Dynamic Coattention Networks for Question Answering

Fangchen Sun, Alexander Jermann, Yuanhang Zhang

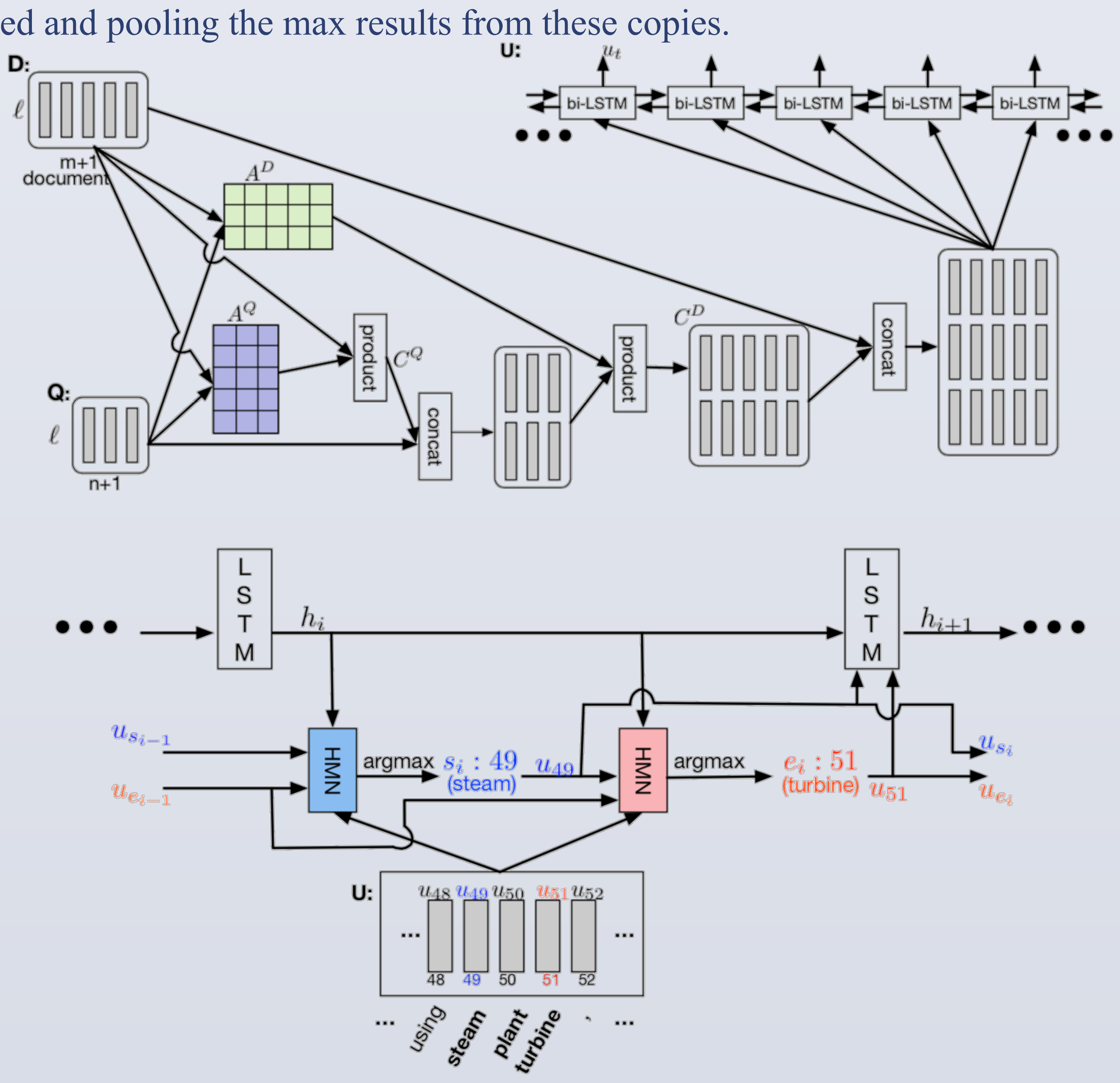
University of Oxford, Computer Science Department

INTRODUCTION

The objective of the final project is to replicate the key aspects and the results of a selected research paper. Our team selects the paper "Dynamic Coattention Networks for Question Answering" (Xiong, 2017)[1]. The paper strives to solve the problem of machine question answering on the SQuAD 1.1 dataset released by Stanford University and proposes an end-to-end Dynamic Coattention Network (DCN) model which inputs documents and questions and outputs answers directly. The model includes a coattention encoder to generate co-dependent representations of the documents and questions and a dynamic decoder to iterate through multiple potential answers in order to avoid local maxima. My team also extends the method in the original paper and proposes several improvements such as adding an additional Double Cross Attention layer (Hasan, 2018)[2] in the coattention encoder. After implementing the model, we are able to achieve 64.1% F1, which improves the baseline model by 24.1% from 40.0% F1.

MODELS & METHODS

The Dynamic Coattention Network has two major parts: a coattention encoder and a dynamic decoder. The coattention encoder has two parts. The model first encodes the given document and question separately via the document and question encoder. The document and question encoders are essentially a one-directional LSTM network with one layer. Then it passes both the document and question encodings to another encoder which computes the coattention via matrix multiplications and outputs the coattention encoding from another bidirectional LSTM network. The dynamic decoder is also a one-directional LSTM network with one layer. The model runs the LSTM network through several iterations. In each iteration, the LSTM takes in the final hidden state of the LSTM and the start and end word embeddings of the answer in the last iteration and outputs a new hidden state. Then, the model uses a Highway Maxout Network (HMN) to compute the new start and end word embeddings of the answer in each iteration. HMN has three consecutive maxout layers with a highway connection between the first and third maxout layer. A maxout layer consists of several copies (equal to pool size) of fully connected linear layers connected and pooling the max results from these copies.



IMPLEMENTATION & IMPROVEMENTS

To start we used GloVe 6B word embeddings with dimension 300. Similarly to the original paper, we set out-of- vocabulary words to zero. We set the max document length to 600 and the max question length to 30 and the hidden dimension for recurrent units, maxout layers, and linear layers to 200. To ensure the generality of our model, we used Dropout ratio of 0.15 for regularizing our neural networks. As opposed to the original paper we decided to omit sentinel vectors because in our experiments the performance increase was close to insignificant. For the dynamic decoder we started with a maximum number of iterations of 4, highway maxout pool size of 16. Finally, we started with a batch size of 200 and a learning rate of 0.0003.

We have three improvements to the model in the original paper. One major improvement and two minor improvements. The major improvement is that we use a Double Cross Attention (DCA) to replace the coattention mechanism in the Dynamic Coattention Network (DCN) paper. The DCA performs almost the same operations as the DCN to compute the attention encodings. The only difference is that it adds another attention layer to compute a second affinity score matrix before computing the last attention matrix. By adding an extra layer, it increases the complexity of the model and puts more weights on the question encoding in the coattention encoding.

The first minor improvement is that we convert the LSTM in the document and question encoder to a bidirectional LSTM so that it can also include the encoding from the end to start and add more representational space. The second minor improvement is that we initialize the forget gate bias to 1 for all LSTM networks in the model since it empirically improves performance (Jozefowicz, 2015)[3].

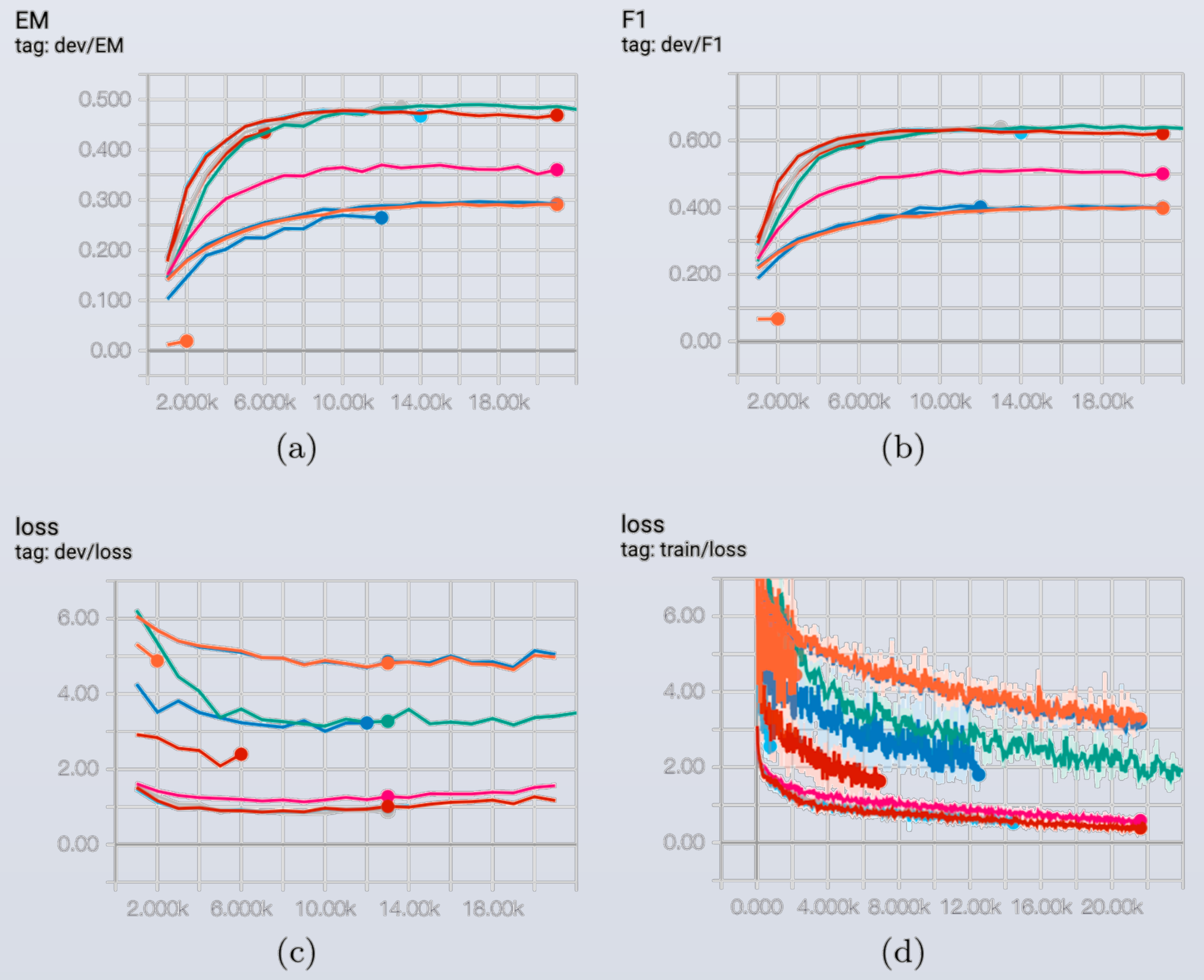
EXPERIMENTS & RESULTS

We ran two types of experiments, one to replicate the results from the original paper and the other to aim to improve the performance by applying new methods. To start we wrote and evaluated a baseline model using a GRU as encoder and a simple fully connected linear layers as decoder to predict the start and end of the answer span and got an Dev EM of 29.5% and Dev F1 of 40.2%. For the first type of experiments, we ran four experiments on the decoder side by experimenting with different pool sizes for the Highway Maxout Network and single decoder iteration. Like the paper, we found that a pool size of 16 provides the best performance with Dev EM 48.8% and Dev F1 of 64.0%, because it is taking into consideration more models in the maxout layer helping with model variety and averaging. For the second type of experiment, we introduced a Double Cross Attention mechanism and got the result of Dev F1 of 58.8%. The decrease in performance is most likely due to the lack of hyperparameter tuning.

We believe that our model does not perform as well as the original experiments in the paper because we ran our experiments with the GloVe 6B pre-trained word embeddings with vocabulary size of 400'000 instead of GloVe 840B pre-trained word embeddings with vocabulary size of 2'200'000. We used a smaller corpus because we did not have enough GPU memory to run a batch size over 10 with GloVe 840B and because we could train the model faster with GloVe 6B and thus run several experiments.

Model	Dev EM	Dev F1	Dev EM	Dev F1
Dynamic Coattention Network (DCN)	*Our Model*		*Xiong et al.*	
HMN pool size 16	**48.8**	**64.0**	**65.4**	**75.6**
HMN pool size 8	47.8	63.3	64.4	74.9
HMN pool size 4	37.0	51.3	65.2	75.2
DCN with double cross attention	43.7	58.8	-	-
Baseline	*Our Model*		*Xiong et al.*	
GRU with MLP layers	29.5	40.2	40.0	51.0

Table 1: Single model ablations on development set of our model compared to Xiong et al. 2018.



(a) (b)

(c) (d)

Figure 1: Visualization of Tensorboard

CONCLUSION & FUTURE PLAN

We are able to replicate the models in the original paper and suggest improvements to the attention mechanism. We believe that the deficiency in our model compared to the original paper is in part due to use of a smaller vocabulary of the pre-trained word embeddings.

For future work, we would attempt to use the GloVe 840B given more GPU memory and time to run experiments and BERT embeddings published by Google. Further we would implement an ensemble method to further increase the performance.

REFERENCES

[1] Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever. "An empirical exploration of recurrent network architectures." International Conference on Machine Learning. 2015.

[2] Xiong, Caiming, Victor Zhong, and Richard Socher. "Dynamic coattention networks for question answering." arXiv preprint arXiv:1611.01604 (2016).

[3] Hasan, Zia, and Sebastian Fischer. "Pay More Attention-Neural Architectures for Question-Answering." arXiv preprint arXiv:1803.09230 (2018).