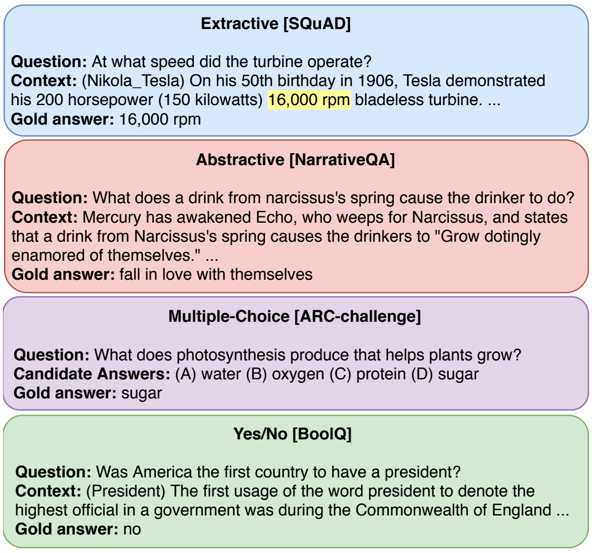
UnifiedQA: Crossing Format Boundaries With a Single QA System

Daniel Khashabi[1] **Tushar Khot**[1] **Ashish Sabharwal**[1]
Oyvind Tafjord[1] **Peter Clark**[1] **Hannaneh Hajishirzi**[1,2]

[1]Allen Institute for AI, Seattle, U.S.A.
[2]University 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 surprisingly well across 17 QA datasets spanning 4 diverse formats. UnifiedQA performs on par with 9 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, simply fine-tuning this pre-trained QA model into specialized models results in a new state of the art on 6 datasets, establishing UnifiedQA as a strong starting point for building QA systems.[1]



Extractive [SQuAD]

Question: At what speed did the turbine operate?
Context: (Nikola_Tesla) On his 50th birthday in 1906, Tesla demonstrated his 200 horsepower (150 kilowatts) 16,000 rpm bladeless turbine. ...
Gold answer: 16,000 rpm

Abstractive [NarrativeQA]

Question: What does a drink from narcissus's spring cause the drinker to do?
Context: 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." ...
Gold answer: fall in love with themselves

Multiple-Choice [ARC-challenge]

Question: What does photosynthesis produce that helps plants grow?
Candidate Answers: (A) water (B) oxygen (C) protein (D) sugar
Gold answer: sugar

Yes/No [BoolQ]

Question: Was America the first country to have a president?
Context: (President) The first usage of the word president to denote the highest official in a government was during the Commonwealth of England ...
Gold answer: no

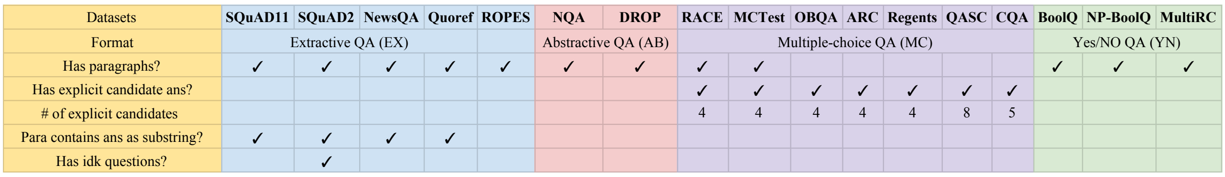
Figure 1: Four formats (color-coded throughout the paper) commonly used for posing questions and answering them: Extractive (EX), Abstractive (AB), Multiple-Choice (MC), and Yes/No (YN). Sample dataset names are shown in square brackets. We study generalization and transfer across these formats.

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 *QA formats*, with four popular formats illustrated in Figure 1. These formats differ not only in how the question is presented but also in some implicit assumptions. For instance, the assumption that the expected answer is either "yes" or "no", or that there is always a unique answer span in the associated paragraph (as opposed to multiple or no spans), etc. These differences have motivated their study in silos, often encoding QA format and assumptions 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 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*

[1] https://github.com/allenai/unifiedqa



Datasets	SQuAD11	SQuAD2	NewsQA	Quoref	ROPES	NQA	DROP	RACE	MCTest	OBQA	ARC	Regents	QASC	CQA	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	4	8	5			
Para contains ans as substring?	✓	✓	✓	✓													
Has idk questions?		✓															

Figure 2: Properties of various QA datasets included in this study: 4 extractive (EX), 3 abstractive (AB), 7 multiple-choice (MC), and 3 yes/no (YN). 'idk' denotes 'I don't know' or unanswerable questions. Regents denotes both 4th and 8th grade datasets. BoolQ represents both the original dataset and its *contrast-sets* extension BoolQ-CS; similarly for ROPES, Quoref, and DROP.

QA system, named UnifiedQA, that exploits information across 4 different QA formats to achieve surprisingly strong performance across 17 different datasets listed in Figure 2.

Our work is enabled by recent progress in text-to-text neural architectures (Raffel et al., 2019; Lewis et al., 2019; Radford et al., 2019). This paradigm conceptually unifies many NLP models that formerly had task-specific designs. While there have been hopes of—and a few attempts at—using this paradigm to achieve strong generalization and transfer across tasks, success so far has been limited. Most approaches fine-tuned a different set of parameters for each end task (Raffel et al., 2019; Radford et al., 2019), and when they have attempted to make a single model for different NLP tasks (Raffel et al., 2019), they have underperformed compared to the standard pre-training plus fine-tuning setup, with a need for explicit task-specific prefixes.

In contrast, by narrowing the scope to tasks that stay within the boundaries of QA, we are able to demonstrate that the text-to-text paradigm can, in fact, be quite powerful for multi-task learning across QA formats. We find that out-of-format training can lead to a single pre-trained QA model that can be applied as-is to different QA tasks, takes in natural text inputs without explicitly specifying a task-specific prefix, generalizes better to other unseen datasets, and with further fine-tuning can achieve state-of-the-art results on many QA tasks.

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 6 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 desirable 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 generalizable models. Similarly, 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. (2019) 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 question-answering formats $F_1, F_2, \ldots, F_k$. For each format F_i, suppose we have k_i datasets sets $D^i_1, D^i_2, \ldots, D^i_{k_i}$ where $D^i_j = (T^i_j, E^i_j)$ includes both training and evaluation examples. In some cases, the training set T^i_j may be empty or we may want to ignore it in order to treat D^i_j 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 possible approach of creating a mixed training pool consisting of all available training instances:

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

Training batches are drawn from this pooled data, $\tilde{T}$, by including each $q \in T^i_j$ with a probability proportional $1/|T^i_j|$. Each batch thus, on average, contains the same number of instances from each training set, regardless of its size. As we will see in the experiments section, our multi-format mixing approach works surprisingly 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 T5 framework (Raffel et al., 2019), a recent text-to-text transformer model. For all experiments, we use token-limits of size 512 and 100 for inputs and outputs sequences, respectively. All models are trained for $100k$ steps, on top of the $1M$ steps pretraining of the T5 model.

We first define a unifying encoding of the instances across various formats (in §3.1). We then introduce UnifiedQA (in §3.2) that is a QA system trained on datasets in multiple formats, indicating new state-of-the-art results on 7 datasets and generalization to unseen datasets.

3.1 Text-to-Text Encoding

We convert each of our target datasets into a text-in/text-out format (Raffel et al., 2019; Lewis et al., 2019; Radford et al., 2019). 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.

The unified model explored in this work incorporates the following four common question-answering formats:

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.

Further details of these formats and specific datasets within them are deferred to Section 4.1.

Table 1 provides examples of the natural input and output encoding for each of these formats. Importantly, 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. The expectation is that a model will learn to *infer* such notions from the raw text training data, just like humans are able to do.

[2] A 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.

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.

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 in Table 1). 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 in Table 1. 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., 2019), 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 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 9 datasets for training UnifiedQA, 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, Regents, ARC, OBQA, MCTest
- YN: BoolQ

One can obviously use other combinations of formats and datsets 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 UnifiedQA. 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 selected 17 existing datasets that target various formats, as well as various complex linguistic phenomena. Table 2 shows different properties for our datasets (whether it comes with a paragraph, whether the paragraph explicitly contains the answer, whether there are candidate-answers as part of the input, etc.) Most importantly, they are grouped into several formats/categories described below. Table 2 gives summary statistics of these datasets.

Extractive QA (EX). All the datasets in this format require models to extract the answer to a given question as a substring from a context paragraph. SQuAD 1.1 (Rajpurkar et al., 2016) contains questions about Wikipedia paragraphs. A later version of this dataset, SQuAD 2 (Rajpurkar et al., 2018), includes unanswerable questions which empirically makes the task much harder. For our evaluation, we use the development sets of SQuAD 1.1 and SQuAD 2. NewsQA (Trischler et al., 2017) dataset focuses on paraphrased questions with predicate-argument structure understanding collected from news articles from CNN/DailyMail articles. Quoref (Dasigi et al., 2019) contains questions that require coreference resolution in Wikipedia articles and can even have disjoint spans as answers. ROPES (Lin et al., 2019) centers around situation understanding, where the model must under the causes and effects implicit in the given situation.

Abstractive QA (AB). All the datasets in this format require models to produce answers that are often not mere substrings of the given context paragraph. NarrativeQA (Kociský et al., 2018)

Dataset	Train set size	Eval. set size	95% CI (as %)	Input length	Output length
SQuAD 1.1	87k	10k	1.0	136.2	3.0
SQuAD 2.0	130k	11k	0.9	139.9	2.6
NewsQA	76k	4k	1.6	606.6	4.0
Quoref	22k	2k	2.2	352.7	1.7
Quoref-CS	-	700	3.7	324.1	2.2
ROPES	10k	1.4k	2.6	169.1	1.4
ROPES-CS	-	974	3.1	182.7	1.3
NQA	65k	21k	0.7	563.6	6.2
DROP	77k	9k	1.0	189.1	1.6
DROP-CS	-	947	3.2	206.0	2.1
RACE	87k	4k	1.6	317.9	6.9
OBQA	4k	0.5k	4.4	28.7	3.6
MCTest	1.4k	0.3k	5.8	245.4	4.0
ARC (easy)	2k	2k	2.2	39.4	3.7
ARC (chal.)	1k	1k	3.1	47.4	5.0
Regents	1k	1k	3.1	51.0	4.9
CQA	9.7k	1.2k	2.8	26.8	1.5
BoolQ	9k	3k	1.8	105.1	1.0
BoolQ-CS	-	461	4.6	108.9	1.0
NP-BoolQ	10k	3k	1.8	106.2	1.0
MultiRC	-	312	5.7	293.3	1.0

Table 2: Dataset Statistics. CQA, OBQA, and NQA refer to CommonsenseQA, OpenBookQA, and NarrativeQA, respectively. The CI column shows the 95% confidence interval for the evaluation set as a percentage, around a mean score of 50%. Input and output representation lengths are measured in the number of tokens and averaged across the dataset.

focuses on understanding various events that happen in a given movie plot, based on summaries of their movie adaptations from various web resources. Many of the answers do not have high overlap with the context. DROP (Dua et al., 2019b) contains questions that involve rudimentary mathematical skills (such as counting, addition, subtraction, maximum, minimum, etc.) and questions query multiple parts of the paragraph. The answer can be either a number or a date that can be inferred from the paragraph, or several spans from the context paragraph.

Multiple-choice QA (MC). All the datasets in this format contain questions that come with candidate answers. MCTest (Richardson et al., 2013) contains questions about simple, fictional stories. RACE (Lai et al., 2017) is a challenging set of English comprehension multiple choice exams given in Chinese middle and high schools. OpenBookQA (Mihaylov et al., 2018), ARC (Clark et al., 2018), Regents Science Exams (Clark et al., 2016), QASC (Khot et al., 2019) are different MC tests focusing on elementary/high school-style science exams. A slightly different dataset within this format is CommonsenseQA (Talmor et al., 2019) which is geared towards activity/concept commonsense questions. Other than MCTest and RACE, the rest of the datasets do not come with accompanying paragraphs. On such datasets, occasionally a retrieval system is used to supplement each question with a relevant retrieved context paragraph. For most of this the work, we keep the questions as is with no additional retrieval (unless otherwise mentioned), except in §6.3 where we use IR to get numbers comparable to earlier work. One other variability among these datasets is their number of candidate answers. While many datasets have four candidates (see Figure 2), others have more. Later, in §6.2 we will see that our approach generalizes to datasets with different number of candidates, even if it's not seen during training.

Yes/No QA (YN). All the datasets in this format contain questions that could be responded with yes/no answers. One can think of these as multiple-choice questions with 2 candidates; however, they're usually treated differently. Several examples we use are BoolQ (Clark et al., 2019a) and a version of this dataset with natural-perturbations, BoolQ-NP (Khashabi et al., 2020), the subset of MultiRC (Khashabi et al., 2018) that have binary(yes/no) answers.

Contrast-sets. Additionally, we use *contrast-sets* (Gardner et al., 2020) corresponding to several of our datasets (indicated by "CS" tag): 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 most often used for it in prior work. Specifically, for the EX format, we use the F1 score for 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 the MC format, we match the generated the text with the closest answer candidate (by token overlap) and measure how often this is the correct answer. 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.

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	NQA	OBQA

Trained on ↓ - Evaluated on →	RACE	OBQA	ARC-easy	ARC-chal	Regents-4th	Reg-8th	MCTest	QASC	CQA
RACE	55.8	26.6	31.8	28.0	32.3	32.7	62.5	17.9	28.3
RACE + X	**59.1**	**32.2**	**32.4**	**28.4**	**32.7**	**34.2**	**69.4**	**23.5**	**36.1**
Best X	SQuAD11	NQA	SQuAD11	NewsQA	NQA	NewsQA	SQuAD11	SQuAD11	SQuAD11

Trained on ↓ - Evaluated on →	BoolQ	MultiRC	NP-BoolQ	BoolQ-CS
BoolQ	76.36	64.10	51.33	53.37
BoolQ + X	**78.93**	**66.03**	**59.38**	**61.00**
Best X	SQuAD2	OBQA	SQuAD2	NQA

Trained on ↓ - Evaluated on →	NQA	DROP	DROP-CS
NQA	51.5	10.2	11.1
NQA + 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).

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^i_1 (the *anchor* dataset) of QA format F_i, is there an out-of-format training set T^j_1 of format F_j such that training jointly on $T^i_1 \cup T^j_1$ improves performance relative to training only on T^i_1? To this end, we evaluate both on the matching evaluation set E^i_1 as well as on ‘unseen’ data $E^i_2, E^i_3, \ldots$ of the same format.

The results are summarized in Table 3. The two rows in each individual table correspond to training on T^i_1 (the *anchor* dataset) and on $T^i_1 \cup X$, where X is an out-of-format dataset corresponding to T^j_1 above. The columns represent various evaluation sets of format F_i. For each column, ‘$X = \ldots$’ 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.[3]

Consider, for example, the case of the anchor set T^i_1 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 in other cases is often not this extreme. Nevertheless, 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 hinted to earlier, we used this empirical data to guide which training sets to include when building UnifiedQA in Section 3.2.

6 Experimental Results

We now discuss our main experimental results, evaluating our proposed UnifiedQA system on seen (used for training the system) and unseen datasets.

6.1 UnifiedQA vs. 9 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., 2019).

Figure 3 summarizes the results of the relevant experiment. As it can be observed from the figure, UnifiedQA performs almost as good as the best single dataset experts. In some cases UnifiedQA performs even better than than the single-dataset experts (e.g., on OBQA or NQA.) On av-

[3] Appendix A.3 reports extended results, including the performance with various choices of X.

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	reeLB-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.

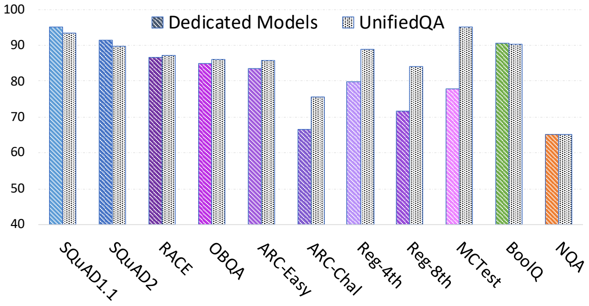


Figure 3: UnifiedQA is on-par with, and often outperforms, 11 different equally-sized T5-based systems tailored to individual datasets. The figure contains separate models for each of the two subsets of the ARC and Regents datasets.

erage (last column) UnifiedQA is doing much better dataset/format-specific systems. In conclusion, UnifiedQA offers flexibility across multiple QA formats while compromising almost nothing compared to dataset-specific experts.

6.2 Generalization to Unseen Datasets

The question we want to explore here is whether UnifiedQA generalizes well to other unseen datasets. Table 4 summarizes the results of experiments during which we evaluate various models on datasets that are not used to train them.

The first few rows of the table shows T5 models trained for individual datasets, 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 (Dasigi et al., 2019), DROP (Dua et al., 2019b), ROPES (Lin et al., 2019), QASC (Khot et al., 2019), CommonsenseQA (Zhu et al., 2020) and x-CS datasets (Gardner et al., 2020).)

The key observations are: (1) On average (last column), UnifiedQA shows a much stronger generalization across a wide range of datasets. (2) on 5 (out of 12) datasets UnifiedQA shows a better generalization than any single-dataset experts. For example, while the system is trained on multiple-choice questions with 4 candidate answers, it does work pretty well on datasets with more than 4 candidate answers (QASC and CommonsenseQA have has 8 and 5 candidate ansers per question, respectively). (3) single-dataset experts are better at generalization only when the source and target datasets are pretty 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 stat-of-art systems (Devlin et al., 2019; Liu et al., 2019). The question we address here is whether 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 question, we fine-tune both UnifiedQA and T5 on several datasets. Table 5 summarizes the results of the experiments. The two last rows of the table show the performance UnifiedQA and T5, both fine-tuned for the target task. The fine-tuning process involves selection of the best checkpoint on the dev set and evaluation on the test set.

The columns indicate the evaluation on the test set corresponding to the data that was used for training. For several multiple-choice datasets that do not come with evidence with paragraphs, we include two variants: use them as is and another variant which uses paragraphs fetched via an Information

Model ↓ - Eval. →	RACE*	OBQA*	OBQA (w/ IR)	ARC-easy*	ARC-easy (w/ IR)	ARC-chal*	ARC-chal (w/ IR)	QASC	QASC (w/ IR)	Common senseQA
Previous best	ALBERT	RoBERTa	KF+SIR	RoBERTa	FreeLB-RoBERTa	RoBERTa	FreeLB-RoBERTa	--	KF+SIR +2Step	FreeLB-RoBERTa
	89.5	75.7	80.0	69.9	80.0	55.9	67.8	--	85.2	72.2
T5 (fine-tuned)	87.1	84.2	84.2	83.8	90.0	65.4	69.7	77.0	88.5	78.1
UnifiedQA	87.3	**86.0**	71.2	85.7	89.2	**75.6**	74.7	68.5	80.1	76.2
UnifiedQA (fine-tuned)	89.4	**86.0**	**87.2**	**86.4**	**92.0**	75.0	**78.5**	**78.5**	**89.6**	**79.1**

Table 5: Simply fine-tuning UnifiedQA (last row) results in new state-of-the-art performance on 6 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	NQA	RACE	OBQA	ARC-easy	ARC-hard	Regents	MCTest	BoolQ	Avg	Δ
UnifiedQA	93.4	89.6	65.2	87.3	86.0	85.7	75.6	86.3	95.0	90.2	85.4	
excluding BoolQ	93.1	90.1	65.0	87.7	85.0	86.1	75.2	85.0	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	84.8	95.3	90.5	81.3	-4.2
excluding OBQA	93.6	89.3	65.2	87.4	77.8	85.7	74.0	83.8	94.7	90.1	84.2	-1.3
excluding NQA	93.6	89.8	52.5	87.7	85.6	86.3	75.9	85.2	95.6	89.9	84.2	-1.2
excluding RACE	93.9	89.0	65.0	78.5	85.2	85.6	74.7	84.6	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	85.0	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	84.4	95.6	90.5	85.1	-0.4
excluding Regents	93.4	89.9	64.8	87.1	84.0	86.2	74.5	86.0	95.3	90.6	85.2	-0.3
excluding MCTest	92.8	90.6	65.0	87.1	84.6	85.6	75.4	85.0	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	85.8	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 the last column: the dataset with biggest contribution appear first. The red highlights indicate the top 3 performance drops at each column.

Retrieval (IR) system as additional evidence, indicated with "w/ IR" tags. We use the same IR sentences as used by the baselines on these datasets: Aristo corpus for ARC and OBQA datasets (Clark et al., 2019c), and 2-step IR for QASC (Khot et al., 2019).

Additionally, we show the best published scores on each dataset: ALBERT (Lan et al., 2019) (on RACE), RoBERTa (Clark et al., 2019c) (on OBQA and ARC), KF+SIR (Banerjee and Baral, 2020) (on OBQA and QASC), FreeLB+RoBERTa (Zhu et al., 2020) (on ARC-easy and CommonsenseQA).

As it can be seen, fine-tuning on UnifiedQA consistently dominates fine-tuning on T5, as well as the best previous scores on each dataset. Intuitively, since UnifiedQA has seen different formats should be positioned to achieve higher scores after a bit of fine-tuning, comparing to fine-tuning on a vanilla T5. This could be especially effective when a user has limited training data for a target QA task.

6.4 Ablation: Training Set Contributions

In this experiment we would like to better understand the contribution of each dataset to UnifiedQA through a leave-one-out experiment.

We take the system from §3.2 and evaluate the model when individual models are dropped from the union. The result of this experiment is summarized in Table 6 compares the performance of UnifiedQA all the default datasets (the first row) followed with ablated versions which exclude one dataset at a time. The rows are sorted based the last column: the dataset with biggest contribution appear first.

The top-few datasets have the highest contributions: BoolQ, SQuAD 2.0, OBQA, NQA are the top four contributing datasets, each with different format. SQuAD 1.1 has the least importance in the union, since its mostly covered by SQuAD 2.0.

The conclusion here is that to build an effective variant of UnifiedQA, one can just use a relatively small number of datasets, so long as that it is trained on representative members of each format.

7 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 T5 text-to-text language model, 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 nearly a dozen 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 (Kwiatkowski et al., 2019) 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 providing suggestions on an earlier version of this work. TPU machines for conducting experiments were provided by Google.

References

Pratyay Banerjee and Chitta Baral. 2020. Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*.

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 NLP models via contrast sets. *ArXiv*, abs/2004.02709.

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. Natural perturbation for robust question answering. *ArXiv*, abs/2004.04849.

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*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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*.

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*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. 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. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

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*.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *ACL*, pages 453–463.

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 Sordoni, 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, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *ICLR*.

A Appendices

A.1 UnifiedQA: Different Sizes

For completeness we're also showing the scores of UnifiedQA of different sizes on each dataset. For these systems each row is a single system.

Trained on ↓ - Evaluated on →	SQuAD11	SQuAD2	NewsQA	Quoref	Quoref-CS	ROPES	ROPES-CS	NQA	DROP	DROP-CS	BoolQ	MultiRC	NP-BoolQ	BoolQ-CS
Small	79.4	67.6	51.1	25.6	27.6	31.0	32.9	53.7	14.6	17.2	77.1	46.9	59.4	58.1
Base	88.2	78.1	54.2	40.0	38.5	33.9	28.4	58.7	19.7	23.7	82.5	64.8	66.3	61.9
Large	91.1	85.9	48.5	45.5	42.1	47.7	37.9	60.8	24.6	30.7	86.1	54.2	72.6	73.0
3B	93.2	87.4	59.6	60.4	54.7	48.7	43.1	63.3	28.5	33.9	89.3	62.6	78.4	77.0
11B	93.4	89.6	58.9	63.5	55.3	67.0	45.6	65.2	32.5	40.9	90.2	59.9	81.3	80.4

Trained on ↓ - Evaluated on →	RACE	OBQA	OBQA	ARC-easy	ARC-easy (w/ IR)	ARC-chal	ARC-hard (w/ IR)	Reg-4th	Reg-8th	MCTest	QASC	QASC (w/ IR)	CQA
Small	56.0	50.4	35.4	42.9	59.5	35.9	35.8	42.1	43.4	80.0	19.1	37.9	32.8
Base	70.3	59.0	38.4	53.0	69.4	42.4	44.2	55.4	48.9	86.9	25.8	50.8	45.0
Large	78.1	68.4	54.6	65.9	77.4	54.4	54.8	67.5	61.7	90.0	43.3	62.6	60.9
3B	83.2	80.8	63.2	78.7	86.2	66.7	64.8	83.8	74.1	95.0	62.2	76.6	71.3
11B	87.3	86.0	71.2	85.7	89.2	75.6	74.7	88.9	84.2	95.0	68.5	80.1	76.2

Table 7: UnifiedQA of different sizes on our datasets.

A.2 Comparison with the Dedicated Models: extended results

Here we summarize an extension of the results in §6.1. Table 8 summarizes the results of the relevant experiment. In the top portion of the table we have evaluations of T5 model fine-tuned for individual datasets, followed by UnifiedQA. As it can be observed from the table, UnifiedQA performs almost as good as the best single dataset experts. In some cases UnifiedQA performs even better than than the single-dataset experts (e.g., on OBQA or NQA.) On average (last column) UnifiedQA is doing much better dataset/format-specific systems. In conclusion, UnifiedQA offers flexibility across multiple QA formats while compromising almost nothing compared to dataset-specific experts.

Trained on ↓ - Evaluated on →	SQuAD11	SQuAD2	RACE	OBQA	ARC-easy	ARC-chal	Reg-4th	Reg-8th	MCTest	BoolQ	NQA	Avg
T5 (SQuAD 11)	**95.2**	47.4	44.1	47.4	60.4	44.1	63.5	55.8	77.8	1.2	57.8	54.1
T5 (SQuAD 2)	91.2	**91.3**	32.6	31.4	50.3	36.0	55.4	48.7	62.5	11.5	50.7	51.0
T5 (RACE)	87.4	42.7	86.6	59.4	77.1	63.1	79.0	71.9	**95.6**	6.7	53.9	65.8
T5 (OBQA)	44.5	22.0	64.9	85.0	75.3	65.7	80.4	72.3	89.7	0.2	18.3	56.2
T5 (ARC-easy)	49.0	24.3	67.5	70.0	83.5	65.0	83.0	75.7	90.6	0.2	27.9	57.9
T5 (ARC-chal)	53.7	26.4	64.6	67.2	76.4	66.5	79.0	71.7	92.8	0.2	28.6	57.0
T5 (Reg-4th)	57.0	28.1	64.5	60.8	76.9	59.7	79.9	69.5	91.3	0.2	31.4	56.3
T5 (Reg-8th)	53.3	26.2	64.9	61.2	78.8	58.5	79.2	71.6	90.6	0.2	29.4	55.8
T5 (MCTest)	78.8	38.6	69.3	42.4	66.5	49.2	66.6	59.6	**95.3**	0.5	39.0	55.1
T5 (BoolQ)	6.3	3.6	21.6	28.0	25.4	22.5	27.1	25.8	23.1	**90.5**	0.3	24.9
T5 (NQA)	90.5	44.8	47.8	38.0	58.2	43.8	58.5	54.5	80.3	46.8	**65.2**	57.1
UnifiedQA	93.4	89.6	**87.3**	**86.0**	**85.7**	**75.6**	**88.9**	**84.2**	95.0	90.2	**65.2**	85.6

Table 8: UnifiedQA is on-par with systems tailored to individual datasets (the diagonal cells vs the last row) while functioning across a wide range of datasets (the last column).

A.3 Pairwise Mixing: extended results

Here we summarize an extension of the results in §5. The question addressed here is whether there is value in mixing datasets with different formats. We evaluated this by adding one dataset of a different format to four different datasets (one for each format). The results are summarized in Table 9. The goal of each sub-table is to measure the *within-format* generalization one can gain via *out-of-format* training. Each sub-table has an *anchor* dataset, indicated in the first column. For example in the first table the anchor dataset is SQuAD. Rows of the table: Each table combines datasets of other formats with the anchor dataset (e.g., SQuAD + RACE, etc). The columns of the sub-tables contain evaluations on the dataset with the same format as the anchor dataset. For example, on the first table, the evaluation is done on SQuAD 1.1/2.0, NewsQA, Quoref which have the same format as SQuaD 1.1, the anchor dataset.

The results show that one can achieve gains for question-answering in a certain format by incorporating resources in other formats. In the first two sub-tables, we see that NQA (AB) and OBQA (MC) help a SQuAD models generalize better to other EX datasets. In the third table where the anchor dataset is NQA (AB), EX datasets help a NQA model generalize better to other AB datasets. In the 4th/5th subtable, EX and AB datasets help a RACE/OBQA (MC) models generalize better to other MC datasets. Similarly, in the final sub-table, MC dataset helps improve the scores on a YN datasets.

Anchor Dataset / Format	Trained on ↓ - Evaluated on →	SQuAD11	SQuAD2	NewsQA	Quoref	Quoref-CS	Avg
SQuAD11	**SQuAD11**	**85.9**	42.8	51.7	28.2	28.11	47.4
	SQuAD11 + RACE	85.6	42.6	51.7	26.6	27.43	46.8
	SQuAD11 + OBQA	85.7	42.8	**52.1**	27.7	**29.84**	**47.6**
	SQuAD11 + BoolQ	85.8	42.7	52.1	27.7	29.42	47.5
	SQuAD11 + NQA	85.6	42.7	51.3	**29.4**	26.56	47.1
SQuAD2	**SQuAD2**	76.5	70.7	46.0	17.7	22.04	46.6
	SQuAD2 + RACE	76.5	70.6	47.9	18.6	20.40	**46.8**
	SQuAD2 + OBQA	76.7	70.8	**48.4**	16.9	19.80	46.5
	SQuAD2 + BoolQ	75.9	**72.0**	45.4	16.3	20.35	46.0
	SQuAD2 + NQA	72.5	70.9	47.3	**20.0**	**23.39**	**46.8**

Anchor Dataset / Format	Trained on ↓ - Evaluated on →	NQA	DROP	DROP-CS	ROPES	ROPES-CS	Avg
NQA	**NQA**	51.5	10.2	11.1	22.8	15.3	22.2
	NQA + SQuAD11	52.7	14.1	14.6	30.5	33.2	29.0
	NQA + SQuAD2	**53.0**	**14.4**	14.6	**31.3**	**33.2**	**29.3**
	NQA + NewsQA	52.5	10.4	12.3	16.6	15.6	21.5
	NQA + RACE	52.0	10.7	13.5	20.0	17.9	22.8
	NQA + OBQA	51.8	10.1	11.3	15.4	17.0	21.1
	NQA + BoolQ	51.8	10.2	10.9	20.7	10.9	20.9

Anchor Dataset / Format	Trained on ↓ - Evaluated on →	RACE	OBQA	ARC-easy	ARC-hard	Regents-4th	Reg-8th	MCTest	QASC	CommonsenseQA	Avg
RACE	**RACE**	55.8	26.6	31.8	28.0	32.3	32.7	62.5	17.9	28.3	35.1
	RACE + SQuAD11	**59.1**	28.0	32.4	28.1	30.8	33.9	**69.4**	**23.5**	**36.1**	**37.9**
	RACE + NewsQA	57.5	28.0	31.6	28.4	30.3	34.2	65.0	19.9	32.1	36.3
	RACE + BoolQ	57.4	26.8	31.8	27.9	31.9	33.7	63.1	18.0	29.6	35.6
	RACE + NQ	55.7	32.2	30.6	28.4	**32.7**	31.5	60.9	17.9	28.1	35.3
OBQA	**OBQA**	28.8	51.8	26.1	**34.8**	29.2	failed	33.1	6.9	17.3	28.5
	OBQA + SQuAD11	29.6	51.6	**27.2**	33.3	30.8	26.2	46.3	**9.5**	**23.3**	30.9
	OBQA + SQuAD2	29.5	**53.2**	27.2	33.5	**31.7**	26.1	46.6	9.3	23.1	**31.1**
	OBQA + NewsQA	30.7	49.4	26.1	32.3	30.4	26.8	37.8	8.9	22.9	29.5
	OBQA + BoolQ	25.0	50.4	26.0	34.3	28.0	25.2	27.2	7.1	18.3	26.8
	OBQA + NQA	29.7	52.8	25.6	33.0	28.6	23.4	**49.1**	8.9	19.1	30.0

Anchor Dataset / Format	Trained on ↓ - Evaluated on →	BoolQ	MultiRC	NP-BoolQ	BoolQ-CS	Avg
BoolQ	**BoolQ**	76.36	64.10	51.33	53.37	61.3
	BoolQ + SQuAD11	78.41	51.28	54.33	58.36	60.6
	BoolQ + SQuAD2	**78.93**	56.89	**59.38**	58.06	63.3
	BoolQ + NewsQA	77.61	54.17	55.46	59.82	61.8
	BoolQ + RACE	75.69	61.22	54.59	56.89	62.1
	BoolQ + OBQA	76.42	**66.03**	52.03	57.77	63.1
	BoolQ + NQA	78.90	59.02	55.33	**61.00**	**63.6**

Table 9: Pairwise mixing of formats: mixing with QA of datasets with different formats helps.