QANet for Question Answering on SQuAD2.0

Stanford CS224N Default Project

Rémy Zawislak
Department of Aeronautics and Astronautics
Stanford University
remzawi@stanford.edu

Abstract

Question answering is a major challenge in Natural Language Processing. Extensive research covers setups where answers always exist. However, being able to differentiate answerable from unanswerable questions is of major importance, and recent datasets such as the Stanford Question Answering Dataset 2.0 SQuAD2.0 include unanswerable questions that models need to handle correctly. Current state-of-the-art models for QA are based on Transformers. In particular, a top scoring model on the first SQuAD dataset, which did not contain unanswerable questions, was QANet, first proposed in 2018. In this paper, we study an implementation of QANet on the SQuAD2.0 dataset to see how it performs on this more challenging task. We demonstrate that QANet can be adapted to the unanswerable setup but that its performance does not compare favorably with the Bi-Directional Attention Flow (BiDAF) model on this task. We then show that simple modifications allow the model to start outperforming the BiDAF model. We also confirm the benefits of model ensembling, which noticeably improves both EM and F1 scores, the two main metrics of interest for question answering. We achieve an EM score of 63.415 and a F1 score of 66.734 on the test dataset.

1 Introduction

Natural Language Processing (NLP), which deals with the processing of language by computers, is a subject of major importance, both from theoretical perspectives, but also for a lot of applications. While improvements were sparse for a long-time pre-2010s, recent advents in machine learning, and more particularly deep learning, allowed significant improvements in NLP and reignited research in a multitude of NLP applications.

One such application is Question Answering (QA), which deals with the ability of models to both understand questions provided in natural language and provide answers to these questions. QA is in itself a large field composed of different applications. For example, in Open-Domain Question Answering, we do not provide the system with a specific context to answer the question so it needs to find the information elsewhere to generate the answer. Conversely, Closed-Domain Question Answering focuses on extracting answers from specific known context. A common approach to define this problem is to provide both a context that contains the answer and a question, and then ask a model to generate the corresponding answer if possible.

In this paper, we'll focus on this specific type of QA. This task is common in the QA field, with multiple existing datasets commonly used to test models such as the TriviaQA [1] dataset. Another such dataset is the original SQuAD [2] dataset, which is commonly used as base for any QA model due to its high-quality. In this dataset, answers are spans of the context that the model should predict. However, this dataset had only answerable question, and it is of major importance for models to be able to differentiate answerable from unanswerable questions to have true understanding of the problem. As such, an updated version that contains around 100000 answerable questions and 50000 unanswerable questions, the SQuAD2.0 [3] dataset, is now a reference for evaluating QA models. In this work, we'll study question answering models that will be evaluated on the SQuAD2.0 dataset.

In particular, we analyze the relative performance of two different models, Bi-Directional Attention Flow [4] (BiDAF) model and the QANet [5] model. We then explore the impact of some changes to the QANet architecture. Finally, we demonstrate that model ensembling, a common technique consisting in using answers from different models, improves generalization and overall performance.

All the code used in this work is available on github[1].

2 Related Work

The original SQuAD [2] paper introduced a baseline model based on a logistic model. It achieved an EM score of 40.4 and a F1 score of 51.0 that where noticeably better than both random guessing or naive approaches such as sliding window algorithms. It demonstrated the usability of the dataset for data-based techniques. However, this score was still quite low compared with human performance (EM and F1 scores of 77.0 and 86.8).

Since then, the apparition of more advanced deep-learning based techniques allowed to get closer to human performance. First, Recurrent Neural Networks and LSTMs [6] based models showed promising results for machine reading comprehension tasks [7]. When combined with attention mechanisms [8], it allowed great leaps in performance. For example, the BiDAF [4] model, based on LSTMs and attention with both self-attention and Context to Query and Query to Context cross-attention, achieved F1 and EM scores of 68 and 77.3 when used alone, or 73.3 and 81.1 with ensemble models, achieving state-of-the-art results at the time.

Later on, the success of the Transformer architecture in [9], which did not used any recurrent model, sparked off interest in recurrent free models, which would allow to parallelize models and as such greatly reduce computational complexity. One such model for question answering was QANet [5], which merged Convolutional Neural Networks with self-attention to remove all recurrent layers. When originally presented, the computational advantage of this model allowed the use of extensive data augmentation to further improve overall performance. This model demonstrated EM and F1 scores of 72.2 and 84.9 was state-of-the-art at the time of release.

However, while such models provided great results, a question that appeared was whether they truly understood the questions. To try to give an answer, the SQuAD dataset was updated with unanswerable question designed to fool models that would have a too limited understanding of the context and questions [3]. They demonstrated that high-performing models such as BiDAF or DocQA [10] on the previous dataset now showed much lower results, while human performance was not much impacted. In particular, with these models, the gap with human performance was now four times higher than on the first version of the dataset. This demonstrated the limitation of prior models and that it constitutes a new challenge for question answering. The SQuAD2.0 dataset remains today a reference for question answering.

Finally, in more recent years, the biggest leap forward was the advent of pre-training, with BERT [11] and subsequent models that built on their findings. These models are pre-trained on tasks for which we can generate a lot of data to learn general language models, and are then finetunned on specific tasks. Most if not all models currently at the top of the SQuAD leaderboard are models using pre-training, and some of them even achieve better than human performance on the dataset.Pre-trained models have significant difference with previous models both in terms of techniques used but also in final performance, and we don't discuss them further in this work.

3 Approach

First, to provide a good baseline for any other models, we complete a provided BiDAF [4] model implementation with character embeddings based on Convolutional Neural Networks (CNNs), as presented in [12]. This model is a very common question answering model, and we refer the reader to the original paper for an in-depth description of its implementation. We also give a graphical overview of the architecture in Figure 3 in the Appendix. We expect character embeddings to improve performance. This implementation is then used in our model of interest, namely QANet [5]. These models, both BiDAF with and without character embeddings, will be used as comparison baseline for the QANet model.

[1] `https://github.com/remzawi/squad`

The QANet model is implemented following the description given in the paper and the overall overview given in Figure 1 (taken directly from the QANet paper). This implementation is made trying to stay close to the original proposed implementation. We refer the reader to the corresponding paper for detailed implementation information. We only provide changes and particularities of our implementation, mostly caused by ambiguities in the original paper. First, for stochastic depth, we consider each encoder layer separately (so for the 7 stacked encoder layers, the first convolution layer has survival probability 1 and the feedforward layer 0.9). Furthermore, the integration of dropout with residual connection is not clear in the original paper. Following implementations of similar models that use LayerNorm [13] before as pre-normalization before the layer, each residual blocks performs the following operation:

$$f(x) = x + Dropout(Layer(LayerNorm(x))).$$

The original paper suggests a size of 128 in all encoders, so resizing after embeddings and after Context-Query attention is necessary but how it is done in practice is not described explicitly. For the case after embeddings, a 1D convolution is suggested so we used this for both without non-linearity or dropout afterwards. We also use a linear warm-up for 2000 steps instead of a exponential warm-up for 1000 steps. Finally, due to smaller computation capability, we use smaller batch sizes and perform gradient accumulation. As in the original paper, convolutions are depthwise separable.

We then propose changes to the original QANet problem. In particular, we experiment with using the AdamW [14] optimizer that exposed issues with traditional weight decay implementation with the Adam [15] optimizer. We also changed the momentum amounts, as it can impact training stability and final results. To further improve training stability, we perform gradient centralization [16] as it allows potential improvements at minimal cost. Additionally, we do not use stochastic depth, as it seemed in our experiment to make training less stable. Finally, in the original Transformer architecture and a lot of subsequent architectures, positional encoding is done right before self-attention or sometimes merged with self-attention. To mimic this, we add a second positional encoding in each encoder group right before the self-attention residual block.

We then experimented with other additional changes changes. In particular, we replaced the relu activation by the Gaussian Linear Unit [17] (GELU) function, which is used in multiple current state-of-the-art NLP models, and we add a one-layer highway network after co-attention before resizing.

Our code is made in a way that mimics the provided BiDAF implementation to be compatible with existing code. We use the same embedding layer as for the BiDAF model. We also reuse the provided attention layer for Context-Question attention, as it is the same in QANet. We list below code either partially or fully not written by us and obtained from open-sources. We use a memory efficient implementation of self-attention inspired from open-source code [18]. This partially alleviates the main limitation of QANet, which is its high memory cost. Using an implementation based on this code allowed noticeable improvements, especially when using multiple heads for attention. We also took the code for the AdamW optimizer from Pytorch source code as the Pytorch version used for development does not include it. The gradient centralization of AdamW is taken from the source code accompanying the original paper [16]. Finally, we use a predefined warm-up scheduler[2] openly available. Everything else is code made from the description in the paper. We also followed the Attention is all you Need [9] paper for the positional encoding implementation. Our model has the same output as the BiDAF baseline and as such no change after that is required in the provided code to generate the predictions. It also takes the same inputs, so no additional data processing is necessary.

For the final results, we know that ensembling typically improve performance, so we tried to average the results of different models. We tested with either two QANet models, a QANet and a BiDAF model or two QANets and one BiDAF model. Using a BiDAF model makes sense as while worse on its own, the great difference in architecture might produce better generalization when ensembling than only QANet models. When ensembling, we give equal weights to all models. An

[2] `https://github.com/ildoonet/pytorch-gradual-warmup-lr`

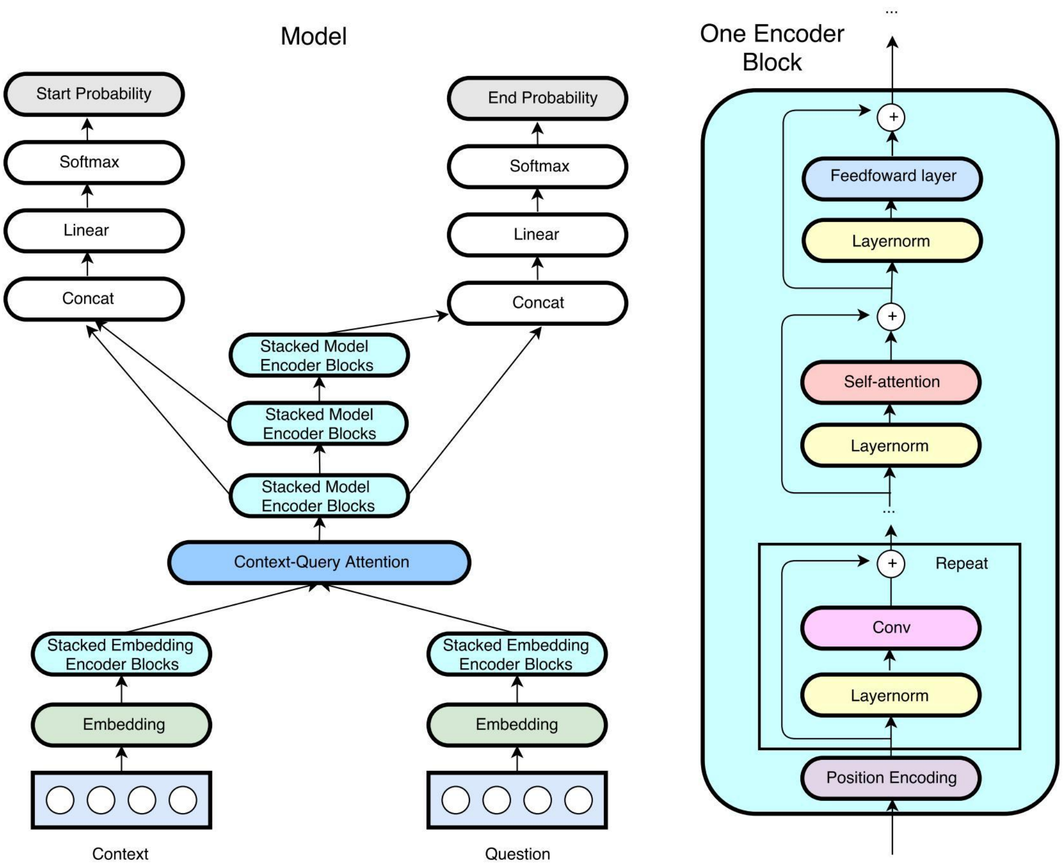

Figure 1: QANet architecture as proposed in the original paper [5].

4 Experiments

4.1 Data

In this work, we use the SQuAD2.0 [3] presented in the previous sections. More specifically, we use a slightly modified version of the dataset for the default project.

4.2 Evaluation method

We'll evaluate the models based on the dev and test sets EM and F1 metrics, two common metrics for question-answering tasks. We'll also compare the training performance, to provide a fair comparison. To do this, we use the number of iterations per second metric as it does not depend on the actual training time and is easier to compute. We also consider a metric called AvNA, for Answer vs. No Answer, which classifies whether models correctly differentiate between answerable and unanswerable questions.

4.3 Experimental details

For the baseline model and baseline with character embeddings, we used the proposed default values. The only change for the version with character embeddings was that we considered the hidden size as the combined size of the projected character and word embeddings, so as we added character embeddings word embeddings were projected to a lower dimension. In particular, we kept a hidden size of 100, used a projected size of 80 for the word embeddings and 20 for the character embeddings. Characters embedding dimension were left at the value proposed in the given code (64). However, as QANet is a bigger model, to compare more fairly, we also experimented with a bigger baseline models, with embeddings size of 300 for words and 200 for characters as in the QANet model. The hidden size is 150 instead of 100 and we use 2 layers in the encoder instead of 1. For the baselines, we use Adadelta with a learning rate of 0.5, dropout probability of 0.2 except for the character embeddings (0.05), no weight decay and EMA decay of 0.999.

For the QANet, we mostly used default values proposed in the original paper: embeddings size of 500 (300 for words and 200 for characters) and every hidden size is 128. In self-attention, we use 8 heads. The number of blocks and convolutions follows the original paper (4 and 2 convolutions, kernel size 7 and 5 and 1 and 7 blocks). We use the Adam optimizer with a learning rate of 0.001, betas of 0.8 and 0.999, weight decay of 3×10^{-7} and epsilon of $1e^{-7}$. As said in the previous section, we also use a linear warm-up for 2000 steps. Dropout probability is 0.1 except for the character embeddings (0.05) and the EMA decay is 0.9999.

For the modified QANets, we have the same sizes, but change the optimizer. We use AdamW with gradient centralization. We use the same parameters as for the original QANet, except the betas that are now 0.9 and 0.999.

For all QANets, we do not perform any data augmentation as done in the original paper and only use the SQuAD2.0 dataset.

We used different number of epochs and batch size for all models due to different models complexities. For batch size, we used 32 for baselines, and for the QANet models we used the maximum size that could fit in memory, typically between 9 and 16, and used gradient accumulation to bring the effective batch size close to 32.

4.4 Results

We give in Table 1 the AvNA, EM and F1 scores on the dev set along with the number of iterations per second for a batch size of 9 for our different models. These models are the baseline model, the baseline with character embeddings, the bigger baseline with character embeddings, the original QANet, and the two modified QANets. We also give in Figure 2 the training history of these models. In Table 2, we give the scores of ensemble models between the two modified QANets, between the second modified QANet and big baseline with character embeddings and between these three models.

Model	AvNA	EM	F1	It/s
BiDAF without char embed	67.11	56.98	60.46	211.99
BiDAF with char embed	68.74	59.03	62.35	207.47
Big BiDAF with char embed	69.47	59.84	63.09	93.41
Original QANet	62.39	56.85	58.69	103.88
First Modified QANet	70.77	60.31	64.04	103.99
Second Modified QANe	72.17	61.54	65.38	95.25

Table 1: AvNA, EM and F1 scores on the dev set for all models, along with the number of iterations per second.

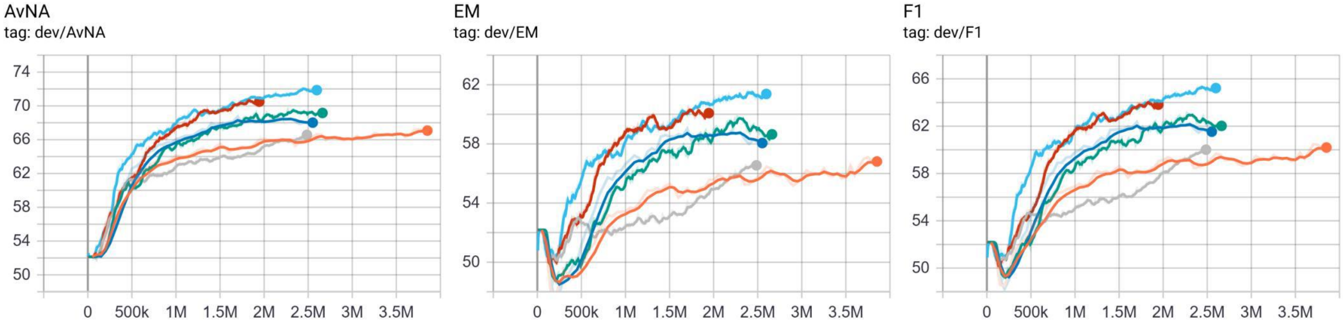

Figure 2: Training history for the different models. The colors for the models are, in the same order as in Table 1, orange, dark blue, green, grey, red and cyan.

Model	AvNA	EM	F1	It/s
First + Second Modified QANets	71.87	62.41	65.97	50.26
Second Modified QANet + big BiDAF	72.37	63.57	66.68	48.01
First + Second Modified QANets + big BiDAF	72.39	63.79	66.90	33.22

Table 2: AvNA, EM and F1 scores on the dev set for all ensemble models, along with the number of iterations per second.

Several observations can be made from the obtained results. First, character embeddings plays an important role, as it noticeably improves performance. However, for the BiDAF model, increasing the size of the model provides limited performance gains compared with the increased computational cost.

Secondly, we can see that the original QANet offers underwhelming performance compared with the BiDAF model with character embeddings. On the other hand, we can clearly observe the improvements in the two modified QANets that outperform both the original QANet and the BiDAF baselines. We believe that the improvement in performance comes from the added positional information, which is one of the main limitation of models without recurrence. We also hypothesize that the added stability from AdamW and gradient centralization might improve the ability of the model to differentiate between answerable and unanswerable question. Indeed, the AvNA metric for the original QANet model is noticeably lower and might explain the difficulties encountered when adapting the model for SQuAD2.0.

Between both modified QANets, performance are similar at the end of training, though the second model has better behavior at the beginning of training but higher cost per iteration. This might come from the added stability of the GELU activation. Furthermore, we can see that the first modified QANet performance incurs no additional performance cost when compared with the original QANet, as the added gradient centralization is quick to compute and the position encoding is a fixed deterministic operation. We experimented with trainable position embeddings, but observed no gains which did not justify the added memory cost. One can also observe that the models did not finish training, as metrics were still improving when training was stopped. We expect even better results when training for longer.

Finally, we can see that as expected, the ensemble model of the BiDAF model and the modified QANet model provides significant improvements though with an additional computational cost. Furthermore, performance is better than with the two modified QANets, despite the higher independent performance, which further strengthens our idea that the combination of different architectures is beneficial. Ensembling the three models provide additional small improvements but with a cost. As the ensemble of the three models provides the best performance on the dev set, we use this model for evaluation on the test set. **On the test set, we obtain an EM score of 63.415 and a F1 score of 66.734 for the IID SQuAD track.**

5 Analysis

We now study cases where the models work or fail and further expose the advantages of ensembling. We give in the table below four examples of context and question, along with the expected and predicted answers from the big BiDAF, second modified QANet and ensemble models.

We can make several observations from these examples. First, both models are fooled by the first example. They both understand that the question expects a person as answer so they both predict the main personage discussed in the paragraph. However, this answer is actually wrong, which shows a limitation of the models: they detect that a person would be expected, but do not evaluate correctly whether it is the correct person. As both models make the same mistake, the ensemble model logically makes the same mistake.

In the second example, the BiDAF model is fooled by the question that asks for a date, while the QANet accurately detects that the date is not the end date. The ensemble model correctly agrees with the QANet model, which can indicate that while BiDAF is wrong, it was uncertain about its prediction so when averaging the no answer prediction won.

In the third example, the BiDAF incorrectly detects the question as unanswerable, while the QANet model accurately detects that it is answerable and produces the correct answer. The ensemble model again agrees with the QANet. In general, we observe that as QANet typically produces better predictions, the ensemble model often agrees with it.

However, in the fourth example, the QANet model makes a greatly incorrect prediction, seemingly only understanding that the question is related to the parliament meeting. On the other hand, the BiDAF correctly identifies the answer and the ensemble model agrees with the BiDAF model. This demonstrates the complementarity of both models in explaining the better performance of the ensemble model.

Generally, these four examples reaffirm the previously obtained quantitative results: the QANet tends to perform better, but ensembling models allows to improve performance as for some specific questions, the BiDAF model outperforms QANet. However, both models can be fooled by some questions, and in particular can be fooled in the same way, so even ensembling has limits.

Context and Question	Answers
Context: In September 1760, and before any hostilities erupted, Governor Vaudreuil negotiated from Montreal a capitulation with General Amherst. Amherst granted Vaudreuil's request that any French residents who chose to remain in the colony would be given freedom to continue worshiping in their Roman Catholic tradition, continued ownership of their property, and the right to remain undisturbed in their homes. The British provided medical treatment for the sick and wounded French soldiers and French regular troops were returned to France aboard British ships with an agreement that they were not to serve again in the present war. Question: What British General negotiated at Toronto?	Truth: N/A BiDAF: Vaudreuil QANet: Governor Vaudreuil Ensemble: Governor Vaudreuil
Context: The war was fought primarily along the frontiers between New France and the British colonies, from Virginia in the South to Nova Scotia in the North. It began with a dispute over control of the confluence of the Allegheny and Monongahela rivers, called the Forks of the Ohio, and the site of the French Fort Duquesne and present-day Pittsburgh, Pennsylvania. The dispute erupted into violence in the Battle of Jumonville Glen in May 1754, during which Virginia militiamen under the command of 22-year-old George Washington ambushed a French patrol. Question: When did violence end in war?	Truth: N/A BiDAF: May 1754 QANet: N/A Ensemble: N/A
Context: For many years, Sudan had an Islamist regime under the leadership of Hassan al-Turabi. His National Islamic Front first gained influence when strongman General Gaafar al-Nimeiry invited members to serve in his government in 1979. Turabi built a powerful economic base with money from foreign Islamist banking systems, especially those linked with Saudi Arabia. He also recruited and built a cadre of influential loyalists by placing sympathetic students in the university and military academy while serving as minister of education. Question: Where did Turbani place students sympathetic to his views?	Truth: university and military academy BiDAF: N/A QANet: university and military academy Ensemble: university and military acedemy
Context: Parliament typically sits Tuesdays, Wednesdays and Thursdays from early January to late June and from early September to mid December, with two-week recesses in April and October. Plenary meetings in the debating chamber usually take place on Wednesday afternoons from 2 pm to 6 pm and on Thursdays from 9:15 am to 6 pm. Chamber debates and committee meetings are open to the public. Entry is free, but booking in advance is recommended due to limited space. Meetings are broadcast on the Parliament's own channel Holyrood.tv and on the BBC's parliamentary channel BBC Parliament. Proceedings are also recorded in text form, in print and online, in the Official Report, which is the substantially verbatim transcript of parliamentary debates. Question: How much does it cost to gain entry to a parliament meeting?	Truth: free BiDAF: free QANet: 2 pm to 6 pm and on Thursdays from 9:15 am to 6 pm Ensemble: free

From these few examples, we also see a tendency for shorter answers from the BiDAF models. To further analyze this phenomenon, we give in Table 3 the average number of words in the answers of the three models both including or not including the no answer predictions. We can indeed observe that the answers produced by the BiDAF model are typically shorter than that of the QANet model, perhaps because of the recurrent layers. When answering, the ensemble produces answers that sizes are between both models, which is expected. However, when including no answers, the average length is lower than both initial models. Looking at the proportion of no answers for the three models given in Table 4, this can be explained by the slightly higher proportion of no answers in the ensemble model, which can itself be explained through cases like the second example were one model might predict no answer and the other an answer for different questions.

Average answer length	BiDAF	QANet	Ensemble
Including no answers	1.55	1.62	1.53
Excluding no answers	3.03	3.12	3.06

Table 3: Average answer length on the dev set.

Model	BiDAF	QANet	Ensemble
Proportion of no answers	0.488	0.481	0.502

Table 4: Proportion of no answer response on the dev set.

6 Conclusion

In conclusion, we found in this paper that designing models for the question answering task with non-answerable questions is challenging and that specific changes in architectures might be necessary to adapt existing models. We observed that the original QANet, at least if not using data augmentation, can perform sub-optimally compared to the BiDAF model, despite being better on the original SQuAD dataset. However, some simple modifications allow substantial gains in performance which reaffirm the overall superiority of this architecture for question answering. We also further expose the advantages of ensembling techniques, especially with different architectures, bringing more than one point improvements in both EM and F1 metrics. Finally, our ensemble model achieves EM and F1 scores of 63.415 and 66.734 on the test set of the IID SQuAD track. Nonetheless, some limitations of our work can be discussed. First, in our experiments, models were not trained until convergence or overfitting, meaning that performance might be further improved. Another limitation is that while the proposed changes improved performance, performing an ablation study might have allowed to differentiate the most useful ones. Future works would include answering the two main limitations, i.e. training models until convergence or overfitting and preforming an ablation study of the proposed modifications.

References

[1] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

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

[4] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension, 2018.

[5] 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, 2018.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[10] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*, 2017.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Yoon Kim. Convolutional neural networks for sentence classification, 2014.

[13] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[16] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. In *European Conference on Computer Vision*, pages 635–652. Springer, 2020.

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[18] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017.

A Appendix

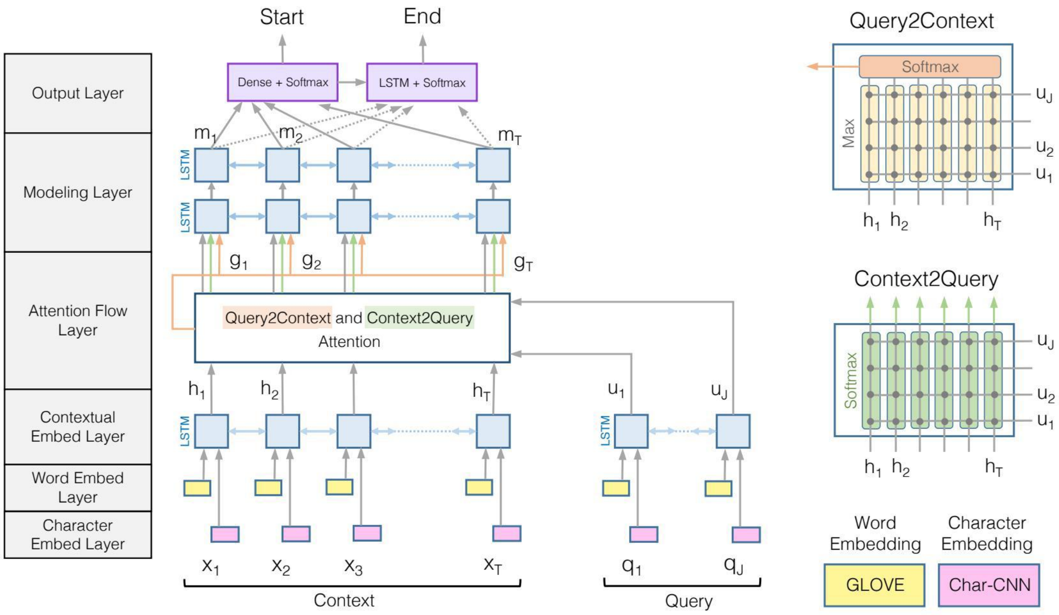

Figure 3: BiDAF architecture as presented in [4].