The SQuAD Challenge - Machine Comprehension on the Stanford Question Answering Dataset

Rohit Prakash Apte
SUID: rapte
Codalab ID: rapte@stanford.edu
CS224N Winter 2017-2018

Abstract

Over the past few years have seen some significant advances in NLP tasks like Named Entity Recognition [1], Part of Speech Tagging [2] and Sentiment Analysis [3]. Deep learning architectures have replaced conventional Machine Learning approaches with impressive results. However, reading comprehension remains a challenging task for machine learning [4][5]. The system has to be able to model complex interactions between the paragraph and question. Only recently have we seen models come close to human level accuracy.[1] For this paper I implemented the Bidirectional Attention Flow model [6], using pretrained word vectors and training my own character level embeddings. Both these were combined and passed through multiple deep learning layers to generated a query aware context representation of the paragraph text. My model achieved 76.553 % F1 and 66.401 % EM on the test set.

1 Introduction

2014 saw some of the first scientific papers on using neural networks for machine translation (Bahdanau et al [7], Kyunghyun et al [8], Sutskever et al [9]). Since then we have seen an explosion in research leading to advances in Sequence to Sequence models, multilingual neural machine translation, text summarization and sequence labeling.

Machine comprehension evaluates a machine's understanding by posing a series of reading comprehension questions and associated text, where the answer to each question can be found only in its associated text [5]. Machine comprehension has been a difficult problem to solve - a paragraph would typically contain multiple sentences and Recurrent Neural Networks are known to have problems with long term dependencies. Even though LSTMs and GRUs address the exploding/vanishing gradients RNNs experience, they too struggle in practice. Using just the last hidden state to make predictions means that the final hidden state must encode all the information about a long word sequence. Another problem has been the lack of large datasets that deep learning models need in order to show their potential. MCTest [10] has 500 paragraphs and only 2,000 questions.

Rajpurkar, et al addressed the data issue by creating the SQuAD dataset in 2016 [11]. SQuAD uses articles sourced from Wikipedia and has more than 100,000 questions. The labelled data was obtained by crowdsourcing on Amazon Mechanical Turk - three human responses were taken for each answer and the official evaluation takes the maximum F1 and EM scores for each one.

[1] based on certain metrics for a specific, constrained task.

Paragraph: The scientific revolution was a period when European ideas in classical Physics, Astronomy, Biology, Human Anatomy, Chemistry, and other classical sciences were rejected and led to doctrines supplanting those that had prevailed from ancient Greece to the middle ages which would lead to a transition to modern science. this period saw a fundamental transformation in scientific ideas across Physics, Astronomy, and Biology, in institutions supporting scientific investigation, and in the more widely held picture of the universe. individuals started to question all manners of things and it was this questioning that led to the scientific revolution, which in turn formed the foundations of contemporary sciences and the establishment of several modern scientific fields.

Question: What did the scientific revolution cause?

Answer: a transition to modern science

Figure 1: Sample SQuAD paragraph, question and answer.

Since the release of SQuAD new research has pushed the boundaries of machine comprehension systems. Most of these use some form of Attention Mechanism [6][12][13] which tell the decoder layer to "attend" to specific parts of the source sentence at each step. Attention mechanisms address the problem of trying to encode the entire sequence into a final hidden state.

Formally we can define the task as follows - given a context paragraph c, a question q we need to predict the answer span by predicting (a_{start}, a_{end}) which are start and end indices of the context text where the answer lies.

For this project I implemented the Bidirectional Attention Flow model [6] - a hierarchical multi-stage model that has performed very well on the SQuAD dataset. I trained my own character vectors [15][16], and used pretrained Glove embeddings [14] for the word vectors. My final submission was a single model - ensemble models would typically yield better results but the complexity of my model meant longer training times.

2 Related Work

Since its introduction in June 2016, the SQuAD dataset has seen lots of research teams working on the challenge. There is a leaderboard maintained at `https://rajpurkar.github.io/SQuAD-explorer/`. Submissions since Jan 2018 have beaten human accuracy on one of the metrics (Microsoft Research, Alibaba, Google Brain and Joint Laboratory of HIT and iFLYTEK Research are on this list at the time of writing this paper). Most of these models use some form of attention mechanism and ensemble multiple models.
For example, the R-Net by Microsoft Research [12] is a high performing SQuAD model. They use word and character embeddings along with Self-Matching attention. The Dynamic Coattention Network [13], another high performing SQuAD model uses coattention.

3 Approach

My model architecture is very closely based on the BiDAF model [6]. I implemented the following layers

- **Embedding layer** Maps words to high dimensional vectors. The embedding layer is applied separately to both the context and question text. I used two methods
    - Word embeddings - Maps each word to pretrained vectors.

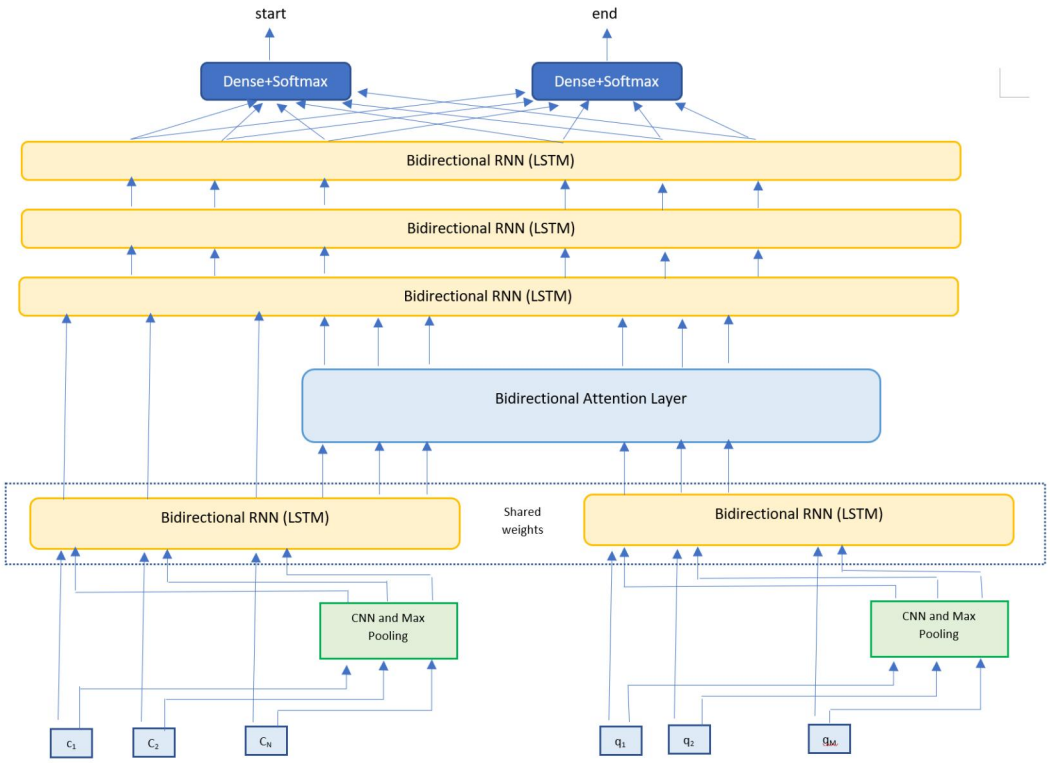

Figure 2: Model Architecture

- Character embeddings - Maps each word to character embedding and run them through multiple layers of Convolutions and Max Pooling layers. I trained my own character embeddings due to challenges with the dataset. These will be explained in the next section.

- **RNN Encoder layer** Takes the context and question embeddings and runs each one through a BiDirectional RNN (LSTM). The Bi-RNNs share weights in order to enrich the context-question relationship.
- **Attention layer** Calculates the BiDirectional attention flow. We concatenate this with the context embeddings.
- **Modeling layer** Runs the attention and context layers through multiple layers of BiDirectional-RNNs (LSTMs).[2]
- **Output layer** Runs an output of the Modeling Layer through two fully connected layers to calculate the start and end indices of the answer span.[3].

4 Experiments

4.1 Dataset

The dataset for this project was SQuAD - a reading comprehension dataset. SQuAD uses articles sourced from Wikipedia and has more than 100,000 questions. Our task is to find the answer span within the paragraph text that answers the questions.

[2]The original BiDAF paper used 2 Bidirectional-RNNs in the modeling layer. I got better performance using 3 Bidirectional-RNNs

[3]This is different from the original BiDAF paper where they use a fully connected layer to calculate the start index and then pass that through an LSTM+softmax layer to calculate the end index

The sentences (all lowercase) are tokenized into words using nltk. The words are then converted into high dimensional vector embeddings using Glove. The characters for each word are also converted into character embeddings and then run through a series of convolutions and max pooling layers. I ran some analysis on the word and character counts in the dataset to better understand what model parameters to use.

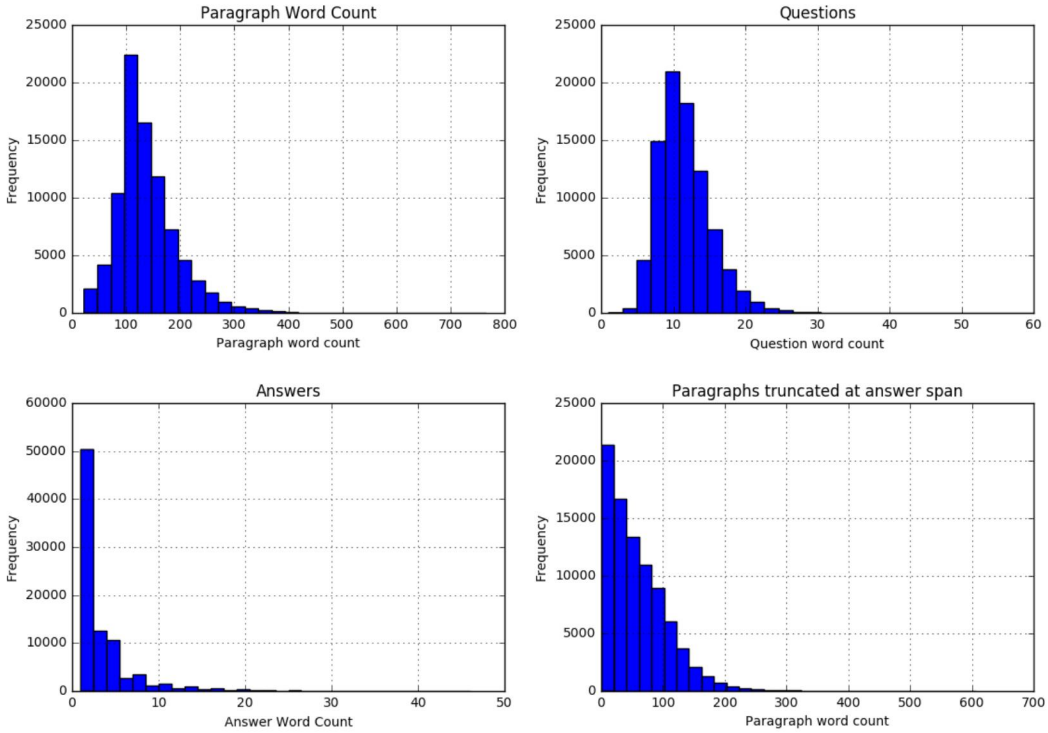

Figure 3: Word Statistics

We see that

99.8 percent of paragraphs are under 400 words
99.9 percent of questions are under 30 words
99 percent of answers are under 20 words (97.6 under 15 words)
99.9 percent of answer spans lie within first 300 paragraph words

We can use these statistics to adjust our model parameters (described in the next section).

For the character level encodings, I did an analysis of the character vocabulary in the training text. We had 1,258 unique characters. Since we are using Wikipedia for our training set, many articles contain foreign characters. Figure 4 shows an example of this scenario. Further analysis suggested that these special characters don't really affect the the meaning of a sentence for our task, and that the answer span contained 67 unique characters. I therefore selected these 67 as my character vocabulary and replaced all the others with a special REPLACEMENT_TOKEN.

Instead of using one-hot embeddings for character vectors, I trained my own character vectors on a subset of Wikipedia. I ran the word2vec algorithm at a character level to get char2vec - 50 dimensional character embeddings. A t-SNE plot of the embeddings shows us results similar to word2vec. I used these trained character vectors for my character embeddings. The maximum length of a paragraph word was 37 characters, and 30 characters for a question word. Since we

Paragraph: Sanskrit has also influenced Sino-Tibetan languages through the spread of Buddhist texts in translation. Buddhism was spread to China by Mahayana missionaries sent by Ashoka, mostly through translations of Buddhist hybrid Sanskrit. Many terms were transliterated directly and added to the Chinese vocabulary. Chinese words like 剎那 chànà (devanagari: क्षण kṣaṇa 'instantaneous period') were borrowed from Sanskrit. many Sanskrit texts survive only in Tibetan collections of commentaries to the Buddhist teachings, the Tengyur.

Question: What in the use of Sanskrit has influenced Sino-Tibetan languages?

Answer: buddhist texts

Figure 4: Foreign characters in paragraph text

are using max pooling, I used these as my character dimensions and padded with zero vectors for smaller words.

4.2 Model Configuration

I used the following parameters for my model. Some of these (context_len, question_len, etc) were fixed based on the data analysis in the previous section. Others were set by trying different parameters to see which ones gave the best results.

Parameter	Description	Value
context_len	Number of words in the paragraph input	300
question_len	Number of words in the question input	30
embedding_size	Dimension of GLoVE embeddings	300
context_char_len	Number of characters in each word for the paragraph input (zero padded)	37
question_char_len	Number of characters in each word for the question input (zero padded)	30
char_embed_size	Dimension of character embeddings	50
optimizer	Optimizer used	Adam
learning_rate	Learning Rate	0.001
dropout	Dropout (used one dropout rate across the network)	0.15
hidden_size_size	Size of hidden state vector in the Bi-RNN layers	200
conv_channel_size	Number of channels in the CNN	128

4.3 Evaluation Metric

Performance on SQuAD was measured via two metrics:

- **Exact Match (EM)** Binary measure of whether the system output matches the ground truth exactly.
- **F1** Harmonic mean of precision and recall.

4.4 Results

My model achieved the following results (I scored much higher on the Dev and Test leaderboards than on my Validation set)

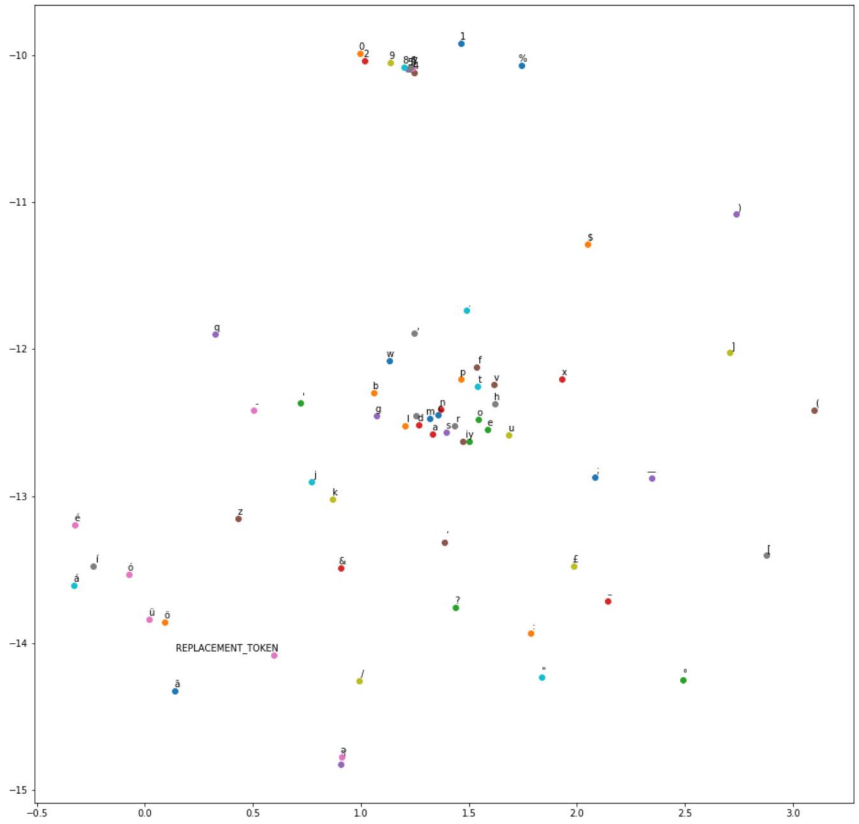

Figure 5: char2vec - character embeddings trained on Wikipedia

Dataset	F1	EM
Train	81.600	68.000
Val	69.82	54.93
Dev	75.509	65.497
Test	76.553	66.401

The original BiDAF paper had an F1 score of 77.323 and EM score of 67.947. My model scored a little lower, possibly because I am missing some details not mentioned in their paper, or I need to tweak my hyperparameters further.
I tracked accuracy on the validation set as I added more complexity to my model. I found it interesting to understand how each additional element contributed to the overall score. Each row tracks the added complexity and scores related to adding that component.[4]

[4]My val scores were lower than both the dev and test scores for all my submissions. The numbers I got would have scored higher on the leaderboard.

Model	F1	EM
Baseline	39.34	28.41
BiDAF	42.28	31
Smart Span	44.61	31.13
1 BiRNN in modeling layer	66.83	51.40
2 BiRNN in modeling layer	68.28	53.10
3 BiRNN in modeling layer	68.54	53.25
Character CNN	69.82	54.93

I also analyzed the questions where we scored zero on F1 and EM scores. The F1 score is more forgiving. We would have a non zero F1 if we predict even one word correctly vs any of the human responses. An analysis of questions that scored zero on the F1 and EM metric were split by question type. The error rates are proportional to the distribution of the questions in the dataset.

Question Type	All Dev set	F1=0	EM=0
what	27.2	28.4	29.3
is	18.4	18.5	18.4
did	9.1	8.8	9
was	8.7	9.1	7.9
do	6.9	6.9	7.9
how	6.2	5.9	6.1
who	6.2	6.7	5.4
are	4.4	3.7	4.2
which	3.3	3.4	3.1
where	2.3	2.5	2.5
when	3.9	2.9	2.3
name	1.8	1.5	1.5
why	0.7	0.6	1.3
would	0.7	0.9	0.9
whose	0.2	0.3	0.2

However, there were some questions where the system was very close to the correct answer, or the correct answer was technically wrong. See Figure 6.

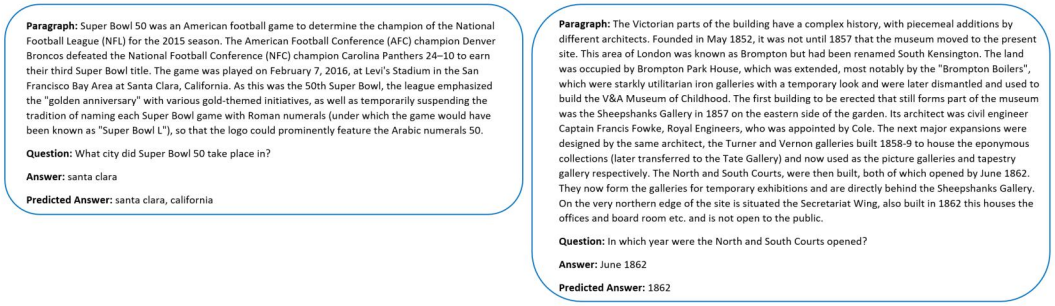

Figure 6: Error Analysis

5 Conclusion

Attention mechanisms coupled with deep neural networks can achieve competitive results on Machine Comprehension. For this project I implemented the BiDirectional attention flow model. My model accuracy was very close to the original paper. In the modeling layer we discovered that deeper networks do increase accuracy, but at a steeper computational cost.

For future work I would like to explore an ensemble of models - using different deep learning layers and attention mechanisms. Looking at the leaderboard (`https:`

//rajpurkar.github.io/SQuAD-explorer/), most of the top performing models are ensembles.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa. *Natural Language Processing (almost) from Scratch.* arXiv reprint arXiv:1103.0398 2011

[2] Peilu Wang, Yao Qian, Frank K. Soong, Lei He, Hai Zhao. *Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network.* arXiv reprint arXiv:1510.06168 2015

[3] Andreea Salinca. Convolutional *Neural Networks for Sentiment Classification on Business Reviews.* arXiv reprint arXiv:1710.05978 2017

[4] Yelong Shen, Po-Sen Huang, Jianfeng Gao, Weizhu Chen *ReasoNet: Learning to Stop Reading in Machine Comprehension.* arXiv reprint arXiv:1609.05284 2016

[5] Mrinmaya Sachan, Avinava Dubey, Eric P. Xing, Matthew Richardson. *Learning Answer-Entailing Structures for Machine Comprehension* 2015

[6] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi. *Bidirectional Attention Flow for Machine Comprehension* arXiv reprint arXiv:1611.01603

[7] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches* arXiv reprint arXiv:1409.1259

[8] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate* arXiv reprint arXiv:1409.0473 2014

[9] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. *Sequence to Sequence Learning with Neural Networks* arVix reprint arXiv:1409.3215 2014

[10] Matthew Richardson Christopher J.C. Burges Erin Renshaw. *MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text.* 2013

[11] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang. *SQuAD: 100,000+ Questions for Machine Comprehension of Text* arVix reprint arXiv:1606.05250 2016

[12] Natural Language Computing Group, Microsoft Research Asia *R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS* 2017

[13] Caiming Xiong, Victor Zhong, Richard Socher *Dynamic Coattention Networks For Question Answering* arXiv reprint arXiv:1611.01604 2016

[14] Jeffrey Pennington, Richard Socher, Christopher D. Manning *GloVe: Global Vectors for Word Representation*

[15] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean *Efficient Estimation of Word Representations in Vector Space* arXiv reprint arXiv:1301.3781 2013

[16] Xiang Zhang, Junbo Zhao, Yann LeCun *Character-level Convolutional Networks for Text Classification* arXiv reprint arXiv:1509.01626 2015