A BERT based model for Multiple-Choice Reading Comprehension

Kegang Xu
Stanford
tosmast

Jung Youn Kim
Stanford
jyk423

Jing Jie Tin
Stanford
jjtin1

Abstract

In this paper, we propose a novel Deep Comatch Network (DCN) based on a Bidirectional Encoder Representations from Transformers (BERT) model which significantly improved the performance compared to baseline model in RACE dataset. In Deep Network, we designed a novel comatch attention layer following 5 layers of coattention after Bert encoder. Our DCN benefits from the question-aware passage representation and answer-aware passage representation through the attentions between Passage/Question and Passage/Answer. We also fine-tuned the hyper-parameters of the baseline model and applied Easy data augmentation. We achieved 66.2% accuracy on our DCN model. Finally, we built the ensemble model based on all the models and reached 67.9% performance which could rank top 6 in RACE leaderboard.

1 Introduction

Automated reading comprehension can be applied to a wide range of commercial applications such as technical support and troubleshooting, customer service, and the understanding of healthcare records and financial reports. This report will explore the various ways in which a computer can be taught to achieve a human-level performance in text understanding task. If intelligence agents can be improved with learning algorithms to understand the text fast and efficiently without human supervision, many industries seeking for automation can benefit from this. Among the diverse types of text understanding tasks, this project will focus on automated multiple-choice reading comprehension on RACE (4) dataset. Compared to SQuAD (5) where the passages are written in a single fixed style, the answers of RACE could not directly be extracted from the passage because answering the question requires higher level of understanding and reasoning. Specifically, it has the following challenges:

1. Created by domain experts to test high and middle students reading comprehension skills and requires a high level of reasoning and calculation techniques.

2. Includes a wide variety of question types, like Summarization, Inference, Deduction and Context Matching.

3. A Broad coverage in various domains and writing styles.

Thus, to tackle these challenges, this report will be approaching machine reading comprehension in four ways: one through a pretrained BERT algorithm, and the other through our novel Deep Comatch Network model, an Attention over Attention (2) algorithm, and lastly, an ensemble model. We will also fine-tune the base model with all kinds of hyper-parameters and explore Easy Data Augmentation. We used an ensemble technique to combine all of the models in a hope to achieve the best accuracy in multiple choice reading comprehension.

1.1 Problem definitions

The input to our algorithms will be a sequence of a passage, a question, and an answer option. We then use a BERT, AoA, DCN, and a ensemble approach to predict the correct answer of the problem.

To test how successful the models perform to predict the correct answer given a passage and a question, we will evaluate the prediction of correct label against the total of questions in the dataset as a percentage score. Hence, throughout the project, the accuracy metric used to evaluate our model performance will be defined as follows:

1. Accuracy = $\frac{\#\text{of correct answers}}{\#\text{of total questions}}$

2. Loss = Cross Entropy Loss

2 Related Work

There have been many attempts to implement machine learning models that can successfully solve the reading comprehension problems. Some of the datasets that are widely used to test these models are SQuAD (5), NEWSQA (7), and MCTest (6). However, the content of these datasets are gathered from one specific field or written in one fixed style, and has span-based answers. Thus, we challenged ourselves by choosing a relatively new dataset, RACE (4) which consists of passages from broad range of topics and styles so the aim of our models is to approach the human's ability of logical thinking and comprehension.

Previous papers mainly focused on pairwise sequence matching to improve the reading comprehension capability of Neural Networks. This is performed by either matching a passage and the concatenation of the passage's questions and answers (10), or by matching a passage with its question before selecting a possible answer (12). In Literature, many types of attentions have been proposed to enhance neural network reasoning in the passage level. Xu et al. (9) used multi-hop reasoning mechanism and proposed the Dynamic Fusion Networks. Zhu et al. (12) proposed the Hierarchical Attention Flow model in order to better model the interactions among passage, questions, and candidates. Bidirectional Encoder Representations from Transformers (BERT) (3) achieved state-of-the art in eleven NLP tasks. Inspired by Dual Comatching Attention (11), we propose Deep Comatch Networks based on the output of BERT's hidden state.

3 Dataset and Features

Dataset	RACE-Middle			RACE-High		
Subset	Train	Dev	Test	Train	Dev	Test
#Passages	6,409	368	362	18,728	1,021	1,045
#Questions	25,421	1,436	1,436	62,445	3,451	3,498

Figure 1: Overview of RACE dataset

RACE dataset includes problems for middle school and high school students. As noted in fig. 1, there are 7,139 passages from RACE dataset for middle school and 20,794 passages from RACE-high dataset. In addition, the total number of passages and questions adds up to 27,933 and 97,687 respectively. If we look at the distribution of number of data in each sets, we can easily notice that 90-5-5 training-validation-test split was used both for RACE-Middle and RACE-High. We also figured that Middle dataset averages about 250 words per passage while the High dataset averages 350 words per passage which will give a great insight in choosing the right max sequence length for the baseline model.

3.1 Data Preprocess

We concatenated a passage, a question, and an option together with special tokens CLS and SEP as the input sequence for BERT model (12). Thus, we will have 4 of such inputs for each question with the correct option as a true label and predict an answer (label number).

```
Input:  [CLS] passage [SEP] question [SEP] option 1 [SEP]
        [CLS] passage [SEP] question [SEP] option 2 [SEP]
        [CLS] passage [SEP] question [SEP] option 3 [SEP]
        [CLS] passage [SEP] question [SEP] option 4 [SEP]
Output: the label of an option
```

3.2 Data Augmentation

Easy Data Augmentation (EDA) (8) was originally developed to enhance text classification on small data sets. EDA intends to create the new and augmented passages by the following operations:

1. **Synonym Replacement:** Randomly choose several non-stop words. Replace each of these words with one of their synonyms randomly.

2. **Word Insertion:** Find a random synonym of a random non-stop word in the sentence. Insert that synonym into a random position in this sentence.

3. **Word Swap:** Randomly choose two words in the sentence and swap their positions.

4. **Word Deletion:** Randomly remove words in a sentence with a specified probability.

Hence, EDA expands the size and diversity of the existing dataset. We hope this could prevent overfitting during training and help to build a more robust model.

4 Methods

As outlined previously, we have explored Attention over Attention (AoA) model and utilized Bidirectional Encoder Representations from Transformers (BERT), Deep Comatch Network (DCN), and ensemble models.

4.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) (5) achieved the state-of-the art performance in 11 NLP tasks in 2018. Our multiple-choice model is conducted upon HuggingFace's BERT implementation with PyTorch and would add one linear layer and a SoftMax layer over the final hidden layer to predict the correct answer. Then we will fine-tune pretrained uncased BERT base model (10) on RACE dataset as our baseline and improve upon it. The total number of encoder layers are 12 and 24 in base and large models respectively.

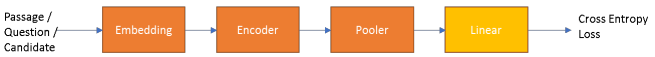

Figure 2: Baseline model

BERT further refines the self-attention layer with multi-headed attention. It improves the performance in two ways. On one hand, it makes the model to capture the contextual correlation of words in long distance. On the other hand, it provides the ability to represent subspaces.

Lastly, there are a total of 15 hyper-parameters in our implementation of BERT model. Among them, we chose to experiment with 5 that impacted the accuracy of model most significantly.

1. **Learning rate:** The initial learning rate for Adam optimizer.

2. **Freeze layers:** Freeze some of BERT encoder layers. Depending on base or large model, the number of freeze layer is 6 or 12 in DCN.

3. **L2 regularization:** Weight decay is set to 0.1, 0.01, or 0.001.

4. **Batch size:** The number of samples from the training set used to estimate the error gradient.

5. **Max sequence length:** The maximum total input sequence length. Sequences longer than this length will be truncated and sequences shorter than this will be padded.

4.2 Attention over Attention

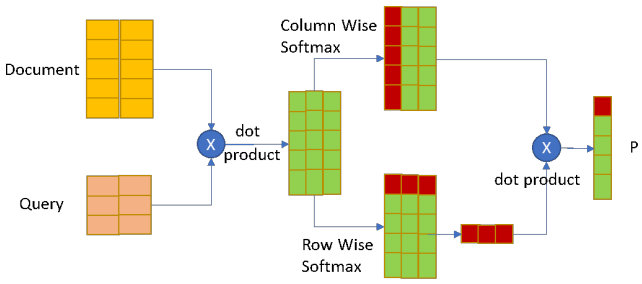

Figure 3: Architecture of AoA layer

Our initial motivation in this project was to push the attention concept in BERT further and introduce the Attention over Attention (AoA) (2) layer to the original BERT model. It is a relatively new NN architecture which aims to place another attention mechanism over the existing document-level attention. Instead of solely summing or averaging individual attentions to get a final attention

score for each word, an additional "importance" distribution on the query is carried out to determine which query words are more important given a single document word. Therefore, it increases the information available to the NN.

We could get the column-wise Softmax (α) and row-wise Softmax (β), we would obtain a new variable $S_i = \alpha^T \beta$. The final probability of each multiple-choice option (w) being the correct answer would then be given by

$$P(w|D,Q) = \sum_{i \in I(w,D)} S_i(w \in V).$$

However, we quickly realized it is difficult to implement AoA integration due to the nature of BERT architecture.

4.3 Deep Comatch Network

Instead, we designed a Deep Comatch Network by introducing the question-aware passage representation and answer-aware passage representation to fully explore the available information in the following triplet {Passage, Question, Answer}. Before getting to the comatch layer (11), network goes through 5 layers of P2Q/Q2P and P2A/A2P coattention. The deep comatch attention inputs the hidden context of passages, question and answer which are output by BERT encoder Layer. In the end, the classifier layer will output the predicted answer after the maxpooling layer in the comatch block.

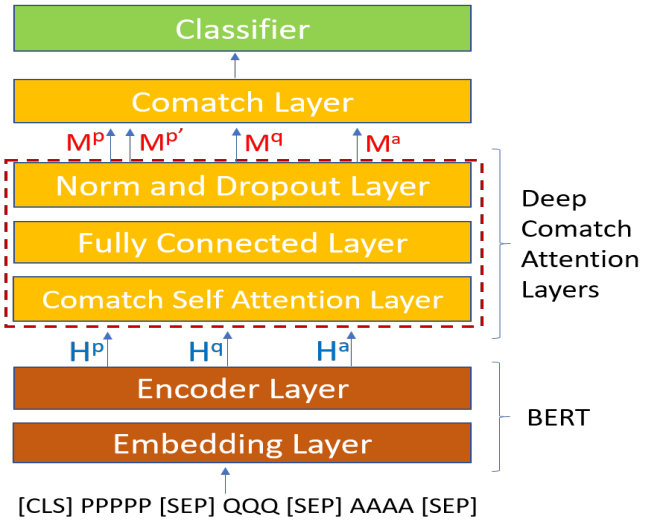

Figure 4: Deep Comatch Network

Firstly, the passage, question and candidate answer are encoded as follows:

$$H^p := BERT(P)$$
$$H^q := BERT(Q)$$
$$H^a := BERT(A)$$

where $H^p \in \mathcal{R}^{P \times L}, H^q \in \mathcal{R}^{Q \times L}, H^a \in \mathcal{R}^{A \times L}$ are sequences of hidden state generated by BERT. Here, P, Q and A are the sequence length of the passage, the question and the candidate answer respectively, and L is the dimension of the BERT hidden state.

Then we handle coattention operation (1) between P2Q/Q2P and P2A/A2P using the following formulas. Note that this is where it differs from the previous papers; in our new model, we also explored the relationship (attention) between a passage and an answer.

$$\begin{aligned}
W &= Coattention(Q,K,V) \\
&= Softmax(\frac{Q \cdot K}{\sqrt{d}})V \\
M^p &= WH^a, M^a = W^T H^p, \\
M^{p'} &= WH^a, M^q = W^T H^p \\
S^p &= Relu([M^a H^a; M^a \cdot H^a]W_1) \\
S^{p'} &= Relu([M^a H^a; M^a \cdot H^a]W_2) \\
S^a &= Relu([M^p H^p; M^p \cdot H^p]W_3) \\
S^q &= Relu([M^q H^q; M^q \cdot H^q]W_4)
\end{aligned}$$

where $-$ and $\cdot$ are the element-wise matrix subtraction and multiplication and $[;]$ is the column-wise matrix concatenation. We use different weights so that we have S^p and $S^{p'}$. Then we concatenate them after maxpooling:

$$C^x = Maxpooling(S^x), x \in (p, p', q, a)$$
$$C = [C^p; C^a; C^{p'}; C^q]$$

Finally, we compute the cross entropy loss after a classifier layer.

4.4 Ensemble

To further improve the performance, we also explored the ensemble model which assembles all the aforementioned techniques. After building different BERT models with different hyper-parameters, we incorporated DCN on EDA dataset to produce a new ensemble model. Thereafter, we will explore the relative weight of each model and tune the weights to achieve a better performance.

5 Experiments/Results

We fine-tuned hyper-parameters of the baseline model and got 62.2%. We applied Easy data augmentation and gained 0.6% performance. We achieved 66.2% accuracy on our DCN model. Finally, we build the ensemble model based on all the models and reached 67.9% performance which could rank top 6 in RACE leader board.

Model	Accuracy
Base BERT	61.6%
Fine-tuned Base BERT	62.6%
Large BERT	65.0%
Easy Data Augmentation	65.6%
Deep Comatch Network	66.2%
Ensemble of models	67.9%

Figure 5: Accuracy of models

5.1 BERT

In this project, we experimented the following hyper-parameters on the BERT base model.

5.1.1 Learning rate

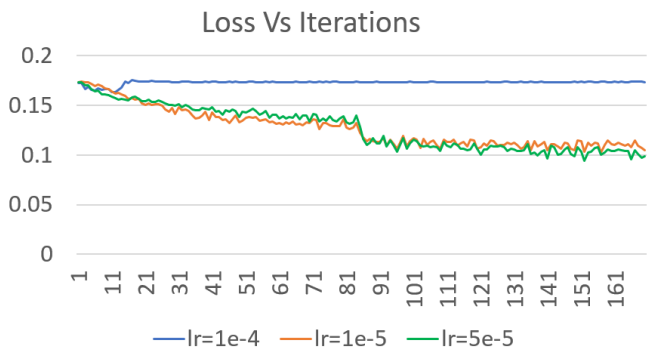

Figure 6: Loss vs. iteration with various Learning Rates

Based on the baseline model, we tried three initial learning rates, 1e-5,5e-5 and 1e -4. The results are shown in Figure 4. A learning rate of 1e-4 is represented by the blue line, and the loss was found to maintain a flat line which indicates that the learning rate is too big. For the learning rates of 1e-5 and 5e-5, both losses are decreasing with increasing iterations. However, a learning rate 5e-5 achieves the least loss and delivers the best performance. The unit of the iteration is 1K samples. In the middle of the figure, there is a big drop because of a programmed change in learning rates.

5.1.2 Freeze BERT encoder layers

Freeze Layers	Dev Accuracy
No Freeze layers	62.2%
Freeze first 1 layer	61.2%
Freeze first 3 layers	60.4%
Freeze first 5 layers	57.9%
Freeze first 7 layers	55.8%

Figure 7: Accuracy vs. Number of freeze layers

We explored how the number of freeze layer impacts the performance. With the max sequence 320 and batch size 8 due to GPU memory limit, the results are shown in the above table. From the table, we know the performance is getting lower as the freeze number increases. And the performance gets much worse once the freeze layer number exceeds 3.

5.1.3 L2 regularization

In addition, we explored weight decay parameters with L2 Regularization coefficients of 0.1, 0.01, and 0.001. We got the corresponding performance, 62.1%, 62.2%, and 62.1%. Hence, the best performance was obtained with an L2 regularization of 0.01.

5.1.4 Batch size

Since the BERT is a huge network, we are using gradient accumulation technique to reduce the GPU memory requirement. We experimented three different batch size: 16, 24, 32, and 64. We started off with batch size of 32 for BERT base model training and we got 61.9% performance result; however, as we decreased the batch size to 24 and 16, accuracy increased a little bit to 62.3% and 62.5% respectively. Also when we increased the batch size to 64, the accuracy decreased. So it seems like batch sizes of 32, 24, and 16 falls into a good range to achieve the best model, but batch size 16 yielded the best performance, which is 0.22% higher than batch size 24 and 0.65% higher than batch size 32.

5.1.5 Max sequence length

Max sequence length is a hyper-parameter that plays a very critical role in the BERT based models. As the max sequence length increases from 320 to 450, the performance increased gradually by 1.5%. We first tested on max sequence length equal to 320 on BERT base model and it resulted in 61.1% accuracy. As we increased the max sequence length, the performance of the model improved. For max sequence lengths 380, 420, and 480, we got accuracies 61.3%, 61.8%, and 62.6%. Thus, we concluded that the larger max sequence length can aid increasing the accuracy of our model. Intuitively, it is true since the length of some passages exceeds 400 words. However, the model size is proportional to max sequence length and leads to more memory requirement. We could not explore more length due to the resource limit.

5.2 Data Augmentation

We augmented the overall training set by 10%, exactly half from RACE-middle and another half from RACE-high. We only applied the augmentation on the randomly selected passages and conducted the same preprocess by concatenating them to original questions and options. In EDA, there is a parameter α which indicates the percentage of words changed in each passage by each augmentation technique. For example, if α is set to 0.1, our EDA implementation would change 10% of words to their synonyms, random word insertion and swap will occur on 10% of the total number of words, and 10% of the words will be randomly deleted. We experimented with this parameter set to 0.1, 0.2, 0.3, and 0.5, and each parameter resulted in 65.2%, 65.6%, 64.9%, and 64.9% accuracy respectively. Thus, $\alpha = 0.2$ gave the best performance although the difference in accuracy wasn't very large. However, compared to the accuracy obtained by large BERT model without EDA, 10% of addition of training data augmented with augmentation parameter 0.2 can improve the model accuracy by 0.6%.

Therefore, we found the best performance reached 65.6% as shown in fig. 5 and compared to the original RACE dataset, it's about 0.6% of gain. There are two possible reasons for not gaining so much. One is we might need to tune more hyper parameter of EDA, using different ratios for different operation, synonym replacement should be less noisy but other 3 operations add more noises. The other reason is Easy Data Augmentation prefers to work well on a small dataset, but our RACE includes 6,400 passages and is a big dataset comparatively.

5.3 Deep Comatch Network

We tried to apply some new attention ideas to Multiple Choice Reading Comprehension on Race. The first idea which was explored was the application of an additional Attention over Attention layer to the base BERT model.

However, a poor accuracy of 25.1% was achieved, which is attributed to the fact that the self-attention layer of BERT conflicts with the AoA layer when both layers are combined in the encoder layer.

Then we turned to designing Deep comatch Network inspired by (11). We created multiple layers of Deep Comatch Network following BERT hidden output. In Deep Network, we designed a novel comatch attention layer following 5 layers of coattention after BERT encoder. Our DCN benefits from the question-aware passage representation and answer-aware passage representation through the attentions between Passage/Question and Passage/Answer. However, due to GPU resource limit, we trained DCN with 3 layers and got 66.2% accuracy in single mode of DCN.

5.4 Ensemble

After exploring all of these different models, we created the ensemble model as we initially planned because we were convinced that an ensemble learning can help to achieve the best performance. We used random search to weigh the accuracies between Base and Large BERT, EDA and DCN while keeping the best performance of the model. Finally, we got 67.9% accuracy in ensemble model as shown fig. 5. In other words, we gained a total of 1.7% increase in accuracy compared to DCN. This result ensured how an ensemble learning with appropriate weights can improve the machine learning results.

6 Analysis

The below figures show the comparison between DCN and Base model. DCN performs better than Base in both high and middle datasets. DCN performance is even much better in long and complex passage in high dataset.

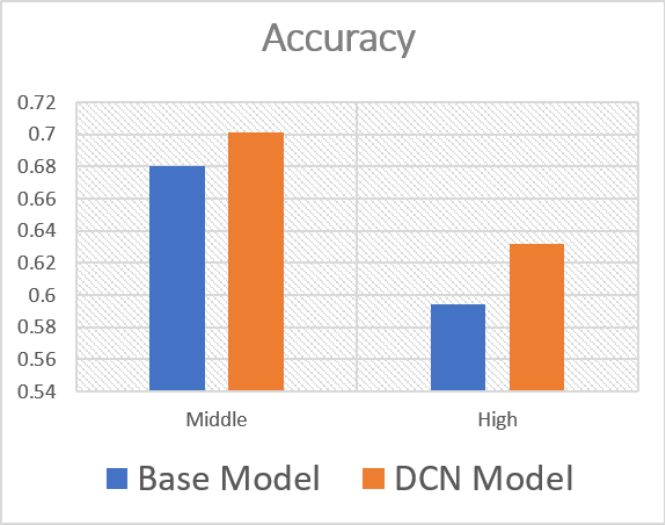

Figure 8: Base Accuracy vs. DCN Accuracy

Crucially, upon analysis of the questions which BERT had answered wrongly, it was discovered that the model was unable to perform simple numerical reasoning. For example, in the following mislabeled case, the first option is correct. Although the predicted (second) option has higher probability, the answer should have been 45-18=27 years ago.

Question: *From the passage we can know that _____*
Options: *"Stewart was depressed at one time", "Stewart lost his left arm 22 years ago", "Stewart never complained about the unfairness of life", "Stewart was persuaded to kayak through the Grand Canyon"*
Passage: *For most of his life, the 45-year-old man has lived with only his right arm. He lost his left arm ... when he was 18. He became a bitter young man, angry at the unfairness of what had happened, and often got into fights.*

The following samples shows DCN correctly predicted the answer "Today's News," but the baseline model failed. It means DCN could more accurately learn what the news means and its relationship with the time.

Question: *In which part of a newspaper can you most probably read the passage?*
Options: *"Today's News", "Culture", "Entertainment", "Science"*
Passage: *"Three cattle farms in Andong... were... infected with... disease, Nov.2 2010, Thursday... On Monday, the disease... detected on two pig farms in Andong... The laboratory tests today showed that all three cattle farms were infected with the disease," an official said. Two newly infected cattle farms... indicating the disease will likely continue... has culled more than 33,000 animals... No suspected cases...*

7 Conclusion/Future Work

We presented a novel Deep Comatch Network by adding several comatch attention layers after BERT hidden output base on Dual Comatch Network. In the end, we finally got 66.2% accuracy in a single model and 67.9% in an ensemble model. Therefore, among the four algorithms that we tested, the ensemble model performed the best.

Compared to the state of the art performance 69.7%, the reason we are a little behind might be that we had to freeze several layers and could not explore with a higher sequence length due to the GPU resource limit. Even the distributed training with multiple Tesla V100 GPUs faced a problem of the memory allocation, so we couldn't explore further although we found using larger number of max sequence was helpful to build a more accurate model.

Our models and datasets were too large that it took a very long time to train and test each experiment. In the future, with more time and GPU memory, it would be interesting to explore more deep layers of DCN with bigger max sequence length. We would also like to spend more time with hyper-parameter tuning EDA, DCN, and ensemble models and to try training on another dataset first to conduct a transfer learning on RACE dataset.

8 Contributions

All three authors equally contributed to the tuning, debugging, and design of the project. Kegang mainly worked on the architecture of the network and baseline model training. Jing Jie focused on the Attention over Attention design. Jung Youn worked on data preprocess and augmentation, and ensemble model training. Kegang and Jung Youn did majority of the work of constructing the poster and synthesizing the results for the final report. Jing Jie presented the work.

9 Code

All code can be found in https://github.com/tosmaster/bert-race.

References

Chadha, H., & Sood, R. (n.d.). *Bertqa - attention on steroids.* Retrieved from https://github.com/ankit-ai/BertQA-Attention-on-Steroids

Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., & Hu, G. (2016, Jul). Attention-over-Attention Neural Networks for Reading Comprehension. *arXiv e-prints*, arXiv:1607.04423.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018, Oct). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, arXiv:1810.04805.

Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017, Apr). RACE: Large-scale ReAding Comprehension Dataset From Examinations. *arXiv e-prints*, arXiv:1704.04683.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016, Jun). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, arXiv:1606.05250.

Richardson, M., Burges, C. J., & Renshaw, E. (2013, October). MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 193–203). Seattle, Washington, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D13-1020

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., & Suleman, K. (2016, Nov). NewsQA: A Machine Comprehension Dataset. *arXiv e-prints*, arXiv:1611.09830.

Wei, J. W., & Zou, K. (2019, Jan). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv e-prints*, arXiv:1901.11196.

Xu, Y., Liu, J., Gao, J., Shen, Y., & Liu, X. (2017, Nov). Dynamic Fusion Networks for Machine Reading Comprehension. *arXiv e-prints*, arXiv:1711.04964.

Yin, W., Ebert, S., & Schütze, H. (2016, Feb). Attention-Based Convolutional Neural Network for Machine Comprehension. *arXiv e-prints*, arXiv:1602.04341.

Zhang, S., Zhao, H., Wu, Y., Zhang, Z., Zhou, X., & Zhou, X. (2019, Jan). Dual Co-Matching Network for Multi-choice Reading Comprehension. *arXiv e-prints*, arXiv:1901.09381.

Zhu, H., Wei, F., Qin, B., & Liu, T. (2018). Hierarchical attention flow for multiple-choice reading comprehension. In *AAAI conference on artificial intelligence.*