Question Answering with Deep Bidirectional Attention Flow and FusionNet

Silviana Maria Ciurea Ilcus, Michal Wegrzynski
Department of Computer Science
Stanford University
Stanford, CA
{smci, michalw} *at stanford.edu*

Abstract

In this paper, we re-implement FusionNet and compare it with a deep version of the Bidirectional Attention Flow model, as applied to the Stanford Question and Answer Dataset (SQuAD). Both the simpler and the more sophisticated architecture leverage some form of LSTM skip connections. We observe that skip connections are responsible for most of the models' performance in the early training stages. Both models achieve 74.0-74.2% F1 on SQuAD.

1 Introduction

Machine comprehension is a critical problem in natural language processing, as it requires an algorithm to exhibit an understanding of a passage of text on multiple semantic levels. A common task used to develop and test machine comprehension is question answering. The variation of the problem that we worked on is a standard formulation as described by Rajpurkar et al. in their 2016 introduction to the Stanford Question and Answer Dataset, and consists of providing the algorithm with a passage of text (context), a question, and a ground truth label in the form of a smaller contiguous passage within the context that constitutes the answer to the question. The model is evaluated on the amount of overlap between its prediction and the true answer to the question, as labeled by three human subjects. Recent advancements in deep-learning architectures have pushed the performance of machine comprehension models closer to a human baseline, but this area of research is still very active; the recently released architecture that we applied on this model is still awaiting publication at ICLR 2018.

2 Dataset

The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) contains over 100,000 question-answer pairs, posed on 500+ Wikipedia articles. The split of the dataset is 80% for the training set, 10% for the development set, and 10% for the hidden test set.

We perform some basic analysis on the dataset in the form of histograms of the context length, question length, and answer length in the training set. We find that in the training set, most paragraphs do not exceed 300 words (Figure 1, left), that there are almost no questions longer than 30 words (Figure 1, center), and that most answers are shorter than 10 words. Based on this information, we use a context length of 300, question length of 30, and maximum length of answer of 15. Reducing the context length and question length results in decreasing the training time.

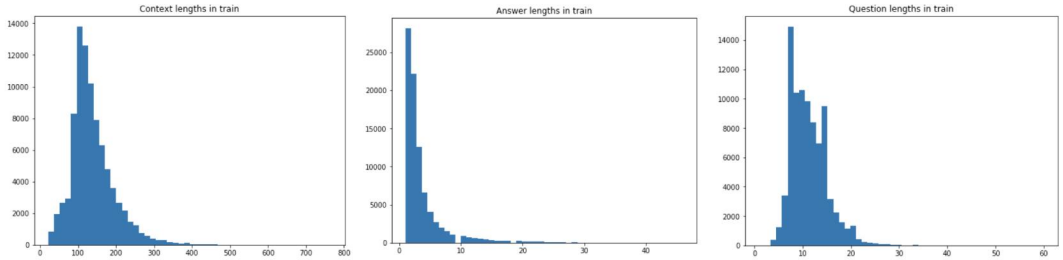


Figure 1: Histogram of context length (left), answer length (center), and question length (right) using the data from the train portion of SQuAD

3 Approach and Related Work

3.1 Baseline

For the baseline, we use the default project baseline code, which employs basic dot-product attention, with the encoded context attending to the encoded question. The resulting attended context embedding is concatenated with the previous encoded context, to produce a blended representation of the context. This blended representation is then passed to a fully connected layer, which features a ReLU non-linearity. The predictions for the start and end tokens are computed independently of each other, by passing the output of the fully connected layer to a softmax layer.

3.2 Bidirectional Attention Flow model (BiDAF)

The BiDAF model (Seo et al., 2017) is a top-scoring model on SQuAD, whose core idea is to avoid the negative effects of early summarization using attention by using bidirectional, context-to-query and query-to-context, memoryless attention to obtain a question-aware representation of the context.

We chose to implement two elements of the original model: the bidirectional attention and the modeling layer. We decided not to implement the convolutional character embedding, since this feature added less than 2% to the F1 score in the original paper. Since we did not use character embeddings, we also did not implement the two-layer Highway Network (Srivastava et al., 2015) that combines the character and word embeddings in the original paper. We chose to integrate the Attention Flow and the Modeling Layer into our model, and then targeted three potential areas for improving our simplified BiDAF model.

3.2.1 Attention Flow and Modeling Layer

We implemented the Attention Flow layer from (Seo et al., 2017), which computes attention from the context to the query, and from the query to the context, producing a query-aware representation of the context. (Figure 2) (Seo et al., 2017). We also implemented the two-layer bi-directional RNN-based modeling layer, which takes as input the query-aware representation of the context and captures the interaction between the context words conditioned on the query (Seo et al., 2017). The outputs of the modeling layer are passed onto the output layer, for the answer prediction step. We re-implemented the architecture of the output layer used by Seo et al, where the end token prediction is not made independently of the start token prediction as in our baseline model.

3.2.2 Modifications to BiDAF

We identified two potential areas of improvement for the simplified BiDAF model we created: using more complex word embeddings, and deepening the modeling layer while simultaneously introducing skip connections in it.

First, we extended the model to use additional input features at the word embedding step; this was inspired by the recent success on SQuAD of the DrQA (Chen et al., 2017) and FusionNet (Huang et al., 2018) models, both of which supplement the word vectors with features such as exact match (i.e. whether a word in the context can be matched exactly to a word in the question), Part-Of-

Speech tag, Named Entity type, and Aligned Question Embedding. We chose to focus on Aligned Question Embedding, based on the feature ablation analysis for DrQA (Chen et al., 2017). The analysis identified both the aligned question embedding and the exact match feature to be equally important; ablating either one of them decreased the model's F1 score by 1.5%, and ablating both led to a decrease in F1 of 19.4%. We decided to only implement one of the two features because of the diminishing returns of implementing the second feature in the context of time constraints.

The Aligned Question Embedding is a function f_{align} on each context word c_i defined as follows:

$$f_{align}(c_i) = \sum_j a_{i,j}\boldsymbol{E}(q_j), \qquad where\ a_{i,j} = \frac{\exp\left(\alpha(\boldsymbol{E}(c_i))\right) * \exp\left(\alpha\left(\boldsymbol{E}(q_j)\right)\right)}{\sum_{j'} \exp\left(\alpha(\boldsymbol{E}(c_i))\right) * \exp\left(\alpha\left(\boldsymbol{E}(q_{j'})\right)\right)}$$

and $\alpha(\bullet)$ is a single dense layer with ReLU nonlinearity. This feature introduces soft alignments between partial synonyms, such as *car* and *vehicle*. (Chen et al., 2017)

Second, we added a third layer to the BiDAF modeling layer, and introduced skip connections in order to avoid potential backpropagation issues that could be caused by deepening the architecture. We investigated how a deeper LSTM architecture performs in the context of the BiDAF model because deep LSTMs have been shown to outperform shallower architectures in language-related tasks such as machine translation and language modeling (Sutskever et al., 2014), and reading comprehension (et al., 2015). Sutskever et al. demonstrated that deep LSTMs significantly outperform shallow LSTMs, employing a four-layer deep LSTM in their Sequence-to-Sequence model for Machine Translation (Sutskever et al., 2014). Since deep stacked RNNs are oftentimes difficult to train, we opted to add shortcut connections across the three different layers. Shortcut connections, also called skip connections, allow unimpeded information flow across different layers (Raiko et al., 2012; Graves, 2013; Hermans and Schrauwen, 2013), and have been shown to be effective in deep stacked models (He et al., 2015; Srivastava et al., 2015; Wu et al., 2016). While this addition increases the computation time, we opted for this improvement as a proof of concept, since optimizing training time is outside the scope of our project. However, recent research has started identifying techniques, such as shortcut blocks (Wu et al., 2017), that allow including shortcuts in deep stacked RNNs without a significant increase in training time.

3.3 FusionNet

FusionNet (Huang et al., 2018) is a new architecture from Microsoft, which performs very well on SquAD, where it achieves an F1 score of 83.9 and EM of 76.9, on Adversarial SquAD (Jia et al., 2017), and on Natural Language Inference tasks on the MultiNLI corpus (Williams et al., 2017). Adversarial SquAD consists of two adversarial evaluation schemes based on SquAD, which aim to test whether the models trained on SquAD truly understand language. Many of the top-scoring models on SquAD perform poorly on these evaluation schemes; sixteen of the published models for SquAD, including BiDAF, drop from an average F1 score of 75% to 36% (Jia et al., 2017). Under both adversarial evaluations (i.e. AddSent and AddOneSent), the FusionNet model outperforms BiDAF's F1 score by ~15%. We chose to implement FusionNet because of its consistently good performance across different tasks that test natural language understanding, and because of the novel concepts it proposes.

The motivation behind this work is that none of the existing fusion mechanisms used by the top-scoring SquAD models employ all levels of word representation jointly, which the authors believe prevents the model from gaining a holistic understanding of each word in a given context and question. As such, FusionNet has three main innovative contributions. First, it introduces a novel way to represent words in language understanding contexts, through a "History of Word" (HoW). This captures the representation of a word from the word embedding level, through the multiple layers of attention, until the final high-level understanding layer.

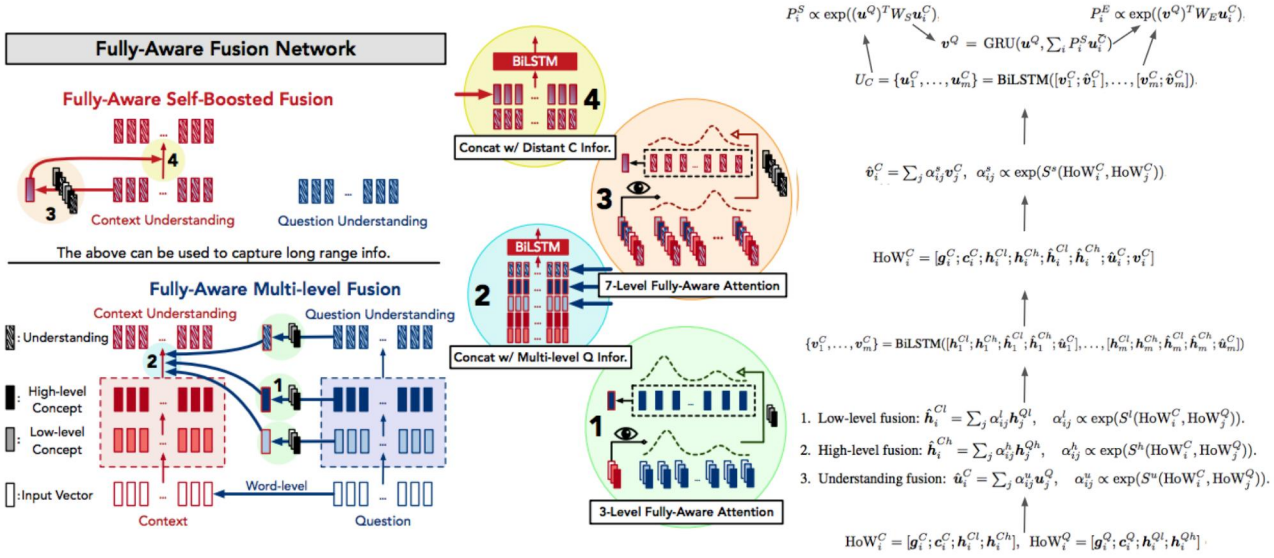
Second, this model also introduces a novel attention scoring function that utilizes the "History of Word" representation and enables multiple levels of attention (i.e. attending at word embedding level, or at later layers corresponding to the words being represented as higher-level, semantic concepts). The attention score is defined as $S_{i,j} = f\left(U(HoW_i^A)\right)^T D\, f(U(HoW_i^B))$, where $f(x)$is an activation function applied element-wise (both in the paper and in our project, $f = ReLU$), D is a diagonal matrix, and HoW_i^A is the History of Word i in text A. For this architecture, this attention score provides much better performance than multiplicative, scaled multiplicative, or

simple symmetric attention, as the authors show in an ablation study (Huang et al., 2018).

Third, the authors of FusionNet propose a multi-level attention mechanism based on this attention scoring function, which helps capture the different conceptual levels of a word. The mechanism, its integration in the network, as well as the high-level architecture of the network are presented in Figure 2.

We implemented the entire Microsoft FusionNet architecture as it was described in the original paper, with the exception of: the CoVE embeddings (McCann et al., 2017), fine-tuning the GloVE embeddings of the top-1000 most frequently occurring words, and the exact match, Named Entity type and Part of Speech tags. We chose to omit the aforementioned features because they were shown in the ablation study performed in the paper to only contribute marginally to the F1 score.

Figure 2: The high-level architecture of the FusionNet we implemented and the History of Word and attention mechanisms at every main step.



4 Experiments and Results

4.1 Embeddings experiments

Throughout our embeddings experiments, we compare the performance of the BiDAF + 2 Modeling Layers model (i.e. the simplified version of the original BiDAF model described in 3.2.1, called simply 'our model' in this section) with that of the same model using each of the extra embedding-related features. All embeddings-related evaluation is done on the tiny-dev data.

4.1.1 Word vector dimensions

To determine which size of GloVE embeddings to use, we tested the performance of our model when using word vectors of dimension 50, 100, and 200. We observed that the model was performing significantly worse with vectors of dimension 50, but that the vectors of dimension 100 and 200 performed very similarly. Therefore, we chose to proceed with using word vectors of dimension 100.

Table 1: F1 score on tiny-dev using different embedding-level features

	F1
Only GLoVE embeddings	66.8
GLoVE + Aligned Question Embeddings	67.1

	F1
Fixed <UNK>	63.1
Trainable <UNK>	66.8
200	66.85

GLoVE dim	F1
50	63.1
100	66.8
200	66.85

4.1.2 Training the <UNK> token

Many top-performing reading comprehension models (DrQA, FusionNet, BiDAF etc.) to train one common word vector for the unknown words encountered. To determine the exact impact on the F1 score of training the <UNK> token, we zero-initialized the <UNK> token and made it a trainable variable (as opposed to the rest of the word vectors). Training the token for unknown words improved the F1 score of our model on the tiny-dev dataset by ~0.2%, at no significant additional computational cost; we decided to train the token in all our subsequent experiments.

4.1.3 Aligned Question Embeddings

Using aligned question embeddings improved the F1 score of our model on tiny-dev by 0.3%. However, using this additional embedding increased training time significantly due to increasing the number of parameters of our RNN Encoders. Since this embedding is only applied to the context, the context embedding size increases while that of the question remains constant, rendering it impossible to use the same RNN Encoder for both the question and the answer, as in the BiDAF + 2 Modeling Layers model. Hence, using this extra embedding requires training two different encoders for the question and the answer. Moreover, the question encoder will have additional parameters, due to the increase in the question embedding size. We considered that an improvement of 0.3% in the F1 score did not warrant slowing the training process this much, so we decided to not include aligned question embeddings in the rest of our BiDAF-related experiments.

4.2 LSTMs and GRUs

Our initial implementation of BiDAF + 2 Modeling Layers model uses exclusively GRUs. Replacing all GRUs in the model with LSTMs renders a 1.5% increase in F1 on tiny-dev, from 66.8% to 68.3%, without a significant increase in training time. We hence decided to proceed using LSTMs instead of GRUs.

4.3 Dropout

Across our different BiDAF experiments, we observed that BiDAF would overfit fairly quickly with dropout=0.15. We used dropout=0.2, as in the original paper, on all the LSTM layers. The higher dropout did prevent overfitting, as can be seen in the dev/loss function, where instead of the train loss decreasing and the dev loss increasing, the train and dev losses remain close in value, with the dev loss flattening out (Figure 4). However, overall the BiDAF model with higher dropout achieved worse results than the one with smaller dropout. For FusionNet, we used dropout=0.3, as in the original paper; we did not tune this parameter further, as we did not observe overfitting.

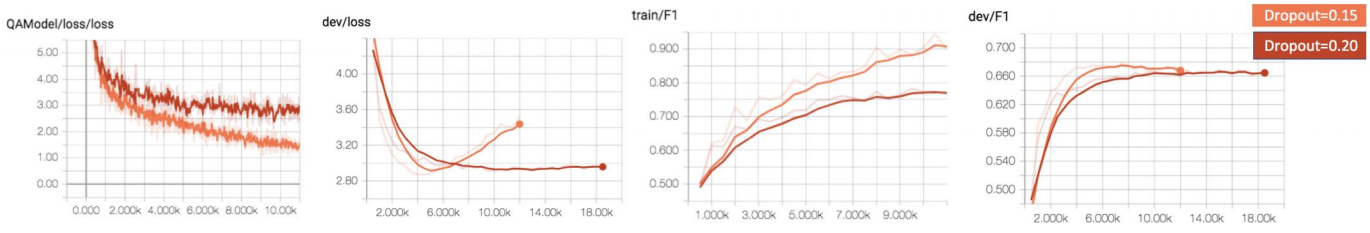


Figure 3: The loss on the training set (left) and on the dev set (middle left), and the F1 on the train set (middle right) and the dev set (right). The model being trained is BiDAF.

4.3 BiDAF with Skip Connections and Extra Modeling Layers

The BiDAF model featuring a deeper modeling layer and skip connections performed best amongst all models we tested. Skip connections improved the F1 score of the BiDAF model with 2-layer Modeling Layer by 1.0%; skip connections and an additional modeling layer increased F1 by 1.5%. We observed overfitting in all BiDAF-based models. As discussed before, dropout was not very effective at preventing overfitting; in the future, we would investigate whether L2 weight regularization can prevent this issue.

Table 2: Results obtained on the development set

Model	**F1**
Baseline	43.4
BA + 2-layer Modeling Layer (GRU, dropout=0.15)	72.5
Aligned Question Embeddings + BA + 2-layer Modeling Layer (GRU, dropout=0.15)	72.9
BA + 2-layer Modeling Layer (dropout=0.15, LSTM cells)	72.8
BA + 2-layer Modeling Layer (dropout=0.2, LSTM cells)	72.7
BA + 2-layer Modeling Layer + shortcut connections (dropout=0.15, LSTM cells)	73.7
BA + 3-layer Modeling Layer + shortcut connections (dropout=0.15, LSTM cells)	74.2
FusionNet (dropout=0.3, LSTM cells)	73.8

4.4 FusionNet

We trained the FusionNet for approx. 7 epochs (18k iterations) and obtained a score of 74.2% F1 on the development set. Our score plateaued after approx. 3.5 epochs. This score is comparable to the score of 76% F1 Huang et al. reported in the paper they achieved after 3.5 epochs, especially since our embeddings are simpler than theirs. Huang et al. used the Adamax optimizer in PyTorch; since this optimizer is not yet natively supported in TensorFlow, we used Adam. Our model did not overfit, which suggests that using a different optimizer and tuning hyperparameter can improve performance. As a reference, the Huang et al. trained FusionNet for 29 epochs.

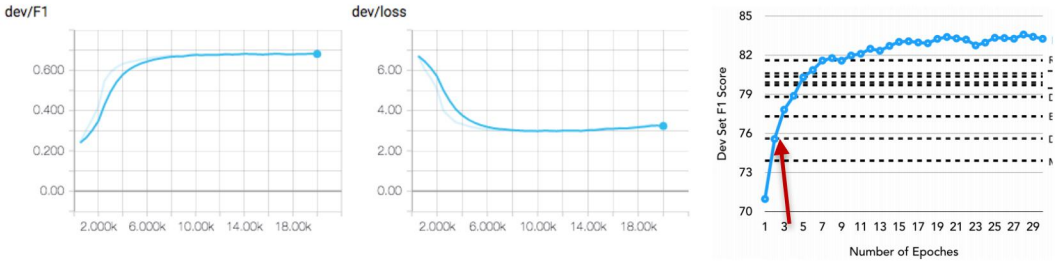


Figure 4: FusionNet learning curve for our implementation (left two panels). Learning curve from the FusionNet paper (Huang et al. 2018); arrow is pointing at the F1 for epoch 3.

5 Discussion and Error Analysis

In this section, we present quantitative and qualitative comparative analysis of the predictions made by our two best-performing models: FusionNet and BiDAF with 3 Modeling layers and skip connections (BiDAF+). On the 10570 predictions generated on dev using the official evaluation script, we obtain the following metrics:

- The number of identical answers produced by the two models: 6025 (57%)
- The number of times BiDAF+ was correct and FusionNet was incorrect: 1226 (11%)
- The number of times both models are correct: 5223 (49%)
- The number of times both models are incorrect: 3129 (29%)

5.1 Types of questions on which models are incorrect

The type of question on which both models fail the most often are ‘why’ questions. This can partly be explained by the fact that ‘why’ questions have, on average, longer answers than ‘when’, ‘who’, or ‘where’ questions (Budianto, 2017). However, ‘why’ questions are also conceptually more difficult to understand. Notably, this category is the only one in which the FusionNet model makes fewer mistakes than the BiDAF+ model; this could be because with FusionNet the model

can grasp higher-level meaning better.

Table 2: Proportion of each question type answered correctly

	What	Who	When	How	Where	Why
Average question length	11.7	10.8	10.9	11.6	10.2	10.8
Number of questions in dev	6071	1290	862	1241	505	158
BiDAF+ errors	0.42	0.33	0.24	0.36	0.45	0.65
FusionNet errors	0.44	0.35	0.26	0.40	0.46	0.63

5.2 Qualitative analysis of a case when BiDAF is correct, and FusionNet is incorrect

Question: according to tesla what had been gone over by the thieves , or spies who entered his room ?

Context: during the period in which the negotiations were being conducted , tesla said that efforts had been made to steal the invention . his room had been entered and his papers had been scrutinized , but the thieves , or spies , left empty-handed . he said that there was no danger that his invention could be stolen , for he had at no time committed any part of it to paper ; the blueprint for the _teleforce_ weapon was all in his mind."

BiDAF+ prediction: his papers *FusionNet prediction:* empty-handed

This example showcases a weakness of attending to the word embeddings at every level of the our FusionNet model. The question contains the span of words "the thieves, or spies", which also appears as-is in the context. We hypothesize that in the cases when there is an area of exact overlap of a few words between the question and the context, FusionNet's multi-level attention mechanism on the History of Word, which contains the word embeddings, will overwhelmingly place most of the attention distribution on the exact match. Moreover, we only partially trained the FusionNet; we hypothesize that at this point in the training process most high-level attention weights have not been optimized as well as the low-level weights which work with lower level representations of the words. Models such as BiDAF appear more robust to this class of errors, because their attention mechanism does not have direct access to the embeddings, being only able to attend to the encoded, hence higher-level, representation of the words.

5.3 Qualitative analysis of a case when BiDAF is incorrect, and FusionNet is correct

Question: what theorem states that the probability that a number n is prime is inversely proportional to its logarithm ?

Context: there are infinitely many primes , as demonstrated by euclid around 300 bc . there is no known simple formula that separates prime numbers from composite numbers . however , the distribution of primes , that is to say , the statistical behaviour of primes in the large , can be modelled . the first result in that direction is the prime number theorem , proven at the end of the 19th century , which says that the probability that a given , randomly chosen number n is prime is inversely proportional to its number of digits , or to the logarithm of n .

BiDAF+ prediction: direction *FusionNet prediction:* prime number theorem

This example showcases the ability of the FusionNet to learn more complex interactions of the context words than FusionNet, due to storing the History of Word. Although the theorem is phrased more lengthily in the question than in the context, the model is able to recognize it as answering the question. Having access to the exact word embeddings for computing attention is helpful here, where the article is full of partial synonyms (formula, result, theorem etc.) that could confuse a higher-level attention mechanism and dilute its signal across too many words in the context.

5.4 Qualitative analysis of a case both BiDAF and FusionNet are incorrect

Question: which architect , famous for designing london 's st. paul cathedral , is represented in the riba collection ?

Context: not only are all the major british architects of the last four hundred years represented , but many european (especially italian) and american architects ' drawings are held in the collection .

the riba 's holdings of over 330 drawings by andrea palladio are the largest in the world , other europeans well represented are jacques _gentilhatre_ and antonio visentini . british architects whose drawings , and in some cases models of their buildings , in the collection , include : inigo jones , sir christopher wren , sir john vanbrugh [etc].

BiDAF+ prediction: british architects *FusionNet prediction:* inigo jones

This example is particularly interesting to examine, since the question cannot be answered correctly using only the information in the context; one must know beforehand who designed St Paul's. In this scenario, FusionNet is able to infer that the names in the list in paragraph represent British architects, thereby denoting a higher level of language understanding that BiDAF+, since a human put in this situation would just guess one of the listed names. BiDAF+ is able to attend to the word 'british' due to 'london' appearing in the question, but does not understand that the question is asking for a name.

5.5 Quantitative analysis of the predicted answer length

The lengths of the predicted answers follow well the distribution of the ground truth answers. In order to better understand our models' mistakes, we investigated whether the length of the ground truth answers is smaller or larger than that of the incorrect predictions. We observe that both BiDAF+ and FusionNet make the majority of their incorrect predictions between 2 and 3 words shorter than the ground truth answer (which can be seen on the right of x=0); BiDAF+ appears to encounter this issue more than FusionNet. However, there are also some cases in which the incorrect predicted answer is longer than the ground truth, as can be seen on the left of x=0.

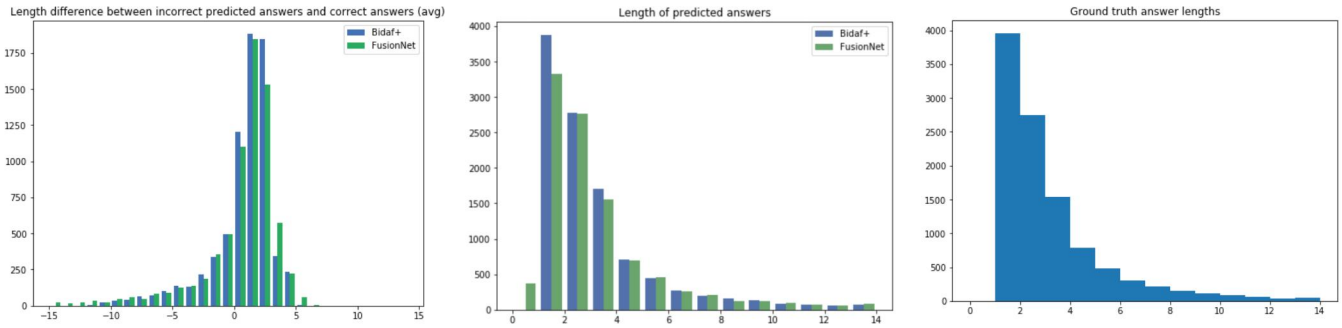


Figure 6: Answer length statistics for the BiDAF+ and FusionNet models, and ground truth

6 Conclusion and Future Work

FusionNet and the BiDAF model with 3 modeling layers and skip connections achived similar performance (74.0-74.2% F1) on SQuAD. FusionNet's "History of Word" representation maintains at each level of the network the inputs to the previous cells; hence, it leverages a very similar mechanism as skip connections. Since we only trained the FusionNet for 3 epochs before its loss flattened out, we explain the remarkably comparable performance of our two top-performing networks by the fact that at this early point in the training of the FusionNet, the skip connections' effect dominates, as the network has not yet learned all the weight matrices to allow it to fully leverage the multi-level attention. The performance of FusionNet suggests that the more sophisticated attention features require longer training to fully contribute to the performance of the algorithm. We also note that while all BiDAF-based architectures started overfitting after a few epochs of training, FusionNet's learning curve flattened out, without exhibiting any sign of overfitting. This suggests that given a more appropriate optimizer, our implementation of FusionNet could potentially achieve the state-of-the-art performance.

In the future, we would like to integrate multi-task learning into our models; one of the tasks would be on SQuAD, while the other one would be predicting whether a passage contains a good answer to a given query. Despite being highly curated, SQuAD contains some questions that are not answerable only using the information in the paragraph; we showed such an example in our Analysis section 5.4. Learning this task could improve the model's performance, as it would know to 'mask' certain training inputs, while also helping it bridge the gap to becoming a real-world

question-answering system (since questions in the real world are not attached to given contexts).

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer opendomain questions. arXiv preprint arXiv:1704.00051, 2017a.

Alex Graves and J¨urgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, 18(5):602–610

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.

Karl Moritz Hermann, Toma´s Ko ˇ cisk ˇ y, Edward ´ Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems (NIPS)

Michiel Hermans and Benjamin Schrauwen. 2013. Training and analysing deep recurrent neural networks. In Advances in Neural Information Processing Systems, pages 190–198.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2018. Fusionnet: Fusing via fullyaware attention with application to machine comprehension. ICLR

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. EMNLP, 2017.

B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in Translation: Contextualized Word Vectors. arXiv preprint arXiv:1708.00107, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In EMNLP, 2014.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. EMNLP, 2016.

Tapani Raiko, Harri Valpola, and Yann LeCun. 2012. Deep learning made easier by linear transformations in perceptrons. In AISTATS, volume 22, pages 924–932.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In ICLR, 2017.

Rupesh Kumar Srivastava, Klaus Greff, and Jurgen Schmidhuber. Highway networks. ¨ arXiv preprint arXiv:1505.00387, 2015.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426, 2017.

Huijia Wu, Jiajun Zhang, and Chengqing Zong. 2016b. An empirical exploration of skip connections for se quentialtagging. arXiv preprint arXiv:1610.03167

Huijia Wu, Jiajun Zhang, and Chengqing Zong. 2017. Shortcut Sequence Tagging, arXiv preprint arXiv:1701.00576