Incorporating Self-Attention and Character Embeddings in a Question Answering System

Stanford CS224N Default Project

Melanie Mei Zhang
Department of Computer Science
Stanford University
melzh@stanford.edu

Abstract

In this project, I extend and improve upon a baseline contextual question-answering BiDAF SQuAD model by using character-level embeddings, GRU encoding layers, and context-to-context self-attention. The existing embedding layer is augmented with the addition of character-level embeddings that are fed to a convolutional layer, which allows the model to deal with out-of-vocabulary tokens and infer more meaning from contexts and questions containing them. With the addition of a self-matching attention layer that receives question-aware context representations as input, the model is able to more effectively gather evidence from the entire context to help determine the answer. In addition, the baseline encoder layers are modified to use GRUs in place of LSTMs. The model is able to improve upon the baseline substantially, achieving EM and F1 scores of X and Y respectively.

1 Key Information to include

- Mentor: Michihiro Yasunaga
- External Collaborators (if you have any): N/A
- Sharing project: No

2 Introduction

Question answering is a problem within NLP whose solutions allow users to formulate questions using natural language and receive an informative response. This has applications in search engines, personal assistants, and more. As an example of a question answering dataset, the SQuAD dataset [1] consists of paragraph, question, and answer triples crowd-sourced using Amazon Mechanical Turk. Each answer is a span of the paragraph text. This project aims to improve on a provided baseline SQuAD model in the task of contexual question answering.

To improve upon the baseline, I make a number of modifications inspired by recent work and techniques learned in class. This includes the addition of character-level embeddings to the model to use alongside the existing word embeddings, which allows the model to better handle out-of-vocabulary words, enabling more meaningful representation for contexts and questions that contain such words. This also includes a context-to-context self-attention layer which refines the context representation by incorporating information from the entire context in each context word representation. To improve speed of iteration, I change the RNN of choice in the model encoder layers from LSTMs to GRUs. These additions allow the model to improve upon the baseline substantially, bringing increases in EM and F1 scores on the provided test and dev datasets.

3 Related Work

BiDAF (Bidirectional Attention Flow for Machine Comprehension) [2] is a SQuAD model that utilizes a bi-directional attention flow mechanism to obtain a question-aware context representation. In particular, the attention layer performs both question-to-context and context-to-question attention. At the time (2016), the model achieved state-of-the-art results on the SQuAD dataset.

Self-attention is a paradigm that has seen great success in both RNN and Transformer [3] models; in fact, it is a key building block of Transformers. R-NET [4] demonstrated incorporating Context-to-Context self-attention on top of a BiDAF [2] model that landed first place on the score leaderboard at the time of inception (2017).

Both BiDAF and R-NET utilize character embeddings, which I add to the baseline model in this project. As done in the original R-NET model, the character embeddings are fed into a CNN layer which produces a representation for each word in each sequence.

4 Approach

My final model is composed of the following layers in order:

Embedding Layer: This layer utilizes both word-level and character-level embeddings which are provided in the baseline project code. The character-level embeddings are combined into a word-level representation using a 2D CNN layer.

Encoder Layer: This layer uses a bi-directional GRU to produce a representation for each word in both the context and question sequences.

Attention Layer: This layer utilizes bi-directional attention flow to produce a question-aware context representation, unchanged from the baseline model.

Modeling Layer: This layer refines the question-aware context representations, using a bi-directional GRU.

Self Attention Layer: This layer performs context-to-context self attention, producing new representations for each context word.

Modeling Layer 2: This layer refines the context-aware representations using a bi-directional GRU.

Output Layer: This layer produces start and end probabilities for each word in the context, unchanged from the original model.

The specific changes I made to the baseline model are expounded in more detail below.

4.1 Character Embeddings

Using the character embeddings provided with the baseline model code, each word in a sequence is split into its respective characters and embedded into vectors. Each sequence can be thought of as an input signal with size (character embedding size, max sequence length, max word length). The input is fed into a 2D convolutional layer, whose input channel size c_{in} is the size of the character embeddings and output channel size c_{out} is the hidden size. The CNN utilizes a kernel of size (1, 5), as done in the original BiDAF model [2]. The outputs are then max-pooled over the width to produce a vector c_{embed} of the desired hidden size for every word in the sequence.

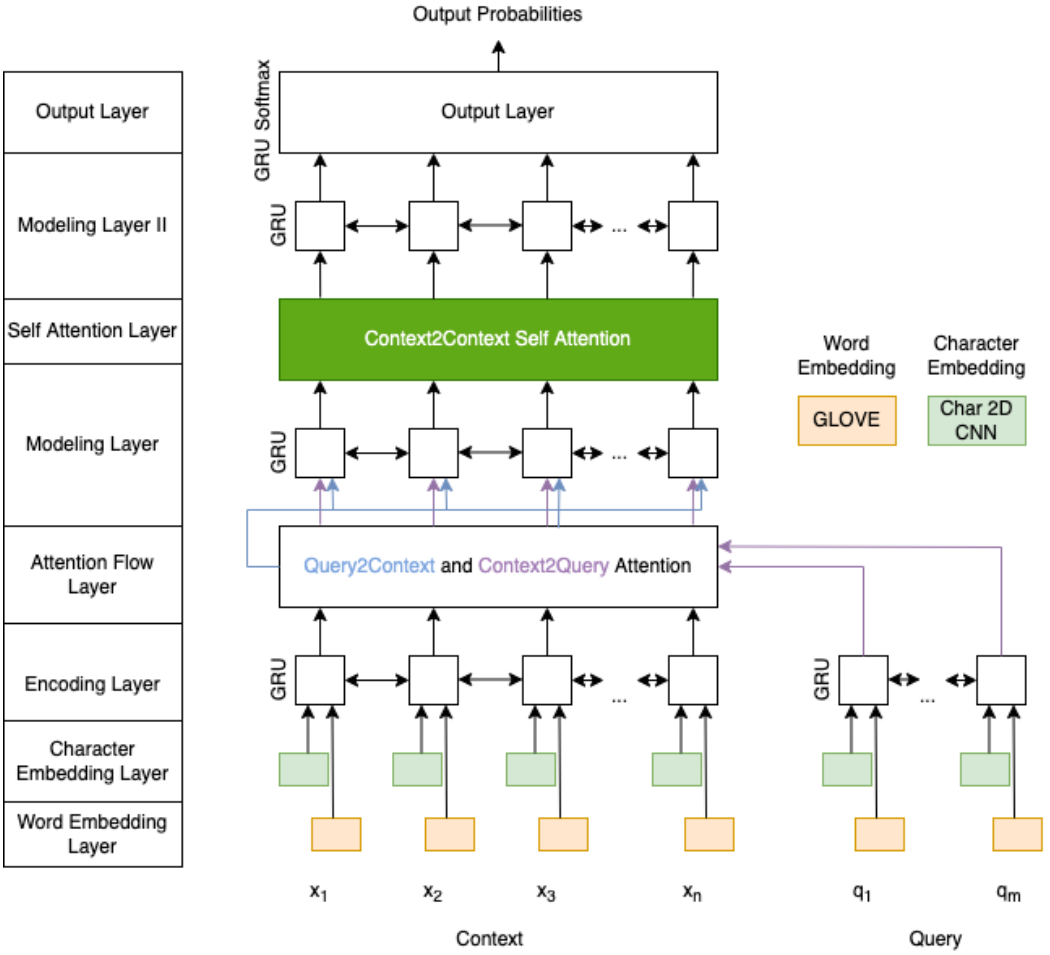
This vector is concatenated with the corresponding word embedding w_{embed} to arrive at the final embedding $[w_{\text{embed}}, c_{\text{embed}}]$ to be used as input to the model's encoding layer. As a result, the size of the hidden states of the revised model is double of that of the baseline.

4.2 Self Attention

I extend the baseline model by adding a Context-to-Context self-attention layer (and second modeling layer) directly before the output layer. This entails directly matching the question-aware context representation (produced by the preceding attention and modeling layer) against itself. The layer does so by computing a similarity score between each pair of context hidden states:

$$S_{ij} = w^T[c_i; c_j; c_i \odot c_j] \in \mathbb{R}$$

Figure 1: Model Architecture Diagram



where $\odot$ is an elementwise product and c_i is the ith context hidden state. These similarity scores form a similarity matrix whose rows are softmaxed and used as weights to calculate a weighted average of context hidden states for each context word.

This attention output is fed into another modeling layer, which is a bi-directional GRU. This modeling layer serves as the final layer before the output layer.

5 Experiments

5.1 Data

I use the provided train, dev, and test sets from the default project, which are in part sourced from the official SQuAD 2.0 training set. The inputs are (context, question) pairs and the outputs are answers, which are spans from the context; in particular, three human-provided answers are given for each question.

5.2 Evaluation method

I used the Gradescope leaderboard provided for the final project, which computes the EM and F1 scores for the provided test and dev sets. The EM score is a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly. The F1 score is the harmonic mean of precision and recall. It can be expressed as $\frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}}$.

The maximum of the F1 and EM scores across all three human-provided answers is taken when evaluating on the dev and test sets.

5.3 Experimental details

I ran my experiments with identical training time as given by the baseline model code. For the final model (character embeddings and self-attention), I used a learning rate of 0.5, dropout probability of 0.2, hidden layer size of 100, and batch size of 64.

I also experimented with an adaptive learning rate using PyTorch's StepLR optimizer, as well as a dropout probability of 0.5, but neither brought improvements to the EM or F1 scores in my testing.

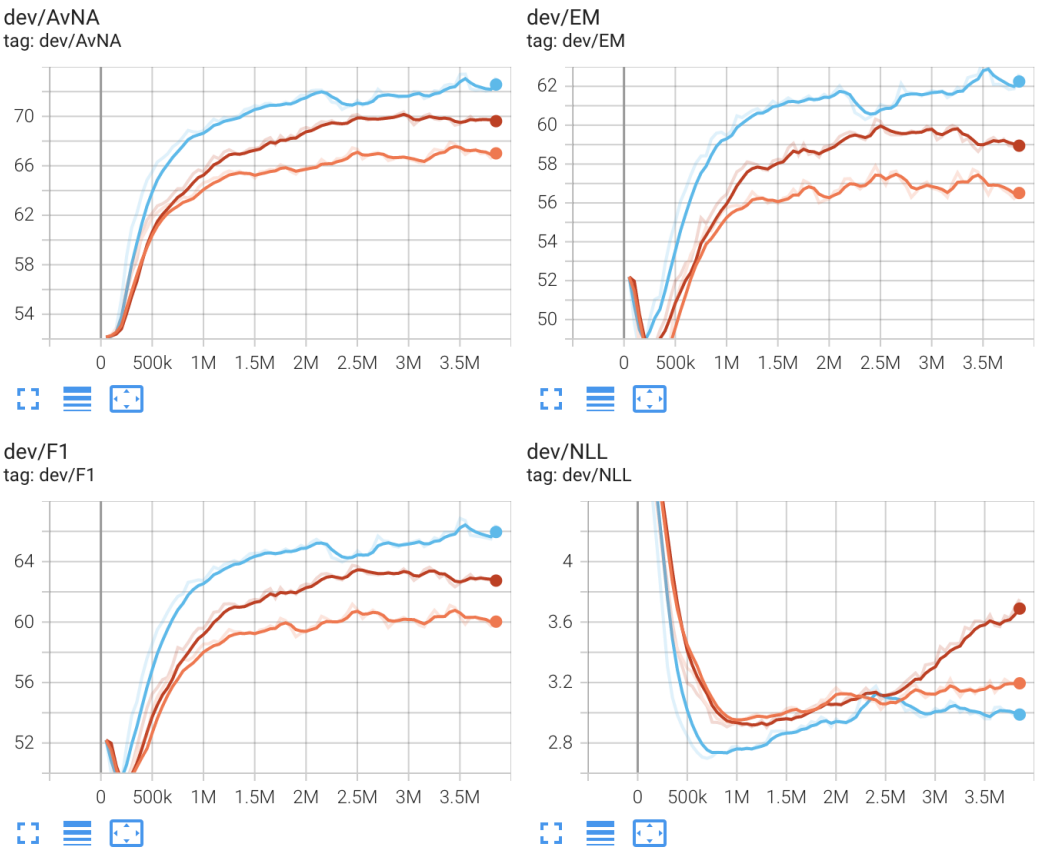
5.4 Results

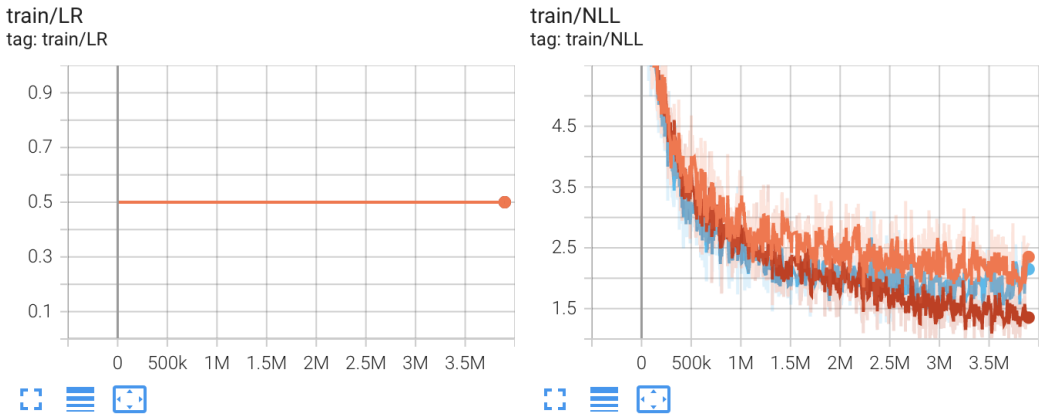
The final model with character embeddings and self-attention achieves EM score of **61.522** and F1 score of **64.799** on the test set.

The below table compares model performance in terms of E1 and F1 scores on the dev set.

Model	Dev set EM	Dev set F1
Baseline	cell2	cell3
CharEmbed	60.309	63.756
CharEmbed + Self-Attention	**63.452**	**66.861**

Below are the Tensorboard graphs of train and dev NLL, as well as EM and F1 dev scores during training. The orange line is the baseline model, the red line is the model with character embeddings, and the blue line is the model with character embeddings and self attention.





6 Analysis

The model extended with character embeddings alone outperforms the baseline, and better handles out-of-vocabulary tokens and punctuation tokens. As an example, the baseline model at times outputs grammatically incorrect answers in terms of punctuation, as demonstrated in the example below:

- **Question:** What is the receptor that killer T cells use to bind to specific antigens that are complexed with the MHC Class 1 receptor of another cell?
- **Context:** Killer T cells are a sub-group of T cells that kill cells that are infected with viruses (and other pathogens), or are otherwise damaged or dysfunctional. As with B cells, each type of T cell recognizes a different antigen. Killer T cells are activated when their T cell receptor (TCR) binds to this specific antigen in a complex with the MHC Class I receptor of another cell. Recognition of this MHC:antigen complex is aided by a co-receptor on the T cell, called CD8. The T cell then travels throughout the body in search of cells where the MHC I receptors bear this antigen. When an activated T cell contacts such cells, it releases cytotoxins, such as perforin, which form pores in the target cell's plasma membrane, allowing ions, water and toxins to enter. The entry of another toxin called granulysin (a protease) induces the target cell to undergo apoptosis. T cell killing of host cells is particularly important in preventing the replication of viruses. T cell activation is tightly controlled and generally requires a very strong MHC/antigen activation signal, or additional activation signals provided by "helper" T cells (see below).
- **Answer:** T cell receptor (TCR)
- **Prediction:** T cell receptor (TCR

The baseline model is able to identify the correct entity for the answer, but truncates too early and removes the closing parentheses, which renders the answer grammatically incorrect. Both (a) the model with added character embeddings and (b) the full model with character embeddings and self attention correctly identify the answer. Given (a), it's clear that adding the character embeddings allows the model to have a better grasp over grammar and punctuation, preventing it from making the same mistake as the baseline.

Below is an example in which the final model outperforms both the baseline and the character-embeddings-only model:

- **Question:** What type of group is The Islamic State?
- **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.
- **Answer:** Wahhabi/Salafi jihadist extremist militant
- **Prediction:** militant

Both the baseline and character-embeddings-only model produce the same answer of "militant", which is somewhat correct, but not complete. In contrast, the final model produces the exact match answer. It seems that the final model's context-to-context self attention layer allows the model to capture all of the leading descriptors for the word "group", since each context word is able to more effectively pool information from the rest of the context in its representation.

Although the final model is able to more effectively encode information across the entire context, it still seems to be easily "misled" by no-answer questions that refer to places, dates, or names in the context but are phrased in a way such that the mention in the context does not answer the question. The example below refers to the action "join" and the date "October 20, 1973" that are mentioned verbatim in the context but have no relation to the entity "Nixon" mentioned in the question.

- **Question:** What did Nixon join on October 20, 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
- **Prediction:** the embargo

This suggests that the model relies too heavily on the appearance of key words/phrases without taking the sentence structure (and its implications on the meaning) into account. To circumvent this weakness, we could benefit from fine-tuning a pre-trained model whose parameters already have been trained on a general language modeling task.

7 Conclusion

This project demonstrates the effectiveness of adding character-level embeddings and self-attention incrementally to improve a baseline SQuAD question answering model. Each iteration of my model improves upon the previous, speaking the power of each separate method. My model is able to improve upon the baseline model, achieving EM and F1 scores of **61.522** and **64.799** respectively on the test set.

Due to limitations in computing power (and specifically, limited Azure credits and the fact that this was a single person project), I was not able to experiment with as much hyperparameter tuning as I would have liked. Extensions for future work include using an adaptive learning rate optimizer, testing more combinations of dropout probabilities and learning rates, and adding more encoder layers and attention blocks to the model. Nevertheless, the improvements made upon the baseline model are substantial and speak to the effectiveness of self-attention and character embeddings in the task of contextual question answering.

References

[1] Konstantin Lopyrev Percy Liang Pranav Rajpurkar, Jian Zhang. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[2] Ali Farhadi Hannaneh Hajishirzi Minjoon Seo, Aniruddha Kembhavi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations (ICLR)*, 2017.

[3] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

[4] Microsoft Research Asia Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. In *Association for Computational Linguistics (ACL)*, 2017.