CS224N Project Report: Bidirectional Attention Flow and Self Attention Mechanisms for Machine Comprehension

Jervis Muindi
Department of Computer Science
Stanford University
353 Serra Mall, Stanford, CA 94305
jmuindi@stanford.edu

Richard Ruiqi Yang
Department of Computer Science
Stanford University
353 Serra Mall, Stanford, CA 94305
richard.yang@stanford.edu

Abstract

In this project, we tackle the challenge of building neural network models that can perform the task of answering questions on SQuAD (Stanford Question Answer Dataset). We explore the effects of bi-directional attention flow (BiDAF) and self attention mechanisms on this dataset, in comparison to a baseline model that only leverages a basic attention mechanism. We report that both techniques show an improvement over our baseline model, and we analyze the results in this report.

1 Introduction

Reading comprehension is a task whereby an entity is quizzed on the knowledge or facts contained in a passage they read in order to judge their level of understanding. In the context of natural language processing (NLP), reading comprehension can serve as an important foundational building block for larger and more complex systems, such as a question-and-answer dialogue system.

The Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset published in 2016[5] to help spur advances in machine understanding and reasoning of text. This dataset utilizes articles from Wikipedia as passages, and human crowd-sourced workers identified questions and potential answers from these passages. Within this dataset, the passage, question, and expected answer are treated as tuples.

1.1 Problem Definition

Formally, we model the machine reading comprehension task as: given a passage **P** and a question query **Q**, the goal is to find an answer **A** that satisfies the question query. In the specific case of the SQuAD dataset, the answer A is guaranteed to be a subset of the passage. That is, the answer to question is always to be found within the bounds of the passage.

1.2 Evaluation Criteria

There are two primary criteria that are commonly used to evaluate performance on the reading comprehension task. The first is the exact match (EM) score, and the second is the F1 score.

1.2.1 EM Score

An Exact Match (EM) score is a strict binary measurement of whether the predicted answer exactly matches what is in the expected/ground truth data. For instance, assuming a ground truth answer of

"Socal", a predicted answer of "So Cal" would be considered incorrect because it does not exactly match what is in the expected answer response due to the extra space in between.

1.2.2 F1 Score

F1 score is a measure of the model performance that is less strict as compared to the EM score. The F1 metric is computed as the harmonic mean of precision and recall [3]. Mathematically, the F1 score is computed as:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Precision measures how much of the computed answer matches the ground truth. Precision would be 100 percent when the answer is a subset of the ground truth answer. Recall, on the other hand, evaluates how many of the words in the ground truth answer the system was able to include. Any excluded words are penalized and result in a lower score. For example, suppose the ground truth answer to a question is "Stephen Hawkings" but the computed answer by a model only included "Hawkings". That model would have 100 percent precision but only 50 percent recall since "Stephen" was mixing. The final F1 score would be 66.67%

2 Baseline Model

Our initial approach to the reading comprehension task is a baseline model that was previously implemented. The model consists of three primary components. First, we use a recurrent neural network (RNN) model to encode the input passage and question into a set of hidden states. The input to this model is a SQuAD tuple of (context, question, answer), represented by 100-dimensional pre-trained GloVe embeddings. The context and question embeddings are input to a bidirectional gated recurrent unit (BiGRU) or a long-short term memory (LSTM) layer to encode the information as hidden states.

Next, the RNN embeddings are served to a basic attention layer. This attention layer generates an attention distribution from the context and question embeddings by taking the dot product, and applying a masked softmax function. This is further converted into an attention output vector, through a weighted sum of the question hidden vector produced by the RNN encoder.

Finally, the output layer is a rectified linear unit (ReLU) non-linearity, which computes a score from a blended representation of the attention output and context hidden states. We then take the softmax of the scores to produce a probability distribution of the answer.

In order to train this model, we use a cross-entropy loss function and optimize the parameters of the model using Adam.

3 SQuAD Model Improvements

The baseline model performs fairly well on the development set of the SQuAD dataset. However, we noted several key improvements to this model for an increased performance. Specifically, we pursue attention mechanisms that are more complicated than a basic dot product between context and question hiddens states. The two modifications we found that were shown to perform well are bi-drectional attention flow (BiDAF) and self attention.

3.1 BiDAF

BiDAF was introduced by Seo et. al [6] to expand on the basic attention mechanism, which heavily focuses on a small region of the context passage, and flows in a uni-directional attention. In BiDAF, the key concept is that attention should flow in both directions, from question to the passage and vice-versa (from the passage to the question).

The first step in our BiDAF model is to update the attention layer to use the attention mechanism from [6]. Briefly, this model involves the following computations:

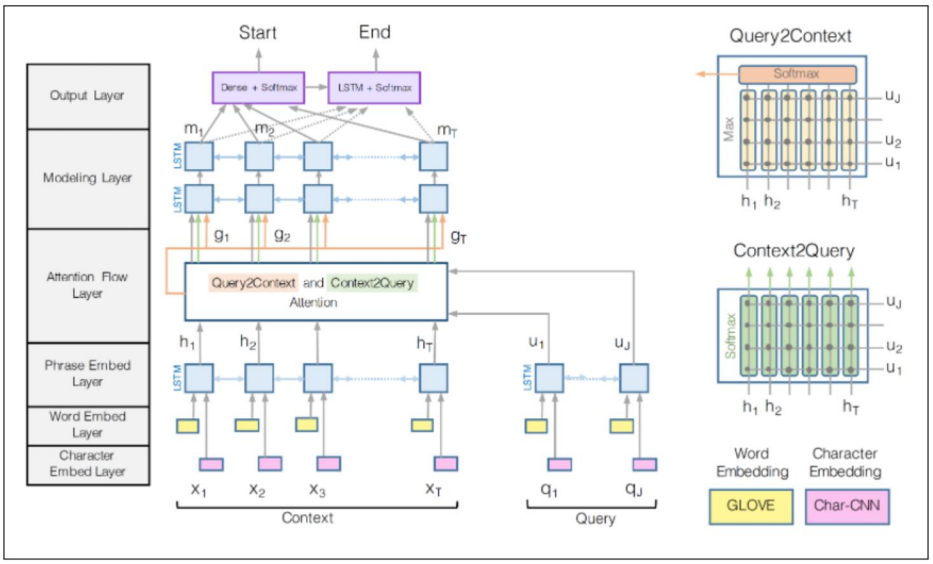

Figure 1: BiDAF Overall Model Network Architecture. Image source: BiDAF Paper[6]

For consistency with the code implementation, context is another name for the passage and it's abbreviated with letter c. First, we compute similarity matrix between the context and question hidden states:

$$\boldsymbol{S}_{ij} = \boldsymbol{w}_{sim}^{T}[\boldsymbol{c}_i; \boldsymbol{q}_j; \boldsymbol{c}_i \circ \boldsymbol{q}_j] \in \mathbb{R}$$

Next, we calculate the attention from Context-to-Question using:

$$\alpha^i = softmax(\boldsymbol{S}_{i,:}) \in \mathbb{R}^M \forall i \in \{1, ..., N\}$$

$$\boldsymbol{a}_i = \sum_{j=1}^{M} \alpha_j^i \boldsymbol{q}_j \in \mathbb{R}^{2h} \forall i \in \{1, ...N\}$$

For the attention to flow in both directions, we also compute the Question-to-Context attention, using:

$$\boldsymbol{m}_i = max_j \boldsymbol{S}_{ij} \in \mathbb{R} \forall i \in \{1, ..., N\}$$

$$\beta = softmax(\boldsymbol{m}) \in \mathbb{R}^N$$

$$\boldsymbol{c}' = \sum_{i=1}^{N} \beta_i \boldsymbol{c}_i \mathbb{R}^{2h}$$

Finally, the output is the combination of attentions and hidden state representations:

$$\boldsymbol{b}_i = [\boldsymbol{c}_i; \boldsymbol{a}_i; \boldsymbol{c}_i \circ \boldsymbol{a}_i; \boldsymbol{c}_i \circ c'] \forall i \in \{1, ..., N\}$$

The output blended representations are then fed to the same output layer as our baseline model. While replacing our baseline model's attention layer with the BiDAF attention layer showed improvements in the F1 and EM scores on the development set, the authors of the BiDAF paper also use an RNN encoder (called the "modeling layer") following the attention layer to scan for contextual information about the word with respect to the context paragraph and query. We implement two models, one with the RNN modeling layer and one without. The comparisons are shown in the results section.

3.2 Self Attention

The other model we looked at is self attention. The idea here is to extend the basic attention that is already included in the baseline model and mix it up with some self attention. For self attention, as described in section 5.1.3 of project description [1], we have some $v_1 ... v_N \in \mathbb{R}^l$ from previous

basic attention layer where where v_i matches a context location and we want each v_i to attend to all the other ones. Mathematically, we can express this as:

$$\boldsymbol{e}_i^j = \boldsymbol{u}^T tanh(\boldsymbol{W_1 v}_j + \boldsymbol{W}_2 \boldsymbol{v}_i) \in \mathbb{R}$$

$$\alpha^i = softmax(\boldsymbol{e}^i) \in \mathbb{R}^N$$

$$\boldsymbol{a}^i = \sum_{j=1}^{N} \alpha_j^i \boldsymbol{v}_j \in \mathbb{R}^l$$

$$SelfAttentionoutput = \{\boldsymbol{h}_1, ..., \boldsymbol{h}_N\} = biRNN(\{[\boldsymbol{v}_1; \boldsymbol{a}_1], ..., [\boldsymbol{v}_N; \boldsymbol{a}_N]\})$$

In above, $\boldsymbol{u}, \boldsymbol{W}_1, \boldsymbol{W}_2$ are trainable parameters, the first being a weight vector and the rest weight matrices. l is the size of the hidden dimensions output from previous (basic) attention layer and our case it is 2h.

4 Experiments

Below is a discussion of our experiments in training the various models for the reading comprehension task on the SQuAD dataset. Note all numbers are normalized between 0-1. We first describe our experiments and the respective evaluation metrics that each model achieves. We summarize our model performances and offer insight on improvement at the end of this section.

4.1 Setup

We perform our model training on an Azure Standard NV6 instance. This has 6 CPU cores with 56GB of RAM. The cloud machine has an Nvidia Tesla M60 GPU with 8GB of available GPU memory. We use a default learning rate of 0.001 and a dropout probability of 0.15.

4.2 Baseline

We trained the provided baseline implementation using basic attention, which achieved an F1 score of 0.3974 and EM score of 0.2837 on the dev set. We performed hyperparameter tuning, and minor adjustments to the model, but did not observe a significant change in the performance after 10,000 iterations. Our first attempt at improving the baseline is using an LSTM instead of BiGRU for the RNN encoding layer. Then, we enabled the input GloVe word embeddings to be trainable along with the model. Finally, we experimented with the learning rate and dropout probability.

4.3 Vanilla BiDAF

For BiDAF, we implemented a basic version that replaces the baseline model's attention layer with the BiDAF attention layer. We used an LSTM for the input RNN encoding layer, as opposed to the original BiGRU, and this model achieved an F1 score of 0.4538 and EM score of 0.3355 on the dev set.

After analyzing intermediate performances results, we noticed that our models were performing much higher on the training set than the development set. We believed this was an issue with overfitting, so we added L2 regularization on the trainable weight vectors to prevent overfitting. We find that training BiDAF with L2 regularization constant of 0.0001 for the trainable weights in the model led to some additional small improvements in model performance. This model achieved an F1 score of 0.4704 and EM score of 0.3501 on the dev set.

4.4 BiDAF with Modeling Layer

Following the original implementation of BiDAF as described by [6], we add an LSTM encoding layer following the attention layer. This LSTM encoding layer attempts to model the contextual information of words in the context passage and question by further encoding the attention outputs.

Table 1: Comparison of our various models and performance on dev set

Model	Dev F1	Dev EM
Baseline	0.3974	0.2837
Baseline with HP tuning, LSTM encoder	0.4013	0.2898
BiDAF	0.4538	0.3355
BiDAF with L2 regularization	0.4704	0.3501
BiDAF with L2, LSTM encoder	0.4729	0.3534
BiDAF with L2, LSTM encoder, modeling layer	**0.6535**	**0.5030**
Self Attention model	0.5338	0.3804

4.5 Self Attention

We implemented the self attention layer from the R-Net paper, and swapped out our baseline model's attention layer with self-attention. We had trouble running the self attention model using default model parameter sizes and encountered Out of memory errors on the GPU while trying to train the model. This challenge arose due to the need to computed the N^2 values as specified in the first equation of the self attention model in section 3.2.

To deal with this limitation, we opted to reduce our context model parameter sizes. To find a reasonable value to select, we wrote code[2] to look at the distribution of passage length, question length, and answer length in the SQuAD training set. Analyzing the passage length distribution, plotted in figure 2 we find that 300 can be a reasonable cut off value and this is what we picked.

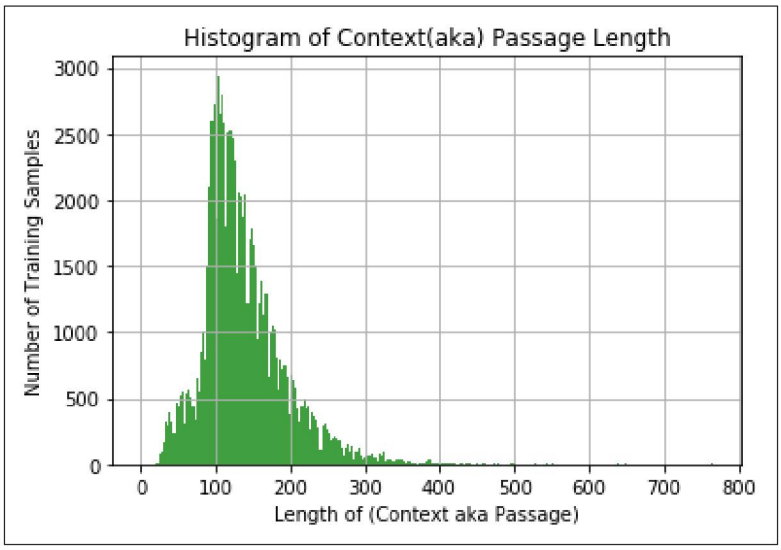

Figure 2: Histogram of Passage Length in SQuAD Training Data.

In addition, to a smaller context size of 300, we also used a batch size of 5 and a hidden layer size of 150. These reduced values resolved our out of memory errors for Self attention and enabled us to train the model. Looking at the performance for dev set, we find that self attention attained an F1 score of 0.5338 and an EM score of 0.3804.

4.6 Results and Discussion

Our top performing model is BiDAF with L2 regularization, using an LSTM encoder and a modeling layer. We were able to achieve the performance from the original BiDAF paper on the dev set, and this model boasts a 0.2561 F1 and 0.2193 EM improvement over the baseline. We report that the RNN modeling layer is crucial to the performance of BiDAF. Including this layer improved our F1 score by 0.1997 and EM score by 0.1675. This LSTM encoding layer attempts to model

the contextual information of words in the context passage and question by further encoding the attention outputs.

From these results, we also deduce that replacing the BiGRU in the RNN encoding layer with an LSTM unit does not impact performance significantly. While hyperparameter tuning is important and usually performed with grid search, we note here that the tradeoffs are insignificant for the performance of these models. Each model can take up to 15 hours to train, performing grid search is infeasible. The best improvement we noticed was a combination of adjusting the dropout rate to 0.25 and using 200 as the GloVe embedding size. We also note that introducing L2 regularization will help the model generalize better to the development set. Additionally, by allowing the GloVe word vectors to be trained in conjunction of the model, the dev performance actually decreases due to overfitting.

Additionally, we plot the training and dev F1 and EM scores across our baseline, self attention, and two BiDAF models in Figure 3. It's interesting to note that although self attention performs well on the dev set, it performs poorly (below the baseline) in regards to the training set, which may mean that our model suffers from underfitting. BiDAF with the modeling layer is an overall better performer on both training and dev sets.

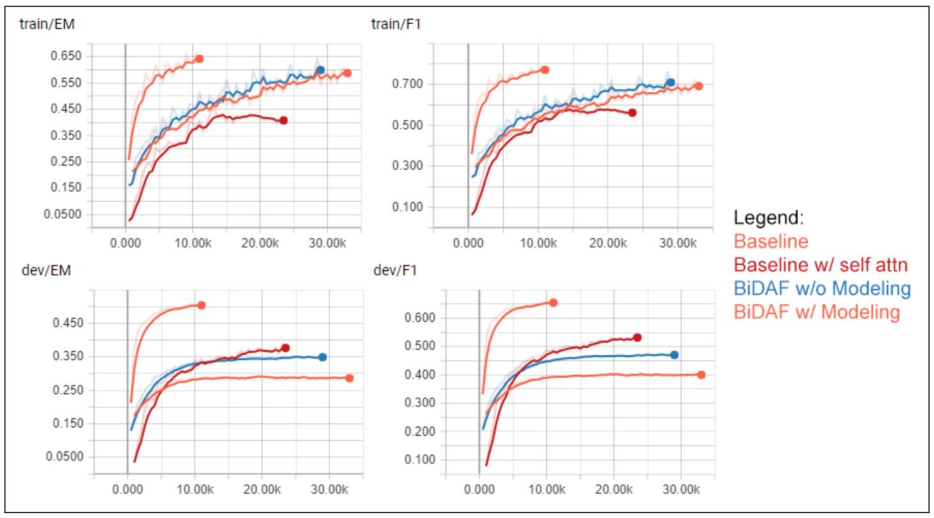

Figure 3: Comparison of training and dev F1 and EM scores across four models.

For additional experiments, we would have liked to use a combination of BiDAF and self attention. BiDAF offers a bi-directional attention flow while self attention offers a global representation of the entire passage. We propose two additional experiments (but could not complete because we ran out of Azure credit): a single model that uses BiDAF for the attention layer, with a self attention layer following using the BiDAF attention outputs, and an ensemble technique using several BiDAF models and several self attention models. Many top performing SQuAD models use an ensemble approach, where multiple models are trained separately. For prediction, a vote of the predicted answer is taken across all models.

5 Error Analysis

Beyond quantitative analysis of the performance of the models, we perform some qualitative analysis of examples where our model still fails. A note that a symbol of "_" surrounding a token means that the word is unknown.

5.1 BiDAF

One type of error that we observe with the BiDAF model is partial comprehension. This type of mistake is when the model is able to predict only part of the true answer correctly. For the sake of

the F1 score, these mistakes would count as partially correct. In the case of EM, they would not be. An example is shown in table 3. This may be due to attention being focused on a specific word, rather than a sequence of words.

Table 2: An example of a partial comprehension error from BiDAF

Passage: prime numbers have influenced many artists and writers . the french composer olivier messiaen used prime numbers to create _ametrical_ music through " natural phenomena " . in works such as la _nativit_ du seigneur (1935) and quatre tudes de rythme (_194950_) , he simultaneously employs motifs with lengths given by different prime numbers to create unpredictable rhythms : the primes 41 , 43 , 47 and 53 appear in the third tude , " neumes _rythmiques_ " . according to messiaen this way of composing was " inspired by the movements of nature , movements of free and unequal durations " . **Question:** in which etude of neumes rythmiques do the primes 41 , 43 , 47 and 53 appear in ? **True Answer:** the third tude **Predicted Answer:** tude

5.2 Self Attention

One of the issues that the self attention model we implemented has is dealing with unknown tokens. In the example given in table 3, we see that the true answer to the question includes the unknown word "merwede-oude".

Since this word is unknown the model was having trouble paying attention to it. For the word "maas" which is the second part of the answer, we see that the model correctly included it. This type of issue can be improved by augmenting the model with components that help it deal with words that it has never seen before.

Table 3: An example error from Self Attention model due to unknown words

Passage: before the st. elizabeth 's flood (1421) , the meuse flowed just south of today 's line _merwede-oude_ maas to the north sea and formed an _archipelago-like_ estuary with waal and lek . this system of n umerous bays , _estuary-like_ extended rivers , many islands and constant changes of the coastline , is hard to imagine today . from 1421 to 1904 , the meuse and waal merged further upstream at gorinchem to form merwede . for flood protection reasons , the meuse was separated from the waal through a lock and diverted into a new outlet called " _bergse_ maas " , then amer and then flows into the former bay hollands diep . **Question:** where did the meuse flow before the flood ? **True Answer:** merwede-oude maas **Predicted Answer:** south of today 's line merwede-oude maas

Another issue we have observed is the model getting confused and paying too much attention on the question in its output such that the answer is merely a regurgitation of the question. Table 4 has an example of a whom-type question which the model gets completely wrong.

6 Related Work

Since the SQuAD dataset was published in 2016, there have been numerous approaches proposed for solving the machine reading comprehension task on this dataset. The BiDAF model approach we pursued as an improvement is directly based on the work by MinJoon Seo et al.[6] in their paper titled Bidirectional Attention Flow for Machine Comprehension.

One of the differences between their model and ours is that ours only includes the bi directional attention piece. The full BiDAF model has other component pieces to it such as using character level embeddings beyond word vector. The character level embeddings are trained using a convolutional neural network and the character encoding is augmented with the traditional word vector encoding to obtain a hybrid representation. The character embedding are useful for making the model to still be robust even when it encounters unknown words.

Table 4: A regurgitation effect example error from Self Attention model

Passage: the church also holds that they " are equally bound to respect the sacredness of the life and well-being of the mother , for whom devastating damage may result from an unacceptable pregnancy . in continuity with past christian teaching , we recognize tragic conflicts of life with life that may justify abortion , and in such cases we support the legal option of abortion under proper medical procedures . " as such , two official bodies of the united methodist church are part of the religious coalition for reproductive choice 's governing coalition , the general board of church and society , and the united methodist women . the church cautions that " governmental laws and regulations do not provide all the guidance required by the informed christian conscience . " the church emphasizes the need to be in supportive ministry with all women , regardless of their choice . **Question:** the church holds that they are equally bound to respect the sacredness of the life and well-being of whom ? **True Answer:** the mother **Predicted Answer:** the church also holds that they "

Similarly, the self attention enhancement we pursued as an improvement is based directly on the work on R-Net from Microsoft Natural Language research group[4]. The self attention in our model is but one piece of the larger R-Net model. One of the additional pieces in R-Net which we did not implement is using an attention-based recurrent neural network to obtain passage representations. In addition, R-Net leverages pointer networks in finding the location of the answer whereas our model has no pointer techniques at all.

The idea of using answer pointers for the SQuAD challenge is also applied by Wang et al. in their Machine Comprehension Using Match-LSTM and Answer Pointer paper[7]. In that paper, Wang. et al show that using a pointer network which leverages attention mechanisms to select start/end tokens for answer as a bounded scenario can lead to worthwhile improvements in model performance.

7 Future Work

For future work, we would like to implement the full extent of the R-Net models as specified in the original paper publication. We would also like to use an ensemble learning technique for our models. We would also like to run more experiments of model feature ablation to get a better understanding of the contribution of a technique to overall model performance. Beyond just optimizing for F1 and EM score metrics, an alternate research direction that is interesting to us is finding ways of reducing the size and complexity of a SQuAD model such that it may be able to fit reasonably well on a resource constrained smartphone device.

8 Conclusion

In this report we have presented our work on the SQuAD machine reading comprehension challenge. Our focus has been on attention techniques and how they can be applied towards this task. For the baseline model, we had only basic dot product attention. Our improvements over the baseline were BiDAF (Bidirectional Attention Flow) and Self-Attention. Of these, the best model was BiDAF with a modeling layer whereby we got a dev F1 score of 0.6535 and an EM score of 0.5030. Even with these simple attention enhancements, we saw some decent performance improvements over the baseline and this shows that attention is a good technique to use for tackling the machine reading comprehension task.

9 Acknowledgments

We would like to extend our thanks to the CS224N teaching staff for an amazing quarter. We would also like to extend a note of thanks to Microsoft for giving us Azure research credits that allowed us to train our SQuAD models in the cloud.

References

[1] Cs244n (2018) squad project description. http://web.stanford.edu/class/cs224n/default_project/default_project_v2.pdf.

[2] Squad training dataset histogram plotting code. https://github.com/jervisfm/cs224n-project/blob/master/Plot-Histogram-Statistics.ipynb.

[3] W. contributors. F1 score — wikipedia, the free encyclopedia, 2018. [Online; accessed 17-March-2018].

[4] M. N. L. C. Group. R-net: Machine reading comprehension with self-matching networks. 2017.

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[6] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[7] S. Wang and J. Jiang. Machine comprehension using match-lstm and answer pointer. *CoRR*, abs/1608.07905, 2016.