Exploring the Impacts of Character Modeling, Co-Attention, and Dual Attention on BiDAF Performance

Stanford CS224N Default Project

Sophia Andrikopoulos
Department of Computer Science
Stanford University
sophiala@stanford.edu

Abstract

Question-answering is a key area of focus in the NLP community, and has been improved in recent years with the implementation of neural attention mechanisms. The BiDirectional attention Flow (BiDAF) has made strides in question-answering. However, more complex attention mechanisms have been developed in recent years that may further improve question-answering models. In this paper, I explore the impacts of added character-level embeddings, a CoAttention mechanism, and a Dual BiDirectional-CoAttention mechanism to the BiDAF model. When evaluated on the Stanford Question Answering Dataset 2.0 (SQuAD 2.0), BiDAF with character-level embeddings and Dual Attention out-perform the simpler baseline.

1 Key Information to include

- Mentor: Kaili Huang
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Question-answering (QA) is a highly relevant area of focus in the Natural Language Processing (NLP) community. QA systems are both useful tools in themselves, and provide us with a better understanding of how well machines encode human language. Massive improvements have been made to QA models in recent years with the introduction of neural attention mechanisms.

In this paper, I focus on exploring one such model, BiDirectional Attention Flow (BiDAF). While I do not propose any significant novel components, I explore the addition of character-level embeddings to BiDAF. Further, I examine the impacts of CoAttention, a secondary attention model, on BiDAF performance in order to better understand the functions of both BiDirectional and CoAttention and the interaction between the two.

3 Related Work

This exploration interprets previous work in the QA domain. Namely, I focus on the BiDAF Model. Previous to the work of Seo et al., QA models typically attended to small portions of the context with uni-directional attention. BiDAF proposed a revolutionary BiDirectional attention mechanism that represents query-to-context and context-to-query attention.[1] BiDAF was originally implemented with word and character embeddings. The character embeddings draw directly from

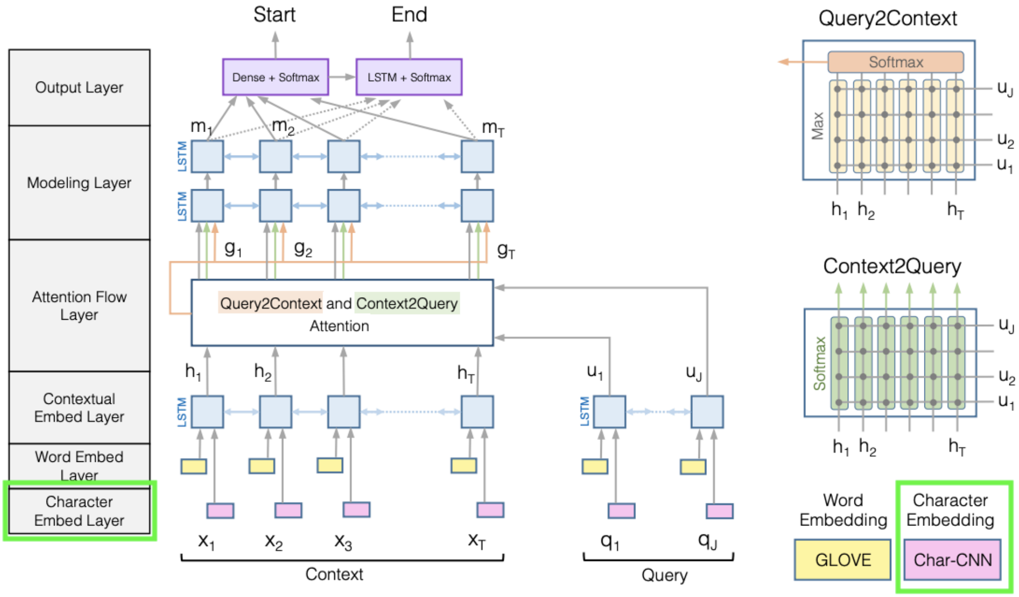

Figure 1: BiDirectional Attention Flow model. This paper's baseline model does not include the character-level embeddings highlighted in green. [1]

Kim's *Convolutional Neural Networks for Sentence Classification*, whcih uses a single convolutional layer and max-pooling to build character representations.[2]

Additionally, I focus on Xiong et al.'s *Dynamic Coattention Networks For Question Answering*.[3] Previously, attention mechanisms including BiDirectional attention only performed a single pass, attending directly to context and question hidden states. Xiong et al. highlight the downsides of single-pass attention mechanisms, namely that they cannot recover well from local maxima, and propose the CoAttention mechanism. CoAttention attends over first-level bidirectional attention, outputting a second-level attention representation.

4 Approach

The baseline model is based on the BiDirectional Attention Flow model (BiDAF). BiDAF consists of a character- and word-Embedding layer, an Encoder layer, a BiDirectional Attention layer, a Modeling layer, and an Output layer.1

In the embedding layer, a pre-trained word embedding for both the questions and contexts is looked up by index and passed through both a dropout and highway layer. The encoding layer pads the embeddings to a fixed length and passes them through a bi-directional LSTM to learn word representation. For N context hidden states $\mathbf{c}$ and M question hidden states $\mathbf{q}$, the BiDirectional attention layer computes a similarity matrix $\mathbf{S}$ and takes the sums of $\mathbf{q}$ weighted by the row-wise Softmax of $\mathbf{S}$ to compute Context-to-Question attention a_i for $i \in N$. Question-to-Context attention b_i is computed by summing $\mathbf{c}$ weighted by the column-wise Softmax of $\mathbf{S}$. Finally the BiDirectional attention output $g_i = [c_i; a_i; c_i \circ a_i; c_i \circ b_i]$ is obtained, where $\circ$ denotes elementwise multiplication and ; denotes concatenation. The modeling layer refines the output of the attention layer and is similar in structure to the encoding layer. Finally, the output layer generates a vector of probabilities that the correct answer lies at each index in the context.

The baseline model's embedding layer only contains word-level embeddings. First, I aim to improve the provided baseline model by re-integrating character-level embeddings. The BiDAF character-level embeddings are implemented by converting the baseline model's provided character indices into the provided pre-trained character embeddings, a nearly identical process to creating the word-level embeddings. The character-embeddings for each word are then passed into a Convolutional Neural Network (CNN) implemented using pytorch and max-pooled to obtain a vector for each word.[4] Both the word-level embeddings and the character-level embeddings are initialized with half the hidden size of the model and ultimately concatenated in the forward pass to generate the final embeddings. The character-level embeddings are used in all further explorations of the model.

Next, I seek to explore the impacts of substituting the BiDirectional attention layer with a CoAttention layer. CoAttention includes a secondary attention computation that attends over the first-level attention output. First, a linear layer with a tanh nonlinearity is applied to the question hidden states q, with learnable weights W and bias vector b to obtain $q^{'}$. Next, randomly-initialized sentinel vectors are concatenated to c and $q^{'}$, allowing a word in the context to attend to none of the words in the question and vice-versa. Next, an affinity matrix $\mathbf{L} = c^T q$ which contains the affinity scores for each context and question hidden state is computed. The Context-to-Question attention distributions α are computed by taking the column-wise Softmax of $\mathbf{L}$. Computing the sum of q' weighted by α results in the Context-to-Question attention outputs a. Computing the sum of c weighted by the row-wise Softmax of $\mathbf{L}$ yields the Qeustion-to-Context attention outputs b. Finally, α is used to take the weighted sum of b, yielding the second-level attention outputs s. s and a are concatenated and fed through a recurrent neural network to produce the final output u. Xiong et al. opt for a bi-directional LSTM. However, I opt for a two-layer gated recurrent unit (GRU) in this implementation.

Both the attention layers discussed in this paper have their shortcomings–BiDirectional attention only performs a single pass at each step and thus cannot recover from local maxima. Conversely, CoAttention provides a second-level attention but may not pass useful aspects of first-level BiDirectional attention to the next layer. In order to try and overcome these shortcomings, I explore the effects of a combination of BiDirectional attention and CoAttention, refered to as Dual Attention in this paper. Both BiDirectional attention and CoAttention as previously described are computed and concatenated in the Dual Attention layer as the layer's output.

5 Experiments

5.1 Data

For this task, I use the Stanford Question Answering Dataset 2.0 (SQuAD 2.0) provided by the CS224N teaching team. Each SQuAD 2.0 example consists of a (context, question, answer) triple. The answers to each question can be selected directly from the context paragraph, but roughly half the questions do not have an associated answer.[5] Each answerable question corresponds to three human-provided answers, which may not be exactly the same per example. The data from the original SQuAD 2.0 test and dev sets has been split into 129,941 training examples, 6078 dev examples, and 5915 test examples. The goal of our model is to produce the correct answer to a given question (either a span from the context paragraph or 'NA').

5.2 Evaluation method

Model performance was measured using the following metrics:

1. **Exact Match (EM) Score** - a binary True/False measure of whether the model output for an example matches the ground truth answer exactly.
2. **F1 Score** - the harmonic mean of precision and recall:

$$F1 = 2 * \frac{precision * recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

3. **Negative Log-Likelihood Loss (NLL)** - the chosen loss function for training (a secondary metric)
4. **Answer vs. No Answer (AvNA)** - a secondary metric, the classification accuracy of the model when only considering its answer vs. no answer prediction

During evaluation, the maximum EM and F1 scores from the three possible correct answers are taken for each example, and then averaged over the dataset to obtain the final metrics.

5.3 Experimental details

All models were initialized with the parameters listed in Table 1 to optimize F1 score. The baseline model took approximately 2 hours to train on a virtual machine, while training the character-level embedding took approximately 5 hours. Both the CoAttention model and the Dual attention model took approximately 10 hours to train.

Parameter	**Value**
batch size	64
dropout probability	0.2
EMA decay rate	0.999
evaluation steps	50000
hidden layer size	100
learning rate	0.5
max answer length	15
max gradient norm	5
epochs	30
random seed	224

Table 1: Training parameters for both BiDAF baseline and added character-level embedding models.

5.4 Results

Adding character-level embeddings to the baseline BiDAF model improved F1, EM, NLL, and AvNA, as shown in Table 2. The CoAttention layer model resulted in better results than the baseline model, but was outperformed by the simple character embedding model. Finally, the Dual Attention model outperformed its competitors on both the dev and test sets.

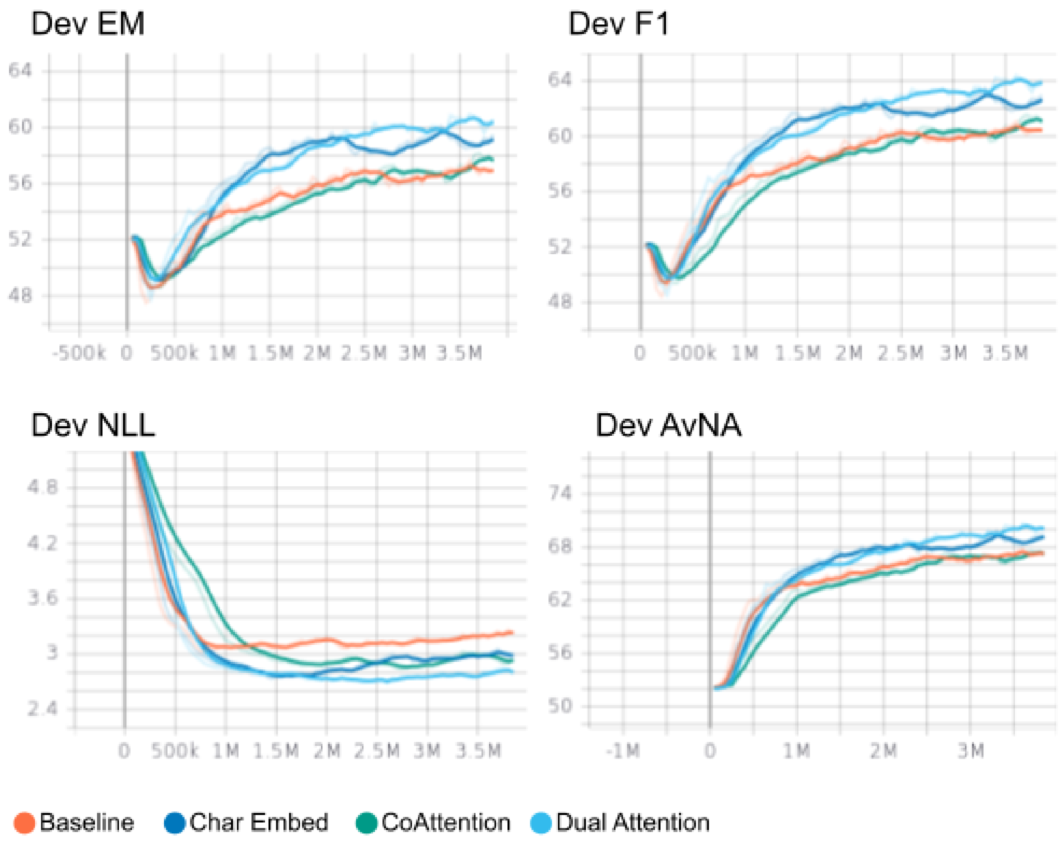

Figure 2: Model performance leveled out around Epoch 30 for all models. Dual Attention outperformed all models, followed by the the character-embedding addition to the baseline. While the CoAttention model displays minor improvement over the baseline, it performs weakly compared to other models explored.

	F1	**EM**
BiDAF Baseline	60.412	56.898
BiDAF with Character-Level Embeddings (dev)	63.445	60.208
BiDAF with Character-Level Embeddings (test)	62.490	58.969
BiDAF with CoAttention	62.012	58.578
BiDAF with Dual Attention (dev)	**64.396**	**60.931**
BiDAF with Dual Attention (test)	**63.926**	**60.338**

Table 2: BiDAF performance on the dev set improves with added character-level embeddings compared to the baseline model, but performance is negatively impacted by substituting Bidirectional Attention with CoAttention. Improvements are seen on the test and dev sets with the Dual attention model over BiDAF with character-level embeddings. Model outputs are evaluated using both F1 and EM scores.

6 Analysis

The incorporation of character embeddings to the BiDAF baseline improved both metrics as expected, given their performance in the original BiDAF paper.[1] Character-level embeddings likely improve upon the model by capturing sub-word meanings, allowing the model to handle previously unseen words and therefore to generalize outside of the training set.

Replacing the BiDirectional attention layer of the character-embedding inclusive model with a CoAttention layer worsened the model's performance. This drop in performance could be attributed to a poor implementation of the CoAttention layer. More likely, CoAttention's second-level attention output passed less useful information to future layers compared to BiDirectional Attention, indicating that first-level attention is an important aspect of the model.

The Dual Attention model outperformed all other models, supporting the theory that the information passed to future layers using first-level attention improves model performance. Further, the concattenation of CoAttention to BiDirectional attention outperformed both attention mechanisms on their own, suggesting that the CoAttention layer was likely implemented correctly, and that CoAttention contains useful secondary information that further supports the information passed in by BiDirectional attention. Perhaps additionally AvNA was improved by the addition of sentinel vectors in CoAttention, allowing the model as a whole to perform a little better when there is no relation between question and context. It is also likely that the size of the layer (double that of the single attention layers) improved the model simply by passing on significantly more information to future layers.

7 Conclusion

This exploration highlights important attributes of the BiDAF model. Primarily, character embeddings allow the model to generalize outside of the training set to unseen words by capturing sub-word meaning, greatly improving BiDAF performance. Secondly, BiDirectional attention is an integral component of BiDAF–as the name suggests–because it provides future layers with important first-level context-to-question and question-to-context information. On its own, CoAttention negatively impacts BiDAF performance, likely because second-level attention does not carry enough information on its own to support the model. This is supported by the high performance of Dual Attention, suggesting the second-level information of CoAttention positively complements the more powerful BiDirectional attention.

While I have improved upon the BiDAF baseline with character embeddings and Dual Attention, there is still great room for exploration and improvement within BiDAF and the SQuAD 2.0 question-answering task. The success of character-level embeddings suggests that BiDAF might be further improved by exploring the effects of larger embeddings, or n-gram embeddings. Further, there are several attention models such as Self Attention that could be substituted into BiDAF or concatenated to the BiDirectional attention output similar to the Dual model that may provide the model with further information not explored in this paper. Finally, with greater resources, it is essential to explore hyperparameter tuning in order to maximize the performance of the model.

References

[1] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. volume abs/1611.01603, 2016.

[2] Yoon Kim. Convolutional neural networks for sentence classification. volume abs/1408.5882, 2014.

[3] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. volume abs/1611.01604, 2016.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc,

E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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