Building a QA system (IID SQuAD track)

Stanford CS224N Default Project

Name
Department of Computer Science
Stanford University
bmeena@stanford.edu

Abstract

The goal of the project is to build a question answering system that works well on SQAD dataset[1]. The system should be able to read a paragraph and answer a question correctly related to the paragraph. This is an interesting task because it measures how well the system can interpret text. The input to the system is a paragraph and a question related to the paragraph and the output from the system is the answer to the question based on the text in the paragraph. We have developed a system implementing character-level embedding using 1D Convolutions on top of the provided baseline code to mimic the BiDAF (Bidirectional Attention Flow) model. We find that the addition of character-level embedding model performs better than the baseline. We also fine-tuned the hyper-parameters and tried different types optimizers to increase the metric scores.

1 Key Information to include

- Mentor: Chris Waites
- Sharing project: N/A

2 Introduction

The project involves building a question answering system specifically on SQUAD dataset. The system will take a paragraph and a question related to the paragraph as inputs and output the answer to the question based on the text paragraph. Question answering is a very important NLP research area since it would enable to computer systems to understand text. Also, we have a massive collection of full-text documents on the web and just displaying the related documents for a question is not very useful especially on mobiles or while using digital assistants like Alexa and Google assistant. Also, Reading Comprehension is an important field and being able to develop systems that can interpret text at human level will be able to lead us to the next revolution in Artificial Intelligence. We will be using the starter code as the baseline and will add character-level embedding layer. Character-level CNN embedding handles out-of-vocabulary words by breaking the words.Along with character-level CNN, we will be trying different improvements like using different types of optimizers, tuning the hyperparameters and using regularization techniques like dropout and Batch normalization.

3 Related Work

Question answering is a part of Reading Comprehension field within NLP and has been researched extensively the last few years. Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowd-workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. Attention mechanisms have been successfully extended to Question answering.

One of the well-known models that work well on the SQUAD dataset is the BiDAF[2] which is a multi-stage hierarchical process that represents the context at different levels of granularity and uses bi-directional attention flow mechanism to obtain a query-aware context representation without early summarization. We will be using the starter code which is based on BiDAF.

Another high-performing SQUAD model is the Dynamic Coattention Networks[3]. Bidirectional Encoder Representations from Transformers (BERT)[4] is a Transformer-based machine learning technique for NLP pre-training developed by Google. QANet[5] adapts ideas from the Transformer[6] and applies them to question answering, doingaway with RNNs and replacing them entirely with self-attention and convolution.

4 Approach

The main approach is to extend the baseline model by adding character-level embedding layer. Character-level embeddings allow us to condition on the internal structure of words, and better handle out-of-vocabulary words. For the Character level embedding, we use a one-dimensional convolutional neural network (Conv1d) [7] to find numeric representation of words by looking at their character-level compositions followed by a Batch Normalization, Relu layer and Max-pool.

The concatenation of the character and word embedding vectors is passed to a two-layer Highway Network. The necessary changes to include the character indexes for the context and query to be passed to the embedding layer has been implemented in the new model. We also tried out a few hyper-parameters and different types of regularization techniques. We also tried different optimizers to check the more efficient one for the given task.

Baseline will the one provided as the starter code for the default project. The baseline model is a based on Bidirectional Attention Flow (BiDAF). The original BiDAF model uses learned character-level word embeddings in addition to the word-level embeddings. Figure 1 shows the BiDAF architecture. Unlike the original BiDAF model, the baseline implementation does not include a character-level embedding layer.

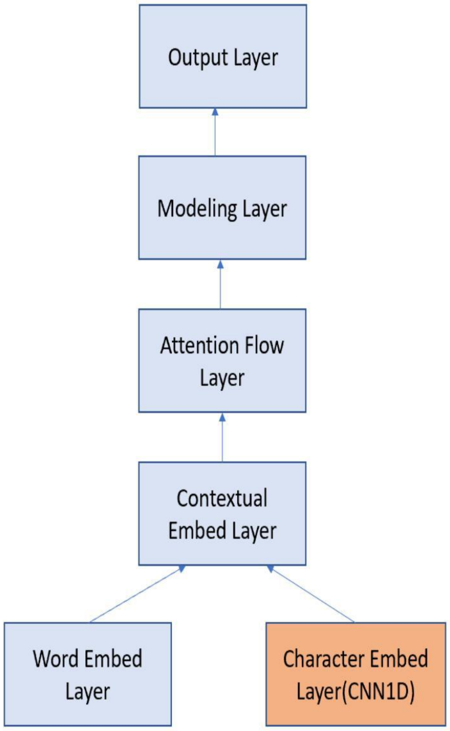

Figure 1. BiDAF model

5 Experiments

5.1 Data

The dataset for this project is SQuAD 2.0. It has three splits: train, dev and test. The train and dev sets are publicly available, and the test set is entirely secret. This is the split between train, dev and test:

- 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.

We will use the train set to train the model and the dev set to tune hyper-parameters and measure progress locally. We will submit the test set solutions to a class leaderboard, which will calculate and display the scores on the test set. Nearly half of the SQuAD examples are No-answer. If the question is answerable, the answer is a chunk of text taken from the paragraph ie, the answer will just be a span of text from the paragraph that answers the question.

5.2 Evaluation method

Performance is measured by these metrics: Exact Match (EM)score and F1 score.

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

When evaluating on the dev or test sets, we take the maximum F1 and EM scores across the three human-provided answers for that question. This makes evaluation more forgiving. Finally, the EM and F1 scores are averaged across the entire evaluation dataset to get the final reported scores.

5.3 Experimental details

First, we ran the baseline model to get the baseline metrics. The batch size was set to 64 and the number of features in the encoder hidden layers was set to 100. The models were run for 30 epochs. The initial dropout rate used was 0.2. The initial learning rate used was 0.5. Then the following different experiments were run adding improvements to the previous best model.

- Character embedding 2 CNN: Character-level embedding with two cnn layers was added.
- Character embedding 1 CNN: Instead of two cnn layers, we used one CNN layer followed by RELU and Max Pool.
- Types of RNN: The baseline uses a bidirectional LSTM. We changed it to GRU to make the system run faster.
- Batch Normalization for regularization: We added BatchNorm layer in the Character embedding.
- Dropout for regularization: We also used different values for dropout ranging from values 0.1 to 0.5.
- Optimizers: The optimizer used in the baseline is Adadelta. We tried experiments with Adam[8] and SGD optimizers as well.
- Learning rate decay: The baseline has a constant Learning rate. We ran experiments on a few learning rate decay values – 0.94, 0.95 and 0.96
- Learning rate: The baseline had a learning rate of 0.5. We tried different learning rates – 0.05, 0.001, 0.0003
- Early stopping: Early stopping is a way to regularize and prevent over-fitting of the training data. We used early stopping for one of the runs by setting the number of epochs to 20.

5.4 Results

The following table lists the metrics for the different models.

Model	NLL	F1	EM	AvNA
Baseline (Adadelta)	3.03	61.33	58.04	68.04
Char-cnn embedding	3.16	63.25	59.91	69.45
Char-cnn embedding with 2 cnn layers	3.12	63.06	59.62	69.32
Char-cnn embedding with batch norm(BN)	3.01	64.13	60.61	70.63
Char-cnn embedding BN/Learning rate decay(0.96)	2.94	65.26	61.77	71.58
Char-cnn embedding BN/Learning rate decay(0.95)	2.81	**66.27**	**62.86**	72.21
Char-cnn embedding Adam 1, LR 0.001	2.91	65.08	61.52	71.52
Char-cnn embedding Adam 2, LR 0.0003	2.94	66.62	63.37	73.23
Char-cnn embedding Adam 3 dropout 0.5, LR 0.0003	3.37	57.81	54.56	65.22
Char-cnn embedding Adam 4 dropout 0.1, LR 0.0003	3.44	65.89	62.19	72.64
Char-cnn embedding Adam 5 dropout 0.13, LR 0.0003	3.10	**66.60**	**63.23**	73.16

Table 1. Metrics for the models from different experiments

We incrementally built the model starting with the baseline and making changes on top it. The addition of the Char embedding to the baseline code improved the performance because the character-level embeddings allow us to condition on the internal structure of words and better handle out-of-vocabulary words. The 2 CNN layers model took a very long time to train. We then changed it to just one CNN layer and added ReLU and Max Pool layer. Using a GRU instead of LSTM also increased the speed of the network.

We then added Batch Norm layer[9] to the Convolutional network and that increased the performance because of its regularization effect. While using a dropout of 0.5 (the suggested NLP dropout [10]), the performance dropped because when using Batch Norm, higher dropout rates do not work well. The reason is that the statistics used to normalize the activations of the prior layer may become noisy given the random dropping out of nodes during the dropout procedure. We tried lower dropouts too. Dropouts below 0.12 were overfitting the training data and did not perform well on the dev sets.

The baseline model uses Adadelta. While using different optimizers like SGD, Adam, we found that Adam optimizer works the best for the given task. In addition to storing an exponentially decaying average of past squared gradients like Adadelta, Adam also keeps an exponentially decaying average of past gradients, like momentum. Adam behaves like a heavy ball with friction, which thus prefers flat minima in the error surface and is computationally efficient.Learning rate decay of 0.95 had the best scores. Very low decay rate had no changes compared to the constant learning rate and very high decay rate reduces the learning rate to a low value to stop the learning.

When using the default learning rate 0.001 for Adam, the dev NLL curve keeps reducing but starts to increase with a steep slope suggesting that the learning rate is too high. The learning rate 0.0003 worked best for Adam and that was the learning rate for the best model. When the learning rate was reduced to 0.0001, there was hardly any training and when using higher learning rates, the loss curve was fluctuating.

The following table shows the Dev/Test F1 and EM scores for the top two models in the Validation and Test leaderboard.

Model	Dev F1	Dev EM	Test F1	Test EM
Char-cnn embedding BN/Learning rate decay(0.95)	66.179	62.662	64.079	60.659
Char-cnn embedding Adam 5 dropout 0.13, LR 0.0003	**68.248**	**65.266**	**66.174**	**63.077**

Table 2. Metrics for the top two models in the Validation and test leaderboard

Figure 2 shows the metrics results from tensorboard. The plots show that the training loss continues to reduce throughout the training for all the three models. The dev loss begins to increase after 2M iterations which is due to overfitting. Also, our model with the highest F1 score starts to overfit the training data more than the other models as shown in the NLL training data. This is because the dropout is lower than the other models. Both the models with the character-level embedding are

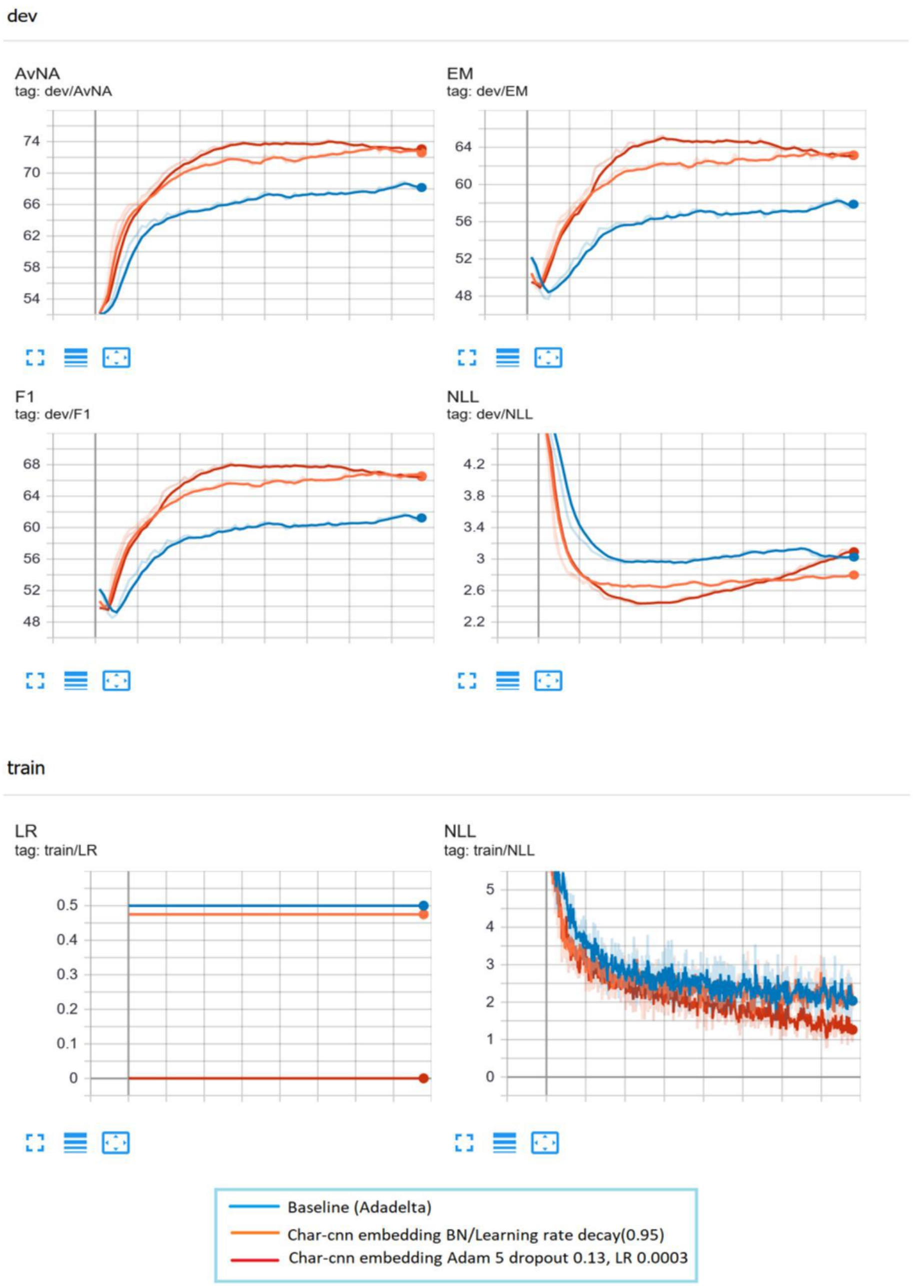

Figure 2. Baseline and the top two model metrics

doing better in all of the metrics than the baseline model which is expected since character-level embedding handles out-of-vocabulary words and help with increasing efficiency of model.

6 Analysis

The following are some samples of input paragraph and the answer and predictions for the baseline and the best model.

6.1 Example 1

Question: During what period did Radcliffe become prominent as a university?

Context: Women remained segregated at Radcliffe, though more and more took Harvard classes. Nonetheless, Harvard's undergraduate population remained predominantly male, with about four men attending Harvard College for every woman studying at Radcliffe. Following the merger of Harvard and Radcliffe admissions in 1977, the proportion of female undergraduates steadily increased, mirroring a trend throughout higher education in the United States. Harvard's graduate schools, which had accepted females and other groups in greater numbers even before the college, also became more diverse in the post-World War II period.

Answer: N/A

Baseline Prediction: post-World War II

Our Model Prediction: N/A

6.2 Example 2

Question: What did Geroge Lenczowski do to the price of oil on October 16, 1973?

Context: In response to American aid to Israel, on October 16, 1973, OPEC raised the posted price of oil by 70%, to $5.11 a barrel. The following day, oil ministers agreed to the embargo, a cut in production by five percent from September's output and to continue to cut production in five percent monthly increments until their economic and political objectives were met. On October 19, Nixon requested Congress to appropriate $2.2 billion in emergency aid to Israel, including $1.5 billion in outright grants. George Lenczowski notes, "Military supplies did not exhaust Nixon's eagerness to prevent Israel's collapse...This [$2.2 billion] decision triggered a collective OPEC response." Libya immediately announced it would embargo oil shipments to the United States. Saudi Arabia and the other Arab oil-producing states joined the embargo on October 20, 1973. At their Kuwait meeting, OAPEC proclaimed the embargo that curbed exports to various countries and blocked all oil deliveries to the US as a "principal hostile country".

Answer: N/A

Baseline Prediction: $5.11 a barrel

Our Model Prediction: N/A

Reason: The baseline model incorrectly predicts an answer because it did not pick the correct context. Even though the price of oil was changed to $5.11 a barrel, the noun subject of that sentence is not George but OPEC. Our model has correctly identified the context for the corresponding sentence and that the question does not have an answer in the given paragraph.

6.3 Example 3

Question: What is one avenue being compensated for by having committees serve such a large role?

Context: Much of the work of the Scottish Parliament is done in committee. The role of committees is stronger in the Scottish Parliament than in other parliamentary systems, partly as a means of strengthening the role of backbenchers in their scrutiny of the government and partly to compensate for the fact that there is no revising chamber. The principal role of committees in the Scottish Parliament is to take evidence from witnesses, conduct inquiries and scrutinise legislation. Committee meetings take place on Tuesday, Wednesday and Thursday morning when Parliament is sitting. Committees can also meet at other locations throughout Scotland.

Answer: no revising chamber

Baseline Prediction: N/A

Our Model Prediction: N/A

Reason: Both the baseline and our model predict N/A incorrectly. The relevant part of the question – 'The role of committees is stronger in the Scottish Parliament than in other parliamentary systems, partly as a means of strengthening the role of backbenchers in their scrutiny of the government and partly to compensate for the fact that there is no revising chamber.' has the answer but it this is a difficult question as avenue is not directly found in the text and could mean different things, and we need a more complex model to understand the language and find the correct answer.

7 Conclusion

Adding the character-level embedding to the baseline started code has given a lot of improvement to the EM and F1 scores. After running a lot of experiments, we found the best performing model to be an Adam optimizer with one char cnn embedding layer. For future work, we would like to combine local convolution with global self-attention and use ensemble techniques to improve the metrics.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

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

[3] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[5] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension, 2018.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[7] Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[10] https://ruder.io/deep-learning-nlp-best-practices/.