Question-Answering with QANet for SQuAD 2.0

Stanford CS224N Default Project

Martin Altenburg
Department of Computer Science
Stanford University
altenbur@stanford.edu

Swathi Iyer
Department of Computer Science
Stanford University
swathii@stanford.edu

Luke Miller
Department of Computer Science
Stanford University
lukedm@stanford.edu

Abstract

Our task for this project is to is to design a question-answering system for the SQuAD 2.0 dataset that improves upon the BiDAF baseline model. To do this, we experiment with QANet, a transformer-based architecture. We also reintroduce a character-level embeddings on top of the provided BiDAF model, as well as a self-attention layer. Our best QANet model achieved 61.47/64.81 EM/F1 scores on the test set.

1 Introduction

Question-answering is an important NLP task, because it can gauge how well NLP systems understand text. Provided a question and a passage of text, the model should correctly output the span of text in the paragraph that answers the question. In recent years, with the production of massive, labeled datasets, researchers have been able to build large neural systems to automate question-answering, leading to significant advancements in the field. Practically, these systems can be used in a variety of information-retrieval applications, such as database lookup or online customer service.

In this project, we build on top of the baseline BiDAF model [3] to design a QA system that performs well on the SQuAD question-answering dataset. The SQuAD dataset is a reading comprehension dataset, where passages are Wikipedia articles, and each passage has a corresponding question. The answer to every question is segment of text from the passage that answers the question. SQuAD 2.0, the version used in this paper, also includes unanswerable questions, for which the system must be able to determine that it is unanswerable rather than reporting an incorrect answer.

Our focus for this paper was to experiment with QANet [8], a transformer-based architecture. The Transformer model has become the state-of-the art for NLP tasks since its introduction in 2017. Prior to this, many models relied on LSTM or GRU architectures. The main problem with these models was their recurrent nature, which causes longer sequences to require very long runtimes. The transformer architecture forgoes this recurrence, and instead leverages an attention mechanism to learn dependencies between the inputs and outputs. In this paper, we experiment with this model for the SQuAD question-answering dataset. We also reintroduce character-level embeddings and multi-headed self-attention to the BiDAF model for comparison.

While the baseline code has been provided for us in the starter code, our implementations of the transformer-based architecture, character-level embeddings and self-attention have been done ourselves here: https://github.com/swathiiyer2/squad.

2 Related Work

There has been rapid advancement in automating question-answering in recent years, largely due to the rise of massive labeled datasets and subsequent design of neural systems around these datasets. We have employed some of these designs in our own question-answering system. Our baseline model is a version of the one presented in the original Bidirectional Attention Flow (BiDAF) paper [3], which can be thought of as a bidirectional LSTM with a unique bidirectional attention flow layer.

The R-Net paper [1] introduces a self-attention mechanism. The self-attention mechanism effectively matches a passage against itself, by collecting evidence from the entire passage for each word in the passage, and encodes the relevant information for each word in the passage into the passage representation. This allows the model to collect context from the entire passage to answer the question.

The QANet paper [8] presents a simple and effective Transformer model, in which the encoder block is composed of only convolutions and a self-attention mechanism. The goal of this design is that convolutions can learn the local structure of the text, while self-attention can learn global context. In comparison to BiDAF and other recurrent models, QANet relies on an attention mechanism to learn dependencies between the inputs and outputs. Thus, large sequence lengths will not lead to exploding runtime. The attention mechanism also allows for more parallelism than other RNNs.

3 Approach

3.1 Baseline

The baseline model provided for us is based on the original Bidirectional Attention Flow (BiDAF) model [3]. While the original model uses both character and word-level embeddings, the provided starter code does not include the former.

The idea of the BiDAF layer is to have attention, as the name suggests, flow bidirectionally from context to the question, as well as from question to context. The attention layer computes a similarity matrix S between context hidden states, $c_1, ..., c_n$ and question hidden states, $q_1, ...q_n$. In similarity matrix S, each element S_{ij} contains a similarity score between each pair of context and question hidden states, (c_i, q_j). This allows us to normalize across rows or columns of the similarity matrix to attend to the questions or context, respectively [3].

3.2 Character-level Embeddings

Character-level embeddings are better equipped to handle out-of-distribution words or words that happen infrequently than word-level embeddings, because the characters in these words have still learned good embeddings. We were able to use a convex neural network to combine the initial character embeddings. The original BiDAF model makes use of character embeddings in addition to word embeddings, and we have reintroduced this layer to the provided baseline. The initial structure came from the starter code of Assignment 5 from 2020. However, the convex neural networks and contents of the character embeddings were implemented ourselves.

3.3 Self-Attention

In practice, traditional recurrent neural networks can only memorize a limited amount of context. Self-attention aims to widen the amount of context usable for question-answering; this is especially important when, as in the SQuAD dataset, the question lexically differs from the passage. To improve the BiDAF model's performance, we introduce the self-attention mechanism originally described in [1]. Namely, our self-attention layer is placed directly after bi-directional attention (replacing our RNN) and imlemented within our transformer encoding. We implement a multi-headed, masked self-attention layer.

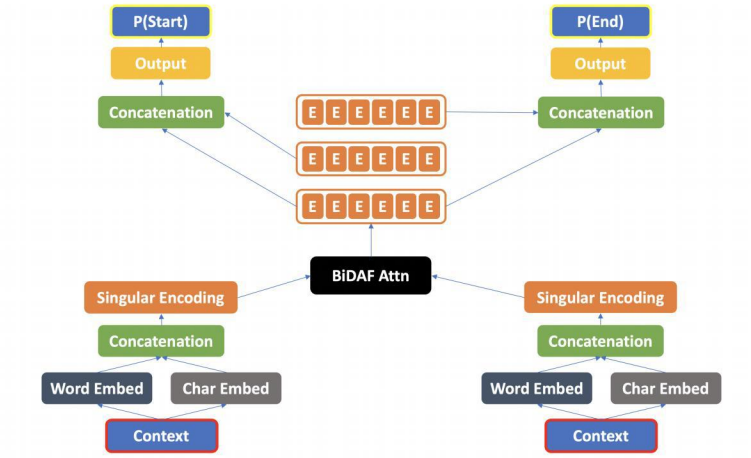


Figure 1: The QANet transformer model

3.4 QANet

3.4.1 Input Embedding Layer

Our input embedding layer includes both word and character embeddings. This layer performs an embedding lookup to convert word indices into word embeddings, for both the context and question. For word embeddings, we use pretrained 300-dimensional GloVE embeddings, which are fixed during training. For the character embeddings, we initialize them to be pretrained vectors, so that we get a matrix of learnable parameters, $W_{proj} \in \mathbb{R}^{HxD}$. This is then passed to convolutional and max pooling layers to get an H-dimensional projection for each embedding. Finally, we concatenate the word and character embeddings, and apply a two-layer Highway Network presented in [4]. Each layer uses distinct learnable parameters. One layer of this network is computed as follows:

$$g = \sigma(W_g h_i + b_g) \in \mathbb{R}^H$$
$$t = ReLU(W_t h_i + b_t) \in \mathbb{R}^H$$
$$h_i' = g \odot t + (1 - g) \odot h_i \in \mathbb{R}^H$$

3.4.2 Embedding Encoder Layer

The embedding encoder layer has four main components: a positional encoding layer, convolution layer, self-attention layer, and a feed-forward network. The positional encoding is added to each word embedding to give the model signal as to the order of the words. The calculation for positional encoding is the same as in the Attention is All you Need paper [6], as follows, where "p" is the position of the word in the sequence, "d" is the size of the embedding, and "i" is the position in the embedding:

$$PE_{2i}(p) = \sin(p/10000^{2i/d})$$
$$PE_{2i+1}(p) = \cos(p/10000^{2i/d})$$

Once our initial position encodings are calculated for the context and query, they remain constant for our remaining sections.

The depthwise separable convolution is a convolution kernel where the number of filters is the depth of the top layer, and each channel is convoluted separately to get the desired number of feature maps. The self-attention layer is the same one used for BiDAF (refer to section 3.3), and the feed-forward network is a standard fully connected network [8].

3.4.3 Context-Query Attention Layer

Let C and Q be the encoded context and query. The context-query attention is computed in the same way as the QANet paper [8]. First, we take the similarity between each pair of context and query

words to form a similarity matrix, $S \in \mathbb{R}^{nxm}$. We then normalize each row of S by applying softmax to form $\bar{S}$. The context-to-query attention is computed by multiplying the normalized similarity matrix with the encoded query: $A = \bar{S} \cdot Q^T$. We also compute the query-to-context attention by normalizing $\bar{S}$ column-wise, and computing the following: $B = \bar{S} \cdot \bar{\bar{S}}^T \cdot C^T$.

3.4.4 Model Encoder Layer

Our model encoding layer uses an archeture that is simliar to our embedding encoding layer except that the set of encoding blocks is repeated and the number of convolutions in each block is lower. The Model Encoding layer can be split into three stages. Within each stage, the embedding encoding block (with self-attention and a feed-forward layer) is applied 6 times. The output of the first two blocks S_1 and S_2 are respectively fed into the second and third blocks S_2 and S_3.

3.4.5 Output Layer

We model our output layer to be similar to [3] for our output layer. Like in the starter code, this layer is created specifically to determine the conditional probability of the start/end of an answer given an index. Our code models these probabilites based on the following equation:

$$p^{begin} = \text{softmax}(U_1[S_1; S_2])$$
$$p^{end} = \text{softmax}(U_2[S_1; S_3])$$

As shown in the diagram, S_1 is the output from the first set of Model Encoders, S_2 is the output from the second set, and S_3 is the output from the third set. These probabilities are then fed into the objective function:

$$L(\theta) = -\frac{1}{N}\sum_i^N [\log(p^1_{y_i^1}) + \log(p^2_{y_i^2})]$$

4 Experiments

4.1 Data and Evaluation Method

Our experiments were conducted on the SQuAD 2.0 dataset. We measured our model's performance using EM and F1 scores as evaluation metrics, which are the metrics used in the SQuAD leaderboard.

4.2 Experiments

We trained and tested four different models: three BiDAF models and one QANet transformer model. For all models, we trained for 30 epochs using the Adam optimizer with $\beta_1 = 0.8$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$ with F1 score as the target metric. We trained the three BiDAF models using a hidden size of 100, and learning rate of 0.5, and dropout rate of 0.2. For QANet, we used a hidden size of 128, learning rate of 0.001, and dropout rate of 0.1.

For the first model, we made use of the provided baseline BiDAF model. Second, we added character-level embeddings to the baseline model by reintroducing the embeddings originally described in [3]. Third, we added the self-attention mechanism described in [1]. We compare these three models to our final model, a re-implementation of the QANet transformer model described in [8]. For the QANet model, we conducted hyperparameter search around the number of attention heads, testing 1, 2, 4, and 8 heads. Our final model made use of 4 attention heads. However, our hyperparameter search reamains in the investigation stage, and given more time we would have liked to conduct a more thorough analysis of the effect of various hyperparameters on the performance of QANet.

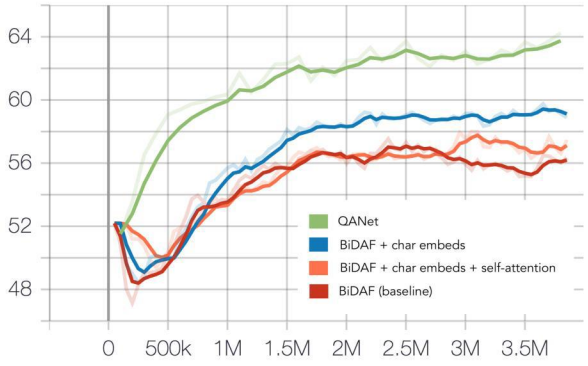
4.3 Results

Table 1 summarizes the results of our training on the four major models, comparing the EM and F1 scores on the development set. We can see that adding character-level embeddings clearly improves upon the baseline model, as expected. Contrary to our expectations, however, adding a self-attention layer to that model actually decreased performance. While we did not conduct a thorough investigation surrounding the reasons for this, we believe that this decrease in performance

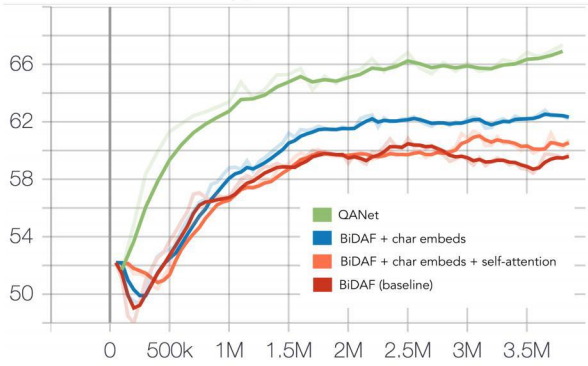
Table 1: The results of each model on the development set

Model	EM	F1
BiDAF (baseline)	57.32	60.84
BiDAF + char embeddings	59.75	62.80
BiDAF + char embeddings + self-attention	58.01	61.35
QANet (4 attention heads)	64.24	67.34

is due to the sensitivity of the model to the number of self-attention layers. Finally, we see that implementing QANet improves significantly upon all the BiDAF models, resulting in EM/F1 scores of 64.24/67.34 on the development set and 61.47/64.81 on the test set.



(a) EM scores



(b) F1 scores

Figure 2: EM and F1 scores throughout training for each model

In addition to reporting the final results for each model, we also analyzed how training progressed over time. We can see that for all three BiDAF models, the training process began with a decrease in EM and F1 scores on the development set for the first 250,000 to 500,000 iterations, followed by a steady increase in scores until plateauing after approximately 2.5 million iterations. For the transformer model, on the other hand, the period of decreasing scores was extremely short, and while the rate of improvement decrease significantly after about 1.5 million iterations, the training process did not seem to plateau completely even by the time training had completed. We attribute this phenomenon to the fact that transformer models are much more flexible than the BiDAF model, and yet in this case did not seem to overfit the training data even after 30 epochs of training. Given more time and computational resources, we would like to have experimented with training the transformer model for longer than 30 epochs.

5 Analysis

5.1 Advantages of Character Embeddings

Question: What is the rarest cause of poor immune function in developing countries?

Context: Immunodeficiencies occur when one or more of the components of the immune system are inactive. The ability of the immune system to respond to pathogens is diminished in both the young and the elderly, with immune responses beginning to decline at around 50 years of age due to immunosenescence. In developed countries, obesity, alcoholism, and drug use are common causes of poor immune function. However, malnutrition is the most common cause of immunodeficiency in developing countries. Diets lacking sufficient protein are associated with impaired cell-mediated immunity, complement activity, phagocyte function, IgA antibody concentrations, and cytokine production. Additionally, the loss of the thymus at an early age through genetic mutation or surgical removal results in severe immunodeficiency and a high susceptibility to infection.

Answer: 'N/A'

Predictions:

- BiDAF : 'malnutrition'
- BiDAF with character embeddings: 'N/A'

Analysis: We found that the BiDAF model augmented with character embeddings regularly performed better than the BiDAF model when the context contained many words mapped to the <UNK> token. While in the baseline BiDAF model these words would simply represent lost information, the model with character embeddings is able to represent these words similarly to morphologically similar words. In the example above, the baseline model represents "immunosenescence" with <UNK>, while the character embeddings model represents it with vectors close in vector-space to words like "immunology" and "immunocompromised." This allows the model to pick up on additional information and achieve better performance.

5.2 Advantages of QANet

Question: Particle physics has created a Unique Model to describe what?

Context: With modern insights into quantum mechanics and technology that can accelerate particles close to the speed of light, particle physics has devised a Standard Model to describe forces between particles smaller than atoms. The Standard Model predicts that exchanged particles called gauge bosons are the fundamental means by which forces are emitted and absorbed. Only four main interactions are known: in order of decreasing strength, they are: strong, electromagnetic, weak, and gravitational.:2–10:79 High-energy particle physics observations made during the 1970s and 1980s confirmed that the weak and electromagnetic forces are expressions of a more fundamental electroweak interaction.

Answer: 'N/A'

Predictions:

- BiDAF: 'forces between particles smaller than atoms'
- BiDAF with character embeddings: 'forces between particles smaller than atoms'
- QANet: 'N/A'

Analysis: The QANet model shows better performance on a variety of question types compared to the BiDAF models. Compared to recurrent models, transformers are able to more effectively model longer dependencies between words and phrases in the input text, which is especially relevant to the QA task. Transformers' multi-headed self-attention mechanism and positional encodings also help the model to better represent relationships between words and thus achieve better performance. In the above example, the QANet model is able to recognize that a 'Standard Model' is not the same thing as a 'Unique Model' and accurately predict that the answer to the question is not found in the text.

5.3 Limitations of All Models

Question: What can the exhaust steam not fully do when the exhaust event is insufficiently long?

Context: The simplest valve gears give events of fixed length during the engine cycle and often make the engine rotate in only one direction. Most however have a reversing mechanism which additionally can provide means for saving steam as speed and momentum are gained by gradually "shortening the cutoff" or rather, shortening the admission event; this in turn proportionately lengthens the expansion period. However, as one and the same valve usually controls both steam flows, a short cutoff at admission adversely affects the exhaust and compression periods which should ideally always be kept fairly constant; if the exhaust event is too brief, the totality of the exhaust steam cannot evacuate the cylinder, choking it and giving excessive compression ("kick back").[citation needed]

Answer: 'evacuate the cylinder'

Predictions:

- BiDAF: 'N/A'
- BiDAF with character embeddings: 'N/A'
- QANet: 'N/A'

Analysis: All of the models we tested had generally poor performance on questions where the correct answer is a verb, and they usually predicted 'N/A' even though an answer could be found in the context. This is likely because, when a verb is the intended answer, finding that answer involves tracing the logic of the question and the context more deeply than most other questions. In this example, we can see that the question asks about the outcome of a specific condition ("when the exhaust event..."), how it relates to an entity ("the exhaust steam"), and finally, the result of that condition on the entity. These kinds of complex questions are difficult to answer for any of the models we tested.

6 Conclusion

In this paper, we compare performance between a recurrent model, the baseline BiDAF model, and a Transformer-based model, QANet. We found that in comparison to the recurrent model, even after the addition of character embeddings and self-attention layers, the Transformer-based model performs better. Our best model was QANet with 4 attention heads, which achieved EM/F1 scores of 61.47/64.81 on the test set. This is likely due to the fact that the recurrent nature of BiDAF limits its ability to learn long-term dependencies between words in longer sequences. In contrast, the QANet model includes positional embeddings and multi-headed self-attention to better learn the relationship between words in a passage.

One avenue we would have liked to explore more is the BiDAF model with self-attention layers. Interestingly, this model performed worse than the BiDAF model with only character embeddings. With more time permitting, we would have liked to conduct a more in-depth hyperparameter search for the self-attention layer. We also found that increasing the number of attention heads in QANet led to better performance. However, at 8 attention heads, we ran into memory issues. It would be interesting to explore larger models with different memory constraints. State-of-the-art models seem to get larger as time goes on (take GPT models for example). It would be also interesting to explore at which point compute costs and training time for these larger models outweigh any performance benefits.

References

[1] Michael F Coulas, Glenn H MacEwen, and Genevieve Marquis. Rnet: A hard real-time distributed programming system. *IEEE Transactions on computers*, 100(8):917–932, 1987.

[2] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

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

[4] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[5] Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. Why self-attention? a targeted evaluation of neural machine translation architectures. *arXiv preprint arXiv:1808.08946*, 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. *arXiv preprint arXiv:1706.03762*, 2017.

[7] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*, 2015.

[8] 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. *arXiv preprint arXiv:1804.09541*, 2018.