One Step At A Time with SQUAD 2.0

Somnath Purkayastha
psomnath@stanford.edu

Abstract

Reading comprehension is one of the fundamental NLP tasks, where model figures out the answer from a given context paragraph and question. The ability to comprehend text has many useful applications, like text summarization, advance search etc. The goal of this project was to develop a reading comprehension model, which can successfully predict the correct answer (or NULL, when the given question is non-answerable) for the SQuAD (Stanford Question Answering Database) 2.0 dataset.

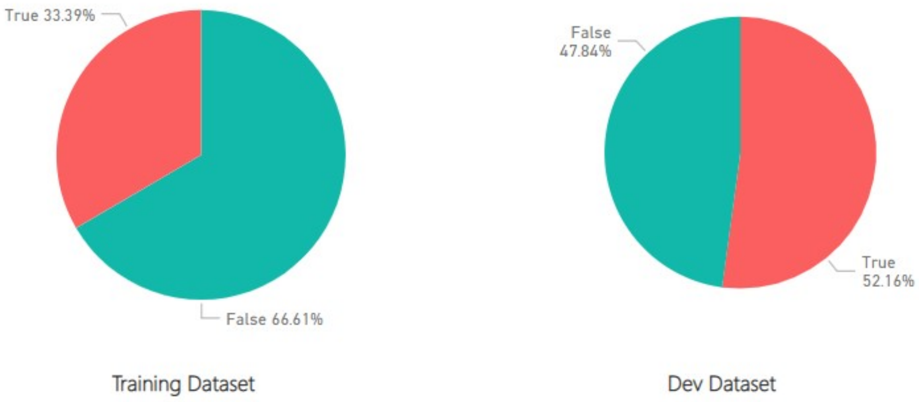
Our final model achieved 63.25 F1 score and 59.50 EM score with dev dataset. With the test dataset those scores were 62.708 F1 and 58.918 EM.

1 Introduction

For this project, our primary approach was to improve the given default model by reviewing the input data (to identify hidden features), by implementing character-level embedding (to have a more robust model), by tuning hyper-parameters (to gain better performance) and finally analyzing the error (to understand the strengths and weaknesses of the model and tune the model further with a feedback loop).

1.1 Review the input data

The SQuAD 2.0 dataset has number of questions, which are non-answerable. In the given training dataset, the proportion of answerable questions to non-answerable questions is approximately 2:1, whereas in the given dev dataset, the same proportion is almost 1:1.



Training Dataset

Dev Dataset

Fig 1: Answerable question distribution in the input (training/dev) data

The context span in the input data is primarily concentrated under 400 (words). The similar span for question is under 40 (words) and for answer is under 35 (words). Note, for obvious reason, answer data is only available for training and dev datasets. This analysis was important to set the model parameters (para_limit, ques_limit etc.).

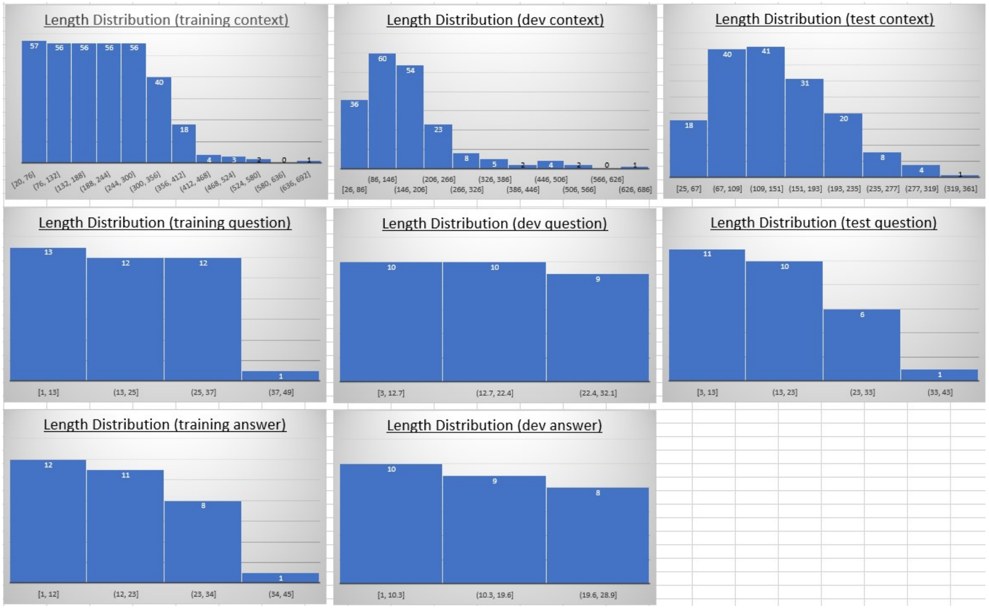


Fig 2: Context/Question/Answer length distribution in the input (training/dev/test) data

Questions in the given SQuAD 2.0 dataset (train/dev/test) are part of 10 major categories.

- **What**: More than 50% of those questions belong to this category. Example of this type of question is *‘What kind of force did Harthacnut establish?’*
- **Who:** Example of this type of question is *‘Who was the Norse leader?’*
- **How:** Example of this type of question is *‘How many men were in Robert’s army?’*
- **When:** Example of this type of question is *‘When were the Normans in Normandy?’*
- **Which:** Example of this type of question is *‘Which bound of time is more difficult to establish?’*
- **Where:** Example of this type of question is *‘Where did Oursel lead the Franks?’*
- **ToBe:** Example of this type of question is *‘Are the sizes of packets variable¿*
- **Why:** Example of this type of question is *‘Why did OPEC dollars depriciate?’*
- **Name:** Example of this type of question is *‘Name a luxury division of Toyota.’*
- **Others:** Example of this type of question is *‘Telnet was sold to’*

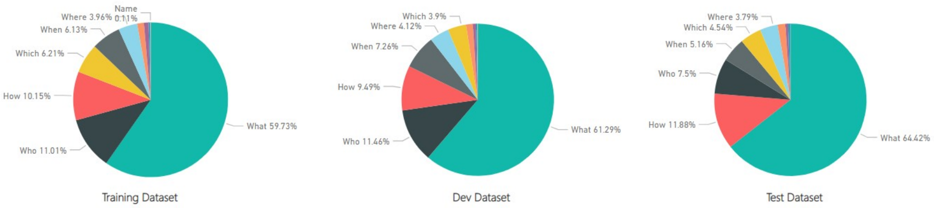


Fig 3: Distribution of question types

1.2 Implement the character-level embedding

The given default BiDAF model had five layers – Embedding Layer, Encoder Layer, Attention Layer, Modeling Layer, Output Layer. As part of this project, we enhanced the Embedding Layer to include character-based embedding, as described below.

- Given some input word indices w_1,. . . ,w_k (dim: N), the word (w_x) was broken into a list of characters ch_x1, ch_x2,. . . .,ch_xn.
- The list of characters was padded (or truncated) with blank character to make it a list of maximum number of characters per word (one of the hyperparameter).
- A character embedding vector was used to convert each character in the list to corresponding embedding vector.
- Passed this list of character embedding vectors through a Conv1D layer, ReLU layer and finally MaxPooling layer. The output was a 300D vector for the whole word.
- This output vector was concatenated with the original output of the word embedding step, producing a 600D vector for the word.
- The above-mentioned steps were performed for each word in context and question, producing 600D vector for each of those words.
- For regularization purpose, those combined vectors were passed through a drop out layer, followed by a projection layer (had to double the input dimension) and a highway encoder layer.
- The output of this enhanced Embedding Layer was passed to the next Encoder Layer. From this point, all subsequent layers stayed same (as provided by the default project).

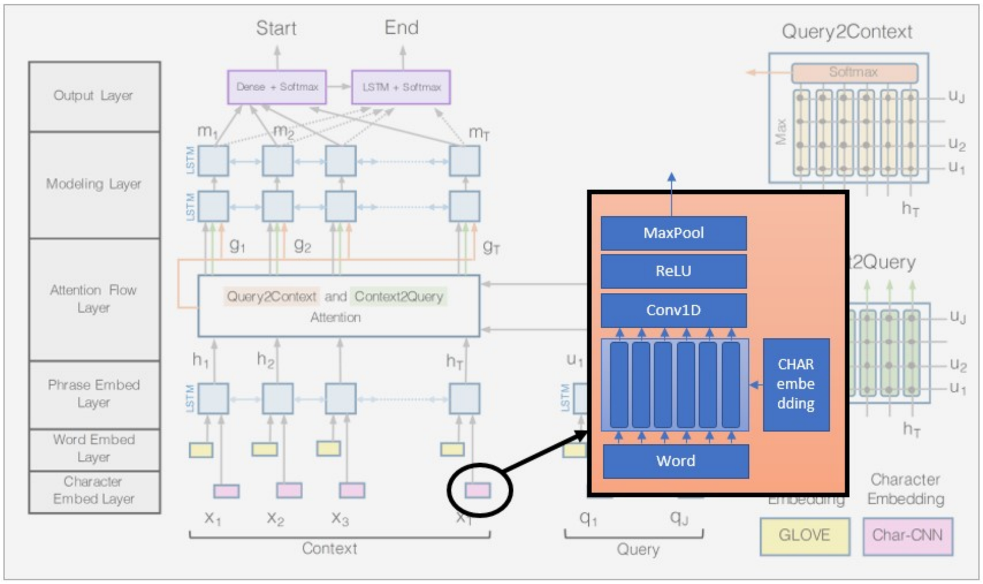


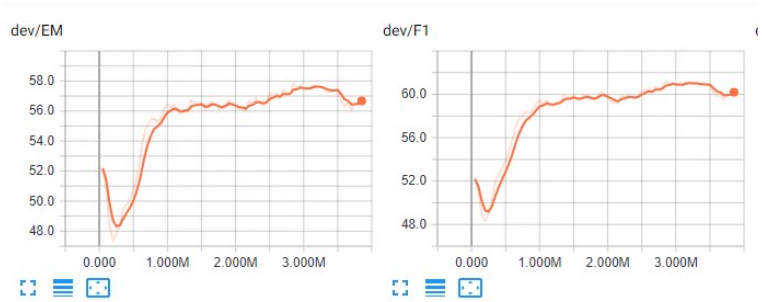
Fig 4: Updated default BiDAF model (with character-level embedding)

2 Experiments

As this is the default project, we have used the provided pre-processed SQUAD 2.0 dataset and default evaluation metrics (EM and F1) to evaluate the model. Here are the summaries of seven (out of many) full training cycles with progressively improved results:

- **Experiment #1: Run the baseline:**

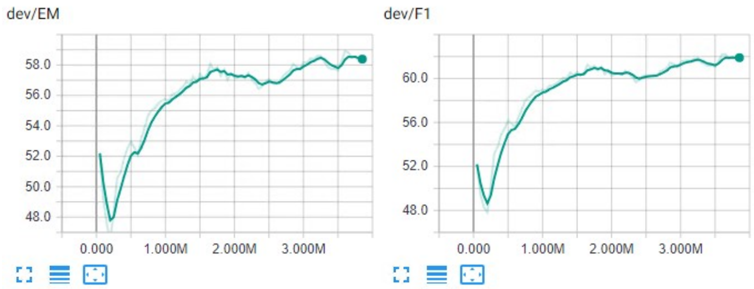
  First, we ran the given model (part of default project) without any change for 30 epochs. The objective of this run was to get a baseline result to compare the performance of future runs.



Dev NLL: 03.22, F1: 60.55, EM: 56.98, AvNA: 67.69

- **Experiment #2: Update the RNN model:**

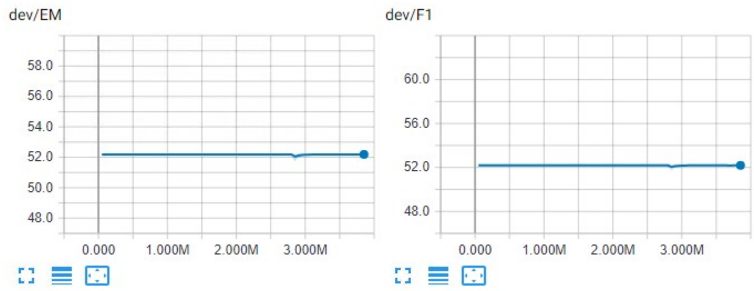
  The default model used LSTM for its RNN layer. In general, GRU as RNN layer is simpler (from execution cost standpoint) than LSTM layer. So, we updated the RNN layer to use GRU architecture. The result was slightly better with improved performance (execution time).



Dev NLL: 03.00, F1: 61.91, EM: 58.31, AvNA: 68.39

- **Experiment #3: Update the Optimizer (hyper-parameter tuning) - Failure:**

  The default model used LSTM for its RNN layer. In general, GRU as RNN layer is simpler (from execution cost standpoint) than LSTM layer. So, we updated the RNN layer to use GRU architecture. The result was slightly better with improved performance (execution time).

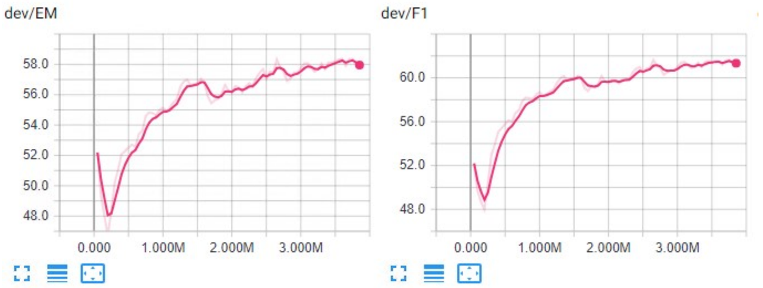


Dev NLL: 07.97, F1: 52.19, EM: 52.19, AvNA: 52.14

- **Experiment #4: Update the Optimizer (hyper-parameter tuning) - Failure:**

  Next, we wanted to try another optimizer Adamax. Unfortunately, just like the last run, the NLL score improved, but the F1 and EM scores didn't improve at all.

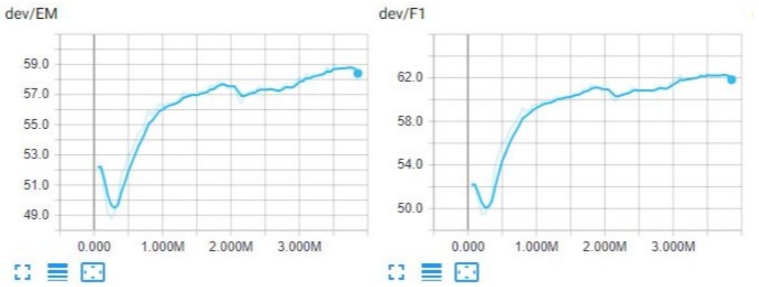
- **Experiment #5: Re-run the baseline:**



Dev NLL: 03.00, F1: 61.18, EM: 57.69, AvNA: 67.77

- **Experiment #6: Implement the character-level embedding:**

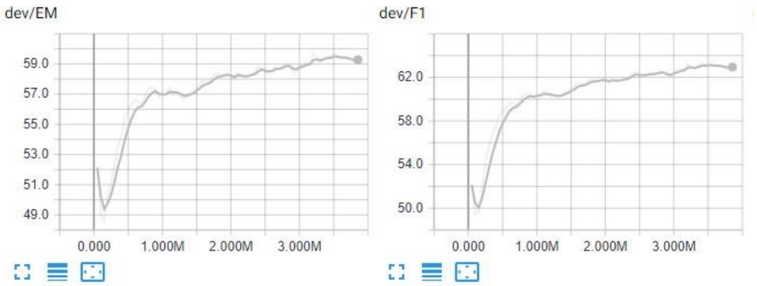
The given BiDAF model was updated with character-level embedding. This change introduced additional execution time (took approximately 50% extra time per iteration), to calculate the character-based word embedding vector, but improved the scores.



Dev NLL: 03.20, F1: 62.41, EM: 59.98, AvNA: 68.68

- **Experiment #7: Update the Optimizer (hyper-parameter tuning) - Success:**

Finally, we figured out the root cause of our earlier failures with new optimizer. The given default learning rate, 0.5, was too high for the Adam optimizer to converge. After applying the recommended learning rate, 0.001, the result converged with better scores.



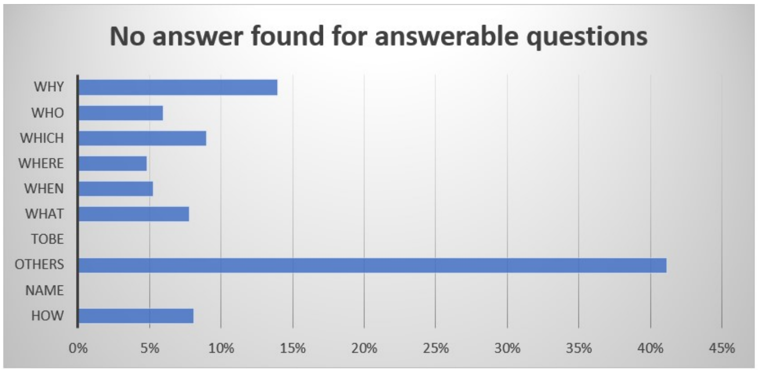
Dev NLL: 03.11, F1: 63.25, EM: 59.50, AvNA: 69.92

The final submission of our model to the Non-PCE Division Test leaderboards had F1 score 62.708 and EM score 58.918. These scores were slightly lower than the evaluation scores we got from the Dev set (F1 score 63.25 and EM score 59.50).

Error Analysis

2.1 Error Type #1: No answer found for answerable question

Context: A conservative force that acts on a closed system has an associated mechanical work that allows energy to convert only between kinetic or potential forms. This means that for a closed system, the net mechanical energy is conserved whenever a conservative force acts on the system. The force, therefore, is related directly to the difference in potential energy between two different locations in space, and can be considered to be an artifact of the potential field in the same way that the direction and amount of a flow of water can be considered to be an artifact of the contour map of the elevation of an area.
Question: What is the only form potential energy can change into?
Answer: kinetic
Prediction: NULL



Observation: Mostly questions from the 'Others' category had this problem. After further analysis it was found to be an issue with those question constructions. Here are few examples to show those challenges:

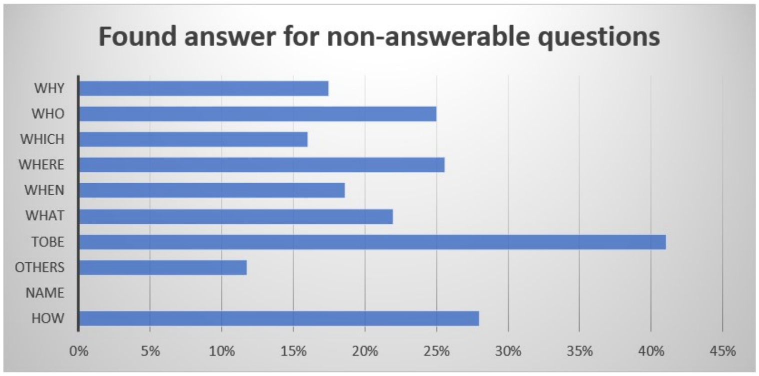
- hat military occupation wasn't opened to men until the 2000s? [**spelling mistake**]
- Street lights help reduce? [**poor question construction**]
- This state grew at a rapid pace during the Porfiriato. [**confusing question**]

Thankfully very small number of questions (only 17 out of 6̃K questions) belong to this category.

2.2 Error Type #2: Non-answerable question was identified as answerable question

Context: The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.
Question: What is France a region of?
Answer: NULL

Prediction: Normandy



Observation: Most of these cases, the model identified one of those plausible answers (SQuAD dataset successfully tricked the model).

2.3 Error Type #3: Attention Mismatch

Context: Bethencourt took the title of King of the Canary Islands, as vassal to Henry III of Castile. In 1418, Jean's nephew Maciot de Bethencourt sold the rights to the islands to Enrique Pérez de Guzmán, 2nd Count de Niebla.
Question: Who bought the rights?
Answer: Enrique Pérez de Guzmán
Prediction: Maciot de Bethencourt
Observation: In our opinion, these are hardest kind of question, which requires solid comprehension to answer the question.

2.4 Error Type #4: Close Call

Context: "The Islamic State", formerly known as the "Islamic State of Iraq and the Levant" and before that as the "Islamic State of Iraq", (and called the acronym Daesh by its many detractors), is a Wahhabi/Salafi jihadist extremist **militant group** which is led by and mainly composed of Sunni Arabs from Iraq and Syria. In 2014, the group proclaimed itself a caliphate, with religious, political and military authority over all Muslims worldwide. As of March 2015[update], it had control over territory occupied by ten million people in Iraq and Syria, and has nominal control over small areas of Libya, Nigeria and Afghanistan. (While a self-described state, it lacks international recognition.) The group also operates or has affiliates in other parts of the world, including North Africa and South Asia.
Question: What type of group is The Islamic State?
Answer: extremist militant; Wahhabi/Salafi jihadist extremist militant; Wahhabi/Salafi jihadist extremist militant group
Prediction: militant group
Observation: Though the prediction didn't match with human prediction, the answers were very close. The model was almost correct with attention to the right set of words.

3 Future Works

3.1 Hyperparameter Tuning:

For the new optimizer, 'Adam', we have used the recommended learning rate 0.001 with better performance. In future we like to tune other related parameters like beta, epsilon, decay rate etc. Also, we like to try using different optimizer like 'Adamax' etc.

3.2 Model Engineering:

As per the original plan for this project, we were developing a CNN based model to predict, if a given question (for a given context) is answerable or not. Unfortunately, the initial results were not very encouraging. Probably because the model was not complex enough to tackle the challenge. Due to time constraint, we had to abandon that plan. In future, we like to develop more comprehensive binary predictor (probably based on QANet architecture, with new final output layer) to determine, if the question is answerable, before trying to predict the answer.

3.3 Ensemble:

In general, Ensembled model always produces better result than one very complex model. There are so many different architectures are available to deal with the reading comprehension task with SQuAD 2.0 dataset. In future, we will explore the option to implement those models and ensembled them for the optimum result.

3.4 Feature Engineering:

Though now-a-days with deep learning models, enough data and processing power, we don't do feature engineering anymore. But we like to exploit few obvious patterns in the data (like question type, context/question span etc.) to improve the performance.

3.5 Error Analysis:

We wrote python script to analyze the input and output data (refer: data_analysis.py) and ported the result to SQL server to run advanced analytics using PowerBI report. It greatly helped us to understand the strengths and weaknesses of the model. Arm with that knowledge we made further changes to the model with improved performance. In future we have plan to enhance this feedback loop, so that we can pinpoint the source of the error more accurately.

4 Conclusion

To tackle one of the most popular and challenging NLP tasks, reading comprehension, we have implemented a bi-directional attention flow model with character-based embedding layer. We have made incremental improvements to the model with new RNN unit, new optimizer, new hyperparameter setting etc. We have analyzed the error in detail and created a feedback loop to update the model accordingly. The results were encouraging. Our final dev score was (F1: 63.25, EM: 59.50). However, we have plenty of opportunities to improve, to match those top entries in the SQuAD 2.0 leaderboard, and eventually the human performance (EM: 86.831, F1: 89.452).

5 Acknowledgments

We would like to thank Chris Chute and the CS224N teaching group for on-going help and guidance with this project.

References

[1] Kim, Yoon (2014) Convolutional Neural Networks for Sentence Classification
(https://arxiv.org/pdf/1408.5882v2.pdf)

[2] Seo, Minjoon & Kembhavi, Aniruddha & Farhadi, Ali & Hajishirzi, Hananneh (2016) Bidirectional Attention Flow for Machine Comprehension
(https://arxiv.org/pdf/1611.01603.pdf)