MULTIQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension

Alon Talmor[1,2] **Jonathan Berant**[1,2]

[1]School of Computer Science, Tel-Aviv University
[2]Allen Institute for Artificial Intelligence
{alontalmor@mail,joberant@cs}.tau.ac.il

Abstract

A large number of reading comprehension (RC) datasets has been created recently, but little analysis has been done on whether they generalize to one another, and the extent to which existing datasets can be leveraged for improving performance on new ones. In this paper, we conduct such an investigation over ten RC datasets, training on one or more *source* RC datasets, and evaluating generalization, as well as transfer to a target RC dataset. We analyze the factors that contribute to generalization, and show that training on a source RC dataset and transferring to a target dataset substantially improves performance, even in the presence of powerful contextual representations from BERT (Devlin et al., 2019). We also find that training on multiple source RC datasets leads to robust generalization and transfer, and can reduce the cost of example collection for a new RC dataset. Following our analysis, we propose MULTIQA, a BERT-based model, trained on multiple RC datasets, which leads to state-of-the-art performance on five RC datasets. We share our infrastructure for the benefit of the research community.

1 Introduction

Reading comprehension (RC) is concerned with reading a piece of text and answering questions about it (Richardson et al., 2013; Berant et al., 2014; Hermann et al., 2015; Rajpurkar et al., 2016). Its appeal stems both from the clear application it proposes, but also from the fact that it allows to probe many aspects of language understanding, simply by posing questions on a text document. Indeed, this has led to the creation of a large number of RC datasets in recent years.

While each RC dataset has a different focus, there is still substantial overlap in the abilities required to answer questions across these datasets. Nevertheless, there has been relatively little work (Min et al., 2017; Chung et al., 2018; Sun et al., 2018) that explores the relations between the different datasets, including whether a model trained on one dataset generalizes to another. This research gap is highlighted by the increasing interest in developing and evaluating the generalization of language understanding models to new setups (Yogatama et al., 2019; Liu et al., 2019).

In this work, we conduct a thorough empirical analysis of generalization and transfer across 10 RC benchmarks. We train models on one or more source RC datasets, and then evaluate their performance on a target test set, either without any additional target training examples (*generalization*) or with additional target examples (*transfer*). We experiment with DOCQA (Clark and Gardner, 2018), a standard and popular RC model, as well as a model based on BERT (Devlin et al., 2019), which provides powerful contextual representations.

Our generalization analysis confirms findings that current models over-fit to the particular training set and generalize poorly even to similar datasets. Moreover, BERT representations substantially improve generalization. However, we find that the contribution of BERT is much more pronounced on Wikipedia (which BERT was trained on) and Newswire, but quite moderate when documents are taken from web snippets.

We also analyze the main causes for poor generalization: (a) differences in the language of the text document, (b) differences in the language of the question, and (c) the type of language phenomenon that the dataset explores. We show how generalization is related to these factors (Figure 1) and that performance drops as more of these factors accumulate.

Our transfer experiments show that pre-training on one or more source RC datasets substantially improves performance when fine-tuning on a tar-

get dataset. An interesting question is whether such pre-training improves performance even in the presence of powerful language representations from BERT. We find the answer is a conclusive yes, as we obtain consistent improvements in our BERT-based RC model.

We find that training on multiple source RC datasets is effective for both generalization and transfer. In fact, training on multiple datasets leads to the same performance as training from the target dataset alone, but with roughly three times fewer examples. Moreover, we find that when using the high capacity BERT-large, one can train a single model on multiple RC datasets, and obtain close to or better than state-of-the-art performance on all of them, without fine-tuning to a particular dataset.

Armed with the above insights, we train a large RC model on multiple RC datasets, termed MULTIQA. Our model leads to new state-of-the-art results on five datasets, suggesting that in many language understanding tasks the size of the dataset is the main bottleneck, rather than the model itself.

Last, we have developed infrastructure (on top of AllenNLP (Gardner et al., 2018)), where experimenting with multiple models on multiple RC datasets, mixing datasets, and performing fine-tuning, are trivial. It is also simple to expand the infrastructure to new datasets and new setups (abstractive RC, multi-choice, etc.). We will open source our infrastructure, which will help researchers evaluate models on a large number of datasets, and gain insight on the strengths and shortcoming of their methods. We hope this will accelerate progress in language understanding.

To conclude, we perform a thorough investigation of generalization and transfer in reading comprehension over 10 RC datasets. Our findings are:

- An analysis of generalization on two RC models, illustrating the factors that influence generalization between datasets.
- Pre-training on a RC dataset and fine-tuning on a target dataset substantially improves performance even in the presence of contextualized word representations (BERT).
- Pre-training on multiple RC datasets improves transfer and generalization and can reduce the cost of example annotation.
- A new model, MULTIQA, that improves state-of-the-art performance on five datasets.
- Infrastructure for easily performing experiments on multiple RC datasets.

Dataset	Size	Context	Question	Multi-hop
SQUAD	108K	Wikipedia	crowd	No
NEWSQA	120K	Newswire	crowd	No
SEARCHQA	140K	Snippets	trivia	No
TRIVIAQA	95K	Snippets	trivia	No
HOTPOTQA	113K	Wikipedia	crowd	Yes
CQ	2K	Snippets	Web queries/KB	No
CWQ	35K	Snippets	crowd/KB	Yes
COMQA	11K	Snippets	WikiAnswers	No
WIKIHOP	51K	Wikipedia	KB	Yes
DROP	96K	Wikipedia	crowd	Yes

Table 1: Characterization of different RC datasets. The top part corresponds to *large datasets*, and the bottom to small datasets.

The uniform format datasets can be downloaded from `www.tau-nlp.org/multiqa`. The code for the AllenNLP models is available at `github.com/alontalmor/multiqa`.

2 Datasets

We describe the 10 datasets used for our investigation. Each dataset provides question-context-answer triples $\{(q_i, c_i, a_i)\}_{i=1}^{N}$ for training, and a model maps an unseen question-context pair (q, c) to an answer a. For simplicity, we focus on the single-turn extractive setting, where the answer a is a span in the context c. Thus, we do not evaluate abstractive (Nguyen et al., 2016) or conversational datasets (Choi et al., 2018; Reddy et al., 2018).

We broadly distinguish *large datasets* that include more than 75K examples, from *small datasets* that contain less than 75K examples. In §4, we will fix the size of the large datasets to control for size effects, and always train on exactly 75K examples per dataset.

We now shortly describe the datasets, and provide a summary of their characteristics in Table 1. The table shows the original size of each dataset, the source for the context, how questions were generated, and whether the dataset was specifically designed to probe multi-hop reasoning.

The large datasets used are:

1. SQUAD (Rajpurkar et al., 2016): Crowdsourcing workers were shown Wikipedia paragraphs and were asked to author questions about their content. Questions mostly require soft matching of the language in the question to a local context in the text.
2. NEWSQA (Trischler et al., 2017): Crowdsourcing workers were shown a CNN article (longer than SQUAD) and were asked to author questions about its content.

3. SEARCHQA (Dunn et al., 2017): Trivia questions were taken from Jeopardy! TV show, and contexts are web snippets retrieved from Google search engine for those questions, with an average of 50 snippets per question.
4. TRIVIAQA (Joshi et al., 2017): Trivia questions were crawled from the web. In one variant of TRIVIAQA (termed TQA-W), Wikipedia pages related to the questions are provided for each question. In another, web snippets and documents from Bing search engine are given. For the latter variant, we use only the web snippets in this work (and term this TQA-U). In addition, we replace Bing web snippets with Google web snippets (and term this TQA-G).
5. HOTPOTQA (Yang et al., 2018): Crowdsourcing workers were shown pairs of related Wikipedia paragraphs and asked to author questions that require multi-hop reasoning over the paragraphs. There are two versions of HOTPOTQA: the first where the context includes the two gold paragraphs and eight "distractor" paragraphs, and a second, where 10 paragraphs retrieved by an information retrieval (IR) system are given. Here, we use the latter version.

The small datasets are:

1. CQ (Bao et al., 2016): Questions are real Google web queries crawled from Google Suggest, originally constructed for querying the KB Freebase (Bollacker et al., 2008). However, the dataset was also used as a RC task with retrieved web snippets (Talmor et al., 2017).
2. CWQ (Talmor and Berant, 2018c): Crowdsourcing workers were shown compositional formal queries against Freebase and were asked to re-phrase them in natural language. Thus, questions require multi-hop reasoning. The original work assumed models contain an IR component, but the authors also provided default web snippets, which we use here. The repartitioned version 1.1 was used. (Talmor and Berant, 2018a)
3. WIKIHOP (Welbl et al., 2017) Questions are entity-relation pairs from Freebase, and are not phrased in natural language. Multiple Wikipedia paragraphs are given as context, and the dataset was constructed such that multi-hop reasoning is needed for answering the question.
4. COMQA (Abujabal et al., 2018): Questions are real user questions from the WikiAnswers community QA platform. No contexts are provided, and thus we augment the questions with web snippets retrieved from Google search engine.
5. DROP (Dua et al., 2019): Contexts are Wikipedia paragraphs and questions are authored by crowdsourcing workers. This dataset focuses on quantitative reasoning. Because most questions are not extractive, we only use the 33,573 extractive examples in the dataset (but evaluate on the entire development set).

3 Models

We carry our empirical investigation using two models. The first is DOCQA (Clark and Gardner, 2018), and the second is based on BERT (Devlin et al., 2019), which we term BERTQA. We now describe the pre-processing on the datasets, and provide a brief description of the models. We emphasize that in all our experiments we use exactly the same training procedure for all datasets, with minimal hyper-parameter tuning.

Pre-processing Examples in all datasets contain a question, text documents, and an answer. To generate an extractive example we (a) *Split*: We define a length L and split every paragraph whose length is $> L$ into chunks using a few manual rules. (b) *Sort*: We sort all chunks (paragraphs whose length is $\leq L$ or split paragraphs) by cosine similarity to the question in tf-idf space, as proposed by Clark and Gardner (2018). (c) *Merge*: We go over the sorted list of chunks and greedily merge them to the largest possible length that is at most L, so that the RC model will be exposed to as much context as possible. The final context is the merged list of chunks $c = (c_1, \ldots, c_{|c|})$ (d) We take the gold answer and mark all spans that match the answer.

DOCQA (Clark and Gardner, 2018): A widely-used RC model, based on BIDAF (Seo et al., 2016), that encodes the question and document with bidirectional RNNs, performs attention between the question and document, and adds self-attention on the document side.

We run DOCQA on each chunk c_i, where the input is a sequence of up to $L(= 400)$ tokens represented as GloVE embeddings (Pennington et al., 2014). The output is a distribution over the start and end positions of the predicted span, and we output the span with highest probability across all chunks. At training time, DOCQA uses a shared-norm objective that normalizes the probability dis-

tribution over spans from all chunks. We define the gold span to be the first occurrence of the gold answer in the context c.

BERTQA (Devlin et al., 2019): For each chunk, we apply the standard implementation, where the input is a sequence of $L = 512$ wordpiece tokens composed of the question and chunk separated by special tokens `[CLS] <question> [SEP] <chunk> [SEP]`. A linear layer with softmax over the top-layer `[CLS]` outputs a distribution over start and end span positions.

We train over each chunk separately, back-propagating into BERT's parameters. We maximize the log-likelihood of the first occurrence of the gold answer in each chunk that contains the gold answer. At test time, we output the span with the maximal logit across all chunks.

4 Controlled Experiments

We now present controlled experiments aiming to explore generalization and transfer of models trained on a set of RC datasets to a new target.

4.1 Do models generalize to unseen datasets?

We first examine generalization – whether models trained on one dataset generalize to examples from a new distribution. While different datasets differ substantially, there is overlap between them in terms of: (i) the language of the question, (ii) the language of the context, and (iii) the type of linguistic phenomena the dataset aims to probe. Our goal is to answer (a) do models over-fit to a particular dataset? How much does performance drop when generalizing to a new dataset? (b) Which datasets generalize better to which datasets? What properties determine generalization?

We train DOCQA and BERTQA (we use BERT-base) on six large datasets (for TRIVIAQA we use TQA-G and TQA-W), taking 75K examples from each dataset to control for size. We also create MULTI-75K, which contains 15K examples from the five large dataset (Using TQA-G only for TRIVIAQA), resulting in another dataset of 75K examples. We evaluate performance on all datasets that the model was not trained on.

Table 2 shows exact match (EM) performance (does the predicted span exactly match the gold span) on the development set. The row SELF corresponds to training and testing on the target itself, and is provided for reference (For DROP, we train on questions where the answer is a span in the context, but evaluate on the entire development set). The top part shows DOCQA, while the bottom BERTQA.

At a high-level we observe three trends. First, models generalize poorly in this zero-shot setup: comparing SELF to the best zero-shot number shows a performance reduction of 31.5% on average. This confirms the finding that models over-fit to the particular dataset. Second, BERTQA substantially improves generalization compared to DOCQA owing to the power of large-scale unsupervised learning – performance improves by 21.2% on average. Last, MULTI-75K performs almost as well as the best source dataset, reducing performance by only 3.7% on average. Hence, training on multiple datasets results in robust generalization. We further investigate training on multiple datasets in §4.2 and §5.

Taking a closer look, the pair SEARCHQA and TQA-G exhibits the smallest performance drop, since both use trivia questions and web snippets. SQUAD and NEWSQA also generalize well (especially with BERTQA), probably because they contain questions on a single document, focusing on predicate-argument structure. While HOTPOTQA and WIKIHOP both examine multi-hop reasoning over Wikipedia, performance dramatically drops from HOTPOTQA to WIKIHOP. This is due to the difference in the language of the questions (WIKIHOP questions are synthetic). The best generalization to DROP is from HOTPOTQA, since both require multi-hop reasoning. Performance on DROP is overall low, showing that our models struggle with quantitative reasoning.

For the small datasets, COMQA, CQ, and CWQ, generalization is best with TQA-G, as the contexts in these datasets are web snippets. For CQ, whose training set has 1,300 examples, zero-shot performance is even higher than SELF.

Interestingly, BERTQA improves performance substantially compared to DOCQA on NEWSQA, SQUAD, TQA-W and WIKIHOP, but only moderately on HOTPOTQA, SEARCHQA, and TQA-G. This hints that BERT is efficient when the context is similar to (or even *part of*) its training corpus, but degrades over web snippets. This is most evident when comparing TQA-G to TQA-W, as the difference between them is the type of context.

Global structure To view the global structure of the datasets, we visualize them with the force-directed placement algorithm (Fruchterman and

	CQ	CWQ	ComQA	WikiHop	DROP	SQuAD	NewsQA	SearchQA	TQA-G	TQA-W	HotpotQA
SQuAD	18.0	10.1	16.1	4.2	2.4	-	**23.4**	9.5	32.0	20.9	**7.6**
NewsQA	14.9	8.2	13.5	4.8	3.0	**41.9**	-	7.7	25.3	19.9	5.3
SearchQA	29.2	16.1	24.6	8.1	2.3	17.4	10.8	-	**50.3**	**28.9**	4.5
TQA-G	**30.3**	**17.8**	**29.4**	**9.2**	3.0	30.2	15.5	**38.5**	-	-	7.2
TQA-W	24.6	14.5	17.9	8.4	2.9	24.8	15.0	20.5	-	-	6.5
HotpotQA	24.6	14.9	21.2	8.5	**7.7**	38.3	16.9	13.5	36.8	26.0	-
Multi-75K	32.8	17.9	26.7	7.4	4.3	-	-	-	-	-	-
Self	24.1	24.9	45.2	41.7	15.6	68.0	36.5	51.3	58.9	41.6	22.5
SQuAD	23.6	12.0	20.0	4.6	5.5	-	**31.8**	8.4	37.8	33.4	**11.8**
NewsQA	24.1	12.4	18.9	7.1	4.4	**60.4**	-	10.1	37.6	28.4	8.0
SearchQA	30.3	18.5	25.8	12.4	2.8	23.3	12.7	-	**53.2**	**35.4**	5.2
TQA-G	**35.4**	**19.7**	**28.6**	6.3	3.6	36.3	18.8	**39.2**	-	-	8.8
TQA-W	30.3	16.5	23.6	**12.6**	5.1	35.5	19.4	27.8	-	-	8.7
HotpotQA	27.7	15.5	22.1	10.2	**9.1**	54.5	25.6	19.6	37.3	34.9	-
Multi-75K	34.0	18.2	30.9	11.7	8.6	-	-	-	-	-	-
Self	30.8	27.1	51.6	52.9	17.9	78.0	46.0	52.2	60.7	50.1	24.2

Table 2: Exact match on the development set for all datasets in a zero-shot training setup (no training on the target dataset). The top of the table shows results for DocQA, while the bottom for BERTQA. Rows correspond to the training dataset and columns to the evaluated dataset. Large datasets are on the right side, and small datasets on the left side, see text for details of all rows. Datasets used for training were not evaluated. In Multi-75K these comprise all large datasets, and thus these cases are marked by "-"

Reingold, 1991). The input is a set of nodes (datasets), and a set of undirected edges representing springs in a mechanical system pulling nodes towards one another. Edges specify the pulling force, and a physical simulation places the nodes in a final minimal energy state in 2D-space.

Let P_{ij} be the performance when training BERTQA on dataset D_i and evaluating on D_j. Let P_i be the performance when training and evaluating on D_i. The force between an unordered pair of datasets is $F(D_1, D_2) = \frac{P_{12}}{P_2} + \frac{P_{21}}{P_1}$ when we train and evaluate in both directions, and $F(D_1, D_2) = \frac{2 \cdot P_{12}}{P_2}$, if we train on D_1 and evaluate on D_2 only.

Figure 1 shows this visualization, where we observe that datasets cluster naturally according to shape and color. Focusing on the context, datasets with web snippets are clustered (triangles), while datasets that use Wikipedia are also near one another (circles). Considering the question language, TQA-G, SearchQA, and TQA-U are very close (blue triangles), as all contain trivia questions over web snippets. DROP, HotpotQA, NewsQA and SQuAD generate questions with crowd workers, and all are at the top of the figure. WikiHop uses synthetic questions that prevent generalization, and is far from other datasets – however this gap will be closed during transfer learning (§4.2). DROP is far from all datasets because it requires quantitative reasoning that is missing from other datasets. However, it is relatively close to HotpotQA and WikiHop, which target multi-hop reasoning. DROP is also close to SQuAD, as both have similar contexts and question language, but the linguistic phenomena they target differ.

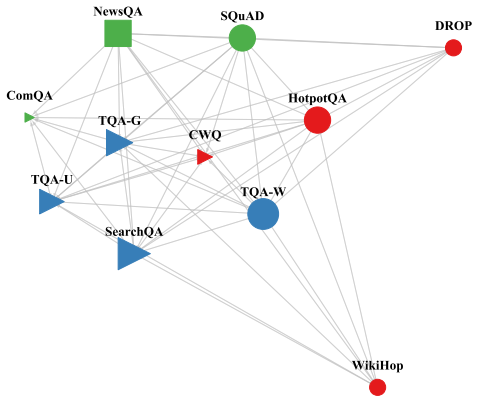

Figure 1: A 2D-visualization of the similarity between different datasets using the force-directed placement algorithm. We mark datasets that use web snippets as context with triangles, Wikipedia with circles, and Newswire with squares. We color multi-hop reasoning datasets in red, trivia datasets in blue, and factoid RC datasets in green.

Does generalization improve with more data? So far we trained on datasets with 75K examples. To examine generalization as the training set size increases, we evaluate performance as the number of examples from the five large datasets grows. Table 3 shows that generalization improves by 26% on average when increasing the number of examples from 37K to 375K.

	CQ	CWQ	ComQA	WikiHop	DROP
Multi-37K	30.9	17.7	28.4	12.3	6.3
Multi-75K	34.0	18.2	30.9	11.7	8.6
Multi-150K	35.0	17.6	30.0	12.4	9.1
Multi-250K	35.6	20.2	31.1	11.9	11.0
Multi-300K	37.6	18.8	31.5	13.5	10.4
Multi-375K	36.1	20.7	31.3	13.3	11.3

Table 3: Exact match on the development set of all small datasets, as we increase the number of examples taken from the five large datasets (zero-shot setup).

4.2 Does pre-training improve results on small datasets?

We now consider transfer learning, assuming access to a small number of examples ($\leq$15K) from a target dataset. We pre-train a model on a source dataset, and then fine-tune on the target. In all models, pre-training and fine-tuning are identical and performed until no improvement is seen on the development set (early stopping). Our goal is to analyze whether pre-training improves performance compared to training on the target alone. This is particularly interesting with BERTQA, as BERT already contains substantial knowledge that might deem pre-training unnecessary.

How to choose the dataset to pre-train on? Table 4 shows exact match (EM) on the development set of all datasets (rows are the trained datasets and columns the evaluated datasets). Pre-training on a source RC dataset and transferring to the target improves performance by 21% on average for DocQA (improving on 8 out of 11 datasets), and by 7% on average for BERTQA (improving on 10 out of 11 datasets). Thus, pre-training on a related RC dataset helps even given representations from a model like BERTQA.

Second, Multi-75K obtains good performance in almost all setups. Performance of Multi-75K is 3% lower than the best source RC dataset on average for DocQA, and 0.3% lower for BERTQA. Hence, one can pre-train a single model on a mixed dataset, rather than choose the best source dataset for every target.

Third, in 4 datasets (ComQA, DROP, HotpotQA, WikiHop) the best source dataset uses web snippets in DocQA, but Wikipedia in BERTQA. This strengthens our finding that BERTQA performs better given Wikipedia text.

Last, we see dramatic improvement in performance comparing to §4.1. This highlights that current models over-fit to the data they are trained on, and small amounts of data from the target distribution can overcome this generalization gap. This is clearest for WikiHop, where synthetic questions preclude generalization, but fine-tuning improves performance from 12.6 EM to 50.5 EM. Thus, low performance was not due to a modeling issue, but rather a mismatch in the question language.

An interesting question is whether performance in the generalization setup is predictive of performance in the transfer setup. Average performance across target datasets in Table 4, when choosing the best source dataset from Table 4, is 39.3 (DocQA) and 43.8 (BERTQA). Average performance across datasets in Table 4, when choosing the best source dataset from Table 2, is 38.9 (DocQA) and 43.5 (BERTQA). Thus, one can select a dataset to pre-train on based on generalization performance and suffer a minimal hit in accuracy, without fine-tuning on each dataset. However, training on Multi-75K also yields good results without selecting a source dataset at all.

How much target data is needed? We saw that with 15K training examples from the target dataset, pre-training improves performance. We now ask whether this effect maintains given a larger training set. To examine this, we measure (Figure 2) the performance on each of the large datasets when pre-training on its nearest dataset (according to $F(\cdot,\cdot)$) for both DocQA (top) and BERTQA (bottom row). The orange curve corresponds to training on the target dataset only, while the blue curve describes pre-training on 75K examples from a source dataset, and then fine-tuning on an increasing number of examples from the target dataset.

In 5 out of 10 curves, pre-training improves performance even given access to all 75K examples from the target dataset. In the other 5, using only the target dataset is better after 30-50K examples. To estimate the savings in annotation costs through pre-training, we measure how many examples are needed, when doing pre-training, to reach 95% of the performance obtained when training on all examples from the target dataset. We find that with pre-training we only need 49% of the examples to reach 95% performance, compared to 86% without pre-training.

To further explore pre-training on multiple datasets, we plot a curve (green) for BERTQA, where at each point we train on a fixed number of examples from all five large datasets (no fine-

	CQ	CWQ	ComQA	WikiHop	DROP	SQuAD	NewsQA	SearchQA	TQA-G	TQA-W	HotpotQA
SQuAD	29.7	25.3	37.1	39.2	14.5	-	33.3	39.2	49.2	34.5	17.8
NewsQA	16.9	26.1	34.7	38.1	14.3	59.6	-	41.6	44.2	33.9	16.5
SearchQA	30.8	28.8	41.3	39.0	**15.0**	57.0	31.4	-	**57.5**	39.6	**19.2**
TQA-G	**41.5**	**30.1**	**42.6**	**42.0**	14.0	57.7	**31.8**	**49.5**	-	**41.4**	19.1
TQA-W	31.3	27.0	38.0	41.4	13.3	57.6	31.7	44.4	50.7	-	17.2
HotpotQA	40.0	27.7	39.5	40.4	14.6	**59.8**	32.4	46.3	54.6	37.4	-
Multi-75K	43.1	27.6	39.1	38.9	14.5	59.8	33.0	47.5	56.4	40.4	19.2
Self	24.1	24.9	45.2	41.7	15.6	56.5	30.0	35.9	41.2	27.7	13.8
SQuAD	36.9	29.0	52.2	48.2	**18.6**	-	**41.2**	47.8	55.2	45.4	20.8
NewsQA	36.9	29.4	52.2	48.4	17.8	**72.1**	-	47.4	55.9	45.2	20.6
SearchQA	**40.5**	30.0	53.4	**50.6**	17.6	70.2	40.2	-	**57.3**	45.5	20.4
TQA-G	40.0	**30.6**	53.4	49.5	17.6	69.9	**41.2**	**50.0**	-	**46.2**	20.8
TQA-W	39.0	30.3	**54.0**	50.0	17.3	71.0	39.2	48.4	55.7	-	**20.9**
HotpotQA	34.4	30.2	53.0	49.3	17.2	71.2	39.5	48.6	56.6	45.6	-
Multi-75K	42.6	30.6	53.3	50.5	17.9	71.5	42.1	48.5	56.6	46.5	20.4
Self	30.8	27.1	51.6	52.9	17.1	70.1	37.9	46.0	54.4	41.9	18.9

Table 4: Exact match on the development set for all datasets with transfer learning. Fine-tuning is done on $\leq 15K$ examples. The top of the table shows results for DocQA, while the bottom for BertQA. Rows are the trained datasets and columns are the evaluated datasets for which fine-tuning was performed. Large datasets are on the right, and small datasets are on the left side

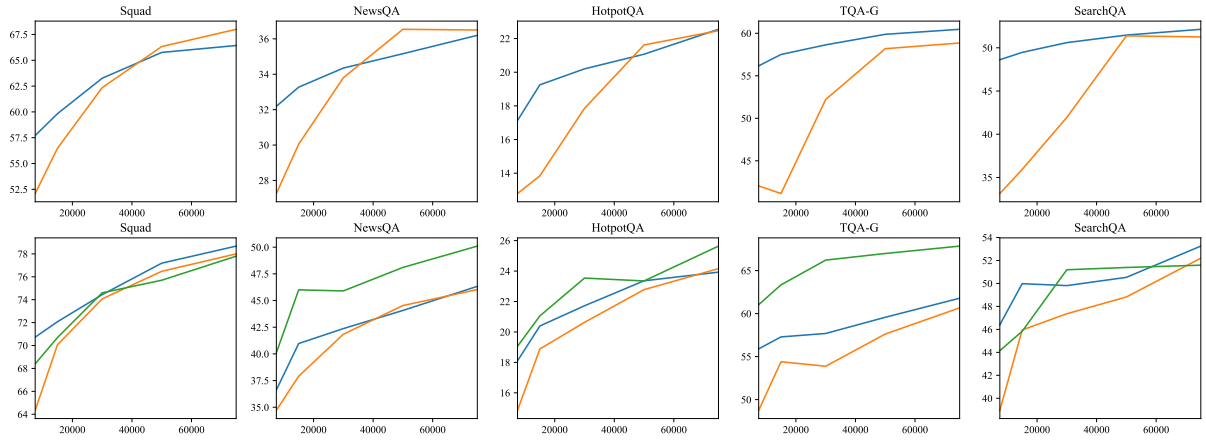

Figure 2: Learning curves for the five large datasets (top is DocQA and bottom is BertQA). The x-axis corresponds to the number of examples from the target dataset, and the y-axis is EM. The orange curve refers to training on the target dataset only, and the blue curve refers to pre-training on 75K examples from the nearest source dataset and fine-tuning on the target dataset. The green curve is training on a fixed number of examples from all 5 large datasets without fine-tuning (MultiQA).

tuning). We observe that more data from multiple datasets improves performance in almost all cases. In this case, we reach 95% of the final performance using 30% of the examples only. We will use this observation further in §5 to reach new state-of-the-art performance on several datasets.

4.3 Does context augmentation improve performance?

For TriviaQA we have for all questions, contexts from three different sources – Wikipedia (TQA-W), Bing web snippets (TQA-U), and Google web snippets (TQA-G). Thus, we can explore whether combining the three datasets improves performance. Moreover, because questions are identical across the datasets, we can see the effect on generalization due to the context language only.

Table 5 shows the results. In the first 3 rows we train on 75K examples from each dataset, and in the last we train on the combined 225K examples. First, we observe that context augmentation substantially improves performance (especially for TQA-G and TQA-W). Second, generalization is sensitive to the context type: performance substantially drops when training on one context type and evaluating on another (60.7 $\rightarrow$ 48.4 for TQA-G, 53.1 $\rightarrow$ 44.6 for TQA-U, and 50.1 $\rightarrow$ 43.3 for TQA-W).

	TQA-G	TQA-U	TQA-W
TQA-G	60.7	53.6	43.3
TQA-U	57.2	53.1	39.9
TQA-W	48.4	44.6	50.1
ALLCONTEXTS	67.7	54.4	54.7

Table 5: EM on the development set, where each row uses the same question with a different context, and ALLCONTEXTS is a union of the other 3 datasets.

5 MULTIQA

We now present MULTIQA, a BERT-based model, trained on multiple RC datasets, that obtains new state-of-the-art results on several datasets.

Does training on multiple datasets improve BERTQA? MULTIQA trains BERTQA on the MULTI-375K dataset presented above, which contains 75K examples from 5 large datasets, but uses BERT-large rather than BERT-base. For small target datasets, we fine-tune the model on these datasets, since they were not observed when training on MULTI-375K. For large datasets, we do not fine-tune. We found that fine-tuning on datasets that are already part of MULTI-375K does not improve performance (we assume this is due to the high-capacity of BERT-large), and thus we use *one model* for all the large datasets. We train on MULTI-375K, and thus our model does not use all examples in the original datasets, which contain more than 75K examples.

We use the official evaluation script for any dataset that provides one, and the SQUAD evaluation script for all other datasets. Table 6 shows results for datasets where the evaluation metric is EM or token F_1 (harmonic mean of the list of tokens in the predicted vs. gold span). Table 7 shows results for datasets where the evaluation metric is average recall/precision/F_1 between the list of predicted answers and the list of gold answers.

We compare MULTIQA to BERT-large, a model that does not train on MULTI-375K, but only fine-tunes BERT-large on the target dataset. We also show the state-of-the-art (SOTA) result for all datasets for reference.[1]

MULTIQA improves state-of-the-art performance on fivedatasets, although it does not even train on all examples in the large datasets.[2] MULTIQA improves performance compared to BERT-large in all cases. This improvement is especially noticeable in small datasets such as COMQA, CWQ, and CQ. Moreover, in NEWSQA, MULTIQA surpasses human performance as measured by the creators of those datasets. (46.5 EM, 69.4 F1) (Trischler et al., 2017)), improving upon previous state-of-the-art by a large margin.

To conclude, MULTIQA is able to improve state-of-the-art performance on multiple datasets. Our results suggest that in many NLU tasks the size of the dataset is the main bottleneck rather than the model itself.

Does training on multiple datasets improve resiliency against adversarial attacks? Finally, we evaluated MULTIQA on the adversarial SQUAD (Jia and Liang, 2017), where a misleading sentence is appended to each context (ADDSENT variant). MULTIQA obtained 66.7 EM and 73.1 F_1, outperforming BERT-large (60.4EM, 66.3F1) by a significant margin, and also substantially improving state-of-the-art results (56.0 EM, 61.3 F_1, (Hu et al., 2018) and 52.1 EM, 62.7 F_1, (Wang et al., 2018)).

6 Related Work

Prior work has shown that RC performance can be improved by training on a large dataset and transferring to a smaller one, but at a small scale (Min et al., 2017; Chung et al., 2018). Sun et al. (2018) has recently shown this in a larger experiment for multi-choice questions, where they first fine-tuned BERT on RACE (Lai et al., 2017) and then fine-tuned on several smaller datasets.

Interest in learning general-purpose representations for natural language through unsupervised, multi-task and transfer learning has been skyrocketing lately (Peters et al., 2018; Radford et al., 2018; McCann et al., 2018; Chronopoulou et al., 2019; Phang et al., 2018; Wang et al., 2019). In parallel to our work, studies that focus on generalization have appeared on publication servers, empirically studying generalization to multiple tasks (Yogatama et al., 2019; Liu et al., 2019). Our work is part of this research thread on generalization in

[1] State-of-the-are-results were found in (Tay et al., 2018) for NEWSQA, in Lin et al. (2018), for SEARCHQA, in Das et al. (2019) for TQA-U, in (Talmor and Berant, 2018b) for CWQ, in Ding et al. (2019) for HOTPOTQA, in (Abujabal et al., 2018) for COMQA, and in Bao et al. (2016) for CQ.

[2] We compare only to models for which we found a publication. For TQA-U, Figure 4 in Clark and Gardner (2018) shows roughly 67 F_1 on the development set, but no exact number. For CQ we compare against SOTA achieved on the web snippets context. On the Freebase context SOTA is 42.8 F_1. (Luo1 et al., 2018)

	BERT-large Dev.		MULTIQA Dev.		MULTIQA Test		SOTA[1]	
Dataset	EM	tok. F1	EM	tok. F1	EM	tok. F1	EM	tok. F1
NEWSQA	51.5	66.2	53.9	68.2	**52.3**	**67.4**	53.1	66.3
SEARCHQA	59.2	66.4	60.7	67.1	**59.0**	**65.1**	58.8	64.5
TQA-U	56.8	62.6	58.4	64.3	-	-	52.0[2]	61.7[2]
CWQ	30.8	-	35.4	-	**34.9**	-	34.2	-
HOTPOTQA	27.9	37.7	30.6	40.3	**30.7**	**40.2**	37.1[2]	48.9[2]

Table 6: Results for datasets where the official evaluation metric is EM and token F_1. The CWQ evaluation script provides only the EM mertic. We did not find a public evaluation script for the hidden test set of TQA-U.

	BERT-large Dev.			MULTIQA Dev.			MULTIQA Test			SOTA		
Dataset	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
COMQA	45.8	42.0	42.9	51.9	47.2	48.2	**44.4**	**40.0**	**40.8**	21.2	38.4	22.4
CQ	-	-	32.8	-	-	46.6	-	-	**42.4**	-	-	39.7[2]

Table 7: Results for datasets where the evaluation metric is average recall/precision/F_1. CQ evaluates with F_1 only.

natural langauge understanding, focusing on reading comprehension, which we view as an important and broad language understanding task.

7 Conclusions

In this work we performed a thorough empirical investigation of generalization and transfer over 10 RC datasets. We characterized the factors affecting generalization and obtained several state-of-the-art results by training on 375K examples from 5 RC datasets. We open source our infrastructure for easily performing experiments on multiple RC datasets, for the benefit of the community.

We highlight several practical take-aways:

- Pre-training on multiple source RC datasets consistently improves performance on a target RC dataset , even in the presence of BERT representations. It also leads to substantial reduction in the number of necessary training examples for a fixed performance.
- Training the high-capacity BERT-large representations over multiple RC datasets leads to good performance on all of the trained datasets without having to fine-tune on each dataset separately.
- BERT representations improve generalization, but their effect is moderate when the source of the context is web snippets compared to Wikipedia and newswire.
- Performance over an RC dataset can be improved by retrieving web snippets for all questions and adding them as examples (context augmentation).

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was completed in partial fulfillment for the PhD degree of Alon Talmor. This research was partially supported by The Israel Science Foundation grant 942/16, The Blavatnik Computer Science Research Fund and The Yandex Initiative for Machine Learning.

References

A. Abujabal, R. S. Roy, M. Yahya, and G. Weikum. 2018. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. *arXiv preprint arXiv:1809.09528*.

J. Bao, N. Duan, Z. Yan, M. Zhou, and T. Zhao. 2016. Constraint-based question answering with knowledge graph. In *International Conference on Computational Linguistics (COLING)*.

J. Berant, V. Srikumar, P. Chen, A. V. Linden, B. Harding, B. Huang, P. Clark, and C. D. Manning. 2014. Modeling biological processes for reading comprehension. In *Empirical Methods in Natural Language Processing (EMNLP)*.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *International Conference on Management of Data (SIGMOD)*, pages 1247–1250.

E. Choi, H. He, M. Iyyer, M. Yatskar, W. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. 2018. Quac: Question answering in context. In *Empirical Methods in Natural Language Processing (EMNLP)*.

A. Chronopoulou, C. Baziotis, and A. Potamianos. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. *arXiv preprint arXiv:1902.10547*.

Y. Chung, H. Lee, and J. Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *North American Association for Computational Linguistics (NAACL)*.

C. Clark and M. Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Association for Computational Linguistics (ACL)*.

R. Das, S. Dhuliawala, M. Zaheer, and A. McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations (ICLR)*.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Association for Computational Linguistics (NAACL)*.

M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Association for Computational Linguistics (ACL)*.

D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *North American Association for Computational Linguistics (NAACL)*.

M. Dunn, , L. Sagun, M. Higgins, U. Guney, V. Cirik, and K. Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv*.

T. M. Fruchterman and E. M. Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.

M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

K. M. Hermann, T. Koisk, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NeurIPS)*.

M. Hu, Y. Peng, F. Wei, Z. Huang, D. Li, N. Yang, and M. Zhou. 2018. Attention-guided answer distillation for machine reading comprehension. In *Empirical Methods in Natural Language Processing (EMNLP)*.

R. Jia and P. Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*.

G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Y. Lin, H. Ji, Z. Liu, and M. Sun. 2018. Denoising distantly supervised open-domain question answering. In *Association for Computational Linguistics (ACL)*, volume 1, pages 1736–1745.

X. Liu, P. He, W. Chen, and J. Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

K. Luo1, F. Lin1, X., L. Kenny, and Q.Zhu1. 2018. Knowledge base question answering via encoding of complex query graphs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

B. McCann, N. S. Keskar, C. Xiong, and R. Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

S. Min, M. Seo, and H. Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Association for Computational Linguistics (ACL)*.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computing at NIPS*.

J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *North American Association for Computational Linguistics (NAACL)*.

J. Phang, T. Fevry, and S. R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

S. Reddy, D. Chen, and C. D. Manning. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

M. Richardson, C. J. Burges, and E. Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 193–203.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv*.

K. Sun, D. Yu, D. Yu, and C. Cardie. 2018. Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*.

A. Talmor and J. Berant. 2018a. Repartitioning of the complexwebquestions dataset. *arXiv preprint arXiv:1807.09623*.

A. Talmor and J. Berant. 2018b. Repartitioning of the complexwebquestions dataset. *arXiv preprint arXiv:1807.09623*.

A. Talmor and J. Berant. 2018c. The web as knowledge-base for answering complex questions. In *North American Association for Computational Linguistics (NAACL)*.

A. Talmor, M. Geva, and J. Berant. 2017. Evaluating semantic parsing against a simple web-based question answering model. In **SEM*.

Y. Tay, L. Tuan, S. Hui, and J. Su. 2018. Densely connected attention propagation for reading comprehension. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. 2017. NewsQA: A machine comprehension dataset. In *Workshop on Representation Learning for NLP*.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*.

W. Wang, M. Yan, and C. Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Association for Computational Linguistics (ACL)*.

J. Welbl, P. Stenetorp, and S. Riedel. 2017. Constructing datasets for multi-hop reading comprehension across documents. *arXiv preprint arXiv:1710.06481*.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*.

D. Yogatama, C. de M. d'Autume, J. Connor, T. Kocisky, M. Chrzanowski, L. Kong, A. Lazaridou, W. Ling, L. Yu, C. Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.