Question Answering on SQuAD2.0

Stanford CS224N {Default} Project
(IID SQuAD track)

Aditya Srivastava
adityaks@stanford.edu

Anirudh Rao
anirao26@stanford.edu)

Abstract

The goal of this project is to build a Question Answering system on the SQuAD 2.0 dataset. Our initial approach to solve this problem focused on implementing the default baseline model that is based on a variant of Bidirectional Attention Flow (BiDAF) with attention. We explored performance after adding character level embeddings to the baseline along with exploring various attention mechanisms. Additionally, we also explored the impact of tuning the hyper-parameters used to train the model. Finally, we studied the effect of using multiple variants of RNN as building blocks in the neural architecture.

1 Introduction

The domain of question answering systems has evolved many fold in recent years. Given a question and a context paragraph, the task is to identify a continuous span of words in the context paragraph that best answers the question. It is a particularly challenging task since human language is subjective in nature and the context in which words are used vary between contexts.

The SQuAD2.0 dataset used for this project is unique because it contains certain questions that do not have a right answer in the passage provided. This means that a proposed model must not only learn to reason the dependencies of words/phrases within questions and answers but also between them. The high penalty of guessing a wrong answer when there is no right answer forces the model to have a higher degree of confidence in its predictions.

The traditional RNN based approaches used to solve this problem were not very effective because they were not efficient in preserving long term dependencies between words in a sentence. The introduction of the attention mechanism and other RNN architectures tremendously help solve for this.

2 Approach

Our approach tries to improve upon the baseline BiDAF implementation provided to us. We start with replication of the original BiDAF paper by adding character level embeddings to both contexts and questions. Additionally, we explore the use of self-attention, CoAttention, hyperparameter tuning and model architecture modifications.

2.1 Baseline

We use the default implementation of the Bidirectional Attention Flow (BiDAF) model to establish the baseline accuracy metrics. The baseline code was provided by the teaching staff and the code and the details can be found in the project handout.

2.2 Our Approach

Embedding layer

- **Word embeddings**: We use pretrained GloVe word vectors to obtain word embeddings for each token in the question and the context.
- **Character embeddings**: We map each character in a word to its corresponding character embedding and pass the output through a 1 dimensional Convolution Neural Network to obtain a word representation. Including character level embedding is particularly useful in handling out of vocabulary words as well as in capturing structural relationships between characters in words.

Encoding Layer Once we obtain the embeddings for the contexts (P) and the questions (Q), we use a bidirectional LSTM to encode the temporal structure and information of each context and question.

$$\mathbf{H}^p = \overrightarrow{LSTM}(\mathbf{P}), \quad \mathbf{H}^q = \overrightarrow{LSTM}(\mathbf{Q}).$$

The resulting matrices H^p and H^q are hidden representations of the passage and the question. p and q are matrices of dimensions lxP and lxQ respectively. P and Q signify the number of tokens in the passage and the question respectively. l is dimensionality of the hidden vectors.

Self-attention In order to capture long term dependencies, we explored a self-attention layer to replace the original BiDAF attention layer implementation. This effectively encoded the information of the whole paragraph.

Coattention We also experimented with CoAttention encoder as a way to capture the simultaneous interaction of context embeddings with question embeddings. This did not lift the performance to the extent we hoped for and it would be interesting to dig into the reasons why in the future.

Modelling layer The output from the attention layer contains representations of the context vector conditioned on the question vector. In this layer, we apply a bidirectional LSTM to refine the order of vectors.

Output layer The output layer consumes the output from the modelling layer and the attention layer to determine the start and end position in the context paragraph that defines the span representing the predicted answer.

3 Experiments

3.1 Data

The dataset is already provided for this project track. It contains about 140,000 example triplets (question, context and answer). Some of the questions are unanswerable. If the question is answerable then the system has to select the span of text in the paragraph that answers the question. The dataset is further split as follows:

- train (129,941 examples): All taken from the official SQuAD 2.0 training set.
- dev (6078 examples): Roughly half of the official dev set, randomly selected.
- test (5915 examples): The remaining examples from the official dev set, plus hand-labeled examples.

3.2 Evaluation methods

We are using two standard metrics to evaluate the performance of our Question Answering system:

- **Exact Match** is a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly.
- **F1 score** is a less strict metric and defined as a harmonic mean of precision and recall.

3.3 Experimental details

The metrics for the baseline models were obtained using the default hyperparameters: evaluation steps = 50,000, learning rate = 0.5, dropout probability = 0.2, exponential moving average decay =

0.999, hidden size = 100 and batch size = 64. We have enhanced the baseline implementation by including character embedding as input to the model.

Our Mentor guided us to experiment with Cottattention layers to improve model performance and we experimented with adding additional layers to BiDAF attention like Self attention as well as experimenting with BiLinear and TriLinear attentions along with GRU and LSTM for RNN. We found GRU to be faster and requiring less memory with 64 batch size to be ok while with LSTM, we found the model to be more accurate however we had to reduce the batch size to 32 as it required a lot more memory.

We also tried Pointer net however we ran into time limitation and if we had more time then we could have compared that with our current best score.

3.4 Results

Baseline scores without and with character embedding after training are reported below:

Model	F1 score	EM score
BiDAF (Without character embedding)	60.65	57.13
BiDAF (With character embedding)	61.07	58.23
BiDAF+ Char Emb + Self attention with GRU	64.88	61.17
BiDAF+ Char Emb + Self attention with LSTM	65.80	62.99

Our best experiment improved the baseline F1 score by +4.41 points

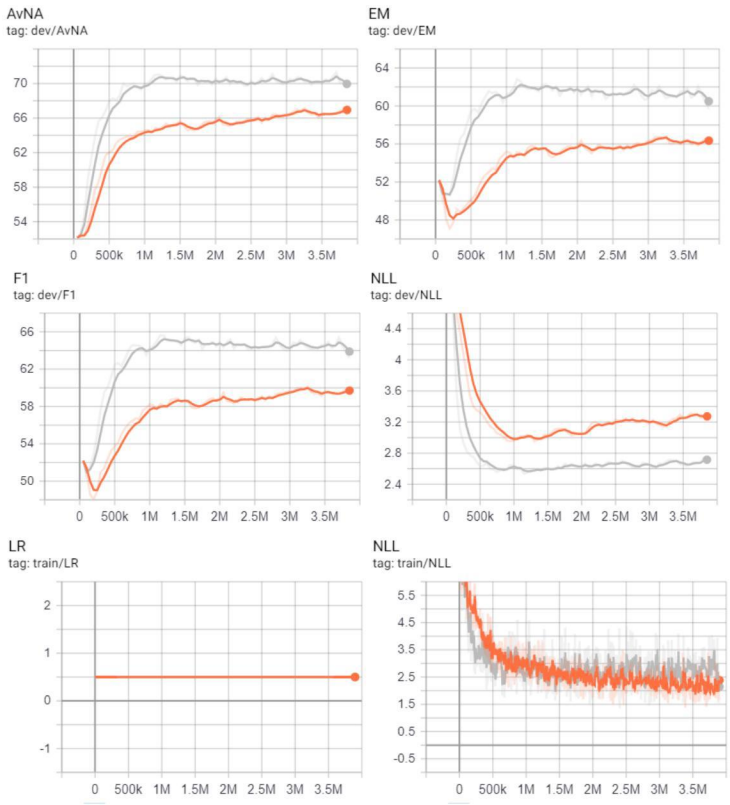

Figure 1: Learning Diagrams as compared to Baseline

4 Hyperparameter tuning

We are currently in the process of implementing the neural architecture discussed above. Going forward, we are going to explore the use of adding a Coattention (Xiong et al.) layer to the architecture.

This second attention layer on top of the first attention layer promises to improve performance. In addition to this, we plan to do extensive hyperparameter tuning w.r.t. the following parameters to further improve model accuracy:

- Types of RNN: we found LSTM to be the best however GRU was significantly faster and consumed less memory.
- Learning rate: We experimented with various learning rates like LR=0.9, 0.4, 0.6 however we settled with default as they were all degrading the F1 score.
- Drop prob: we experimented with values .3, .4 and .5 but in this case also we decided to stick with the default value of .2
- Batch size: this has a significant impact of memory requirement and I ended up training with 32 for LSTM.
- Max Grad Norm: in this case also we reverted back to default of 5 after playing with 3 and 4 which reduced the F1 score in the first few Epoch.
- We also looked at hits and misses and we found that increased length of context was not a problem for the model as it could handle it well by focusing on sections containing answers. However length of Answer was counter-productive.

5 Error Analysis

Context Length:

We find that our model picked the right answer even if the Context length was big. e.g. "The Norman dynasty had a major political, cultural and military impact on medieval Europe and even the Near East. The Normans were famed for their martial spirit and eventually for their Christian piety, becoming exponents of the Catholic orthodoxy into which they assimilated. They adopted the Gallo-Romance language of the Frankish land they settled, their dialect becoming known as Norman, Normaund or Norman French, an important literary language. The Duchy of Normandy, which they formed by treaty with the French crown, was a great fief of medieval France, and under Richard I of Normandy was forged into a cohesive and formidable principality in feudal tenure. The Normans are noted both for their culture, such as their unique Romanesque architecture and musical traditions, and for their significant military accomplishments and innovations. Norman adventurers founded the Kingdom of Sicily under Roger II after conquering southern Italy on the Saracens and Byzantines, and an expedition on behalf of their duke, William the Conqueror, led to the Norman conquest of England at the Battle of Hastings in 1066. Norman cultural and military influence spread from these new European centres to the Crusader states of the Near East, where their prince Bohemond I founded the Principality of Antioch in the Levant, to Scotland and Wales in Great Britain, to Ireland, and to the coasts of north Africa and the Canary Islands.",
Question: "Who ruled the duchy of Normandy"
Expected Answer: Richard I
Model Answer: Richard I
Analysis: Correct.

Answer Length:

"An early important political response to the opening of hostilities was the convening of the Albany Congress in June and July, 1754. The goal of the congress was to formalize a unified front in trade and negotiations with various Indians, since allegiance of the various tribes and nations was seen to be pivotal in the success in the war that was unfolding. The plan that the delegates agreed to was never ratified by the colonial legislatures nor approved of by the crown. Nevertheless, the format of the congress and many specifics of the plan became the prototype for confederation during the War of Independence.",

Question: "Was the plan formalized?"
Expected answers: ["The plan that the delegates agreed to was never ratified by the colonial legislatures nor approved of by the crown","was never ratified","never ratified","never ratified","The plan that the delegates agreed to was never ratified"]
Model Output: the format of the congress and many specifics of the plan became the prototype

Analysis: We found that our model didn't pick the right answer if the length of the answer was big and this is an area of improvement.

List of Values as Answers:

Context:
"After the revocation of the Edict of Nantes, the Dutch Republic received the largest group of Huguenot refugees, an estimated total of 75,000 to 100,000 people. Amongst them were 200 clergy. Many came from the region of the Cvennes, for instance, the village of Fraissinet-de-Lozre. This was a huge influx as the entire population of the Dutch Republic amounted to ca. 2 million at that time. Around 1700, it is estimated that nearly 25 of the Amsterdam population was Huguenot.[citation needed] In 1705, Amsterdam and the area of West Frisia were the first areas to provide full citizens rights to Huguenot immigrants, followed by the Dutch Republic in 1715. Huguenots intermarried with Dutch from the outset.",
Question:"What was the population of the Dutch Republic before this emigration?"
Expected answers: ["ca. 2 million","2 million","2 million"]
Model output: 75,000
Analysis: Our model was not so accurate for which a number of answers were provided. This is another area of improvement.

6 Conclusion

We could improve the model performance on both dev and test sets by at least 4 points which was among top 25 in both evaluation and test leaderboards. the baseline F1 and EM scores without character embedding were 60.65 and 57.13 while our best improvements with BiDAF, Character Embedding, Self attention with LSTM were 65.80 and 62.99 respectively. The scores would have been better with pre-trained models however, for our track it was prohibited. Even if we could improve the performance by a bit, the question answering remains a challenging problem with a lot of scope of improvement. Also we need to make sure that the current model generalises beyond SQuAD dataset.

7 Future work

We initially planned to do more extensive hyperparameter tuning. However, given the limited we had and the long training time on VMs, we could not fully exhaust our list of experiments. If we had more time then we wanted to explore some more values. We also wanted to experiment with other attention mechanisms as well as implementing more recent papers on Transformer Architecture.

8 Contribution and Acknowledgement

Both the team members contributed equally in studying literature, model implementation, diagnosing and analyzing model outputs and errors, Hyperparameter tuning.
We also want to express our heartfelt gratitude to our mentor Gita, teaching staff and instructors for their kind help and guidance. We also want to acknowledge the value added by the topics covered in the CS224N course to our understanding of NLP as a subject.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805v2 [cs.CL] 24 May 2019

[2] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603, 2016.

[3] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. arXiv preprint arXiv:1608.07905, 2016.

[4] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604, 2016.

[5] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," arXiv preprint arXiv:1611.01604, 2016.