoolQ	MultiRC	NP-BoolQ	BoolQ-CS
BoolQ	76.4	64.1	51.3	53.4
BoolQ + X	78.9	66.0	59.4	61.0
Best X	SQuAD2	OBQA	SQuAD2	NarQA

Trained on ↓ - Evaluated on →	NarQA	DROP	DROP-CS
NarQA	51.5	10.2	11.1
NarQA + X	53.0	14.4	14.6
Best X	SQuAD2	SQuAD2	SQuAD2

Table 3: Pilot study showing that out-of-format training can help improve performance. Each table compares training on just the anchor dataset (e.g., BoolQ in the top-left table) with training also on an out-of-format dataset denoted ‘X’. Evaluation is on the anchor dataset as well as unseen datasets of that format. The last row identifies the out-of-format dataset that helped most on each evaluation dataset. All results are based on the “small” size T5 model. Color denotes QA format (see Table 2).

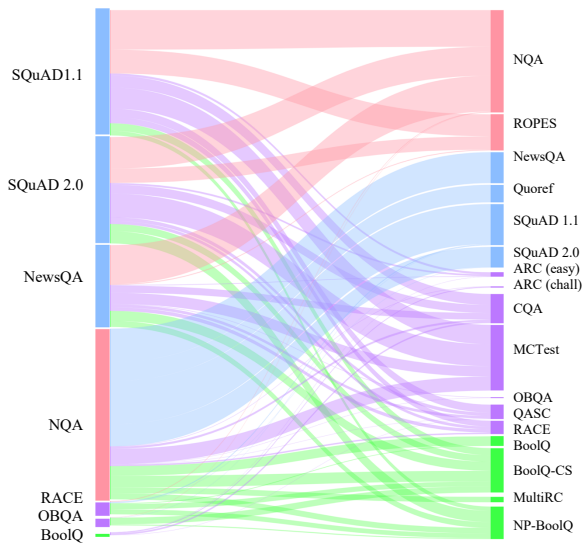


Figure 3: Bipartite graph showing the value of various datasets. The datasets on the left were used for training and on the right for evaluation. The wider the edge from a dataset ℓ (on the left) to a dataset r (on the right), the higher the contribution of adding the out-of-format dataset ℓ to the training set of questions in r 's format.

Fig. 3. In this bipartite graph, the datasets used for training are on the left hand side and the evaluation datasets are on the right hand side. The weight of each edge $w(\ell, r)$ indicates the contribution of a dataset ℓ when used for training jointly with an anchor dataset d , when evaluated on r (d and r have the same format.) Specifically,

$$w(\ell, r) = \text{avg}_d [S(\ell \cup d; r) - S(d; r)],$$

where $S(d, r)$ is the score achieved on r after training on d . Since we only focus on *gains* from out-of-format training, we drop the edges that are negative or between two datasets of the same format.

As expected, there are strong connections between the AB and EX datasets in Fig. 3 since their definitions are quite similar. Apart from the

edge weight, the overall width of a dataset ℓ on the left also depicts how much it contributes to out-of-format datasets. E.g., NQA (NarrativeQA) is the most helpful dataset and even helps multiple formats. Similarly our extractive datasets (SQuAD11.1, SQuAD 2, and NewsQA) are also relatively more helpful. While large datasets generally appear to help, RACE, another large-scale dataset, doesn't help that much. The least helpful dataset in the mix is BoolQ which focuses on yes/no questions.

In a similar vein, the wider the dataset on the right hand side, the more it can be benefit from out-of-format datasets. Among these beneficiary datasets, all four formats are equally represented.

6 Experimental Results

We now discuss our main experimental results, evaluating UNIFIEDQA on seed datasets (used for training the system) as well as unseen datasets.

6.1 UNIFIEDQA vs. 8 Dedicated Models

Is UNIFIEDQA, a single pre-trained multi-format QA system, as good as dedicated systems trained for individual datasets? We emphasize that the answer to this question is not as simple as it may seem, since earlier works have observed that a system addressing multiple tasks often *underperforms* a focused system (Raffel et al., 2020).

Fig. 4 summarizes the results of the relevant experiment. The gray bars belong to UNIFIEDQA (a single system for multiple datasets of different formats). The colored bars are different T5-based systems tailored to individual datasets (a different system for each dataset). The results show that UNIFIEDQA performs almost as good as individual T5 models targeted to each dataset. In some cases UNIFIEDQA performs even better than the

Seen dataset?	Model ↓ - Evaluated on →	NewsQA	Quoref	Quoref-CS	ROPES	ROPES-CS	DROP	DROP-CS	QASC	Common senseQA	NP-BoolQ	BoolQ-CS	MultiRC	Avg
No	UnifiedQA [EX]	58.7	64.7	53.3	43.4	29.4	24.6	24.2	55.3	62.8	20.6	12.8	7.2	38.1
	UnifiedQA [AB]	58.0	68.2	57.6	48.1	41.7	30.7	36.8	54.1	59.0	27.2	39.9	28.4	45.8
	UnifiedQA [MC]	48.5	67.9	58.0	61.0	44.4	28.9	37.2	67.9	75.9	2.6	5.7	9.7	42.3
	UnifiedQA [YN]	0.6	1.7	1.4	0.0	0.7	0.4	0.1	14.8	20.8	79.1	78.6	91.7	24.2
	UnifiedQA	58.9	63.5	55.3	67.0	45.5	32.5	40.1	68.5	76.2	81.3	80.4	59.9	60.7
Yes	Previous best	66.8	86.1	55.4	61.1	32.5	89.1	54.2	85.2	79.1	78.4	71.1	--	
		Retro Reader	TASE	XLNet	ROBERTa	RoBERTa	ALBERT	MTMSN	KF+SIR+2Step	reel.B-RoBERT	RoBERTa	RoBERTa	--	

Table 4: Generalization to unseen datasets: Multi-format training (UNIFIEDQA) often outperforms models trained the same way but solely on other in-format datasets (e.g., UNIFIEDQA [EX], which is trained on all extractive training sets of UNIFIEDQA). When averaged across all evaluation datasets (last column), UNIFIEDQA shows strong generalization performance across all formats. Notably, the “Previous best” models (last row) were trained on the target dataset’s training data, but are even then outperformed by UnifiedQA (which has never seen these datasets during training) on the YN tasks.

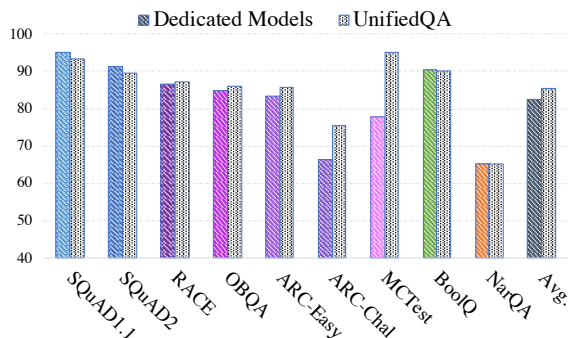


Figure 4: UNIFIEDQA is on-par with, and often outperforms, 9 different equally-sized T5-based systems tailored to individual datasets. The figure contains separate models for each of the two subsets of the ARC and Regents datasets.

single-dataset experts (e.g., on OBQA or NQA). On average (last column) UNIFIEDQA clearly outperforms the ensemble of dataset/format-specific systems. UNIFIEDQA thus offers flexibility across multiple QA formats while compromising almost nothing compared to dataset-specific experts.

6.2 Generalization to Unseen Datasets

We now explore whether UNIFIEDQA generalizes well to other, unseen datasets. Table 4 summarizes the results of experiments where we evaluate various models on datasets that are not used to train them. It compares UNIFIEDQA (training on multiple formats) with training on various datasets of a *single* format (e.g., UNIFIEDQA [EX], built by training the model on only extractive datasets).

The first few rows of the table show T5 models trained for individual formats, followed by UNIFIEDQA. For completeness, we include the highest previous scores for each dataset; one must be careful when reading these numbers as the best previous numbers follow the fully *supervised* protocol (for NewsQA (Zhang et al., 2020),

Quoref (Segal et al., 2019), DROP (Lan et al., 2019), ROPES (Lin et al., 2019), QASC (Khot et al., 2019), CommonsenseQA (Zhu et al., 2020) and x-CS datasets (Gardner et al., 2020).)

We make three key observations: (1) On average (last column), UNIFIEDQA shows much stronger generalization across a wide range of datasets. (2) on 9 (out of 12) datasets, UNIFIEDQA shows a better generalization than any single-format expert. For example, while the system is trained on multiple-choice questions with 4 candidate answers, it works quite well on datasets with more than 4 candidate answers (QASC and CommonsenseQA have 8 and 5 candidate answers per question, respectively). (3) Single-format experts are better at generalization only when the source and target datasets are very similar (for instance SQuAD and Quoref).

6.3 State-of-the-Art via Simple Fine-tuning

Fine-tuning of pre-trained language models has become the standard paradigm for building dataset-specific state-of-the-art systems (Devlin et al., 2019; Liu et al., 2019). The question we address here is: when it comes to QA, is there a value in using UNIFIEDQA as a starting point for fine-tuning, as opposed to a vanilla language model that has not seen other QA datasets before?

To address this question, we fine-tune each of UNIFIEDQA, T5, and BART on several datasets by selecting the best checkpoint on the dev set, and evaluating on the test set. Table 5 summarizes the results of the experiments. The table shows two variants: UNIFIEDQA_{T5} and UNIFIEDQA_{BART}. All results are based on the 11B version of T5.

The columns indicate the evaluation on the test set corresponding to the data that was used for training. For each dataset, the first line of the table

Model ↓ - Eval. →	OBQA *	OBQA (w/ IR)	ARC-easy *	ARC-easy (w/ IR)	ARC-chal *	ARC-chal (w/ IR)	QASC	QASC (w/ IR)
Previous best published	RoBERTa (Clark et al.,2019c)	KF+SIR (Mitra et al., 2020)	RoBERTa (Clark et al.,2019c)	FreeLB-RoBERTa (Zhu et al., 2020)	RoBERTa (Clark et al.,2019c)	FreeLB-RoBERTa (Zhu et al., 2020)	--	KF+SIR +2Step (Mitra et al., 2020)
BART _{large} - FT	67.8	66.2	64.1	79.6	36.6	40.4	50.0	75.3
UnifiedQA _{BART} - FT	63.8	70.0	68.0	82.7	52.1	55.0	53.2	78.2
T5 - FT	84.2	84.2	83.8	90.0	65.4	69.7	77.0	88.5
UnifiedQA - FT	86.0	87.2	86.4	92.0	75.0	78.5	78.5	89.6

Model ↓ - Eval. →	RACE *	ComQA	WG	PIQA	SIQA	ROPES	NatQ (w/ IR)
Previous best published	ALBERT (Lan et al.,2019)	FreeLB-RoBERTa (Zhu et al.,2020)	RoBERTa (Sakaguchi et al.,2019)	RoBERTa (Bisk et al., 2019)	RoBERTa (Mitra et al., 2020)	RoBERTa (Lin et al., 2019)	DPR+BART (Min et al.,2020)
BART _{large} - FT	78.8	62.5	62.4	77.4	74.0	60.5	42.1
UnifiedQA _{BART} - FT	79.4	64.0	63.6	77.9	73.2	60.0	44.5
T5 - FT	87.1	78.1	84.9	88.9	81.4	74	49.3
UnifiedQA - FT	89.4	79.1	85.7	89.5	81.4	75.2	49.3

Table 5: Fine-tuning UNIFIEDQA (last row) results in new state-of-the-art performance on 11 datasets. Further, it consistently improves upon fine-tuned T5 (2nd last row) by a margin ranging from 1% for CommonsenseQA (CQA) to as much as 13% for ARC-challenge. ‘(w/ IR)’ denotes relevant information is retrieved and appended as context sentences in the input encoding. Datasets marked with * are used in UNIFIEDQA’s original training.

Model ↓ - Evaluated on →	SQuAD11	SQuAD2	NarQA	RACE	OBQA	ARC-easy	ARC-hard	MCTest	BoolQ	Avg	Δ
UnifiedQA	93.4	89.6	65.2	87.3	86.0	85.7	75.6	95.0	90.2	85.4	
excluding BoolQ	93.1	90.1	65.0	87.7	85.0	86.1	75.2	94.7	8.3	77.0	-8.4
excluding SQuAD 2	95.3	47.3	65.4	87.7	84.8	85.9	75.5	95.3	90.5	81.3	-4.2
excluding OBQA	93.6	89.3	65.2	87.4	77.8	85.7	74.0	94.7	90.1	84.2	-1.3
excluding NarQA	93.6	89.8	52.5	87.7	85.6	86.3	75.9	95.6	89.9	84.2	-1.2
excluding RACE	93.9	89.0	65.0	78.5	85.2	85.6	74.7	95.9	90.1	84.3	-1.2
excluding ARC-easy	93.4	89.8	65.0	87.0	83.8	84.0	75.9	94.7	89.9	84.9	-0.6
excluding ARC-hard	93.6	90.1	64.9	87.3	85.2	85.1	73.8	95.6	90.5	85.1	-0.4
excluding MCTest	92.8	90.6	65.0	87.1	84.6	85.6	75.4	95.6	90.2	85.2	-0.2
excluding SQuAD 1.1	92.6	90.3	65.3	87.4	85.8	86.5	75.9	95.3	90.7	85.6	0.1

Table 6: The results of a leave-one-out ablation. The first row indicates the performance of UNIFIEDQA on each dataset it was trained on. The rest of the rows exclude one dataset at a time. The rows are sorted based on the last column: the dataset with biggest contribution appear first. The red highlights indicate the top 3 performance drops for each column.

reports the best previously published work. For several MC datasets that do not come with evidence paragraphs, we include two variants: one where we use them as-is and another that uses paragraphs fetched via an Information Retrieval (IR) system as additional evidence, indicated with “w/ IR” tags. We use the same IR sentences as used by the baselines: Aristo corpus for ARC and OBQA datasets (Clark et al., 2019c), and 2-step IR for QASC (Khot et al., 2019). For NatQA, following (Min et al., 2020), we use the DPR retrieval engine (Karpukhin et al., 2020) to augment each question with additional paragraphs.

We see that fine-tuning on UNIFIEDQA consistently dominates fine-tuning on T5 and BART, respectively. It also dominates the best previous scores on the datasets. Intuitively, since UNI-

FIEDQA has seen different formats, it should be positioned to achieve higher scores after a little fine-tuning, compared to fine-tuning on a vanilla T5 or BART model. This could be especially effective when a user has limited training data for a target QA task (also shown in Appendix A.6.) This also highlights that the effectiveness of cross-format training is not limited only to T5, but is rather a general trend for text-to-text architectures.

6.4 Ablation: Training Set Contributions

We now perform a leave-one-out experiment to better understand the contribution of each seed dataset to UNIFIEDQA. We take the system from §3.2 and assess how strong the model is when individual seed training datasets are dropped from the union. The result of this experiment is summarized

in Table 6. It compares the performance of full UNIFIEDQA (the first row) with ablated variants that exclude one seed dataset at a time. The rows are sorted based on the last column: datasets with higher contributions appear first.

Looking at first few rows of the table, BoolQ, SQuAD 2.0, OBQA, NarQA are the top four contributing datasets, each with a different format. SQuAD 1.1 has the least importance, presumably because it is mostly covered by SQuAD 2.0.

This study suggests that in order to build an effective unified QA system, it suffices to have a relatively small set of datasets as long as the set includes representatives from each format.

7 Discussion

The key motivation for this work is the observation that nearly all prior efforts on QA research were limited to the boundaries defined by narrow *formats*. A *format-specific* design would not generalize across QA datasets with slightly different definitions (e.g., a model built for SQuAD would not work for RACE). Additionally, such a design would prevent us from benefiting from the labeled data available in other formats. We challenge this view by advocating for approaches that combine seemingly different datasets. We believe that developing QA systems targeted to a specific format is a conceptual barrier for progress in the field.

Factors affecting generalization. Format is not the only factor affecting generalization across datasets. We additionally studied the value of other factors including *dataset size* and *domain* (vocabulary, topic, and style) in improving generalization. We observed that larger datasets often help with generalization, but not always (§5); e.g., RACE or OBQA show similar benefits (Fig. 3), even though RACE is much larger than OBQA. We observed a similar phenomenon with domain: similar domains help with transfer, but that is not always the case. For example, while BoolQ questions, similar to SQuAD, are accompanied with Wiki paragraphs, they barely benefit each other. Overall, the factors affecting generalization are not well-understood, leaving room for future investigations.

Unifying QA formats and text-to-text models. While UNIFIEDQA is built based using existing text-to-text models (Radford et al., 2019a; Raffel et al., 2020), we emphasize that the choice of tasks for multi-task learning plays a crucial role

in achieving successful results. Previous studies (Raffel et al., 2020) did *not* observe gains when mixing *tasks* that are very different. The key intuition is that a more coherent choice of *tasks* is more likely to succeed. Further, focusing on a coherent space of QA tasks/formats allows us to simplify the input by not requiring “prefixes” to explicitly define tasks/formats.

8 Conclusion

The question-answering community has fruitfully explored the design of strong models, but while staying within the boundaries of individual QA formats. We argued that such boundaries are artificial and can even limit the performance of systems, because the desired reasoning abilities being taught and probed are not tied to specific formats. Training data in one format should, in principle, help QA systems perform better even on questions in another format.

With this intuition in mind, we presented UNIFIEDQA, a single pre-trained QA system based on the text-to-text paradigm, seeking to bring unification across four common QA formats. We showed that even with its simple multi-format training methodology, UNIFIEDQA achieves performance on par with 8 dataset-specific expert models (§6.1), while also generalizing well to many unseen datasets of seen formats (§6.2). At the same time, we demonstrated that UNIFIEDQA is a strong starting point for building QA systems: it can achieve state-of-the-art performance by simply fine-tuning on target datasets (6.3).

We hope this effort will inspire a future line of work in the QA and NLP communities, moving towards more general and broader system designs. We leave extensions of UNIFIEDQA to other formats such as to direct-answer questions (Roberts et al., 2020) as a promising avenue for future work.

Acknowledgments

The authors would like to thank Collin Raffel, Adam Roberts, and Nicholas Lourie for their help with the T5 framework and for providing feedback on an earlier version of this work. The authors would like to acknowledge grants by ONR N00014-18-1-2826 and DARPA N66001-19-2-403, and gifts from the Sloan Foundation and the Allen Institute for AI. Moreover, the authors would like to thank members of the Allen Institute for AI, UW-NLP, and the H2Lab at the University of Wash-

ington for their valuable feedback and comments. TPU machines for conducting experiments were provided by Google.

References

- Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. In *NAACL*.
- Pratyay Banerjee and Chitta Baral. 2020. Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.
- Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT*.
- Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc Le. 2019b. BAM! Born-again multi-task networks for natural language understanding. In *ACL*, pages 5931–5937.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *ArXiv*, abs/1803.05457.
- Peter Clark, Oren Etzioni, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, et al. 2019c. From 'F' to 'A' on the NY Regents science exams: An overview of the Aristo project. *ArXiv*, abs/1909.01958.
- Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*.
- Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *EMNLP/IJCNLP*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- Dheeru Dua, Ananth Gottumukkala, Alon Talmor, Matt Gardner, and Sameer Singh. 2019a. Comprehensive multi-dataset evaluation of reading comprehension. In *2nd Workshop on Machine Reading for Question Answering*.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019b. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *2nd Workshop on Machine Reading for Question Answering, at EMNLP*.
- Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In *EMNLP - Findings*.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.
- Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Unifying question answering, text classification, and regression via span extraction. *arXiv preprint arXiv:1904.09286*.
- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL-HLT*.
- Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. More bang for your buck: Natural perturbation for robust question answering. In *EMNLP*.
- Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2019. QASC: A dataset for question answering via sentence composition. In *AAAI*.
- Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *TACL*, 6:317–328.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *TACL*, 7:453–466.

- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: Large-scale reading comprehension dataset from examinations. In *EMNLP*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2011. The winograd schema challenge. In *KR*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.
- Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An information-theoretic approach to automatic evaluation of summaries. In *NAACL*.
- Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *2nd Workshop on Machine Reading for Question Answering, at EMNLP*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.
- Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *EMNLP/IJCNLP*.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *EMNLP*.
- Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2020. How additional knowledge can improve natural language commonsense question answering. *arXiv: Computation and Language*.
- Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop qa with evidence extraction. In *ACL*.
- Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. In *NAACL*, pages 809–819.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019a. Language models are unsupervised multitask learners.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- 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. *JMLR*.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. In *ACL*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.
- Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *EMNLP*.
- Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *ACL*, pages 453–463.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WINOGRANDE: an adversarial winograd schema challenge at scale. In *AAAI*.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *EMNLP-IJCNLP*, pages 4453–4463.
- Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2019. A simple and effective model for answering multi-span questions. In *EMNLP*.
- Alon Talmor and Jonathan Berant. 2019. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. In *ACL*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*.
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Rep4NLP@ACL*.

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

Chen Zhu, Yu Cheng, Zhe Gan, Siqu Sun, Tom Goldstein, and Jingjing Liu. 2020. Freeb: Enhanced adversarial training for natural language understanding. In *ICLR*.