Machine Comprehension using BiDAF

Bimal Parakkal
Stanford Center for Professional Development
Stanford University
Stanford, CA 94305
bimal84@stanford.edu

Abstract

Bi-directional Attention Flow(BiDAF) network is a recent deep learning model which has been quite successful in the Machine Comprehension problem domain. This report details the study I did using a re-implementation of the complete BiDAF network as laid out in Bi-Directional Attention Flow for Machine Comprehension (Minjoon Seo et al). My BiDAF implementation got an F1 score 72.6 and EM 62.8 on the Test leaderboard. The effect of changing word vector embedding size was investigated and the experiments showed that while Glove embedding size impacted learning time of BiDAF model, the final performance metrics of the models were quite close to one another. The observation was interpret qualitatively using examples. Finally an ensemble of the different models with varying word vector dimensions was created which improved upon the performance of individual models as anticipated.

1 Introduction

The Wikipedia article on Reading comprehension states that *the fundamental skills required in efficient reading comprehension are knowing meaning of words, ability to understand meaning of a word from discourse context, ability to follow organization of passage and to identify antecedents and references in it, ability to draw inferences from a passage about its contents, ability to identify the main thought of a passage, ability to answer questions answered in a passage, ability to recognize the literary devices or propositional structures used in a passage and determine its tone, to understand the situational mood (agents, objects, temporal and spatial reference points, casual and intentional inflections, etc.) conveyed for assertions, questioning, commanding, refraining etc. and finally ability to determine writer's purpose, intent, and point of view, and draw inferences about the writer.*

Thanks to recent advances in Deep learning, Machines have been able to make progress on many of the tasks outlined above independently. The problem of Machine comprehension combines many of the tasks under a single umbrella. Until recent years it was difficult to make progress on Machine comprehension due to unavailability of a high quality dataset. This domain has been made vastly accessible after Stanford Question Answering dataset (SQuAD) was released.

SQuAD consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. With 100,000+ question-answer pairs on 500+ articles, SQuAD is significantly larger than previous reading comprehension datasets.

Bidirectional Attention Flow (BiDAF) network is a new deep learning model which has been quite successful in the Machine Comprehension Domain. I've attempted re-implementing BiDAF model and studied the effects of varying word embedding vector size on the performance of the model.Two metrics are used to evaluate the model: Exact Match (EM) and a softer metric, F1 score, which measures the weighted average of the precision and recall rate at character level. T

2 Background and Related Works

The open sourcing of Stanford Question Answering Dataset (SQuAD) by Rajpurkar et al. [4] represents a landmark event in the progression of Machine Comprehension research. The SQuAD leaderboard (https://rajpurkar.github.io/SQuAD-explorer/) is a living, breathing document of the on-going research in the field of Machine Comprehension research. The best performing models in there are only a couple of months old at the time of writing this document. Some of the other good models which have been around slightly longer have inspired more research in the area.

The Dynamic Coattention Networks(DCN) [1] is an end-to-end neural network architecture for question answering. The DCN consists of a coattention encoder which learns co-dependent representations of the question and of the document, and a dynamic decoder which iteratively estimates the answer span. The encoder's purpose is to create coattention mechanism that attends to the question and document simultaneously and fuses both attention contexts in the output. The Dynamic decoder is similar to a state machine whose state is maintained by an LSTM-based sequential model.

The Bidirectional Attention Flow (BiDAF) network[1] includes character-level, word-level, and contextual embeddings, and uses bi-directional attention flow to obtain a query-aware context representation. This report outlines study based on re-implementation of the complete network outlined in the original paper.

3 BiDAF Model

3.1 Architecture

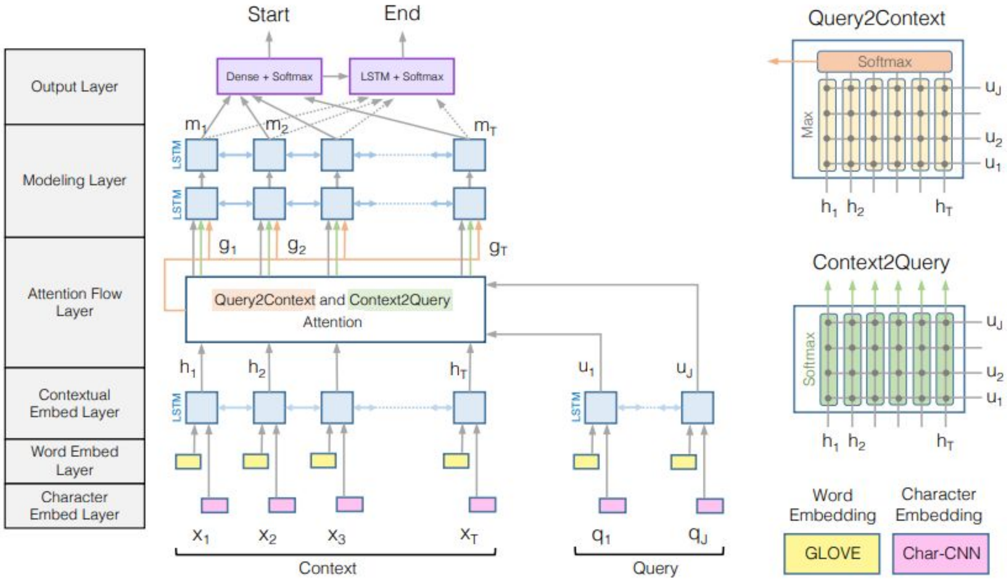

Figure 1: (Seo et al.,2017)

Figure-1 is a graphical representation of the multi-stage architecture from Seo et., 2017. The purpose the different layers are briefly described below. For additional details on the network architecture please refer to the original paper

1. **Character Embedding Layer** maps each word to a vector space using character-level CNNs.
2. **Word Embedding Layer** maps each word to a vector space using a pre-trained word embedding model.
3. **Contextual Embedding Layer** utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context.
4. **Attention Flow Layer** couples the query and context vectors and produces a set of query aware feature vectors for each word in the context.
5. **Modeling Layer** employs a Recurrent Neural Network to scan the context.
6. **Output Layer** provides an answer to the query

3.2 Implementation details

For CNN char embedding 100 1D filters, each with a width of 5 were used. Glove representations of varying sizes 50/100/200/300 were used for Word embedding in the different iterations of the model. The hidden state size (d) of the model was set of 200. I used the Adam[2] (Kingma & Ba, 2014) optimizer, with a minibatch size of 60 and learning rate of 0.001. A dropout (Srivastava et al., 2014)[5] rate of 0.15 was used for the CNN, all LSTM layers, and the linear transformation before the softmax for the answers. The training process was run for 15K iteration takes roughly 18 hours on a single Nvidia Tesla P100 GPU. Also an ensemble model was trained consisting of the four training iteration with the identical architecture and hyper-parameters except for Glove word vector embedding size. The ensemble was based on a voting strategy with a tie broken by the model iteration with Glove vector embedding size 200, which happened to the best performing single model based on F1 score.

4 Results and Discussion

4.1 Model F1 & EM score comparison with Prior work

In spite of implementing all layers on the original BiDAF network, I was not able to replicate the stellar performance of the model on SQuAD dataset. This could have been partially due to lack of hyper parameter tuning or due to any model implementation errors which I have not been able to identify. Due to lack of training time, hyperparameter search was restricted to choices of Glove word vector embedding sizes and optimizer learning rates alone.

Table 1: Scores on SQuAD dev dataset

	Single Model		**Ensemble**	
	EM	F1	EM	F1
Match-LSTM	64.7	73.7	67.9	77.0
Dynamic Co-attention network	66.2	75.9	71.6	80.4
R-Net	68.4	77.5	72.1	79.7
BiDAF(original paper)	68.0	77.3	73.3	81.1
BiDAF(my implementation)	**62.0**	**72.2**	**66.7**	**76.0**

4.2 Impact of Glove word vector embedding size

Iterations of BiDAF network were trained using all the four available Glove word vector embedding size - 50/100/200/300. It was found that with a higher word embedding size, the model was faster to train at the cost of additional parameters in the model of-course. However when allowed to near convergence(Figure-2), the model with embedding size 50 eventually caught up with the performance of model with embedding size of 300. This probably highlights the shortcoming in my model that my implementation of the BiDAF network and was unable to lock in the information in the additional dimensions into network parameters and reach a better state of performance. Further investigation needs to be done if the model was suffering from case of diminishing gradients especially towards to initial layers.

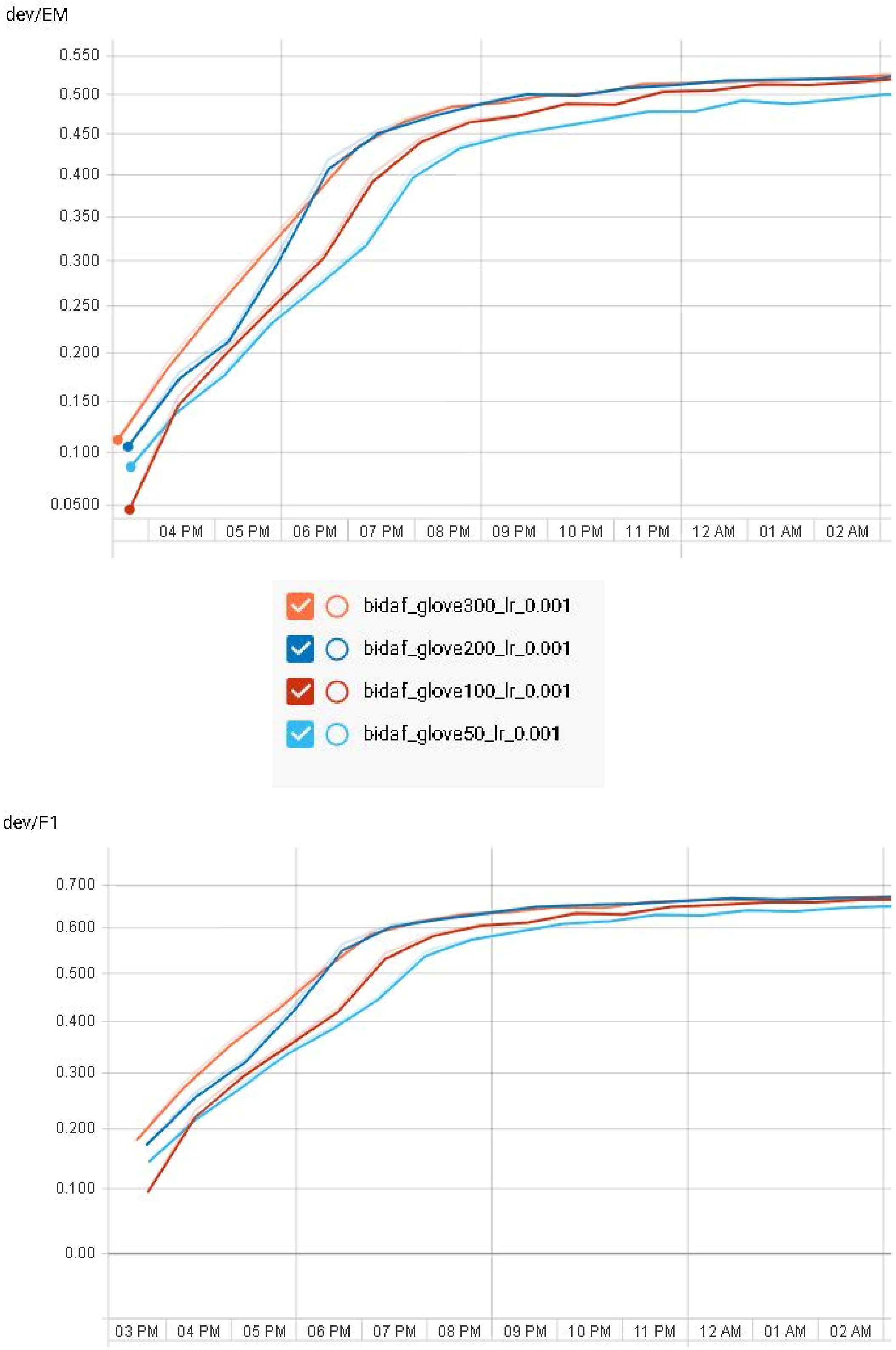
dev/EM
0.550
0.500
0.450
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.0500
04 PM
05 PM
06 PM
07 PM
08 PM
09 PM
10 PM
11 PM
12 AM
01 AM
02 AM
bidaf_glove300_lr_0.001
bidaf_glove200_lr_0.001
bidaf_glove100_lr_0.001
bidaf_glove50_lr_0.001
dev/F1
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.00
03 PM
04 PM
05 PM
06 PM
07 PM
08 PM
09 PM
10 PM
11 PM
12 AM
01 AM
02 AM

Figure-2

5 Conclusion

BiDAF implementation improved the performance significantly over baseline but achieving the performance outlined in the paper requires additional effort involving hyper parameter tuning and monitoring gradient updates in the different layers of the network. Higher Glove vector word embedding sizes resulting in faster training of the model but did not lead to better performance when allowed to run to convergence

Acknowledgments

The author likes to thank TAs for timely guidance and tips on Piazza for managing the project and improving the model.

References

[1] Caiming Xiong, Victor Zhong, and Richard Socher. “Dynamic coattention networks for question answering”. arXiv preprint arXiv:1611.01604, 2016b.

[2] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[3] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” arXiv preprint arXiv:1611.01603, 2016.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” arXiv preprint arXiv:1606.05250, 2016.

[5] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.