Building a QA System (IID SQuAD Track

Aaron Li [1] Edgar Roman [1] Tanish Jain [1]

[1]Computer Science, Stanford

Introduction

In this project, we aim to develop a more complex model from a starting baseline model, namely BiDAF without character embeddings, to perform well on the SQuAD 2.0 dataset, which is relevant to evaluating the model's text-comprehension ability. We ultimately find that the integration of character embeddings into the baseline model enhances performance as does the use of the Dynamic Coattention Network on "answerable" questions.

Background: SQuAD 2.0

The SQuAD2.0 dataset combines the existing Stanford Question Answering Dataset (SQuAD) with "unanswerable" questions written by crowdsourced workers to look similar to answerable ones. In order for our question answering model to perform successfully on this dataset, it must first determine if the question is answerable at all, and if so, answer it appropriately.

Article: Super Bowl 50
Passage (from Wikipedia): *"Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.*
Question 1: *"What is the name of the quarterback who was 38 in Super Bowl XXXIII?"*
Answerability for Q1: Unanswerable (Number Swap)
Question 2: *"Who is the youngest quarterback ever to play in a Super Bowl?"*
Answerability for Q2: Unanswerable (Antonym)

Figure 1. SQuAD 2.0: Example Unanswerable Questions

Methods

For our approach, we wanted to test out a few different techniques from prior work, and then build upon the most promising one or combine them in a meaningful way.

Baseline

Our baseline model is based on Bidirectional Attention Flow (BiDAF). However, unlike the original model, the implementation of this baseline model does not include a character-level embedding layer.

BiDAF with Character Embeddings

We extend the baseline model to match the original BiDAF-No-Answer model. To implement character embeddings, we added a character embedding layer which utilizes a simple Convolutional Neural Network (CNN), whose output is then max-pooled in order to obtain a fixed-size vector for each word. To integrate the embeddings into our model, we later concatenate both the character and word embeddings, which is ultimately passed into a two-layer Highway Network just as in the original BiDAF model.

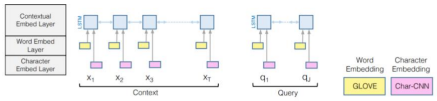

Figure 2. Integration of Character Embeddings

Source: Seo et al.

Dynamic Coattention Network

We also implemented a Dynamic Coattention Netowrk Model This approach was picked for its ability to use bidirectional attention, much like the first approach, alongside the introduction of a dynamic pointing decoder which goes over answer spans. This can be particularly useful, since this allows the model to "explore local maxima corresponding to multiple plausible answers". The decoder can then produce start and end indices of the answer, as shown in Figure **??**.

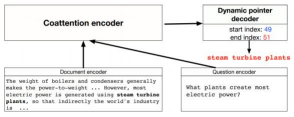

Figure 3. Overview of DCN Encoder-Decoder Setup

Source: Xiong, Zhong and Socher.

The implementation was built on top of the provided baseline implementation, with a few key modifications.

- **Addition of coattention layer.** The attention layer of the baseline model was modified into a coattention layer.
- **Addition of a dynamic pointing decoder.** Another key feature of the DCN is the dynamic pointing decoder. At each iteration, the decoder state is updated using previous estimates of the start and end positions to generate the new estimates of the start and end positions.

Experiments

Data & Metrics

The evaluation was done using the SQuAD 2.0 dataset, which consists of triples containing questions, contexts and answers [5]. Specifically, the models were evaluated in the dev set in the provided data splits. The metrics used to evaluate were the F1 and EM scores.

Hyperparameters

Hyperparameter	Value
Learning Rate	0.5
Size of Char Embedding	1376
Size of Hidden Units	100
Dropout	0.2
Optimizer	Adam
Batch Size	64

Table 1. BiDAF (baseline + character embeddings) Hyperparameters

Hyperparameter	Value
Learning Rate	0.001
Size of Embedding	100
Size of Hidden Units	100
Dropout	0.3
Optimizer	Adam
Batch Size	64

Table 2. DCN Hyperparameters

Results & Analysis

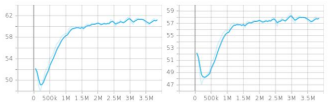

Figure 4. Baseline: F1 & EM vs Number of Iterations

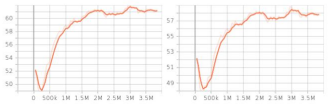

Figure 5. BiDAF: F1 & EM vs Number of Iterations

The results of each of the models are summarized in Table 3.

Model	F1 Score	EM Score
Baseline (BiDAF w/o Character Embeddings)	61.769	58.629
BiDAF	62.107 (+0.338)	58.814 (+0.185)
Dynamic Coattention Network	58.704 (-3.065)	57.269 (-1.361)

Table 3. Summary of model performance

The BiDAF model performs only marginally better than the baseline model. By introducing character-level embeddings, enabling the conditioning on the internal structure of words and the handling of out-of-vocabulary words (OOV). Instead of simply allowing the model to deal with OOV words by having GloVe assign them a random embedding, character level embeddings allow us to find better embeddings by focusing on the character-level compositions of these OOV words.

The Dynamic Coattention Network model's performance falls below that of the baseline. This is worse than expected, since the addition of the coattention layer and dynamic pointer were seen as improvements in prior work. Interestingly, most of the prediction error appears in the "unanswerable" questions, where the model performs poorly. This may be indicative of the inability of the model to generalize well to unanswerable questions, especially since the original model was developed using the SQuAD dataset, which only contained answerable questions.

Conclusion

At the time of the relase of SQuAD2.0, strong neural systems that performed well on SQuAD 1.1 performed nowhere near as well on SQuAD 2.0. Now, systems such as IE-NET, achieve 90%+ F1 scores, so it is incredible to think about the progress that was made in such a short amount of time. Nonetheless, the models in this particular project are not quite as impressive in terms of the results. However, we learn that although character embeddings certainly help models understand language better, they are insufficient. We also learn that a Dynamic Coattention Network model can work well if it can also determine whether a question is unanswerable or not due to the sole model's inability to generalize well to unanswerable questions.