SQuAD Model Comparison

Jiehan Zhu*
Department of Computer Science
Stanford University
jzhu01@stanford.edu

Abstract

This project explores and compares different NLP architectures performance on The Stanford Question Answering Data(SQuAD). The basest performances model is BERT based model with addition layer on original BERT SQuAD implementation. This model's F1 score and Exact Match score is 73.77. But this model takes long time to train each epoch, and does not improve after trained 3 epochs. QANet based model provided slightly lower F1 and EM score, but continuous improving with training.

1 Introduction

Recent development in Deep Learning in Nature Language Processing (NLP) has reach better than human performance on multiple NLP tasks. Recurrent neural networks(RNN), long short-term memory(LSTM)[2] are widely use NLP model. The sequence nature of RNN and LSTM limited the neural network model to parallel calculation, and requires long training time for large dataset. This leads to application of Convolution Neural Networks(CNN) and Attention layer that enable faster training and smaller model with similar or higher preference. In 2018, the application of Bidirectional Encoder Representations from Transformers(BERT) has reached new state of art in various tasks, including reading comprehension, and encourage a multitask model structure.

Reading comprehension or question answering task has gained great popularity in NLP task due to its wide application. Stanford Question Answering Dataset (SQuAD) is one of most widely used reading comprehension dataset and base of data used in this paper. The official SQuAD 2.0 dataset consists of 100,000 questions posed by crowd-workers on a set of Wikipedia articles. About half of questions are answerable, and the answer to answerable every question is a segment of text, or span, from the corresponding reading context.

This paper focused on comparing architectures with different complexities level on performance matrix, training time, and memory requirement. This is important in industry or research application because it shows the balance of model performance and cost, as well as sensitivity models to change in layer and hyper parameters. The results are useful when try to apply these architectures to solve different problems , or transfer learning to different data sets.

2 Approach

Baseline model in this project is a BiDAF model without character level embedding. The three architectures are compared and analysis in addition to baseline model. The smallest and largest architectures have more than one models with slight layer differences. Baseline mode is Bidirectional Attention Flow for Machine Comprehension(BiDAF)[5] without character level embedding. The smallest set of models are BiDAF based model with various embedding layers; the median

*Ameriprise Information Management Department, Ameriprise Financial (jiehan.zhu@gmail.com)

Preprint. Work in progress.

size model is based on Combining Local Convolution with Global Self-Attention for Reading Comprehension(QANet)[8], and the largest model is a BERT[1] based model with additional layers on original SQuAD specific implementation in the paper.

2.1 Bidirectional Attention Flow Based Models

The baseline model has five block, Embedding Layer, Encoder Layer, Attention Layer, Model Layer and Output Layer. The Embedding Layer project the word embedding from GloVe embedding with a Fully Connected layer with H output neurons, and two Highway layers with H output neurons as well. The Encoder Layer, Attention Layer, Model Layer and Output Layer are implemented based on BiDAF[5].

- EMBEDDING LAYER The embedding layer project the word embedding from GloVe embedding with a Fully Connected layer with H output neurons, and two Highway layers[6] with H output neurons as well. This output a vector of H for each word in question and answer.
- ENCODER LAYER The encoder layer uses a bidirectional LSTM without sharing parameters between left-to-right and right-to-left LSTM. Thus, the output for this layer is $2H$ for each word.
- ATTENTION LAYER The two Attention layer allow in this part allow the attentions flow both way, Context-to-Question (C2Q) Attention and Question-to-Context(Q2C) Attention. The output of this layer had dimension of $8H$.
- MODEL LAYER The model layer have same structure as encoder layer, a bidirectional LSTM without sharing weights.
- OUTPUT LAYER The output layer applies a bidirectional LSTM to the modeling layer outputs, and Fully Connected layers with softmax to product distribution probability of answer start and end positions.

2.1.1 Embedding Layer

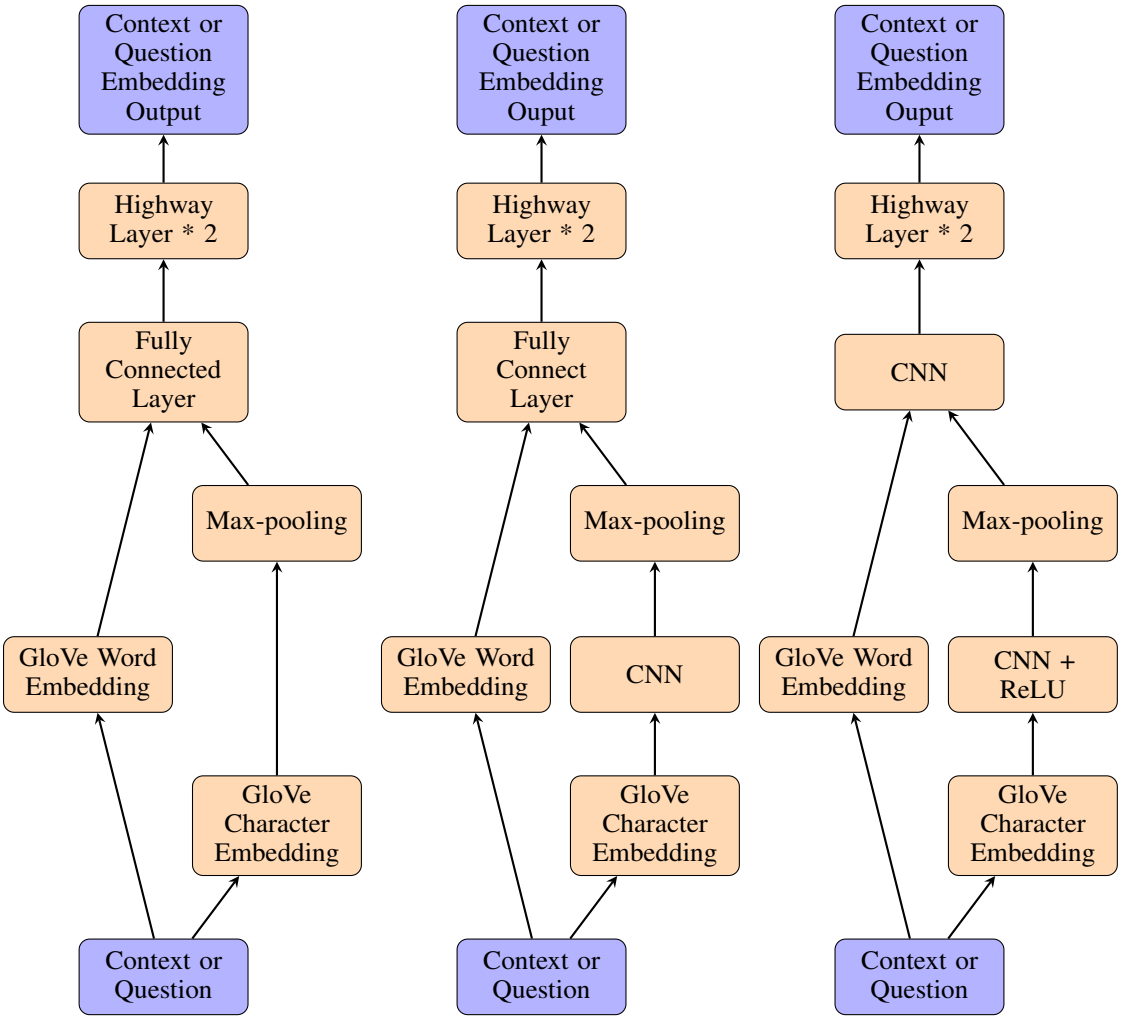
Though the model Encoder, Attention and Model layer is similarly with BiDAF, the Embedding layer are different from the original paper implementation. The embedding layer take in to context, question as a word string, look up words' index in GloVe word embedding[4] $w_1, w_2, w_3, ..., w_n \in \mathbb{R}$, and append pretrained GloVe word embedding to convert context and question to $c_1, c_2, c_3, ..., c_{n_1} \in \mathbb{R}^D, n_1 = 400$ and $q_1, q_2, q_3, ..., q_{n_2} \in \mathbb{R}^D, n_2 = 50$. Similar to word embedding, character level embedding split each word in context, question as a character string $s_1, s_2, s_3, ..., s_m \in \mathbb{R}^D$ and then and append pretrained GloVe character level embedding[3]. Both word and character level embedding used in this project are 300 dimensions, $D = 300$. The max number of character to keep from a word is 16, $m = 16$.

Models with fixed or training character level embedding has been tested. Fixed GloVe pretrained character level embedding provided better development dataset result, and the later results in the experiments section are based on model with word and character embedding are not trained during the training process. Context and question are calculated separately in this layer.

The character level embedding $s_1, s_2, s_3, ..., s_m \in \mathbb{R}^D$ are then passes though following three architecture and lead to three different model. Detailed flow charts of three embedding layer are shown in the appendix.

- 1^{st} CHARACTER EMBEDDING STRUCTURE The model applies max-pool layer on character level embedding to create 16 neurons for each word, merger with word embedding. Word and character level embedding vector are trained with dropout rate of 0.1, a Fully Connected layer with output of 100 neurons, and then two Highway layer[6] with 100 neurons.
- 2^{nd} CHARACTER EMBEDDING STRUCTURE The model adds a 1D Convolutional Neural Network (CNN) on character level embedding before max-pool layer. The CNN layer has a kernel size of 5, 16 filters, and padded size of 2. The same layers as the 1^{st} Character embedding model are applies to output of CNN.
- 3^{rd} CHARACTER EMBEDDING STRUCTURE The model used a CNN layer as prior model on character level embedding, but it uses 100 filters, instead of 16. It also adds dropout on the output

of CNN with dropout rate of 0.1, and applies ReLU activation on the output before max-pool layer. This also use to with 100 filters on character level embedding before max-pool layer. The output from max-pool layer are merged with word embedding, and a 1D CNN with kernel size of 5, padding size of 2, and output channel of 100. Two Highway layer same as prior models are applies at the end of embedding layer.



Bidirectional Attention Based Model Embedding Layer Flow Charts

2.2 QANet with Pretrained Character Level Embedding

Built on Transformer[7], QANet[8] used only CNN and self-attention layer in order to train parallel on GPU and obtain much faster training process.

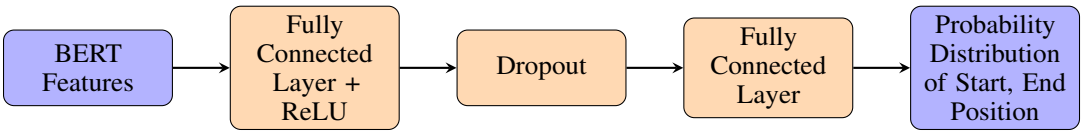
The QANet based model used in this project are slightly different from the implementation in original paper. The original QANet used random initialization 200 dimension character level embedding, while this model uses pretrained 300 dimension GloVe character level embedding.

2.3 BERT Based Models

BERT base model is a fine tuning implementation of BERT base model with lower cased text before WordPiece tokenization is implemented as suggested in BERT paper[1] on SQuAD.

In addition, BERT with additional layer model, a model with additional layers specific for SQuAD has been trained and compares. A Fully Connected layer with H hidden units, ReLU activation, and

dropout layers have been added before question answering output linear layer that calculating start and end position probability distribution.



BERT SQuAD Model with Additional Layers

3 Experience

3.1 Data

This report use CS224N default final project's data split of SQuAD dataset. Each example in training and development data sets has (context, question, answer) triples, and each example in testing data set has (context, question) triples. The model take context and question as input, and try to output the start and end position of the answer.

The train, development, and test data set have 129,941, 6078, and 5915 examples, respectfully. One context has multiple questions related to it; and one question has only one answer in the train dataset, but might have slightly different answers in development dataset.

> **Context**: Following the disbandment of Destiny's Child in June 2005, she released her second solo album, B'Day (2006), which contained hits "Déjà Vu", "Irreplaceable", and "Beautiful Liar". Beyoncé also ventured into *acting*, with a Golden Globe-nominated performance in Dreamgirls (2006), and starring roles in The Pink Panther (2006) and Obsessed (2009). Her marriage to rapper *Jay Z* and portrayal of Etta James in Cadillac Records (2008) influenced her third album, I Am... Sasha Fierce (2008), which saw the birth of her alter-ego Sasha Fierce and earned a record-setting six Grammy Awards in 2010, including Song of the Year for "Single Ladies (Put a Ring on It)". Beyoncé took a hiatus from music in 2010 and took over management of her career; her fourth album 4 (2011) was subsequently mellower in tone, exploring 1970s funk, 1980s pop, and 1990s soul. Her critically acclaimed fifth studio album, Beyoncé (2013), was distinguished from previous releases by its experimental production and exploration of darker themes.
>
> **First Question**: After her second solo album, what other entertainment venture did Beyoncé explore?
>
> **Answers**: acting
>
> **Second Question**: Which artist did Beyoncé marry?
>
> **Answers**: Jay Z

The length of context various in the data set, most of the context have 50 to 250 words, while some of the context have more than 600 words. Due to the memory restriction, the maximum number of word loaded for each context is limited at 400, and this only impact less then one percent of data. And the maximum number of word loaded for each question is limited at 50, which should not impact the model at all.

3.2 Pretrained Character Level Embedding verse Train Character Level Embedding

Random initialization and training character level embedding are tested in BiDAF and QANet based models. It uses 100 dimensions character level embedding with 94 characters.

GloVe pretrained character level embedding is used without future training in the same models. Since this embedding are trained on a larger data set with longer training time, this 300 dimensions embedding with 94 character are frozen during the training process.This should lead to better generalization cross train, development, and test data sets.

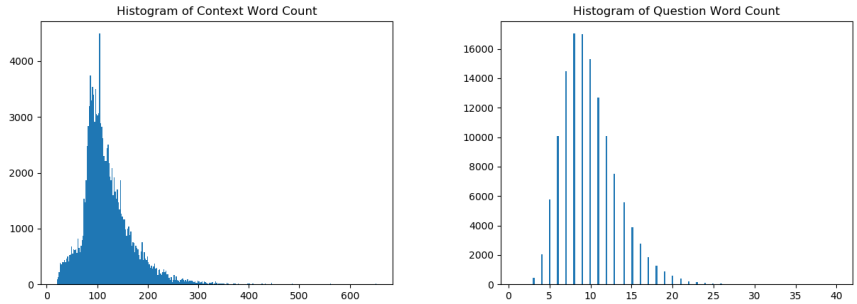


Figure 1: **Left**: Histogram of number of word in context in train dataset. The 99th percentile of number of word in context is 285 in train dataset. **Right**: Histogram of number of word in question in train dataset. The maximum number of word loaded for question in less than 40.

3.3 Batch Size

This project compared models that are trainable on single GPU with 8GB GPU memory. This is consistent with resource that is available to this project and to a most average data scientist team. For example, this limited the project to use BERT base model instead of BERT Large model that require 12 to 16 GB GPU memory to load.

In addition, depended on number of parameters in each model, the batch size is also limited by model. The batch size of each model are select by largest of batch size as long as the models run on VM does not have out of memory issue. The BiDAF based models are trained with batch size of 64, QANet based models are trained with batch size of 8, and BERT based model are trained with batch size of 6.

3.4 Number of Epochs

In addition to the RAM limitation, this comparison also limit on the time of training for each model. The larger model require more RAM, so it can only run on a smaller batch size, and takes longer to train one epoch. In addition, the model with more parameter take longer to calculate each forward and backward propagation,

The baseline and QANet models are train on 30 epochs, BiDAF based models are trained on 20 epochs. BERT based original model is trained with 3 epoch, and BERT with additional layer is trained on 4 epoch.

3.5 Hidden Size and Number of Heads

All hidden size are 100 in BiDAF based models other than embedding layer that are detailed explained above.

Consistent with the original implementation, the model uses multi-head attention with 8 heads, and hidden size and the convolution filter number are all 128. In addition, the embedding layer are trained with dropout rate of 0.1 for both word and character level embedding, while original model is trained with 0.05 dropout rate in character level embedding.

BERT base model are uses multi-head attention with 12 heads, and hidden size of 768, same as original paper. BERT with additional layer model used 100 hidden size in the additional Fully Connected layer.

3.6 Evaluation Metrics

The best model are evaluated with F1 score on development data set, which is harmonic mean of precision and recall. In addition to F1 score, Exact Match(EM) score is also use to evaluate the models.

Table 1: Performance Summary

Model Name	Epochs	Train Time	Dev EM	Dev F1
Baseline	30	9 hours	58.58	61.88
Baseline with L2	30	7.5 hours	57.47	60.78
BiDAF with 1^{st} char embed	20	4.5 hours	57.64	60.63
BiDAF with 2^{nd} char embed	20	6 hours	59.23	62.26
BiDAF with 3^{rd} char embed	20	5 hours	57.03	60.27
QANet with GloVe char embed	30	31 hours	64.24	67.84
Bert Base	3	17 hours	69.84	73.09
Bert with add. layers	3	11 hours	70.32	73.77

Answerable questions and unanswerable questions in development data set are tracked separately in BERT based model to provide a better understanding of model training process.

3.7 Other Training Detail

Baseline model is trained with and without L2 weight decay, and the decay rate used is 0.01. All other modes are trained without L2 weight decay.

Learning rate in BiDAF based models and QANet model are 0.5, and BERT based models have train with $5e-5$ for 3 epoch, and then trained with $2e-5$.

Bert based models are trained with Adam optimizer with linear learning rate warm up on 0.1 of training data, while other models use Adadelta optimizer.

4 Analysis

Overall, the BERT with addition layers model results in best performance with F1 score of 73.77 and 72.95 on development and test data sets. The deeper the Neural Network and the more parameters the model have, the better the performance, and the longer the training time.

4.1 Regularization

The baseline with L2 weight decay model has no significant improvement on model performance from baseline model, but stable at similar EM and F1 scores. This is as expected since regularization limits the degree of change in parameters during training.

4.2 Training Time

Large models like BERT takes much longer time to train each epoch. BERT based models that have no RNN and LSTM took about 3 hours to train each epochs, while BiDAF based model only take half hour. This majorly due to the complexity of calculation in each step, and the smaller batch size.

Baseline model is finished 20 epochs training in less than 5 hours with best F1 and EM score of 60.23 and 56.90 on development data set; additional 10 epochs training that took an other 2 hours only slightly improved the model performances, resulting F1 and EM score of 61.88 and 58.58.

4.3 Embedding Layer

The best BiDAF based model is using 2^{nd} character level embedding structure. It reaches F1 score of 62.26 on development dataset with 20 epochs training. Comparing with the 1^{st} BiDAF with character level embedding models, the 2^{nd} model has additional CNN layer on character level embedding; comparing with 3^{rd} model, the 2^{nd} model uses Fully Connected layer instead of CNN layer on merged embedding output of word and character level embedding, which have more parameters.

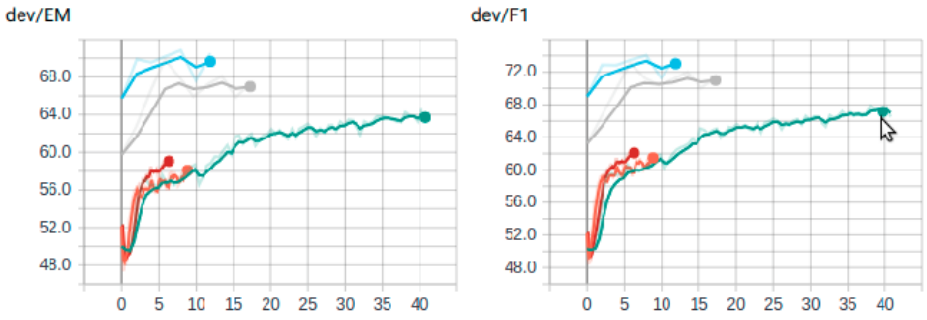


Figure 2: Exact Match and F1 score on development data set during by training hour. Red: BiDAF with 2^{nd} character embedding structure model, Blue: BERT with addition layers model, Gray: BERT base model, Orange: Baseline model, Green: QANet with pretrained character embedding model,

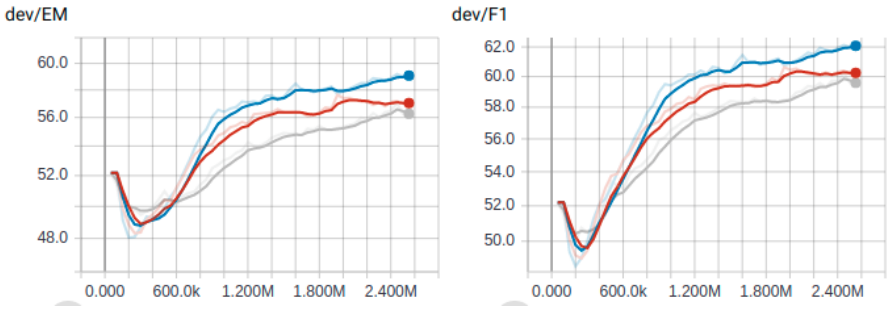


Figure 3: Exact Match and F1 score on development data set during 20 epochs training by step. Red: BiDAF with 1^{st} Character Embedding Structure, Blue: BiDAF with 2^{nd} Character Embedding Structure, Gray: BiDAF with 3^{rd} Character Embedding Structure. The 3^{rd} model consistently has better EM and F1 score during the training process, and stabilize at a higher score.

4.4 Pretrained verse Non-Pretrained Contextual Embedding

QANet with pretrained character level embedding has the highest F1 and EM score in non-pretrained contextual embedding(Non-PCE) models in this project. It results F1 and EM score of 64.24 and 67.84. But this model took a very long time to train, 31 hours for 20 epochs. The development data set's F1 and EM scores are still slightly trending up after 31 hours train, thus, with additional resource, the model might lead to even higher performance metrics.

One the other hand, pretrained contextual embedding(PCE) models, BERT base model and BERT with addition layers model, have much higher initial F1 and EM scores. The BERT model with additional layers at first evaluation point, half epochs, has F1 and EM score of 68.34 and 65.43, and it reaches F1 score of 73.77 and 72.95 on development and test data set in 3 epochs training. The model takes very long time to train, 17 hours for 3 epochs in this case; and the model does not response as well with more training, an other 2 epochs training results in decreasing F1 and EM score on both development and test set.

PCE model starts at higher initial F1 and EM score, since it used pretrained weight from other task or other dataset, and PCE reaches it's best performance very quickly.

5 Conclusion

Largest pretrained contextual embedding model, a BERT based model with additional layers for SQuAD, provides best performance among the models that have been tested., F1 score of 73.77 and 72.95 on development and test sets. Median size model QANet provides second best and BiDAF based have lowest performances. But BERT take very long time to train for each epoch, and reach

performance cap within 3 epoch. On the other hand, Non-PCE models have lower initial performance, but can improve with more train epochs.

Using pretrained character level embedding, regularization improves model performance slightly, but not a significant amount. Model size and the depth are the determination factors of model performance.

Best model requires a long training time and large GPU RAM. This could be a bottle neck when apply the-state-of-art model in industry.

Acknowledgement

Thanks Professor Chris Manning as well as CS224N teaching assistant team who provided this wonderful course on NLP and project on SQuAD, which helps me understand fundamental of NLP and inspire me to explore industry application cases in my career.Secondly I would also like to thank Microsoft for credit on Azure Virtual Machines that makes this project possible.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805, Oct 2018.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[3] minimaxir. A repository containing 300d character embeddings derived from the glove 840b/300d dataset, and uses these embeddings to train a deep learning model to generate magic: The gathering cards using keras. `https://github.com/minimaxir/char-embeddings`, 2017.

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

[5] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *arXiv e-prints*, page arXiv:1611.01603, Nov 2016.

[6] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway Networks. *arXiv e-prints*, page arXiv:1505.00387, May 2015.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, Jun 2017.

[8] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv e-prints*, page arXiv:1804.09541, Apr 2018.