MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension

Adam Fisch[♦] **Alon Talmor**[♠◇] **Robin Jia**[♣] **Minjoon Seo**[♥△] **Eunsol Choi**[♥□] **Danqi Chen**[♡]
[♦] Massachusetts Institute of Technology [♠] Tel Aviv University [♣] Stanford University
[♥] University of Washington [△] NAVER [♡] Princeton University
[□] Google AI [◇] Allen Institute for Artificial Intelligence

Abstract

We present the results of the Machine Reading for Question Answering (MRQA) 2019 shared task on evaluating the generalization capabilities of reading comprehension systems.[1] In this task, we adapted and unified 18 distinct question answering datasets into the same format. Among them, six datasets were made available for training, six datasets were made available for development, and the final six were hidden for final evaluation. Ten teams submitted systems, which explored various ideas including data sampling, multi-task learning, adversarial training, and ensembling. The best system achieved an average F1 score of 72.5 on the 12 held-out datasets, 10.7 absolute points higher than our initial baseline based on BERT.

1 Introduction

Machine Reading for Question Answering (MRQA) has become an important testbed for evaluating how well computer systems understand human language. Interest in MRQA settings—in which a system must answer a question by reading one or more context documents—has grown rapidly in recent years, fueled especially by the creation of many large-scale datasets (Rajpurkar et al., 2016; Joshi et al., 2017; Kwiatkowski et al., 2019). MRQA datasets have been used to benchmark progress in general-purpose language understanding (Devlin et al., 2018; Yang et al., 2019). Interest in MRQA also stems from their use in industry applications, such as search engines (Kwiatkowski et al., 2019) and dialogue systems (Reddy et al., 2019; Choi et al., 2018).

While recent progress on benchmark datasets has been impressive, MRQA systems are still primarily evaluated on in-domain accuracy. It remains challenging to build MRQA systems that generalize to new test distributions (Chen et al., 2017; Levy et al., 2017; Yogatama et al., 2019) and are robust to test-time perturbations (Jia and Liang, 2017; Ribeiro et al., 2018). A truly effective question answering system should do more than merely interpolate from the training set to answer test examples drawn from the same distribution: it should also be able to extrapolate to test examples drawn from different distributions.

In this work we introduce the MRQA 2019 Shared Task on Generalization, which tests extractive question answering models on their ability to generalize to data distributions different from the distribution on which they were trained. Ten teams submitted systems, many of which improved over our provided baseline systems. The top system, which took advantage of newer pre-trained language models (Yang et al., 2019; Zhang et al., 2019), achieved an average F1 score of 72.5 on our hidden test data, an improvement of 10.7 absolute points over our best baseline. Other submissions explored using adversarial training, multi-task learning, and better sampling methods to improve performance. In the following sections, we present our generalization-focused, extractive question-answering dataset, a review of the official baseline and participating shared task submissions, and a meta-analysis of system trends, successes, and failures.

2 Task Description

The MRQA 2019 Shared Task focuses on generalization to *out-of-domain* data. Participants trained models on a fixed training dataset containing examples from six QA datasets. We then evaluated their systems on examples from 12 held-out test datasets. For six of the test datasets, we provided participants with some development data; the other six datasets were entirely hidden—

[1] https://github.com/mrqa/MRQA-Shared-Task-2019.

Proceedings of the Second Workshop on Machine Reading for Question Answering, pages 1–13
Hong Kong, China, November 4, 2019.

participants did not know the identity of these datasets.

We restricted the shared task to English-language extractive question answering: systems were given a question and context passage, and were asked to find a segment of text in the context that answers the question. This format is used by several commonly-used reading comprehension datasets, including SQuAD (Rajpurkar et al., 2016) and TriviaQA (Joshi et al., 2017). We found that the extractive format is general enough that we could convert many other existing datasets into this format. The simplicity of this format allowed us to focus on out-of-domain generalization, instead of other important but orthogonal challenges.[2]

The datasets we used in our shared task are given in Table 1. The datasets differ in the following ways:

- **Passage distribution:** Context passages come from many different sources, including Wikipedia, news articles, Web snippets, and textbooks.
- **Question distribution:** Questions are of different styles (e.g., entity-centric, relational) and come from different sources, including crowdworkers, domain experts, and exam writers.
- **Joint distribution:** The relationship between the passage and question also varies. Some questions were written based on the passage, while other questions were written independently, with context passages retrieved afterwards. Some questions were constructed to require multi-hop reasoning on the passage.

Evaluation criteria Systems are evaluated using exact match score (EM) and word-level F1-score (F1), as is common in extractive question answering tasks (Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018). EM only gives credit for predictions that exactly match (one of) the gold answer(s), whereas F1 gives a partial credit for partial word overlap with the gold answer(s). We follow the SQuAD evaluation normalization rules and ignore articles and punctuation when computing EM and F1 scores. While more strict evaluation (Kwiatkowski et al., 2019) computes scores based on the token indexes of the provided context, we compute scores based on answer string match (i.e., the prediction doesn't need to come from exact same annotated span as long as the predicted answer string matches the annotated answer string). We rank systems based on their macro-averaged test F1 scores across the 12 test datasets.

[2]Notatably, the task does *not* test unanswerable (Rajpurkar et al., 2018), multi-turn (Reddy et al., 2019), or open-domain (Chen et al., 2017) question types.

3 Dataset Curation

The MRQA 2019 Shared Task dataset is comprised of many sub-domains, each collected from a separate dataset. The dataset splits and sub-domains are detailed in Table 1. As part of the collection process, we adapted each dataset to conform to the following unified, extractive format:

1. The answer to each question must appear as a span of tokens in the passage.
2. Passages may span multiple paragraphs or documents, but they are concatenated and truncated to the first 800 tokens. This eases the computational requirements for processing large documents efficiently.

The first requirement is motivated by the following reasons:

- Extractive settings are easier to evaluate with stable metrics than abstractive settings.
- Unanswerable questions are hard to synthesize reliably on datasets without them. We investigated using distant supervision to automatically generate unanswerable questions, but found it would introduce a significant amount of noise.
- It is easier to convert multiple-choice datasets to extractive datasets than converting extractive datasets to multiple-choice, as it is difficult to generate challenging alternative answer options.
- Many of popular benchmark datasets are already extractive (or have extractive portions).

3.1 Sub-domain Splits

We partition the 18 sub-domains in the MRQA dataset into three splits:

Split I These sub-domains are available for model training and development, but are not included in evaluation.

	Dataset	Question (Q)	Context (C)	|Q|	|C|	Q ⫫ C	Train	Dev	Test
I	SQuAD	Crowdsourced	Wikipedia	11	137	✗	86,588	10,507	-
	NewsQA	Crowdsourced	News articles	8	599	✓	74,160	4,212	-
	TriviaQA♠	Trivia	Web snippets	16	784	✓	61,688	7,785	-
	SearchQA♠	Jeopardy	Web snippets	17	749	✓	117,384	16,980	-
	HotpotQA	Crowdsourced	Wikipedia	22	232	✗	72,928	5,904	-
	Natural Questions	Search logs	Wikipedia	9	153	✓	104,071	12,836	-
II	BioASQ♠	Domain experts	Science articles	11	248	✓	-	1,504	1,518
	DROP♢	Crowdsourced	Wikipedia	11	243	✗	-	1,503	1,501
	DuoRC♢	Crowdsourced	Movie plots	9	681	✓	-	1,501	1,503
	RACE♡	Domain experts	Examinations	12	349	✗	-	674	1,502
	RelationExtraction♠	Synthetic	Wikipedia	9	30	✓	-	2,948	1,500
	TextbookQA♡	Domain experts	Textbook	11	657	✗	-	1,503	1,508
III	BioProcess♡	Domain experts	Textbook	9	94	✗	-	-	219
	ComplexWebQ♠	Crowdsourced	Web snippets	14	583	✓	-	-	1,500
	MCTest♡	Crowdsourced	Crowdsourced	9	244	✗	-	-	1,501
	QAMR♢	Crowdsourced	Wikipedia	7	25	✗	-	-	1,524
	QAST	Domain experts	Transcriptions	10	298	✗	-	-	220
	TREC♠	Crowdsourced	Wikipedia	8	792	✓	-	-	1,021

Table 1: MRQA sub-domain datasets. The first block presents six domains used for training, the second block presents six given domains used for evaluation during model development and the last block presents six hidden domains used for evaluation. $|\cdot|$ denotes the average length in tokens of the quantity of interest. $Q \perp\!\!\!\perp C$ is true if the question was written independently from the passage used for context. ♠-marked datasets used distant supervision to match questions and contexts, ♡-marked datasets were originally multiple-choice, and ♢-marked datasets are other datasets where only the answer string is given (rather than the exact answer span in the context).

Split II These sub-domains are not available for model training, but are available for model development. Their hidden test portions are included in the final evaluation.

Split III These sub-domains are not available for model training or development. They are completely hidden to the participants and only used for evaluation.

Additionally, we balance the testing portions of Splits II and III by re-partitioning the original sub-domain datasets so that we have 1,500 examples per sub-domain. We partition by context, so that no single context is shared across both development and testing portions of either Split II or Split III.[3]

3.2 Common Preprocessing

Datasets may contain contexts that are comprised of multiple documents or paragraphs. We concatenate all documents and paragraphs together. We separate documents with a `[DOC]` token, insert `[TLE]` tokens before each document title (if provided), and separate paragraphs within a document with a `[PAR]` token.

Many of the original datasets do not have labeled answer spans. For these datasets we provide all occurrences of the answer string in the context in the dataset. Additionally, several of the original datasets contain multiple-choice questions. For these datasets, we keep the correct answer if it is contained in the context, and discard the other options. We filter questions that depend on the specific options (e.g., questions of the form *"which of the following..."* or *"examples of ... include"*). Removing multiple-choice options might introduce ambiguity (e.g., if multiple correct answers appear in the context but not in the original options). For these datasets, we attempt to control for quality by manually verifying random examples.

3.3 Sub-domain Datasets

In this section we describe the datasets used as sub-domains for MRQA. We focus on the modifications made to convert each dataset to the unified MRQA format. Please see Table 1 as well as the associated dataset papers for more details on each sub-domain's properties.

SQuAD (Rajpurkar et al., 2016) We used the SQuAD (**S**tanford **Qu**estion **A**nswering **D**ataset)

[3] We draw examples from each dataset's original test split until it is exhausted, and then augment if necessary from the train and dev splits. This preserves the integrity of the original datasets by ensuring that no original test data is leaked into non-hidden splits of the MRQA dataset.

dataset as the basis for the shared task format.[4] Crowdworkers are shown paragraphs from Wikipedia and are asked to write questions with extractive answers.

NewsQA (Trischler et al., 2017) Two sets of crowdworkers ask and answer questions based on CNN news articles. The "questioners" see only the article's headline and summary while the "answerers" see the full article. We discard questions that have no answer or are flagged in the dataset to be without annotator agreement.

TriviaQA (Joshi et al., 2017) Question and answer pairs are sourced from trivia and quiz-league websites. We use the web version of TriviaQA, where the contexts are retrieved from the results of a Bing search query.

SearchQA (Dunn et al., 2017) Question and answer pairs are sourced from the Jeopardy! TV show. The contexts are composed of retrieved snippets from a Google search query.

HotpotQA (Yang et al., 2018) Crowdworkers are shown two entity-linked paragraphs from Wikipedia and are asked to write and answer questions that require multi-hop reasoning to solve. In the original setting, these paragraphs are mixed with additional distractor paragraphs to make inference harder. We do not include the distractor paragraphs in our setting.

Natural Questions (Kwiatkowski et al., 2019) Questions are collected from information-seeking queries to the Google search engine by real users under natural conditions. Answers to the questions are annotated in a retrieved Wikipedia page by crowdworkers. Two types of annotations are collected: 1) the HTML bounding box containing enough information to completely infer the answer to the question (Long Answer), and 2) the sub-span or sub-spans within the bounding box that comprise the actual answer (Short Answer). We use only the examples that have short answers, and use the long answer as the context.

BioASQ (Tsatsaronis et al., 2015) BioASQ, a challenge on large-scale biomedical semantic indexing and question answering, contains question and answer pairs that are created by domain experts. They are then manually linked to multiple related science (PubMed) articles. We download the full abstract of each of the linked articles to use as individual contexts (e.g., a single question can be linked to multiple, independent articles to create multiple QA-context pairs). We discard abstracts that do not exactly contain the answer.

DROP (Dua et al., 2019) DROP (Discrete Reasoning Over the content of Paragraphs) examples were collected similarly to SQuAD, where crowdworkers are asked to create question-answer pairs from Wikipedia paragraphs. The questions focus on quantitative reasoning, and the original dataset contains non-extractive numeric answers as well as extractive text answers. We restrict ourselves to the set of questions that are extractive.

DuoRC (Saha et al., 2018) We use the ParaphraseRC split of the DuoRC dataset. In this setting, two different plot summaries of the same movie are collected—one from Wikipedia and the other from IMDb. Two different sets of crowdworkers ask and answer questions about the movie plot, where the "questioners" are shown only the Wikipedia page, and the "answerers" are shown only the IMDb page. We discard questions that are marked as unanswerable.

RACE (Lai et al., 2017) ReAding Comprehension Dataset From Examinations (RACE) is collected from English reading comprehension exams for middle and high school Chinese students. We use the high school split (which is more challenging) and also filter out the implicit "fill in the blank" style questions (which are unnatural for this task).

RelationExtraction (Levy et al., 2017) Given a slot-filling dataset,[5] relations among entities are systematically transformed into question-answer pairs using templates. For example, the $educated\_at(x, y)$ relationship between two entities x and y appearing in a sentence can be expressed as *"Where was x educated at?"* with answer y. Multiple templates for each type of relation are collected. We use the dataset's zero-shot benchmark split (generalization to unseen relations), and only keep the positive examples.

TextbookQA (Kembhavi et al., 2017) TextbookQA is collected from lessons from middle school Life Science, Earth Science, and Physical

[4] A few paragraphs are long, and we discard the QA pairs that do not align with the first 800 tokens (1.1% of examples).

[5] The authors use the WikiReading dataset (Hewlett et al., 2016) for the underlying slot-filling task.

Science textbooks. We do not include questions that are accompanied with a diagram, or that are "True or False" questions.

BioProcess (Berant et al., 2014) Paragraphs are sourced from a biology textbook, and question and answer pairs about those paragraphs are then created by domain experts.

ComplexWebQ (Talmor and Berant, 2018) ComplexWebQuestions is collected by crowdworkers who are shown compositional, formal queries against Freebase, and are asked to rephrase them in natural language. Thus, by design, questions require multi-hop reasoning. For the context, we use the default web snippets provided by the authors. We use only single-answer questions of type "composition" or "conjunction".

MCTest (Richardson et al., 2013) Passages accompanied with questions and answers are written by crowdworkers. The passages are fictional, elementary-level, children's stories.

QAMR (Michael et al., 2018) To construct the Question-Answer Meaning Representation (QAMR) dataset, crowdworkers are presented with an English sentence along with target non-stopwords from the sentence. They are then asked to create as many question-answer pairs as possible that contain at least one of the target words (and for which the answer is a span of the sentence). These questions combine to cover most of the predicate-argument structures present. We use only the filtered[6] subset of the Wikipedia portion of the dataset.

QAST (Lamel et al., 2008) We use Task 1 of the Question Answering on Speech Transcriptions (QAST) dataset, where contexts are taken from manual transcripts of spoken lectures on "speech and language processing." Questions about named entities found in the transcriptions are created by English native speakers. Each lecture contains around 1 hour of transcribed text. To reduce the length to meet our second requirement ($\leq$ 800 tokens), for each question we manually selected a sub-section of the lecture that contained the answer span, as well as sufficient surrounding context to answer it.

TREC (Baudiš and Šedivý, 2015) The Text REtrieval Conference (TREC) dataset is curated from the TREC QA tasks (Voorhees and Tice, 2000) from 1999-2002. The questions are factoid. Accompanying passages are supplied using the Document Retriever from Chen et al. (2017), if the answer is found within the first 800 tokens of any of the top 5 retrieved Wikipedia documents (we take the highest ranked document if multiple documents meet this requirement).

[6]The questions that are valid and non-redundant.

4 Baseline Model

We implemented a simple, multi-task baseline model based on BERT (Devlin et al., 2018), following the MultiQA model (Talmor and Berant, 2019). Our method works as follows:

Modeling Given a question q consisting of m tokens $\{q_1, \ldots, q_m\}$ and a passage p of n tokens $\{p_1, \ldots, p_n\}$, we first concatenate q and p with special tokens to obtain a joint context $\{\texttt{[CLS]}, q_1, \ldots, q_m, \texttt{[SEP]}, p_1, \ldots, p_n, \texttt{[SEP]}\}$. We then encode the joint context with BERT to obtain contextualized passage representations $\{\mathbf{h}_1, \ldots, \mathbf{h}_n\}$. We train separate MLPs to predict start and end indices independently, and decode the final span using $\arg\max_{i,j}\{p_{start}(i) \times p_{end}(j)\}$.

Preprocessing Following Devlin et al. (2018), we create p and q by tokenizing every example using a vocabulary of 30,522 word pieces. As BERT accepts a maximum sequence length of 512, we generate multiple chunks $\{p^{(1)}, \ldots, p^{(k)}\}$ per example by sliding a 512 token window (of the joint context, including q) over the entire length of the original passage, with a stride of 128 tokens.

Training During training we select only the chunks that contain answers. We maximize the log-likelihood of the first occurrence of the gold answer in each of these chunks, and back-propagate into BERT's parameters (and the MLP parameters). At test time we output the span with the maximal logit across all chunks.

Multi-task Training We sample up to 75K examples from each training dataset, combine them, and create mixed batches of examples from all of the data. We then follow the same training procedure as before on all the composed training dataset batches.

5 Shared Task Submissions

Our shared task lasted for 3 months from May to August in 2019. All submissions were handled through the CodaLab platform.[7] In total, we received submissions from 10 different teams for the final evaluation (Table 2). Of these, 6 teams submitted their system description paper. We will describe each of them briefly below.

5.1 D-Net (Li et al., 2019)

The submission from Baidu adopts multiple pre-trained language models (LMs), including BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), and ERNIE 2.0 (Zhang et al., 2019). Unlike other submissions which use only one pre-trained LM, they experiment with 1) training LMs with extra raw text data drawn from science questions and search snippets domains, and 2) multi-tasking with auxiliary tasks such as natural language inference and paragraph ranking (Williams et al., 2017). Ultimately, however, the final system is an ensemble of an XLNet-based model and an ERNIE-based model, without auxiliary multitask or augmented LM training.

5.2 Delphi (Longpre et al., 2019)

The submission from Apple investigates the effects of pre-trained language models (BERT vs XLNet), various data sampling strategies, and data augmentation techniques via back-translation. Their final submission uses XLNet (Yang et al., 2019) as the base model, with carefully sampled training instances from negative examples (hence augmenting the model with a no-answer option) and the six training datasets. The final submission does not include data augmentation, as it did not improve performance during development.

5.3 HLTC (Su et al., 2019)

The submission from HKUST studies different data-feeding schemes, namely shuffling instances from all datasets versus shuffling dataset-ordering only. Their submission is built on top of XLNet, with a multilayer perceptron layer for span prediction. They also attempted to substitute the MLP layer with a more complex attention-over-attention (AoA) (Cui et al., 2017) layer on top of XLNet, but did not find it to be helpful.

[7] https://worksheets.codalab.org

5.4 CLER (Takahashi et al., 2019)

The submission from Fuji Xerox adds a mixture-of-experts (MoE) (Jacobs et al., 1991) layer on top of a BERT-based architecture. They also use a multi-task learning framework trained together with natural language inference (NLI) tasks. Their final submission is an ensemble of three models trained with different random seeds.

5.5 Adv. Train (Lee et al., 2019)

The submission from 42Maru and Samsung Research proposes an adversarial training framework, where a domain discriminator predicts the underlying domain label from the QA model's hidden representations, while the QA model tries to learn to arrange its hidden representations such that the discriminator is thwarted. Through this process, they aim to learn domain (dataset) invariant features that can generalize to unseen domains. The submission is built based on the provided BERT baselines.

5.6 HierAtt (Osama et al., 2019)

The submission from Alexandria University uses the BERT-Base model to provide feature representations. Unlike other models which allowed fine-tuning of the language model parameters during training, this submission only trains model parameters associated with the question answering task, while keeping language model parameters frozen. The model consists of two attention mechanisms: one bidirectional attention layer used to model the interaction between the passage and the question, and one self-attention layer applied to both the question and the passage.

6 Results

6.1 Main Results

Table 3 lists the macro-averaged F1 scores of all the submissions on both the development and testing portions of the MRQA dataset. The teams are ranked by the F1 scores on the hidden testing portions of the 12 datasets (Split II and III in Section 3.1). As seen in Table 3, many of the submissions outperform our BERT-Large baseline significantly. The best-performing system, D-Net (Li et al., 2019), achieves an F1 score of 72.5, which is a 10.7 point absolute improvement over our baseline, and 11.5 and 10.0 point improvements, respectively, on Split II (with the development por-

Model	Affliation
D-Net (Li et al., 2019)	Baidu Inc.
Delphi (Longpre et al., 2019)	Apple Inc.
FT_XLNet	Harbin Institute of Technology
HLTC (Su et al., 2019)	Hong Kong University of Science & Technology
BERT-cased-whole-word	Aristo @ AI2
CLER (Takahashi et al., 2019)	Fuji Xerox Co., Ltd.
Adv. Train (Lee et al., 2019)	42Maru and Samsung Research
BERT-Multi-Finetune	Beijing Language and Culture University
PAL IN DOMAIN	University of California Irvine
HierAtt (Osama et al., 2019)	Alexandria University

Table 2: List of participants, ordered by the macro-averaged F1 score on the hidden evaluation set.

Model	Split I	Split II	Split II	Split III	Split II + III
Portion (# datasets)	Dev (6)	Dev (6)	Test (6)	Test (6)	Test (12)
D-Net (Li et al., 2019)	**84.1**	**69.7**	**68.9**	**76.1**	**72.5**
Delphi (Longpre et al., 2019)	82.3	68.5	66.9	74.6	70.8
FT_XLNet	82.9	68.0	66.7	74.4	70.5
HLTC (Su et al., 2019)	81.0	65.9	65.0	72.9	69.0
BERT-cased-whole-word	79.4	61.1	61.4	71.2	66.3
CLER (Takahashi et al., 2019)	80.2	62.7	62.5	69.7	66.1
Adv. Train (Lee et al., 2019)	76.8	57.1	57.9	66.5	62.2
Ours: BERT-Large	76.3	57.1	57.4	66.1	61.8
BERT-Multi-Finetune	74.2	53.3	56.0	64.7	60.3
Ours: BERT-Base	74.7	54.6	54.6	62.4	58.5
HierAtt (Osama et al., 2019)	71.1	48.7	50.5	61.7	56.1

Table 3: Performance as F1 score on the shared task. Each score is macro-averaged across individual datasets. The last column (test portion of Split II and III) is used for the final ranking. Our baselines are shaded in yellow, and the submissions which did not present system description papers are shaded in grey.

		#	Best	BERT Large	Impr.
Question Type	Crowdsourced	6	69.9	58.5	11.5
	Synthetic	1	88.9	84.7	4.2
	Domain experts	5	71.5	60.5	11.5
Context Type	Wikipedia	4	73.4	62.3	11.1
	Education	4	68.2	56.2	12.0
	Others	4	76.1	66.8	9.3
$\mathbf{Q} \perp\!\!\!\perp \mathbf{C}$	✓	5	73.0	63.8	9.2
	✗	7	72.2	60.3	11.9

Table 4: Macro-averaged F1 scores based on the dataset characteristics as defined in Table 1. Best denotes the best shared task result and Base denotes our BERT-Large baseline.

tions provided) and Split III datasets (completely hidden to the participants).

We evaluate all the submissions on the in-domain datasets (Split I) in Table 3 and find that there is a very strong correlation between in-domain and out-of-domain performance. The top submissions on the out-of-domain datasets also obtain the highest scores on the six datasets that we provided for training.

We present per-dataset performances for 12 evaluation datasets in the appendix. Across the board, many submitted systems greatly outperform our baselines. Among the 12 datasets, performance on the DROP dataset has improved the most—from 43.5 F1 to 61.5 F1—while performance on the RelationExtraction dataset has improved the least (84.9 F1 vs. 89.0 F1). The models with higher average scores seemed to outperform in most datasets: the performance rankings of submissions are mostly preserved on individual datasets.

6.2 Summary of Findings

Improvements per data types We analyzed the average performance across the various types of datasets that are represented in Table 1. Table 4 summarizes our observations: (1) the datasets with naturally collected questions (either crowdsourced or curated by domain experts) all obtain large improvements; (2) The datasets collected from Wikipedia or education materials (textbooks and Science articles) receive bigger gains compared to those collected from Web snippets or transcriptions; and (3) There is a bigger improvement for datasets in which questions are posed dependent on the passages compared to those with independently collected questions (11.9 vs. 9.2 points).

Pre-trained language models The choice of pre-trained language model has a significant impact on the QA performance, as well as the generalization ability. Table 5 summarizes the pre-trained models each submission is based on, along with its evaluation F1 score. The top three performing systems all use XLNet instead of BERT-Large—this isolated change in pre-trained language model alone yields a significant gain in overall in- and out-of-domain performance. Li et al. (2019) argues that XLNet shows superior performances on datasets with discrete reasoning, such as DROP and RACE. Su et al. (2019), however, also use XLNet, but does not show strong gains on the DROP or RACE datasets.

The winning system ensembled two *different* pre-trained language models. Only one other submission (Takahashi et al., 2019) used an ensemble for their final submission, merging the same LM with different random seeds.

Model	**Base Language Model**	**Eval F1 (II + III)**
D-Net	XLNet-L + ERNIE 2.0	72.5
Delphi	XLNet-L	70.8
HLTC	XLNet-L	69.0
CLER	BERT-L	66.1
Adv. Train	BERT-L	62.2
BERT-Large	BERT-L	61.8
HierAtt	BERT-B	56.1

Table 5: Pretrained language models used in the shared task submissions. *-L and *-B denote large and base versions of the models.

Data sampling Our shared task required all participants to use our provided training data, compiled from six question answering datasets, and disallowed the use of any other question-answering data for training. Within these restrictions, we encouraged participants to explore *how* to best utilize the provided data.

Inspired by Talmor and Berant (2019), two submissions (Su et al., 2019; Longpre et al., 2019) analyzed similarities between datasets. Unsurprisingly, the performance improved significantly when fine-tuned on the training dataset most similar to the evaluation dataset of interest. Su et al. (2019) found each of the development (Split II) datasets resembles one or two training datasets (Split I)—and thus training with all datasets is crucial for generalization across the

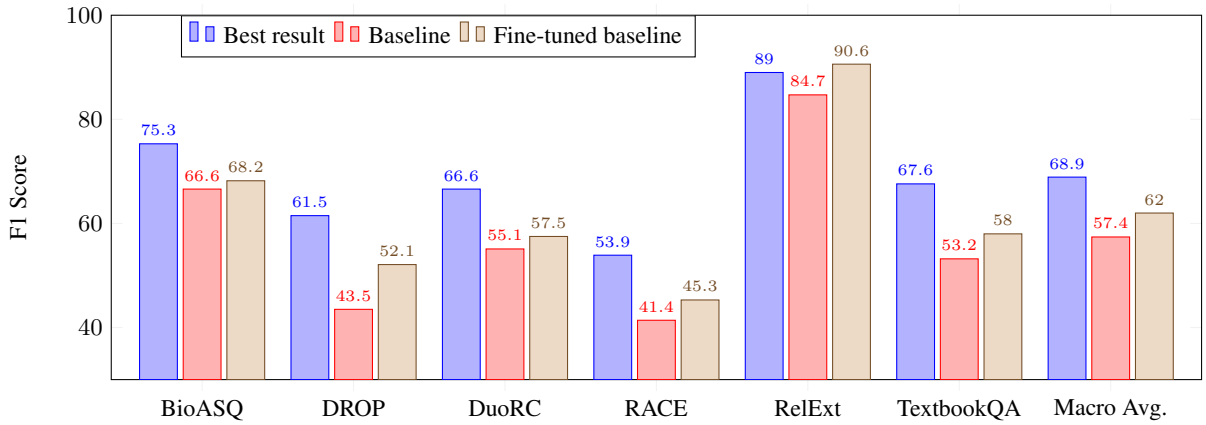

Figure 1: F1 scores on Split II sub-domains (test portions) comparing the best submitted system (D-Net) against our BERT-Large baseline. The third result for each dataset is from individually fine-tuning the BERT-Large baseline on the in-domain dev portion of the same dataset (i.e., Split II (dev)).

multiple domains. They experimented with data-feeding methodologies, and found that shuffling instances of all six training datasets is more effective than sequentially feeding all examples from each dataset, one dataset after another.

Additionally, Longpre et al. (2019) observed that the models fine-tuned on SearchQA and TriviaQA achieve relatively poor results across all the evaluation sets (they are both trivia-based, distantly supervised, and long-context datasets). Downsampling examples from these datasets increases the overall performance. They also found that, although our shared task focuses on answerable questions, sampling negative examples leads to significant improvements (up to +1.5 F1 on Split II and up to +4 F1 on Split I). Since most systems follow our baseline model (Section 4) by doing inference over *chunks* of tokens, not all examples fed to these models are actually guaranteed to contain an answer span.

Multi-task learning Two submissions attempted to learn the question answering model together with other auxiliary tasks, namely natural language inference (Takahashi et al., 2019; Li et al., 2019) or paragraph ranking (Li et al., 2019) (i.e., classifying whether given passages contains an answer to the question or not). This could improve the generalization performance on question answering for two reasons. First, the additional training simply exposes the model to more diverse domains, as the entailment dataset (Williams et al., 2017) contains multiple domains ranging from fiction to telephone conversations. Second, reasoning about textual entailment is often necessary for question answering, while passage ranking (or classification) is an easier version of extractive question answering, where the model has to identify the passage containing the answer instead of exact span.

Both systems introduced task-specific fully connected layers while sharing lower level representations across different tasks. While Takahashi et al. (2019) showed a modest gain by multi-tasking with NLI tasks (+0.7 F1 score on the development portion of Split II), Li et al. (2019) reported that multitasking did not improve the performance of their best model.

Adversarial Training One submission (Lee et al., 2019) introduced an adversarial training framework for generalization. The goal is to learn domain-invariant features (i.e., features that can generalize to unseen test domains) by jointly training with a domain discriminator, which predicts the dataset (domain) for each example. According to Lee et al. (2019), this adversarial training helped on most of the datasets (9 out of 12), but also hurt performance on some of them. It finally led to +1.9 F1 gain over their BERT-Base baseline, although the gain was smaller (+0.4 F1) for their stronger BERT-Large baseline.

Ensembles Most extractive QA models, which output a logit for the start index and another for the end index, can be ensembled by adding the start and end logits from models trained with different random seeds. This has shown to improve performances across many model classes, as can be seen from most dataset leaderboards. The results from the shared task also show similar trends. A

few submissions (Takahashi et al., 2019; Li et al., 2019) tried ensembling, and all reported modest gains. While ensembling is a quick recipe for a small gain in performance, it also comes at the cost of computational efficiency—both at training and at inference time.

Related to ensembling, Takahashi et al. (2019) uses a mixture of experts (Jacobs et al., 1991) layer, which learns a gating function to ensemble different weights, adaptively based on the input.

6.3 Comparison to In-domain Fine-tuning

Lastly, we report how the best shared task performance compares to in-domain fine-tuning performance of our baseline. Section 6.1 shows large improvements by the top shared task model, D-Net, over our baseline. We analyze to what extent the reduced performance on out-of-domain datasets can be overcome by exposing the baseline to only a few samples from the target distributions. As suggested by Liu et al. (2019), if the model can generalize with a few examples from the new domain, poor performance on that domain is an indicator of a lack of training data diversity, rather than of fundamental model generalization weaknesses.

Figure 1 presents our results on the six datasets from Split II, where we have individually fine-tuned the BERT-Large baseline on each of the Split II dev datasets and tested on the Split II test datasets. We see that while the gap to D-Net shrinks on all datasets (overall performance increases by 4.6 F1), surprisingly it is only completely bridged in one of the settings (RelationExtraction). This is potentially because this dataset covers only a limited number of relations, so having in-domain data helps significantly. This suggests that D-Net (and the others close to it in performance) is an overall stronger model—a conclusion also supported by its gain on in-domain data (Split I).

7 Conclusions

We have presented the MRQA 2019 Shared Task, which focused on testing whether reading comprehension systems can generalize to examples outside of their training domain. Many submissions improved significantly over our baseline, and investigated a wide range of techniques.

Going forward, we believe it will become increasingly important to build NLP systems that generalize across domains. As NLP models become more widely deployed, they must be able to handle diverse inputs, many of which may differ from those seen during training. By running this shared task and releasing our shared task datasets, we hope to shed more light how to build NLP systems that generalize beyond their training distribution.

Acknowledgements

We would like to thank Jonathan Berant, Percy Liang, and Luke Zettlemoyer for serving as our steering committee. We are grateful to Baidu, Facebook, and Naver for providing funding for our workshop. We thank Anastasios Nentidis and the entire BioASQ organizing committee for letting us use BioASQ shared task data for our task, and for hosting the data files. We also thank Valerie Mapelli and ELRA for providing us with the QAST data: CLEF QAST (2007-2009) – Evaluation Package, ELRA catalogue (http://catalog.elra.info), CLEF QAST (2007-2009) – Evaluation Package, ISLRN: 460-370-870-489-0, ELRA ID: ELRA-E0039. Finally, we thank the CodaLab Worksheets team for their help with running the shared task submissions.

References

Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the yodaqa system. In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283*.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *EMNLP*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *ACL*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. pages 2174–2184.

Yiming Cui, Zhipeng Chen, Si Wei, Ting Liu Shijin Wang, , and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*.

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

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over wikipedia. In *ACL*.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *TACL*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Lori Lamel, Sophie Rosset, Christelle Ayache, Djamel Mostefa, Jordi Turmo, and Pere Comas. 2008. Question answering on speech transcriptions: the QAST evaluation in CLEF. In *LREC 2008*.

Seanie Lee, Donggyu Kim, and Jangwon Park. 2019. Domain-agnostic question-answering with adversarial training. In *Proceedings of 2nd Machine Reading for Reading Comprehension Workshop at EMNLP*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*.

Hongyu Li, Xiyuan Zhang, Yibing Liu, Yiming Zhang, Quan Wang, Xiangyang Zhou, Jing Liu, Hua Wu, and Haifeng Wang. 2019. D-NET: A simple framework for improving the generalization of machine reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension Workshop at EMNLP*.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. 2019. MRQA 2019 shared task: Fine-tuned xlnet with negative sampling for multi-domain question answering. In *Proceedings of 2nd Machine Reading for Reading Comprehension Workshop at EMNLP*.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. In *NAACL*.

Reham Osama, Nagwa El-Makky, and Marwan Torki. 2019. Question answering using hierarchical attention on top of bert features. In *Proceedings of 2nd Machine Reading for Reading Comprehension Workshop at EMNLP*.

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*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *TACL*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *ACL*.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In *ACL*.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing question answering system with pre-trained language model fine-tuning. In *Proceedings of 2nd Machine Reading for Reading Comprehension Workshop at EMNLP*.

Takumi Takahashi, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. 2019. CLER: Cross-task learning with expert representation to generalize reading and understanding. In *Proceedings of 2nd Machine Reading for Reading Comprehension Workshop at EMNLP*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL*.

Alon Talmor and Jonathan Berant. 2019. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. In *ACL*.

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

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

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *arXiv: 1901.11373*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *ACL*.

Appendix

We present the per-dataset performances in Table 6 and Table 7 for shared task submissions and our baselines.

Model	BioASQ		DROP		DuoRC		RACE		RelExt		TextbookQA	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
D-Net	**61.2**	**75.3**	**50.7**	**61.5**	**54.7**	**66.6**	**39.9**	53.5	**80.1**	**89.0**	**57.2**	**67.6**
Delphi	60.3	72.0	48.5	58.9	53.3	63.4	39.4	**53.9**	79.2	87.9	56.5	65.5
FT_XLNet	59.3	72.9	48.0	58.3	52.7	63.8	39.4	53.8	79.0	87.2	53.6	64.2
HLTC	59.6	74.0	41.0	51.1	51.7	63.1	37.2	50.5	76.5	86.2	55.5	65.2
BERT-cased-whole-word	57.8	72.9	43.1	53.2	42.3	53.5	35.0	48.7	78.5	87.9	43.9	51.9
CLER	53.2	68.8	37.7	47.5	51.6	62.9	31.9	45.0	78.6	87.7	53.5	62.9
Adv. Train	45.1	60.5	34.8	43.8	46.2	57.3	29.6	42.8	74.3	84.9	48.8	58.0
Ours: BERT-Large	49.7	66.6	33.9	43.5	43.4	55.1	29.0	41.4	72.5	84.7	45.6	53.2
BERT-Multi-Finetune	48.7	64.8	30.4	40.3	43.7	54.7	26.4	38.7	75.3	85.0	44.0	52.4
Ours: BERT-Base	46.4	60.8	28.3	37.9	42.8	53.3	28.2	39.5	73.3	83.9	44.3	52.0
HierAtt	43.0	59.1	24.4	34.8	38.5	49.6	24.6	37.4	67.9	81.3	32.1	40.5

Table 6: Performance on the six datasets of Split II (test portion). EM: exact match, F1: word-level F1-score.

Model	BioProcess		ComWebQ		MCTest		QAMR		QAST		TREC	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
D-NET	**61.3**	**75.6**	**67.8**	**68.3**	67.8	**80.8**	60.4	**76.1**	75.0	88.8	51.8	**66.8**
Delphi	58.9	74.2	55.1	62.3	**68.0**	80.2	**61.0**	75.3	**78.6**	**89.9**	**55.0**	65.8
FT_XLNet	62.6	75.2	54.8	62.7	66.0	79.6	56.5	73.4	76.8	90.0	51.8	65.5
HLTC	56.2	72.9	54.7	61.4	64.6	78.7	56.4	72.5	75.9	88.8	49.9	63.4
BERT-cased-whole-word	56.2	71.5	52.4	60.7	63.8	76.4	56.1	71.5	69.6	85.3	43.6	61.6
CLER	48.0	68.4	52.6	61.2	59.9	73.1	54.3	71.4	65.0	84.3	42.7	60.0
Adv. Train	46.1	62.9	48.7	56.9	57.2	70.9	56.8	71.7	56.8	77.8	42.6	58.8
Ours: BERT-Large	46.1	63.6	51.8	59.1	59.5	72.2	48.2	67.4	62.3	80.8	36.3	53.6
BERT-Multi-Finetune	43.4	58.8	49.6	57.7	59.2	72.2	48.6	67.0	60.0	80.1	34.6	52.3
Ours: BERT-Base	38.4	57.4	47.4	55.3	54.2	66.1	47.8	64.8	58.6	77.0	36.7	54.0
HierAtt	44.3	60.8	41.9	51.2	54.2	67.9	48.0	66.0	50.9	75.5	27.7	48.7

Table 7: Results on the six datasets of Split III. EM: exact match, F1: word-level F1-score.