Avengers: Achieving Superhuman Performance for Question Answering on SQuAD 2.0 Using Multiple Data Augmentations, Randomized Mini-Batch Training and Architecture Ensembling

Stanford CS224N {Default} Project

Li Yi
Stanford University
liyi@stanford.edu

Abstract

This project aims to achieve above human performance (EM/F1: 86.9/89.5) on the SQuAD 2.0 dataset. We first augmented the training data by randomly substituting with WordNet's synonyms, followed by paraphrasing using the larger PPDB database. Then, in a novel application of GPT-2, we generated new sentences to augment the context paragraphs in a realistic and coherent manner. We further experimented with randomizing the mini-batches, which increased learning difficulty by sampling from different articles and topics. Our best single model, based on a fine-tuned ALBERT-xxlarge-v2 with data augmentation, achieved a Dev EM/F1 score of **87.5/90.4**. By further ensembling multiple models and architectures, specifically across ALBERT, RoBERTa and XLNet, our final ensemble achieved the top EM/F1 score of **89.4/91.7** on the Dev PCE leaderboard and the top EM/F1 score of **89.1/91.5** on the Test PCE leaderboard *(as of Mar-17-2020)*.

1 Introduction

Question Answering (QA) is a challenging task for machines, as it requires both understanding of natural language and knowledge about the world. It has been, however, a central problem in NLP and the broader AI research, since QA is a good proxy for machines' ability to understand the complex reasoning in human languages and a gateway to achieving *strong AI*. It also has numerous real-world applications ranging from improving search to building virtual assistants.

As discussed in a recent survey [1], the field has progressed from the paradigm of information retrieval, which relied on pattern matching and other traditional statistical methods, to the paradigm of neural NLP using end-to-end deep learning architectures. The difficulty of the QA tasks has also increased from picking from several possible options, to span prediction, to free-form generative QA.

The focus of this paper is SQuAD 2.0 [2], a span prediction dataset released by the Stanford NLP Group with 100,000 answerable questions and 50,000 unanswerable questions derived from 536 Wikipedia articles. We achieved above human performance using a mix of techniques including textual data augmentation, randomized mini-batches and curriculum learning [3], and ensembling multiple models and architectures.

2 Related Work

In the original SQuAD 1.0 paper [4], the baseline model used logistic regression on a large set of 180 million lexicalized features and dependency tree path features. Its EM/F1 score of **40.0/51.0** was significantly below the human baseline of **80.3/90.5**. In the subsequent SQuAD 2.0 paper [2], the authors employed a stronger baseline "DocQA No-Answer with ELMo" [5], which mostly closed the

human-machine gap with its score of **78.6/85.8**. However, on the more challenging 2.0 dataset, the DocQA only achieved a EM/F1 of **63.4/66.3**, well below the human baseline of **86.9/89.5**.

The baseline model provided for the default project was based on Bidirectional Attention Flow (BiDAF) [6], a hierarchical multi-stage architecture that allows modeling of the context paragraph at different levels of granularity. The given baseline has word-level and contextual embeddings, but not the character-level embedding layer. The BiDAF model uses bi-directional attention flow to obtain a query-aware context representation. Unfortunately, since all the model's parameters have to be trained from scratch, running the BiDAF baseline only achieved a weak EM/F1 of **57.5/61.0**.

In late 2018, BERT [7] was introduced and achieved SOTA result on SQuAD 2.0 among 11 other downstream NLP tasks. The key idea was pre-training the contextual embeddings on two tasks—Masked Language Model (MLM) and Next Sentence Prediction (NSP)—using a deep Transformer architecture and a massive corpus (Wikipedia 2.5B + BookCorpus), before fine-tuning the final layer to compute the start and end token probabilities for SQuAD. We were also inspired by a few of the top default projects from last Winter's CS224N, where both the best poster [8] and best report [9] were awarded to teams using BERT and data augmentation. By fine-tuning BERT-large ourselves, we achieved an EM/F1 of **78.9/81.6** on CS224N Dev, which was slightly below the best result from last year's class (EM/F1: 79.9/82.4) [8] and within only a point from human performance.

3 Approach

The starting point of our approach was to understand and fine-tune all the top models currently on the SQuAD 2.0 leaderboard[1], namely ALBERT (#1 to #7), XLNet (#9) and RoBERTa (#13). The exception was ELECTRA [10], as its code was not yet available publicly and its addition could be investigated in a future project. After some basic hyper-parameter tuning, for a single model, we found that albert-xxlarge has the highest EM/F1 of **86.3/88.9**, compared to xlnet-large's **85.8/88.4** and roberta-large's **85.7/88.4**.

Then, using ALBERT as the baseline for ablation, we proceeded with four extensions to build our high-performing model—nicknamed "Avengers" given its superhuman performance. First, we augmented the SQuAD 2.0 dataset by randomly substituting with WordNet's synonyms, followed by paraphrasing using the larger PPDB [11] database. Second, we further augmented the dataset with a novel application of GPT-2 [12], by generating entirely new sentences to augment the context paragraphs in a realistic and coherent manner. Third, we experimented different training strategies, such as randomizing the mini-batches, which increased learning difficulty by sampling from different articles and topics. Finally, we ensembled multiple models both within and across architectures, and tested different ways to combine not just `nbest_predictions` but also the `null_odds`, and optimizing for the "no answer" threshold.

3.1 Applying Pre-Trained Model Architectures to SQuAD

3.1.1 ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

ALBERT [13] used two parameter reduction techniques to achieve better results than BERT-large while having fewer parameters. First, "factorized embedding parameterization" decomposed the large vocabulary embedding matrix into two small matrices, which makes it easier to grow the hidden size without increasing the parameter size of the vocabulary embeddings. Second, "cross-layer parameter sharing" prevented the parameter from growing with the depth of the network. The default decision for ALBERT is to share all parameters across layers. It also replaced BERT's NSP task with a new auxiliary task "sentence-order prediction" (SOP). By using the positive examples in the same way as BERT, but swapping the orders of the negative examples, the SOP loss avoided the easier task of topic prediction and instead focused on modeling inter-sentence coherence. Of particular interest to us, on SQuAD 2.0, Zhenzhong Lan et al. reported +2.8 in F1 score when using SOP vs. NSP. Lastly, when fine-tuning for "no answer" in SQuAD 2.0, ALBERT used the same treatment as BERT, where both the start and end of the answer span has the `[CLS]` token.

[1] `https://rajpurkar.github.io/SQuAD-explorer/`

3.1.2 XLNet: Generalized Autoregressive Pretraining for Language Understanding

XLNet [14] increased the sampling efficiency of BERT, where instead of masking 15% of the words at random, XLNet permuted the order for every training sentence at random. This autoregressive approach maximized the expected log likelihood of a sequence with respect to all possible permutations of the factorization order, while also capturing bidirectional context as the permutation operation can choose the context for each position from tokens from either left or right. Another important difference with BERT and its other variants is how XLNet used relative position embeddings, which has empirically improved the performance for tasks involving a longer text sequence. XLNet also dropped the NSP task, as it has not shown any empirical improvements. Finally, when fine-tuning for SQuAD 2.0, XLNet applied a logistic regression loss for answerability prediction similar to classification tasks and a standard span extraction loss for QA.

3.1.3 RoBERTa: A Robustly Optimized BERT Pretraining Approach

RoBERTa [15] found that BERT was significantly under-trained, and the same MLM objective can achieve significantly better result with longer training (500K steps vs. BERT's 100K steps) and on a larger dataset (160GB vs. BERT's 16GB). It also offered enhancement in dynamic masking as RoBERTa will generate the masking pattern every time the sequence was fed to the model, while masking was done only once during data preprocessing in BERT. Similar to ALBERT and XLNet, RoBERTa also removed the NSP task and found that it improved downstream task performance. RoBERTa also optimized some of the hyper-parameters, notably increasing the batch size from BERT's 2K to 8K as this improved the LM's perplexity while making the training easier to parallelize. Lastly, for SQuAD 2.0, RoBERTa utilized an additional binary classifier to predict whether the question is answerable, which was trained jointly by summing the classification and span loss terms.

3.2 Original Work

3.2.1 Word and Phrase-Level Data Augmentation

We first applied random word substitution with their synonyms in WordNet, with a 25% probability of replacement. Then, we used the much larger PPDB [11] (with millions of paraphrases compared to $\approx$200K tokens in WordNet), and did the same 25% replacement. We trained one epoch over the augmented data followed by another epoch over the original data. We have also updated the `answer_start` accordingly and marked questions as unanswerable if the answer span is no longer found in the augmented context. Unfortunately, we did not see any improvements and our error analysis suggested that the replacement probability was too high. After lowering the probability to 10%, we saw an increase in EM/F1 score by +0.48/+0.25 over the baseline albert-xxlarge.

In **Appendix A**, we showcased some successful and unsuccessful synonym replacements in their respective contexts. For example, Wordnet replaced "include" with "incorporating" while PPDB replaced "countries" with "rural area". This result was expected since the `SynonymAug` class in the nlpaug package [16] has been implemented to leverage semantic meaning in word substitution, using techniques such as "stopword dropout" and "data-level paraphrasing" in the paper by Tong Niu et al. [17]. Our further experiments using random swaps, insertions and deletions did not yield sensible outcomes. Substituting with word embeddings such as Word2Vec [18] and GloVe [19] have also produced poor results, since words with high co-occurrence (e.g. "Europe" with "United States" and "backgrounds" with "color") was evidently incorrect in the context of QA.

3.2.2 Sentence-Level Data Augmentation

We found limited past attempts at neural generation augmentation on SQuAD, with only a past CS224N paper reporting a low F1 score of 32.3 [20]. Moreover, after analyzing the SQuAD 2.0 dataset, we found that most of the contexts were fairly short, with the 75th percentile at just 139 words. Hence, instead of padding them, we hypothesized that we could augment the contexts and expand their diversity using NLG. Using the Transformers package by HuggingFace [21], we first manually tweaked the hyper-parameters for NLG using GPT-2 [12] (see **Appendix B**), to ensure that the machine-generated context paragraphs are realistic and coherent (see **Appendix C**). Then, we augmented all short paragraphs with fewer than 100 words with sentences up to another 100 words. We trained our model for one epoch over the GPT-2-augmented data followed by another epoch over the original, which improved the EM/F1 score by +0.31/+0.25 over the baseline albert-xxlarge.

3.2.3 Randomized Mini-Batches and Curriculum Learning

We observed that the SQuAD dataset was organized hierarchically, by the Wikipedia article followed by the paragraphs sequentially within it. As such, we grew curious if the order of training examples in each mini-batch would make any difference. By extracting the paragraphs and flattening the article hierarchy, and randomly shuffling the questions before training, we hypothesized that this would increase the training difficulty and efficiency since each mini-batch of gradient accumulation will now contain questions from different contexts and topics. Intuitively, this is similar to a student rotating his or her study session between computer science, followed by economics, followed by history, before going back to computer science. We were pleasantly surprised by the sizable +0.81/+0.57 gain in EM/F1 score over the baseline with this simple tweak.

Another related idea was "curriculum learning" [3], where instead of sampling mini-batches uniformly, we sampled training examples with increasing complexity, which allowed our model to exploit previously learnt concepts and thus ease the acquisition of new ones. In the guest lecture by Richard Socher during last Winter's CS224N [22], he also discussed an "anti-curriculum" pre-training strategy which we have also tried. While more sophisticated methods probably exist, we simply used the length of the context as a proxy for difficulty, and by turning off shuffling and training our model from the shortest to the longest context, we gained +0.94/+0.69 in EM/F1. While the reverse training from the longest to the shortest context produced a smaller EM/F1 gain of +0.81/+0.63.

3.2.4 Ensembling Multiple Models and Architectures

Ensemble is a proven method for improving the robustness and accuracy of most predictive models. For instance, even within the same model family such as ALBERT, by combining multiple runs/checkpoints and also different specification of 12-layer and 24-layer architectures, Zhenzhong Lan et al. was able to improve the SQuAD 2.0 Dev EM/F1 from **87.4/90.2** to **88.9/91.4**.

By bagging and averaging predictions from different pre-trained architectures, we would expect an even stronger result than just combining multiple runs of the same architecture. In particular, we saw in Section 3.1 that ALBERT / XLNet / RoBERTa have different LM objectives (e.g. masking vs. permutation), auxiliary tasks (e.g. SOP), varying hyper-parameters and training data, and different approach to handling unanswerable questions in SQuAD 2.0. Thus, we proposed to combine their `nbest_predictions` and `null_odds` via majority voting, before separately optimizing for the "no answer" threshold again on the combined prediction.

4 Experiments

4.1 Data

We used the SQuAD 2.0 dataset, split into the official Train Set of 129,941 questions, and CS224N custom Dev Set and Test Set, with 6,078 and 5,915 questions respectively. We also conducted a deep-dive analysis of the dataset in **Appendix D**, which gave us a much better understanding of the challenges and some inspirations for our various approach in Section 3.

4.2 Evaluation Method

We use the two standard metrics of Exact Match (EM) and F1 Score (F1) for SQuAD. EM measures the percentage of predictions that match any one of the ground truths exactly. F1 measures the average overlap between the prediction and the ground truth, using a harmonic mean of precision and recall, where each prediction/answer pair is treated as a bag of tokens. The maximum F1 score is taken for questions with multiple answers and the final score is the average F1 across all questions.

4.3 Experimental Details

4.3.1 Code and Infrastructure

We used a mix of PyTorch and TensorFlow, and relied heavily on the Transformers package [21] for its implementation of ALBERT, RoBERTa and GPT-2. Given the outstanding issue with XLNet[2],

[2] `https://github.com/huggingface/transformers/issues/2651`

however, we used the original codebase by Zhilin Yang et al.[3]. To speed up our experiments using TPU, we also used the original code by Zhenzhong Lan et al[4] for ALBERT. For WordNet and PPDB synonym replacement, we used the nlpaug package [16]. Computation were performed across our local machine (GPU), AWS (p3.8xlarge) and GCP (TPU-v3-8).

4.3.2 Hyper-Parameter Tuning

We reviewed and summarized the hyper-parameters used by the original authors in Table 1. Then, we did a grid search and found that `batch_size` seemed to work best around 24 to 48, where anything lower than 4 and higher than 128 produced lower scores. We then varied `learning_rate` between 1e-5 to 5e-5, and saw no notable difference, especially if we ran for larger `max_steps` for the slower learning rates. The scores were highest when `batch_size * max_steps / num_examples` was around 1.5 to 2.5 epochs, and started to deteriorate beyond 4 epochs. Finally, while gains were small, `max_seq_length` at 512 worked best compared to shorter ones such as 256 or 384 (as there are only 21 paragraphs in the training data longer than 384 words, and only 3 longer than 512 words).

Model	BERT	ALBERT	RoBERTa	XLNet	Avengers
Batch Size	32	48	48	48	24 to 48
Learning Rate	5e-5	3e-5	1.5e5	3e-5	3e-5
Max Epochs	3	3	2	3	2 to 4
Max Sequence	384	512	-	512	512
Warmup Ratio	-	-	0.10	0.06	0.01
Weight Decay	-	-	0.01	0.01	0.01
Augmentation	TriviaQA	-	-	NewsQA	Multiple

Table 1: Comparison of Hyper-Parameters for SQuAD 2.0

4.4 Results

Our final ensemble Avengers achieved the top EM/F1 score of **89.4/91.7** on the Dev PCE leaderboard and the top EM/F1 score of **89.1/91.5** on the Test PCE leaderboard *(as of Mar-17-2020)*.

4.4.1 Overall Comparison

#	Model	Remarks	Dev EM	Dev F1	Test EM	Test F1
1	BiDAF	-	57.5	61.0	-	-
2	BERT-large	-	78.9	81.6	-	-
3	Human	Official Dev	86.9	89.5	-	-
4	ALBERT	xxlarge Baseline	86.7	89.7	-	-
5	ALBERT	WordNet/PPDB Aug	87.2	90.0	-	-
6	ALBERT	GPT2 Aug	87.0	90.0	-	-
7	ALBERT	Random Training	87.5	90.3	-	-
8	ALBERT	Short to Long	87.5	90.4	-	-
9	ALBERT	Long to Short	87.6	90.3	-	-
10	ALBERT (Ensemble)	4+5+6+7+8+9	88.8	91.3	88.6	91.1
11	XLNet	Single Large	85.8	88.4	-	-
12	RoBERTa	Single Large	85.5	88.4	-	-
13	RoBERTa	2 x Large	85.9	88.9	-	-
14	Avengers	10 + 11 + 13	**89.4**	**91.7**	**89.1**	**91.5**

Table 2: Overall Performance Comparison

Table 2 shows the result of our key models. The lack of test scores was due to submission limits to Test PCE leaderboard. Among the single ALBERT models, the various approaches we have tried generally produced a F1 gain between +0.3 to +0.7. More importantly, by combining them into an ALBERT ensemble, we achieved an EM/F1 score of **88.8/91.3** on Dev and **88.6/91.1** on Test.

[3] `https://github.com/zihangdai/xlnet`

[4] `https://github.com/google-research/ALBERT`

For XLNet and RoBERTa, their baseline F1 scores trained on the original data were behind ALBERT by 1.3 points. We didn't have time to run both of them on our augmented data, but we saw a small improvement by averaging two single RoBERTa in #13. As discussed in Section 3.2.4, when we combined the components of #10, #11 and #13, our final ensemble has the strongest result as expected, with an EM/F1 score of **89.4/91.7** on Dev and **89.1/91.5** on Test. This was +2.5 EM and +2.2 F1 over human performance on SQuAD 2.0, and validated that our approach was fairly successful.

4.4.2 Training Progression of Augmentation and Randomized Training

As we controlled the hyper-parameters to be consistent across the various ALBERT runs, we can examine the effect of training speed vs. the different augmentation and randomized training approaches. Table 3 highlighted the best Dev F1 score at different training steps, and suggested that training on our augmented data (#5 and #6) have lower scores initially, but reached higher peak values with more training steps. Notably, the "curriculum learning" in #8 (Short-2-Long) was able achieve strong results as early as 1500 Step (less than 1/3 epoch), and peaked at 90.39 F1. Further research is needed to better understand and validate this finding, possibly by averaging multiple runs to reduce fluctuations due to random noise and to establish a confidence interval for the gains.

Step	#4 Baseline	#5 Word Aug	#6 GPT2 Aug	#7 Random	#8 S2L	#9 L2S
500	86.97	85.34	86.58	87.17	**87.68**	86.28
1000	87.38	87.91	87.89	87.86	87.59	**88.16**
1500	89.01	87.99	88.92	88.74	**89.44**	88.83
2000	89.02	87.95	89.15	**89.62**	89.07	89.43
2500	89.35	88.67	**89.57**	89.12	89.43	89.50
3000	89.12	88.73	89.54	89.38	89.66	**89.87**
3500	89.27	88.81	89.49	**89.91**	89.88	89.79
4000	89.31	89.11	89.68	89.96	90.05	**90.12**
4500	88.71	89.85	89.92	90.13	**90.34**	90.13
5000	89.61	89.73	89.95	90.27	**90.39**	90.30
5500	89.70	89.95	89.83	90.26	90.12	**90.33**

Table 3: Dev F1 Score Across Training Checkpoints

4.4.3 ALBERT vs. XLNet vs. RoBERTa vs. Avengers

We further breakdown the EM/F1 scores of the different models by their performance on answerable and unanswerable questions in Table 4. While it is clear that Avengers has the best performance across all categories, XLNet was almost 2 points higher than RoBERTa for unanswerable questions, despite having comparable overall F1 scores. This might be due to its better implementation of the answerability prediction classification task. We also observed that Avengers was able to correctly identify 92.9% of all unanswerable questions, which was pretty remarkable given these were written *"adversarially by crowdworkers to look similar to answerable ones [with the] existence of plausible answers to avoid type-matching heuristics"* [2].

Dev Metrics	ALBERT	XLNet	RoBERTa	Avengers
Has Ans EM	82.9	81.4	83.7	**85.4**
Has Ans F1	89.1	86.9	89.5	**90.3**
No Ans EM	90.2	89.9	88.0	**92.9**
No Ans F1	90.2	89.9	88.0	**92.9**
Best EM	86.7	85.8	86.2	**89.4**
Best EM Threshold	*-2.88*	*-3.35*	*-2.20*	*-2.56*
Best F1	89.7	88.4	88.9	**91.7**
Best F1 Threshold	*-2.80*	*-3.35*	*-1.01*	*-2.31*

Table 4: Further Breakdown of Performance by Architectures

5 Analysis

We inspected the predictions of our final model and also its component models to qualitatively understand when they succeed and when they fail. Given their high accuracy rates, we will focus on examples when they produced the wrong predictions.

5.1 Ambiguous / Questionable Ground Truth

- **Relevent Context:** *"The Norman dynasty had a major political, cultural and military impact on medieval Europe and even the Near East. The Normans were famed for their martial spirit and eventually for their Christian piety, becoming exponents of the Catholic orthodoxy into which they assimilated..."*
- **Question:** What religion were the Normans?
- **Avengers Prediction:** "Christian"
- **ALBERT / XLNet / RoBERTa Prediction:** "Christian" / "Christian" / "Christian"
- **Possible Ground Truth:** "Catholic" / "Catholic orthodoxy"
- **Analysis:** All our models gave the same answer "Christian", which appeared earlier in the context, but failed to pick up "Catholic". In this case, however, we believe "Christian" should also be accepted as a possible answer, so the ground truth itself is ambiguous.

5.2 Distraction Span with No Answer

- **Context:** *"Although lacking historical connections to the Middle East, Japan was the country most dependent on Arab oil. 71% of its imported oil came from the Middle East in 1970. On November 7, 1973, the Saudi and Kuwaiti governments declared Japan a 'nonfriendly' country to encourage it to change its noninvolvement policy. It received a 5% production cut in December, causing a panic. On November 22, Japan issued a statement 'asserting that Israel should withdraw from all of the 1967 territories, advocating Palestinian self-determination, and threatening to reconsider its policy toward Israel if Israel refused to accept these preconditions'. By December 25, Japan was considered an Arab-friendly..."*
- **Question:** What did Israel do to Japan's imported oil to force their involvement in the crisis?
- **Avengers Prediction:** No Answer *(Correct)*
- **ALBERT / XLNet / RoBERTa Prediction:** No Answer *(Correct)* / "5% production cut" / "a 5% production cut"
- **Possible Ground Truth:** NIL, but plausible answer include "5% production cut"
- **Analysis:** It was the Saudi and Kuwaiti governments that forced Japan's involvement, and "Israel" was disguised as a distractor in the question. This is a good example where both XLNet and RoBERTa were fooled, while Avengers was able to produce the correct no answer largely by averaging the `null_odds` which at 2.1 was fairly close to the threshold.

In the same context, another question was asked but none of our models gave the correct answer.

- **Question:** What action was taken against Japan on December 25th to make them change their policy?
- **Avengers Prediction:** "5% production cut"
- **ALBERT / XLNet / RoBERTa Prediction:** All "5% production cut'
- **Possible Ground Truth:** NIL, but plausible answer include "the Saudi and Kuwaiti governments declared Japan a 'nonfriendly' country"
- **Analysis:** Our models seemed confused by the logic and failed to understand the sequence of events. In other words, we suspected that they did not understand December 25 occurred after November 7, which was the actual date when the action was taken against Japan, while December 25 was the date Japan was considered Arab-friendly again.

5.3 Complex Domain Knowledge

- **Relevent Context:** *"Where is the relevant cross-sectional area for the volume for which the stress-tensor is being calculated. This formalism includes pressure terms associated with forces that act normal to the cross-sectional area (the matrix diagonals of the tensor) as well as shear terms associated with forces that act parallel to the cross-sectional area (the off-diagonal elements). The stress tensor accounts for forces that cause all strains (deformations) including also tensile stresses and compressions."*
- **Question:** What includes pressure terms when calculating area in volume?
- **Avengers Prediction:** No Answer *(Incorrect)*
- **ALBERT / XLNet / RoBERTa Prediction:** No Answer *(Incorrect)* / No Answer *(Incorrect)* / "stress-tensor"
- **Possible Ground Truth:** "formalism"
- **Analysis:** This is an article on mechanics and our models clearly do not understand it. In fact, for most of the questions they were just guessing between "stress-tensor", "shear" and "pressure terms". In defense of our model, however, looking through some of the ground truths, we suspected that not all the crowdworkers understood the topic either.

6 Conclusion

In this paper, we successfully designed and implemented a high-performing ensemble model—nicknamed "Avengers"—for QA on SQuAD 2.0. It surpassed human performance by +2.5 EM and +2.2 F1, and achieved the top EM/F1 score of **89.4/91.7** on the Dev PCE leaderboard and the top EM/F1 score of **89.1/91.5** on the Test PCE leaderboard.

Our multi-pronged approach involved first data augmentation by randomly substituting with WordNet's synonyms and paraphrasing using the PPDB database. Then, we generated new sentences to augment the context paragraphs in a novel application of GPT-2. We further optimized the training performance by randomizing mini-batches and curriculum learning, before creating an ensemble across ALBERT, RoBERTa and XLNet.

In our pursuit to maximize performance, we learnt that it was hard to beat using more data and training larger networks. For future studies, perhaps combining other data augmentation techniques such as back-translation and pre-training with external QA datasets such as TriviaQA and NewsQA could push the performance boundary further. That said, it was somewhat discouraging to see that the differences between the BERT extensions we have implemented could also be explained by data and compute. For instance, short of the more efficient sampling strategy in XLNet, we saw similar performance in RoBERTa when BERT's MLM was trained with just more data. Likewise, while ALBERT was created as a model with fewer parameters, it was not computationally cheaper and its performance gain was diminished substantially once normalized for training time.

Given that over one hundred hours of combined GPU/TPU computation was used to train Avengers, one limitation of our work was the time and resources required and the inconvenience of running several architectures. Another limitation was the lack of interpretability, while this is a common problem for most neural NLP models, it was exacerbated by our use of a large ensemble. Nonetheless, future projects would likely achieve better performance by creating an even larger ensemble, perhaps with new PCE models such as ELECTRA [10] and specialist QA model such as SpanBERT [23]. In addition, instead of majority voting, a second-level stacker model could be built for ensembling, perhaps combined with a multi-stage verifier such as the retrospective reader [24].

Appendix A Examples of Word Augmentation via Synonym Replacement

Successful Word Augmentation Example (Wordnet)

"Some definition of southern Europe, ~~also~~ as well known as Mediterranean Europe, include the state of the Iberian peninsula (Spain and Portugal), the Italian peninsula, southern France and Greece. Other definitions sometimes ~~include~~ incorporating the Balkan countries of southeast Europe, which are geographically in the southern part of Europe, but which have different historical, political, economic, and cultural backgrounds."

Successful Word Augmentation Example (PPDB)

"Some definitions of southerly Europe, also known as Mediterranean Europe, include the ~~countries~~ rural area of the Iberian peninsula (Spain and Portugal), the Italian peninsula, southerly France and Greece. Other definitions sometimes ~~include~~ comprise the Balkan countries of southeast Europe, which are geographically in the southern part of Europe, but which have different historical, political, economic, and cultural backgrounds."

Unsuccessful Word Augmentation Example (word2vec)

"Some definitions of southerly ~~Europe~~ United_States, also known as Mediterranean Europe, include the countries of the Iberian peninsula (Spain and Portugal), the Italian peninsula, southerly France and Greece. Other ~~definitions~~ synonyms sometimes include the Balkan countries of southeast Europe, which are geographically in the southern part of Europe, but which have different historical, political, economic, and cultural backgrounds."

Unsuccessful Word Augmentation Example (glove)

"Some ~~definitions~~ common of southerly Europe, also known as Mediterranean Europe, include the countries of the Iberian peninsula (Spain and Portugal), the Italian peninsula, southerly France and Greece. Other definitions sometimes include the Balkan countries of southeast Europe, which are geographically in the ~~southern~~ southwestern part of Europe, but which have different historical, political, economic, and cultural ~~backgrounds~~ color."

Appendix B Hyper-Parameters for GPT-2

Augment if existing paragraph has **<100** words, which affected 41% (7834/19035) of all contexts.

- `max_length = min(prompt_len*2, prompt_len+100)` *# max length of generated seq*
- `top_k = 50` *# number of highest probability vocabulary tokens to keep for top-k-filtering*
- `top_p = 0.9` *# highest probability vocabulary tokens to keep for nucleus sampling*
- `length_penalty = 1.5` *# exponential penalty to the length*
- `repetition_penalty = 1.2` *# parameter for repetition penalty*
- `do_sample = True` *# If set to False greedy decoding is used. Otherwise sampling is used.*
- `num_return_sequences = 1` *# number of independently computed returned sequences*
- `temperature = 1.2` *# value used to module the next token probabilities*

Appendix C Examples of NLG-Augmented Context via GPT-2

Successful Sentence Augmentation Example 1

"Since 1996, they have been archiving cached pages of web sites onto their large cluster of Linux nodes. They revisit sites every few weeks or months and archive a new version if the content has changed. Sites can also be captured on the fly by visitors who are offered a link to do so. The intent is to capture and archive content that otherwise would be lost whenever a site is changed or closed down. Their grand vision is to archive the entire Internet. One interesting aspect: Google's search engines keep track at least twice when searching for 'the original website' before being taken offline after viewing them in response (or something similar). That data could prove crucial later as users will not immediately get an accurate sense just how frequently different information was available with regards online history such time stamps from what appears right about where you searched most recently."

Successful Sentence Augmentation Example 2

"The earthquake had a magnitude of 8.0 Ms and 7.9 Mw. The epicenter was in Wenchuan County, Ngawa Tibetan and Qiang Autonomous Prefecture, 80 km west/northwest of the provincial capital of Chengdu, with its main tremor occurring at 14:28:01.42 China Standard Time (06:28:01.42 UTC), on May 12, 2008 lasting for around 2 minutes, in the quake almost 80% of buildings were destroyed. Earthquake caused major damage to residential structures but still more than 90-75000 people have been left without power since that."

Unsuccessful Sentence Augmentation Example 1

"Some definitions of southern Europe, also known as Mediterranean Europe, include the countries of the Iberian peninsula (Spain and Portugal), the Italian peninsula, southern France and Greece. Other definitions sometimes include the Balkan countries of southeast Europe, which are geographically in the southern part of Europe, but which have different historical, political, economic, and cultural backgrounds. As many Canadians were thinking for a moment before Monday night's presidential election when two leading candidates had made history by running their own races while campaigning on behalf that country."

Appendix D SQuAD 2.0 Dataset Analysis

To better find augmentation techniques and relevant data samples, we carefully analyzed the SQuAD 2.0 dataset, which consists of crowdsourced questions and answers from English Wikipedia. Specifically, the authors sampled 536 articles randomly from the top 10000 articles based on their *internal* PageRanks[5]. As the CS224N project further splits the official dev set into dev and test, we will focus on the 442 articles sampled for the train set.

The 442 articles (labeled as *title*) were further divided into 19,035 paragraphs (labeled as *context*) and 130,319 questions. Out of these questions, 43,498 were "negative examples" which are unanswerable, and the authors have also taken measures to make sure they are still relevant and plausible answers could exist. This is relevant since augmenting negative examples automatically, such as selecting or even generating random questions, might be too easy. Some ideas we have are using other questions from the same article but from a different paragraph, and replacing some words in the question with antonyms and marking them unanswerable.

Out of the remaining 86,821 answerable questions, the typical answer span was very short. There were 31,464 (36.2%) single-word answers, with the median answer at two words and the 75th percentile answer at just four words. Nonetheless, there were also some curiously long answers, with the longest at 43 words[6]. Similarly, the questions also tend to be fairly short, with the 25th/50th/75th percentile at 7/9/12 words respectively. Manually inspecting the very short questions, especially those with just one word[7], seemed to suggest potential errors in the dataset.

Of the 19,035 paragraphs, the 25th/50th/75th percentile number of words were at 87/107/139 respectively. While the shortest (about "preaspirated consonants") only has 20 words, the longest (about "Sahara desert") has 653 words. This is relevant given that the hyper-parameter `max_seq_length` in BERT was defaulted to 384, where *"sequences longer than this will be truncated"*. Since there are only 21 paragraphs (0.1%) longer than 384 words, the impact should be fairly small although in persuit of SOTA results, both XLNet and ALBERT used `max_seq_length=512`, as there are now only 3 paragraphs longer than 512 words.

Separately, we observed that the rate of unanswerable question go up for shorter spans (e.g. 5 out of the 7 questions for "preaspirated consonants" are unanswerable), but this is less interesting from a NLU perspective although it could be explored/exploited as an ancillary feature to climb the SQuAD leaderboard. Finally, we noted that the 25th/50th/75th percentile of number of questions per paragraph was 5/5/9 respectively, although there are contexts with just 1 question (about "autosomal SNPs") and also 30 questions (about "Queen Victoria").

[5] `https://www.nayuki.io/page/computing-wikipedias-internal-pageranks`

[6] The question "What was concluded about the construction?" is regarding the 2008 Sichuan Earthquake, with the answer "that the sudden shift of a huge quantity of water into the region could have relaxed the tension between the two sides of the fault, allowing them to move apart, and could have increased the direct pressure on it, causing a violent rupture".

[7] For instance, Question 57262473271a42140099d4ed is "d" and Question 57262473271a42140099d4ec is "dd", both should not have a reasonable answer.

References

[1] Boyu Qiu, Xu Chen, Jungang Xu, and Yingfei Sun. A Survey on Neural Machine Reading Comprehension. *arXiv:1906.03824*, 2019. https://arxiv.org/abs/1906.03824.

[2] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv:1806.03822*, 2018. https://arxiv.org/abs/1806.03822.

[3] Guy Hacohen and Daphna Weinshall. On The Power of Curriculum Learning in Training Deep Networks. *arXiv:1904.03626*, 2019. https://arxiv.org/abs/1904.03626.

[4] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv:1606.05250*, 2016. https://arxiv.org/abs/1606.05250.

[5] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. *arXiv:1802.05365*, 2018. https://arxiv.org/abs/1802.05365.

[6] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *arXiv:1611.01603*, 2016. https://arxiv.org/abs/1611.01603.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805v2*, 2018. https://arxiv.org/abs/1810.04805.

[8] Wen Zhou, Xianzhe Zhang, and Hang Jiang. Ensemble BERT with Data Augmentation and Linguistic Knowledge on SQuAD 2.0. *CS224N Final Project Reports*, 2019. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15845024.pdf.

[9] Sina J. Semnani, Kaushik Ram Sadagopan, and Fatma Tlili. BERT-A: Fine-tuning BERT with Adapters and Data Augmentation. *CS224N Final Project Reports*, 2019. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15848417.pdf.

[10] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*, 2020. https://openreview.net/pdf?id=r1xMH1BtvB.

[11] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2015. "https://www.aclweb.org/anthology/P15-2070".

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Better Language Models and Their Implications. 2018. https://openai.com/blog/better-language-models/.

[13] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*, 2020. https://arxiv.org/abs/1909.11942.

[14] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*, 2019. https://arxiv.org/abs/1906.08237.

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019. https://arxiv.org/abs/1907.11692.

[16] Edward Ma. NLP Augmentation, 2019. https://github.com/makcedward/nlpaug.

[17] Tong Niu and Mohit Bansal. Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models. *arXiv:1809.02079*, 2018. https://arxiv.org/abs/1809.02079.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*. Neural Information Processing Systems Foundation, 2013. https://arxiv.org/abs/1310.4546.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014. https://www.aclweb.org/anthology/D14-1162.

[20] Alex Lu. Semi-Supervised Question Answering: Generative Augmentation in SQuAD 2.0. *CS224N Final Project Reports*, 2019. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15831492.pdf.

[21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771*, 2019. https://arxiv.org/abs/1910.03771.

[22] Richard Socher. The Natural Language Decathlon: Multitask Learning as Question Answering. 2018. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/slides/cs224n-2019-lecture17-multitask.pdf.

[23] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *1907.10529*, 2019. https://arxiv.org/abs/1907.10529.

[24] Hai Zhao Zhuosheng Zhang, Junjie Yang. Retrospective Reader for Machine Reading Comprehension. *2001.09694*, 2020. https://arxiv.org/abs/2001.09694.