An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering

Shayne Longpre[*] **Yi Lu**[*] **Zhucheng Tu**[*] **Chris DuBois**

Apple Inc.

{slongpre, ylu7, zhucheng_tu, cdubois}@apple.com

Abstract

To produce a domain-agnostic question answering model for the Machine Reading Question Answering (MRQA) 2019 Shared Task, we investigate the relative benefits of large pre-trained language models, various data sampling strategies, as well as query and context paraphrases generated by back-translation. We find a simple negative sampling technique to be particularly effective, even though it is typically used for datasets that include unanswerable questions, such as SQuAD 2.0. When applied in conjunction with per-domain sampling, our XLNet (Yang et al., 2019)-based submission achieved the second best Exact Match and F1 in the MRQA leaderboard competition.

1 Introduction

Recent work has demonstrated that generalization remains a salient challenge in extractive question answering (Talmor and Berant, 2019; Yogatama et al., 2019). It is especially difficult to generalize to a target domain without similar training data, or worse, any knowledge of the domain's distribution. This is the case for the MRQA Shared Task.[1] Together, these two factors demand a representation that generalizes broadly, and rules out the usual assumption that more data in the training domain will necessarily improve performance on the target domain. Consequently, we adopt the overall approach of curating our input data and learning regime to encourage representations that are not biased by any one domain or distribution.

As a requisite first step to a representation that generalizes, transfer learning (in the form of large pre-trained language models such as Peters et al. (2018); Howard and Ruder (2018); Devlin et al. (2019); Yang et al. (2019)), offers a solid foundation. We compare BERT and XLNet, leveraging Transformer based models (Vaswani et al., 2017) pre-trained on significant quantities of unlabelled text. Secondly, we identify how the domains of our training data correlate with the performance of "out-domain" development sets. This serves as a proxy for the impact these different sets may have on a held-out test set, as well as evidence of a representation that generalizes. Next we explore data sampling and augmentation strategies to better leverage our available supervised data.

To our surprise, the more sophisticated techniques including back-translated augmentations (even sampled with active learning strategies) yield no noticeable improvement. In contrast, much simpler techniques offer significant improvements. In particular, negative samples designed to teach the model when to abstain from predictions prove highly effective out-domain. We hope our analysis and results, both positive and negative, inform the challenge of generalization in multi-domain question answering.

We begin with an overview of the data and techniques used in our system, before discussing experiments and results.

2 Data

We provide select details of the MRQA data as they pertain to our sampling strategies delineated later. For greater detail refer to the MRQA task description.

Our training data consists of six separately collected QA datasets. We refer to these and their associated development sets as "in-domain" (ID). We are also provided with six "out-domain" (OD) development sets sourced from other QA datasets. In Table 1 we tabulate the number of "examples" (question-context pairs), "segments" (the question combined with a portion of the context), and "no-answer" (NA) segments (those without a valid answer span).

To clarify these definitions, consider examples

---

[*] equal contribution

[1] https://mrqa.github.io/shared

Dataset	Examples	Segments	NA (%)
SQuAD (Rajpurkar et al., 2016)	87K	87K	0.1
SearchQA (Dunn et al., 2017)	117K	657K	56.3
NaturalQuestions (Kwiatkowski et al., 2019)	104K	189K	36.3
TriviaQA (Joshi et al., 2017)	62K	337K	57.3
HotpotQA (Yang et al., 2018)	73K	73K	0.3
NewsQA (Trischler et al., 2017)	74K	214K	49.0
Total	**517K**	**1557K**	**47.3**

Table 1: Number of examples (question-context pairs), segments (question-context chunks), and the percentage of No Answer (NA) segments within each dataset.

with long context sequences. We found it necessary to break these examples' contexts into multiple segments in order to satisfy computational memory constraints. Each of these segments may or may not contain the gold answer span. A segment without an answer span we term "no-answer". To illustrate this pre-processing, consider question, context pair (q, c) where we impose a maximum sequence length of M tokens. If $len(c) > M$ then we create multiple overlapping input segments (q, c_1), (q, c_2), ..., (q, c_k) where each c_i contains only a portion of the larger context c. The sliding window that generates these chunks is parameterized by the document stride D, and the maximum sequence length M, shown below in Equation 1.

$$(q, c) \rightarrow \left\{ (q, c_{i \cdot D : M + i \cdot D}), \forall i \in [0, k] \right\} \quad (1)$$

The frequencies presented in Table 1 are based on our settings of $M = 512$ and $D = 128$.

3   System Overview

3.1   XLNet

While we used BERT Base (Devlin et al., 2019) for most of our experimentation, we used XLNet Large (Yang et al., 2019) for our final submission. At the time of submission this model held state-of-the-art results on several NLP benchmarks including GLUE (Wang et al., 2018). Leveraging the Transformer-XL architecture (Dai et al., 2019), a "generalized autoregressive pretraining" method, and much more training data than BERT, its representation provided a strong source of transfer learning. In keeping with XLNet's question answering module, we also computed the end logits based on the ground truth of the start position during training time, and used beam search over the end logits at inference time. We based our code

on the HuggingFace implementation.[2] of BERT and XLNet, and used the pre-trained models in the GitHub repository.

3.2   Domain Sampling

For the problem of generalizing to an unseen and out-domain test set, it's important not to overfit to the training distribution. Given the selection of diverse training sources, domains, and distributions within MRQA we posed the following questions. Are all training sources useful to the target domains? Will multi-domain training partially mitigate overfitting to any given training set? Is it always appropriate to sample equally from each?

To answer these questions, we fine-tuned a variety of specialized models on the BERT Base Cased (BBC) pre-trained model. Six models were each fine-tuned once on their respective in-domain training set. A multi-domain model was trained on the union of these six in-domain training sets. Lastly, we used this multi-domain model as the starting point for fine-tuning six more models, one for each in-domain training set. In total we produced six dataset-specialized models each fine-tuned once, one multi-domain model, and six dataset-specialized models each fine-tuned twice.

There are a few evident trends. The set of models which were first fine-tuned on the multi-domain dataset achieved higher Exact Match (EM) almost universally than those which weren't. This improvement extends not just to in-domain datasets, but also to out-domain development sets. In Figure 1 we observe these models on the Y-axis, and their Exact Match (EM) scores on each in-domain and out-domain development set. This confirms the observations from Talmor and Berant (2019) that multi-domain training improves robustness and generalization broadly, and suggests that a variety of question answering domains is significant across domains. Interestingly, the second round of fine-tuning, this time on a specific domain, did not cause models to significantly, or catastrophically forget what they learned in the initial, multi-domain fine-tuning. This is clear

---

[2] https://github.com/huggingface/pytorch-transformers Our implementation modifies elements of the tokenization, modeling, and training procedure. Specifically, we remove whitespace tokenization and other pre-processing features that are not necessary for MRQA-tokenized data. We also add subepoch checkpoints and validation, per dataset sampling, and improved post-processing to select predicted text without special tokens or unusual spacing.
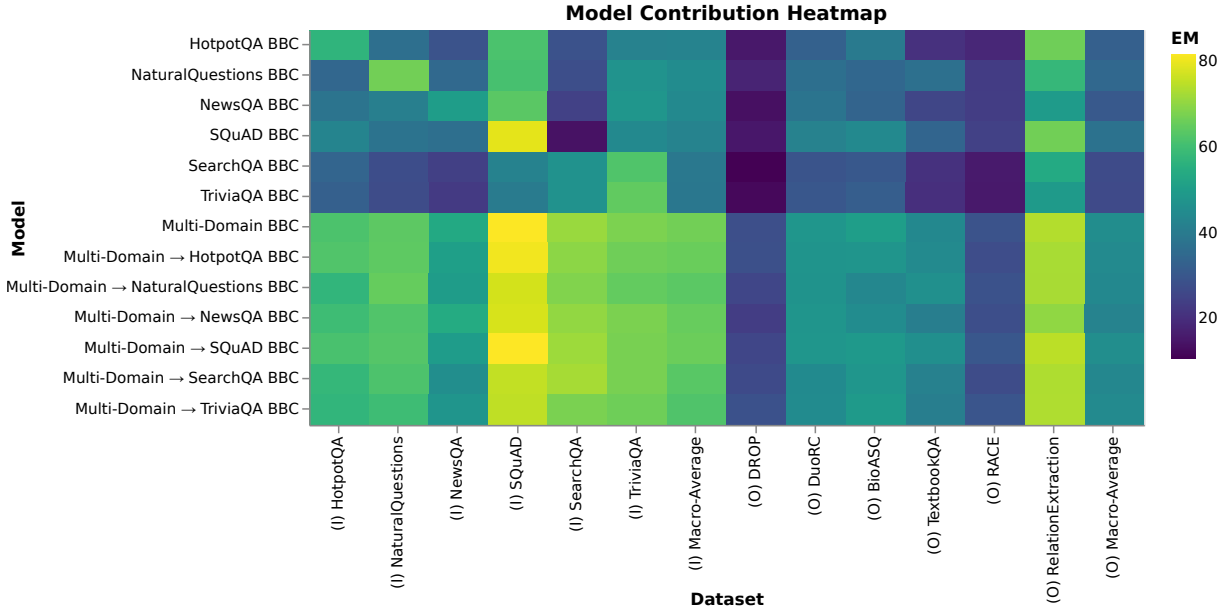
Figure 1: Heatmap of Exact Match (EM) for BERT Base Cased (BBC) models, the top six fine-tuned directly on each training dataset, and the bottom six fine-tuned on multi-domain before being fine-tuned on each training dataset.

from comparing the generic "Multi-Domain BBC" to those models fine-tuned on top of it, such as "Multi-Domain → SQuAD FT BBC".

Secondly, we observe that the models we fine-tune on SearchQA (Dunn et al., 2017) and TriviaQA (Joshi et al., 2017) achieve relatively poor results across all sets (in-domain and out-domain) aside from themselves. The latter datasets are both Jeopardy-sourced, distantly supervised, long context datasets. In contrast, the SQuAD (Rajpurkar et al., 2016) fine-tuned model achieves the best results on both in and out-domain "Macro-Average" Exact Match. Of the models with multi-domain pre-fine-tuning NewsQA, SearchQA, and TriviaQA performed the worst on the out-domain (O) Macro-Average. As such we modified our sampling distribution to avoid oversampling them and risk degrading generalization performance. This risk is particularly prevalent for SearchQA, the largest dataset by number of examples. Additionally, its long contexts generate 657K segments, double that of the next largest dataset (Table 1). This was exacerbated further when we initially included the nearly 10 occurrences of each detected answer. TriviaQA shares this characteristic, though not quite as drastically. Accordingly, for our later experiments we chose not to use all instances of a detected answer, as this would further skew our multi-domain samples towards SearchQA and TriviaQA, and increase the num-

NA	Model	MSL	In-Domain		Out-Domain	
			EM	F1	EM	F1
No	BBC	200	65.70	75.98	45.80	56.78
	BBC	512	65.29	76.01	45.59	57.40
	XBC	200	43.78	65.24	43.78	52.12
	XBC	512	65.91	74.93	49.59	59.61
Yes	BBC	200	66.11	76.41	46.19	57.51
	BBC	512	66.20	76.77	46.28	58.00
	XBC	200	68.67	77.69	50.04	59.68
	XBC	512	**70.04**	**79.15**	**50.71**	**61.16**

Table 2: Model performance including or excluding No-Answer (NA) segments in training. We examine how these results vary with the max sequence length (MSL). BBC refers to BERT Base Cased and XBC refers to XLNet Base Cased.

ber of times contexts from these sets are repeated as segments. We also chose, for many experiments, to sample fewer examples of SearchQA than our other datasets, and found this to improve F1 marginally across configurations.

3.3 Negative Sampling

While recent datasets such as SQuAD 2.0 (Rajpurkar et al., 2018) and Natural Questions (Kwiatkowski et al., 2019) have extended extractive question answering to include a No Answer option, in the traditional formulation of the problem there is no notion of a negative class. Formulated as such, the MRQA Shared Task guarantees the presence of an answer span within each exam-

ple. However, this is not guaranteed within each segment, producing NA segments.

At inference time we compute the most probable answer span for each segment separately and then select the best span across all segments of that ($\mathbf{q}$, $\mathbf{c}$) example to be the one with the highest probability. This is computed as the sum of the start and end span probabilities. At training time, typically the NA segments are discarded altogether. However, this causes a discrepancy between train and inference time, as "Negative" segments are only observed in the latter.

To address this, we include naturally occurring "Negative" segments, and add an abstention option for the model. For each Negative segment, we set the indices for both the start and end span labels to point to the [CLS] token. This gives our model the option to abstain from selecting a span in a given segment. Lastly, at inference time we select the highest probability answer across all segments, excluding the No Answer [CLS] option.

Given that 47.3% of all input segments are NA, as shown in Table 1, its unsurprising their inclusion significantly impacted training time and results. We find that this simple form of Negative Sampling yields non-trivial improvements on MRQA (see Table 2). We hypothesize this is primarily because a vaguely relevant span of tokens amid a completely irrelevant NA segment would monopolize the predicted probabilities. Meanwhile the actual answer span likely appears in a segment that may contain many competing spans of relevant text, each attracting some probability mass. As we would expect, the improvement this technique offers is magnified where the context is much longer than M. To our knowledge this technique is still not prevalent in purely extractive question answering, though Alberti et al. (2019) cite it as a key contributor to their strong baseline on Google's Natural Questions.

3.4 Paraphrasing by Back-Translation

Yu et al. (2018) showed that generating context paraphrases via back-translation provides significant improvements for reading comprehension on the competitive SQuAD 1.1 benchmark. We emulate this approach to add further quantity and variety to our data distribution, with the hope that it would produce similarly strong results for out-domain generalization. To extend their work,

we experiment with both query and context paraphrases generated by back-translation. Leveraging the same open-sourced TensorFlow NMT codebase,[3] we train an 8-layer `seq2seq` model with attention on the WMT16 News English-German task, obtaining a BLEU score of 28.0 for translating from English to German and 25.7 for German to English, when evaluated on the `newstest2015` dataset. We selected German as our back-translation language due to ease of reproducibility, given the public benchmarks published in the `nmt` repository.

For generating query paraphrases, we directly feed each query into the NMT model after performing tokenization and byte pair encoding. For generating context paraphrases, we first use SpaCy to segment each context into sentences,[4] using the `en_core_web_sm` model. Then, we translate each sentence independently, following the same procedure as we do for each query. In the course of generating paraphrases, we find decoded sequences are occasionally empty for a given context or query input. For these cases we keep the original sentence.

We attempt to retrieve the new answer span using string matching, and where that fails we employed the the same heuristic described in Yu et al. (2018) to obtain a new, estimated answer. Specifically, this involves finding the character-level 2-gram overlap of every token in the paraphrase sentence with the start and end token of the original answer. The score is computed as the Jaccard similarity between the sets of character-level 2-grams in the original answer token and new sentence token. The span of text between the two tokens that has the highest combined score, passing a minimum threshold, is selected as the new answer. In cases where there is no score above the threshold, no answer is generated. Any question in each context without an answer is omitted, and any paraphrased example without at least one question-answer pair is discarded.

3.4.1 Augmentation Strategy

For every query and context pair (q, c), we used our back-translation model to generate a query paraphrase q' and a context paraphrase c'. We then create a new pair that includes the paraphrase q' instead of q with probability $P_q(x)$, and independently we choose the paraphrase c' over c with

---

[3] https://github.com/tensorflow/nmt
[4] https://spacy.io/

probability $P_c(x)$. If either q' or c' is sampled, we add this augmented example to the training data. This sampling strategy allowed us flexibility in how often we include query or context augmentations.

3.4.2 Active Learning

Another method of sampling our data augmentations was motivated by principles in active learning (Settles, 2009). Rather than sampling uniformly, might we prioritize the more challenging examples for augmentation? This is motivated by the idea that many augmentations may not be radically different from the original data points, and may consequently carry less useful, repetitive signals.

To quantify the difficulty of an example we used $1 - F1$ score computed for our best model. We chose F1 as it provides a continuous rather than binary value, and is robust to a model that may select the wrong span, but contains the correct answer text. Other metrics, such as loss or Exact Match do not provide both these benefits.

For each example we derived its probability weighting from its F1 score. This weight replaces the uniform probability previously used to draw samples for query and context augmentations. We devised three weighting strategies, to experiment with different distributions. We refer to these as the hard, moderate and soft distributions. Each distribution employs its own scoring function S_x (Equation 2), which is normalized across all examples to determine the probability of drawing that sample (Equation 3).

$$S(x) = \begin{cases} 1 - F1(x) + \epsilon & \text{Hard Score} \\ 2 - F1(x) & \text{Moderate Score} \\ 3 - F1(x) & \text{Soft Score} \end{cases} \quad (2)$$

$$P(x) = \frac{S(x)}{\Sigma_{i=1..n} S(i)} \quad (3)$$

The hard scoring function allocates negligible probability to examples with $F1 = 1$, emphasizing the hardest examples the most of the three distributions. We used an ϵ value of 0.01 to prevent any example from having a zero sample probability. The moderate and soft scoring functions penalize correct predictions less aggressively, smoothing the distribution closer to uniform.

4 Experiments and Discussion

During our experimentation process we used our smallest model BERT Base Cased (BBC) for the most expensive sampling explorations (Figure 1), XLNet Base Cased (XBC) to confirm our findings extended to XLNet (Table 2), and XLNet Large Cased (XLC) as the initial basis for our final submission contenders (Table 3).

Our training procedure for each model involved fine-tuning the Transformer over two epochs, each with three validation checkpoints. The checkpoint with the highest Out-Domain Macro-Average (estimated from a $2,000$ dev-set subsample) was selected as the best for that training run. Our multi-domain dataset originally consisted of 75k examples from every training set, and using every detected answer. We modified this to a maximum of 120k samples from each dataset, 100k from SearchQA, and using only one detected answer per example; given our findings in Section 3.2.

We trained every model on 8 NVIDIA Tesla V100 GPUs. For BBC and XBC we used a learning rate of $5e - 5$, single-GPU batch size of 25, and gradient accumulation of 1, yielding an effective batch size of 200. For XLC we used a learning rate of $2e - 5$, single-GPU batch size of 6, and gradient accumulation of 3, yielding an effective batch size of $6 \cdot 8 \cdot 3 = 144$. We found the gradient accumulation and lower learning rate critical to achieve training stability.

We conduct several experiments to evaluate the various sampling and augmentation strategies discussed in Section 3. In Table 2 we examine the impact of including No Answer segments in our training set. We found this drastically outperformed the typical practice of excluding these segments. This effect was particularly noticeable on datasets with longer sequences. As expected, the improvement is exaggerated at the shorter max sequence length (MSL) of 200, where including NA segments increases Out-Domain EM from 43.78 to 50.04 on the XBC model.

Next, we evaluate our back-translated query and context augmentations using the sampling strategies described in Section 3.4.2. To select the best $P_q(x)$, $P_c(x)$ and sampling strategy we conducted the following search. First we explored sampling probabilities 0.2, 0.4, 0.6, 0.8, 1.0 for query and context separately, using random sampling, and subsequently we combined them using values informed from the previous exploration, this time

			In-Domain F1							Out-Domain F1						
Mode	$P_q(x)$	$P_c(x)$	HotpotQA	Natural Questions	NewsQA	SearchQA	SQuAD	TriviaQA	Macro-Average	BioASQ	DROP	DuoRC	RACE	Relation Extraction	TextbookQA	Macro-Average
–	0	0	82.62	82.15	72.52	82.80	**94.50**	78.28	82.14	**73.00**	63.52	**65.68**	**53.25**	88.49	64.38	**68.07**
R	0.2	0.2	82.42	82.29	72.45	83.20	94.09	**79.44**	82.32	70.45	63.97	62.75	52.66	88.09	63.28	66.87
	0.2	0.4	82.59	**82.51**	72.30	84.50	94.35	79.09	**82.56**	72.02	**64.29**	63.61	52.32	**88.85**	64.12	67.54
	0.4	0.4	82.58	82.28	71.72	83.80	94.02	77.78	82.03	69.60	63.45	63.56	52.74	88.22	63.67	66.87
S	0.2	0.2	82.44	82.10	72.06	83.67	94.32	76.58	81.86	70.47	64.14	63.15	52.61	88.37	63.60	67.06
	0.2	0.4	82.50	81.69	72.43	84.46	93.98	76.80	81.98	70.79	60.62	63.48	52.38	87.38	62.07	66.12
	0.4	0.4	82.07	82.15	72.07	84.20	93.99	77.20	81.95	71.34	62.64	62.81	50.65	87.60	63.12	66.36
M	0.2	0.2	**82.72**	82.26	72.22	83.45	94.12	76.55	81.89	71.46	63.89	63.29	51.67	87.98	**64.85**	67.19
	0.2	0.4	82.41	82.15	**72.60**	**84.88**	93.85	77.34	82.20	71.66	63.89	62.12	52.67	88.03	64.05	67.07
	0.4	0.4	82.55	82.09	72.57	84.30	94.19	76.97	82.11	71.13	63.03	62.58	51.65	87.76	64.67	66.80
H	0.2	0.2	81.68	81.15	70.55	80.51	94.50	74.80	80.46	70.60	62.55	61.96	52.23	87.87	61.16	66.06
	0.2	0.4	82.05	81.45	70.84	81.92	94.18	75.49	80.99	72.89	62.29	63.30	51.66	87.63	62.00	66.63
	0.4	0.4	81.93	81.45	71.67	81.71	93.92	75.96	81.11	71.26	61.52	62.06	51.36	86.91	60.18	65.55

Table 3: F1 scores for data augmentation using different proportions of query and context paraphrasing and different sampling distributions on XLNet Large Cased, on individual datasets. R, S, M, H refer to random, soft, moderate, and hard modes from Section 3.4.2 respectively.

	Out-Domain	
Dataset	EM	F1
BioASQ (Tsatsaronis et al., 2015)	60.28	71.98
DROP (Dua et al., 2019)	48.50	58.90
DuoRC (Saha et al., 2018)	53.29	63.36
RACE (Lai et al., 2017)	39.35	53.87
RelationExtraction (Levy et al., 2017)	79.20	87.85
TextbookQA (Kembhavi et al., 2017)	56.50	65.54
Macro-Average	56.19	66.92

Table 4: Breakdown of hidden development set results by dataset using our best XLNet Large model.

Submission	EM	F1
D-NET (Baidu)	**60.39**	**72.55**
Ours (Apple)	59.47	70.75
FT_XLNet (HIT)	58.37	70.54
HLTC (HKUST)	56.59	68.98
BERT-cased-whole-word (Aristo@AI2)	53.52	66.27
XERO (Fuji Xerox)	52.41	66.11
BERT-large + Adv. Training (Team 42-alpha)	48.91	62.19
BERT large baseline (MRQA Organizers)	48.20	61.76
BERT base baseline (MRQA Organizers)	45.54	58.50

Table 5: Macro-Average EM and F1 on the held-out leaderboard test sets.

searching over sampling strategies: random, soft, moderate and hard. We present the best results in Table 3 and conclude that these data augmentations did not help in-domain or out-domain performance. While we observed small boosts to metrics on BBC using this technique, no such gains were found on XLC. We suspect this is because (a) large pre-trained language models such as XLC already capture the linguistic variations in language introduced by paraphrased examples quite well, and (b) we already have a plethora of diverse training data from the distributions these augmentations are derived from. It is not clear if the boosts QANet Yu et al. (2018) observed on SQuAD 1.1 would still apply with the additional diversity provided by the five additional QA datasets for training. We notice that SearchQA and TriviaQA benefit the most from some form of data augmentation, both by more than one F1 point. Both of these are distantly supervised, and have relatively long contexts.

Our final submission leverages our fine-tuned XLC configuration, with domain and negative sampling. We omit the data augmentation and active sampling techniques which we did not find to aid out-domain performance. The results of the leaderboard Out-Domain Development set and final test set results are shown in Table 4 and Table 5 respectively.

5 Conclusion

This paper describes experiments on various competitive pre-trained models (BERT, XLNet), domain sampling strategies, negative sampling, data augmentation via back-translation, and active learning. We determine which of these strategies help and hurt multi-domain generalization, finding ultimately that some of the simplest techniques offer surprising improvements. The most significant benefits came from sampling No Answer segments, which proved to be particularly important for training extractive models on long sequences. In combination these findings culminated in the second ranked submission on the MRQA-19 Shared Task.

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A BERT baseline for the Natural Questions. *arXiv preprint arXiv:1901.08634*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384. IEEE.

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. *Transactions of the Association of Computational Linguistics*.

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

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 785–794.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Amrita Saha, Rahul Aralikatte, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. *arXiv preprint arXiv:1905.13453*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. *ACL 2017*, page 191.

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):138.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all

you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *EMNLP 2018*, page 353.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.