QaN We Pay More Attention?

Stanford CS224N Default Project

Akshita Agarwal, Qianli Song, Shafat Rahman
Department of Computer Science
Stanford University
`akshita@stanford.edu, qlsong@stanford.edu, shafatr@stanford.edu`

Abstract

Question answering (QA) is one of the hardest challenges in NLP. Unlike information retrieval tasks like named-entity recognition, QA requires a model to develop deep syntactic and semantic understanding of text as well as efficiently represent relationships between context and query. In our project, we are given the SQuAD 2.0 dataset (Stanford Question Answering Dataset) [5] and a baseline that implements BiDAF model (Bidirectional Attention Flow) with word-level embeddings. Our goal is experiment with new deep learning techniques that can improve over the baseline. For this project, we first improved the Baseline BiDAF model by adding Character-Level Embeddings. Then we implemented the QANet architecture and performed analysis on how the complexity of various modules of the system affect its performance. We also proposed a new architecture for the QANet Output Layer which helped to improve the model performance. Our best model *QANet-Ensemble* achieves **F1 score of 71.995** and **EM score of 68.678** on the Dev leaderboard and **F1 score of 69.943** and **EM score of 66.712** on the test leaderboard, placing us at the 6th position on the test leaderboard. Our submissions can be found under the name *squad_trainers*.

1 Introduction

In question answering a model is given a context paragraph and a question as input. The goal is to select a span from the context that answers the question correctly. In recent years, this has become an important benchmark for evaluating the capability of deep learning architectures and methods to understand text. Before BERT started dominating this area, BiDAF and QANet were two very important models for solving the question answering task. The BiDAF model (Bidirectional Attention Flow) is a closed-domain, extractive Q&A model that can only answer factoid questions. It has two key designs. The first one is a recurrent model to process sequential inputs. The second one is an attention component to model long-term interactions [7]. The QANet uses the same layered design as the BiDAF model but improves the performance in both accuracy and speed. The key difference between BiDAF and QANet is their encoder blocks. Rather than using a bidirectional LSTM (Long Short-term Memory) in the encoder layer to incorporate temporal dependencies between timesteps of the embedding layer's output like in BiDAF, the QANet replaces them entirely with self-attention and convolution. The convolutions model local interactions and the self-attention models global interaction[10].

2 Related Work

BiDAF [7] introduces the idea of bi-directional attention that helps to model relationships between context and query. BiDAF is a multi-stage hierarchical process that represents the context at different levels of granularity and uses bi-directional attention flow to obtain a query-aware context representation. It has two key components: a recurrent model to process sequential inputs, and an attention component to understand long-term interactions. The core idea of bidirectional flow is that attention should flow in both directions between the query and context.

The Transformer model [9] introduced multi-headed self-attention as an alternative to the recurrence commonly used in encoder-decoder architectures. QANet [10] adapts ideas from the Transformer and tries to tackle the question answering by applying self-attention and convolutions only. The motivation of this work is that Convolutions perform well at capturing local context while attention captures global relationships in text. By combining both convolutions and attention, QANet achieves 3 to 13 times faster training time and 4 to 9 times faster inference time, while achieving similar accuracy to previous recurrent models on the SQuAD1.0 dataset [6]. Since QANet has not been applied to SQuAD2.0 and SQuAD2.0 has unanswerable questions unlike SQuAD1.0 we believe that our implemented models will provide a good baseline for QANet's performance on unanswerable questions.

One drawback of QANet is that it does not predict the end token based on the start token, which we think could improve the accuracy of the predictions. We propose a conditional output layer for QANet that could yield better results by taking the predicted start of the answer span into account while predicting the end of the span.

3 Approach

3.1 BiDAF

BiDAF Baseline: We use the BiDAF model implementation provided by CS224n staff as our baseline. This implementation differs from the original BiDAF network suggested by [7] in that it does not include the character embedding layer. In the embedding layer the model looks up pre-trained word embeddings for both the context and question based on the word or character index, and then applies a two-layer highway network to improve model robustness. In the encoder layer the model applies a bidirectional LSTM to incorporate temporal dependencies between time-steps of the embedding layer's output. The model uses the attention layer to learn context-to-query and query-to-context attention. In the modeling layer the model refines the sequence of vectors after the attention layer and integrates temporal information between context representations conditioned on the question. Lastly, in the output layer produces a vector of probabilities corresponding to each position in the context.

BiDAF + Character Level Embedding: We add character-level embeddings proposed in the original BiDAF model [7], but was left out of the baseline model. Character-level embeddings allow the model to condition on morphology and handle out-of-vocabulary words better. The character level embedding of each word is obtained using a Convolutional Neural Network (CNN)[4]. The characters are first embedded into vectors, which are 1D inputs to a CNN. The outputs of the CNN are max-pooled over the entire width to obtain a fixed-size vector for each word . We experiment with two primary variations of character embedding:

- **1 Layer Char-CNN** - We apply one convolutional layer with 50 (*CharEmbed-50D*) or 100 filters (*CharEmbed-100D*) and kernel size of 3. The output of this layer is the final character embedding.

- **2 Layers Char-CNN** - We apply two convolutional layers each with 100 filters and kernel size of 3 and 5 to get *CharEmbed-200D*. The output of these two layers are concatenated to obtain the final character embedding.

We modify the input embedding layer, which takes as input word and character indexes of the question and context words and transforms them into distributed embedding vectors. We use Glove-300 Dimensional word embeddings. For character embeddings, we use the same methodology as described in the BiDAF with CharEmbedding model. The dimensions of the word embeddings and character embeddings are $d_{word} = 300$ and $d_{char} = 200$ respectively. The word and character embeddings are then concatenated to form a vector of dimension $d_{embedtotal} = 500$. The embeddings are then reduced using a 1-dimensional convolution to the input size of the model $d_{model} = 128$ and then fed into a Highway Network [8].

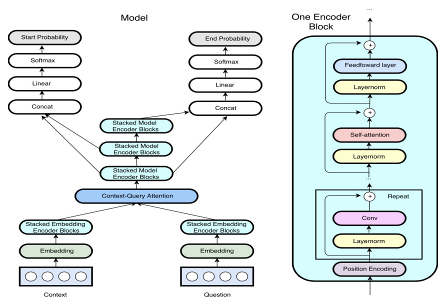


Figure 1: Original QANet Architecture

3.2 QANet

The QANet model has five layers: input embedding, embedding encoder, context-query attention, model encoder, and output [10]. Since some of these layers are quite similar to the original BiDAF model, we reuse implementations of those layers with slight modifications. QANet's encoder block, however, is quite different from the BiDAF encoder layer and is more similar to the Transformer ([9]) in terms of applying positional encodings, residual connections, layer normalization, self-attention and feed-foward sub-layers. We implement the encoder block as described in the original QANet architecture from scratch.

Input Embedding Layer: This layer is similar to the input embedding layer of section 3.1. We adopt the standard techniques to obtain the embedding by concatenating the word-level embedding and the character-level embedding. The exact details of the word-level and character-level embeddings can found in section 3.1. We also adopt a two-layer highway network on top of this representation.

Embedding Encoder Block The core component of QANet is the encoder block. We implement the encoder block proposed by [10] and shown in Figure 1. It consists of convolution-layers, followed by a self-attention-layer and a feed-forward-layer. We use 2 convolutional layers with kernel size 7 in our implementation. For each of these layers we also implement layer normalization to stabilize the hidden state dynamics and residual connections for improved gradient flow through the network. We use Depthwise separable convolutions in the convolutional layers because it is memory efficient and can generalization better [10]. We adopt the multi-head attention mechanism for the self-attention layer. For each query, the multi-head attention computes a weighted sum of all the positions, or keys, in the input based on the similarity between the query and key, which is measured using a dot product [10]. The input of the encoder layer is first mapped to a dimension of 128 by a one-dimensional convolution. The output of this layer also has dimension 128.

Context-Query Attention This layer is similar to the context-query attention layer in BiDAF. Its responsibility is in linking and fusing information from the context and the query words. Here, we need to compute the similarity matrix, which contains a score for each pair of context and question hidden states. We compute attentions in two directions: from context to query as well as from query to context. The context-to-query attention signifies which query words are most relevant to each context word. The query-to-context attention signifies which context words have the closest similarity to one of the query words and are hence critical for answering the query. [10][7]

Model Encoder Layer The model encoder layer uses the same encoder block as the embedding encoder layer. However, instead of one encoder block, this layer has 3 stacks of 7 encoder blocks. The encoder blocks share weights. [10].

Output layer The output layer is task-specific. Each example in SQuAD is labeled with a span in the context containing the answer. Let M_0, M_1, and M_2 denote the outputs of the three model encoder stacks. QANet computes the probability of each position in the context being the start and end position of the answer span as: $p_{\text{start}} = \text{softmax}(W_1[M_0; M_1])$ and $p_{end} = \text{softmax}(W_2[M_0; M_2])$, where W_1 and W_2 are two trainable variables [10].

3.3 QANet Extensions

Conditional Output Inspired by work on conditional output in [2] and BIDAF output, we experimented with the output layer of QANet. We implement an output layer in which the probability of prediction of end token is influenced by the start token prediction. The results can be found under the name *QANet-Conditional-Output* in the Experiment Results section. The equations defining the architecture can be seen below 3.3.

$$\begin{aligned} A &= W_0[M_0; M_1] \\ B &= W_1[M_0; M2] \\ p_{start} &= \text{softmax}(W_2(A)) \\ p_{end} &= \text{softmax}(W_3[A; B]) \end{aligned}$$

Conditional Output with Extra Attention In addition to the conditional output layer from the previous section, we also observed that the BiDAF output not only took the input of the modelling layers but also took the input from the attention layer directly for the output prediction. This is our **original** contribution on experimenting with adding direct attention connections along with conditional dependence in the output layer. The architecture can be seen in 2 and equations can be found below 3.3. p_{start} and p_{end} remain the same as above.

$$\begin{aligned} A &= W_0[M_0; M_1; CQAtt] \\ B &= W_1[M_0; M2; CQAtt] \end{aligned}$$

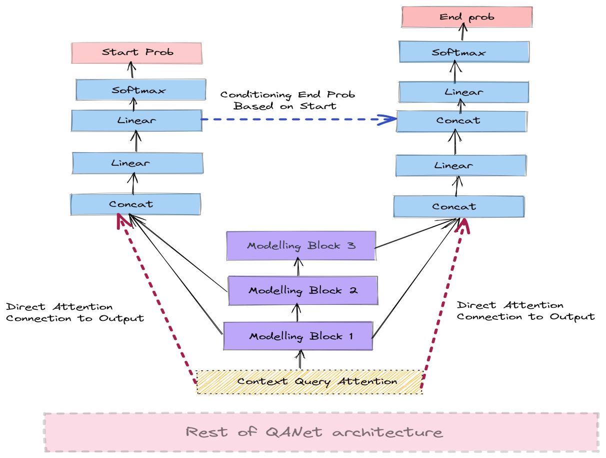


Figure 2: QANet-ExtraAttentionInOutput Architecture

4 Experiments

4.1 Data

We train and test our models on a slightly modified Stanford Question Answering Dataset (SQuAD2.0), which has around $150,000$ questions with about half of the questions unanswerable given the context. A question answering model predicts a span of the context that answers the question, or predicts "no answer". Every answerable SQuAD question has three human-generated answers provided. The training split has $129,941$ examples. The dev set we use is created by selecting roughly half of the official SQuAD2.0 dev set randomly and has 6078 examples. The test set has 5915 examples, which includes the official SQuAD test set and the remaining examples from the official dev set.[1].

4.2 Evaluation method

We evaluate our model based on the EM and F1 scores, which are the official SQuAD evaluation metrics. The EM score stands for Exact Match, which is a binary measure of whether the system output matches the ground truth answer exactly. The F1 score is the harmonic mean of precision and recall.

4.3 Experimental details

In the training phase, all experiments were run for 3 million iterations or 30 epochs. A full training run takes around 15 hours. In the evaluation phase, the number of steps between successive evaluations is 50000. The dropout rate for the CharEmbed layer is 0.05. For all the other layers, the dropout rate is 0.1. We use a batch size of 24 because the VM we use from Microsoft Azure cannot handle batch size larger than this. We use a learning rate of 0.001 with an exponential warm up of 1000 steps. We have an ADAM optimizer with $\beta_1 = 0.8$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$ and weight decay rate of $5 * 10^{-8}$. We apply an exponential moving average of parameters with a decay rate of 0.9999. The number of attention heads is either 4 or 8 depedning on our experiment. The dimension of the Convolution layers in the Encoder Blocks is 128. The size of char vectors is 64. The size of GloVe word vectors to use is 300. The number of sub-processes to use per data loader is 4. We implement our model in Python. For training, we have the Standard NC6s v3 with 6 vcpus and 112 GiB memory provided by Microsoft Azure. These VMs are powered by NVIDIA Tesla V100 GPUs. These GPUs can provide 1.5x the computational performance of the NCv2-series.

Model Complexity and Layers: We experimented with the following configurations of QANet:

- **No. of Convolutional Layers**: In *QANet-Lite* we used 2 Conv layers in the Embedding Encoder block and 1 Conv layer in the Modelling Encoder blocks. For all other models, we used 4 layers in Embedding Encoders and 2 layers in Modeling encoders.
- **No. of Encoder Blocks**: We experimented with using 5 and 7 encoder blocks in the Modeling encoder layers. The results can be seen in the Results table. 1
- **No. of Heads in Self Attention**: We experimented with using 1,4,8 Self Attention heads. Further analysis on the effects of these is present in the Analysis section.

Architecture Extensions: We experimented with two new architectures for the Output layer of QANet (as mentioned in Section 3.3). The resultant models are *QANet-Conditional-Output*, *QANet-XtraAtt* and *QANet-CharEmbed200D-Xtra-Att*. The last model *QANet-CharEmbed200D-Xtra-Att* has 200D CharEmbeddings as well as opposed to all other QANet models we trained which has 128 dimensional Char Embeddings.

Ensemble: We performed an ensemble of all 9 QANet variants in Table 1 using the majority-voting and weighted-average (based on F1 score) technique. The results produced through the weighted-average technique was our best performing model on the dev set.

4.4 Regularization and HyperParameter Tuning

Our initial iterations of the QANet model have seen the training line flat-lined and even underperform the baseline model. This is most likely because, in our initial implementation of the Encoder Block, we were not using dropout as aggressively as recommended in the training details section of the QANet paper: "We additionally use dropout on word, character embeddings and between layers, where the word and character dropout rates are 0.1 and 0.05 respectively, and the dropout rate between every two layers is 0.1" [10]. After adding aggressive dropout, the performance improved.

4.5 Results

Our basic QANet architecture achieves 7% improvement over the baseline. Through further experimentation we arrived at **QANet-Ensemble**, which is our best model and obtains 10.66% improvement over the baseline. The results can be seen in Table 1. We have the following observations from our experimentation:

Model	Dev Loss	F1	EM	AvNA
Baseline	02.96	61.33	58.24	67.87
BiDAF-200D-CharEmbed	02.82	66.68	63.28	72.69
QANet-Lite	02.96	68.25	64.75	74.91
QANet-1-AttHeads-7-Encoders	02.64	67.95	64.78	74.14
QANet-4-AttHeads-5-Encoders	02.69	68.22	64.48	74.54
QANet-4-AttHeads-7-Encoders	02.70	68.52	65.05	74.79
QANet-8-AttHeads-5-Encoders	02.83	68.38	64.49	74.81
QANet-8-AttHeads-7-Encoders	02.75	68.39	64.46	74.83
QANet-Conditional-Output	02.82	67.09	63.75	73.65
QANet-Xtra-Att	02.62	68.62	65.32	74.59
QANet-CharEmbed200D-Xtra-Att	**02.79**	**69.76**	**66.54**	**75.82**
QANet-Ensemble	**02.82**	**71.99**	**68.68**	**77.21**

Table 1: Experiment Results

Adding character embeddings to baseline model improves the F1 score by 5.35%. Character embeddings increases the model's effectiveness at handling out-of-vocabulary words by capturing the morphological structure. QANet-Lite, our simplest version of QANet with 4 attention heads, 5 encoder blocks in modeling layer, 2 convolution layers in embedding encoders and 1 convolution layer in modeling encoder outperforms BIDAF with included character embeddings. We attribute this improvement to the Transformer-like architecture of QANet which also includes the multihead self-attention.

Increasing the number of attention heads or number of blocks in encoder layer improves performance sightly, but not yield significant gains (about 0.4% only). This has also been observed in some previous works [2].

Adding the extra attention input helped to achieve a slight improvement in performance(0.1 % gain in F1 score). Again we had expected a higher gains, as we believed that the context-query attention layer has the most information about the correlation between the context and the query words and feeding this directly to the output layers, should help the model make better predictions. Further analysis on this can be found in section 5.

Increasing the CharEmbedding dimension from 128 to 200 for our final model *QANet-CharEmbed200D-Xtra-Att* obtains highest score improvements among all our experiments on QANet. This is expected since character embeddings capture morphological structure of words and increasing their dimensionality allows to capture more information at the embedding layer. This also improves the modelling capacity of the layers higher up in the network.

Overall, our QANet Ensemble model outperforms all other models that we experiment with. The QANet ensemble uses information from several QANet predictions and can take advantage of the strengths of each variant while dampening their drawbacks.

4.5.1 Leaderboard Results

Our best model ***QANet-Ensemble*** achieves **F1 score of 71.995** and **EM score of 68.678** on the Dev leaderboard and **F1 score of 69.943** and **EM score of 66.712** on the test leaderboard.The difference between Dev and Test accuracy can be attributed to a slight difference in the data distribution. Also it might be due to a slight overfitting on the Dev dataset due to the heavy hyperparameter tuning based on the Dev dataset performance.

5 Analysis

5.1 Performance by Question type

Figure 3 demonstrates that among the different question types, our models performs the worst on "why" questions. "Why" questions require logical reasoning and deeper semantic understanding of the context. This is unlike "who" questions, which require the model to simply identify entities

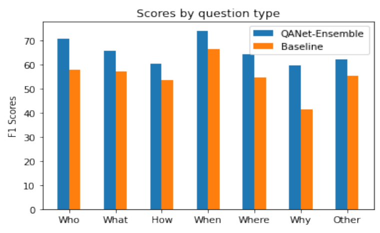


Figure 3: F1 score comparison with Baseline across various question types

in the context. "Reasoning-Driven" QA techniques [3] along with QANet might help to improve performance here.

5.2 Effect of Conditional Dependence in Output

Adding Conditional Dependence from Start to End predictions impacted the model performance negatively. We attribute this to the fact that we added extra input to the End Prediction output layer without adding any regularization such as Dropout. This lead to overfitting on the Training set and impacted the Dev performance negatively. Due to time constraint, we couldnt experiment with adding regularization to this layer.

5.3 Effect of XtraAttention

Extra attention improved the performance of our in the QANet-XtraAtt model, but not as significantly as we expected. The lower gain in performance can be attributed to the architecture of the encoder blocks. The encoder blocks have residual connections that help to stabilize the training. The residual connections present in the modelling encoder block, helped the model to "remember" the context-query attention all through to the output layer. Hence adding more attention, though helped to add some more information to the output layer, but most of the information it added, was already being fed to the output layer through the residual connections, hence adding more of the same information, did not result in a significant change.

5.4 Effect of Number of Heads in Self Attention

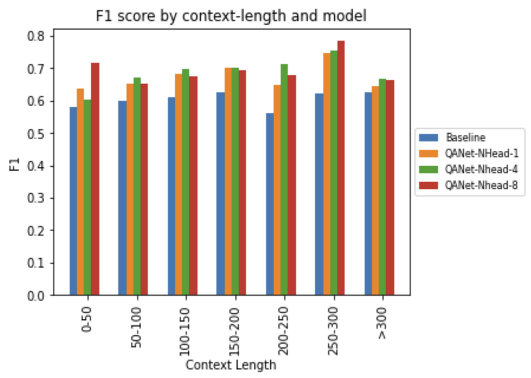
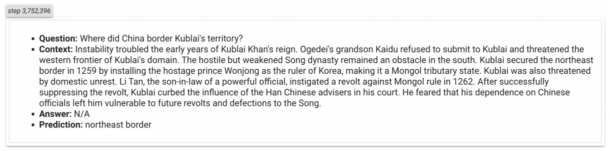


Figure 4: F1 score based on varying number of heads in Self Attention

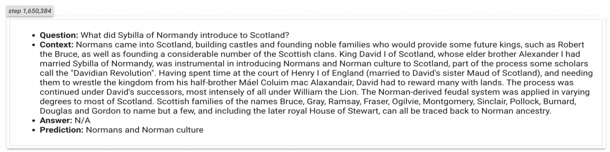
Figure 4 shows how the F1 score changes based on the number of heads in the Self Attention layer against the Context Length. We can see that all the QANet models significantly outperform the baseline. In addition, the model with the highest number of attention heads (8), performs the best.There is a siginificant difference between 1 and 4 attention heads, but the same is not observed for 4 vs 8 heads. This makes us feel that given our dataset, 4 attention heads does a good job of capturing the context dependencies.

5.5 Exploration of individual questions


step 3,752,396

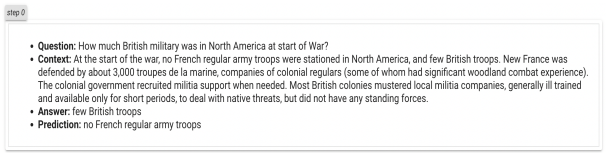
- **Question:** Where did China border Kublai's territory?
- **Context:** Instability troubled the early years of Kublai Khan's reign. Ogedei's grandson Kaidu refused to submit to Kublai and threatened the western frontier of Kublai's domain. The hostile but weakened Song dynasty remained an obstacle in the south. Kublai secured the northeast border in 1259 by installing the hostage prince Wonjong as the ruler of Korea, making it a Mongol tributary state. Kublai was also threatened by domestic unrest. Li Tan, the son-in-law of a powerful official, instigated a revolt against Mongol rule in 1262. After successfully suppressing the revolt, Kublai curbed the influence of the Han Chinese advisers in his court. He feared that his dependence on Chinese officials left him vulnerable to future revolts and defections to the Song.
- **Answer:** N/A
- **Prediction:** northeast border

Figure 5: Baseline model predicts an answer when the ground-truth is "no answer": The model understands the question and knows what to look for probably with the help of context-to-query attention. But is fails to make a well-informed prediction based on deeper semantic understanding of the context.


step 1,650,384

- **Question:** What did Sybilla of Normandy introduce to Scotland?
- **Context:** Normans came into Scotland, building castles and founding noble families who would provide some future kings, such as Robert the Bruce, as well as founding a considerable number of the Scottish clans. King David I of Scotland, whose elder brother Alexander I had married Sybilla of Normandy, was instrumental in introducing Normans and Norman culture to Scotland, part of the process some scholars call the "Davidian Revolution". Having spent time at the court of Henry I of England (married to David's sister Maud of Scotland), and needing them to wrestle the kingdom from his half-brother Máel Coluim mac Alaxandair, David had to reward many with lands. The process was continued under David's successors, most intensely of all under William the Lion. The Norman-derived feudal system was applied in varying degrees to most of Scotland. Scottish families of the names Bruce, Gray, Ramsay, Fraser, Ogilvie, Montgomery, Sinclair, Pollock, Burnard, Douglas and Gordon to name but a few, and including the later royal House of Stewart, can all be traced back to Norman ancestry.
- **Answer:** N/A
- **Prediction:** Normans and Norman culture

Figure 6: QANet predicts an answer when there is no answer: The QANet model fails to gain an understanding of the use of commas to separate phrases. It assumes that Sybilla of Normandy introduces Normans and Norman culture to Scotland just because the two phrases appear together, but fails to understand how the use of the comma has effects on the semantics of the sentence. This might also show that the self-attention span of the model may not be large enough.


step 0

- **Question:** How much British military was in North America at start of War?
- **Context:** At the start of the war, no French regular army troops were stationed in North America, and few British troops. New France was defended by about 3,000 troupes de la marine, companies of colonial regulars (some of whom had significant woodland combat experience). The colonial government recruited militia support when needed. Most British colonies mustered local militia companies, generally ill trained and available only for short periods, to deal with native threats, but did not have any standing forces.
- **Answer:** few British troops
- **Prediction:** no French regular army troops

Figure 7: QANet ensemble predicts wrong answer span: The model fails to recognize the difficult syntax used in this sentence. The sentence containing the answer span is "... no French regular army troops were stationed in North America, and few British troops." The model correctly identifies the the sentence containing the answer span, but since the answer is placed in a phrase after a phrase that could have a standalone meaning, the model finds it difficult to identify the correct answer. This might also demnstrate that for "how" questions, the model is not paying much attention to named entities such as "British" or "French."

We look through some of our model's predictions and try to identify common sources of error to get a better sense of where our model is making mistakes.

In Figure 5, 8 and 9 we notice that the context-to-query and query-to-context attention of the BiDAF baseline performs well. The model is able to identify relevant spans from the context, but lacks semantic understanding of the context and sometimes to question to make the correct prediction.

In Figure 6, 10 and 11 we notice that the QANet model does a better job at developing a semantic understanding of both the context and query, but it still fails to understand difficult syntax and sentence structures.

In Figure 7, 12 and 13 show mistakes made by the ensemble model. Surprisingly, even though the ensemble model performs much better than the standalone QANet models, it tends to make similar mistakes. It is apparent that some of the syntactical shortcomings of standalone QANets are propagated on to the ensembles.

6 Conclusion

From the analysis and experiments above, we can draw the following conclusions. First, increasing complexity of QANet encoder blocks by increasing the number of attention heads or number of encoder blocks does not improve performance significantly on the SQuAD2.0 dataset. Second, increasing complexity of other parts of the model, including features like character-embedding dimension and changing the output layer architecture are key factors in the performance improvements achieved by our models. Third, ensembling helps to produce effective models that attain 2.23 % F1 score gain over our single best model as it reduces the variance and feature noise from the base predictors. Future work could involve investigating whether our models achieves similar performance on other the datasets such as TriviaQA as well as investigating effects the back-translation data augmentation technique utilised by the original QANet [10] on our models.

References

[1] Stanford CS224N Class. "CS 224N Default Final Project: Building a QA system". **in**(): URL: `https://web.stanford.edu/class/cs224n/project/default-final-project-handout-squad-track.pdf`.

[2] Stanford CS224N Class. "QANet+ :Improving QANéet for Question Answering". **in**(): URL: `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report269.pdf`.

[3] Daniel Khashabi. "Reasoning-Driven Question-Answering for Natural Language Understanding". **in***CoRR*: abs/1908.04926 (2019). arXiv: `1908.04926`. URL: `http://arxiv.org/abs/1908.04926`.

[4] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". **in***CoRR*: abs/1408.5882 (2014). arXiv: `1408.5882`. URL: `http://arxiv.org/abs/1408.5882`.

[5] Pranav Rajpurkar **andothers**. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". **in***CoRR*: abs/1606.05250 (2016). arXiv: `1606.05250`. URL: `http://arxiv.org/abs/1606.05250`.

[6] Pranav Rajpurkar **andothers**. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". **in***CoRR*: abs/1606.05250 (2016). arXiv: `1606.05250`. URL: `http://arxiv.org/abs/1606.05250`.

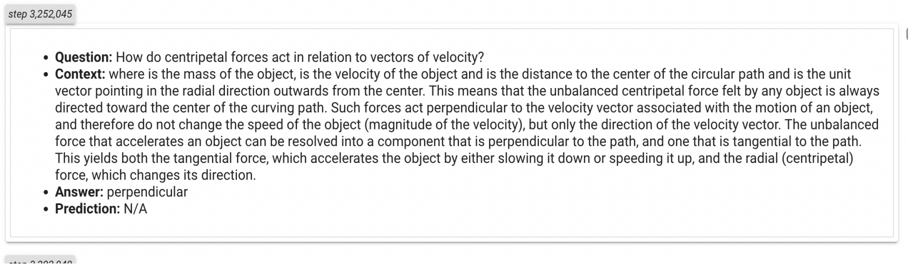
[7] Min Joon Seo **andothers**. "Bidirectional Attention Flow for Machine Comprehension". **in***CoRR*: abs/1611.01603 (2016). arXiv: `1611.01603`. URL: `http://arxiv.org/abs/1611.01603`.

[8] Rupesh Kumar Srivastava, Klaus Greff **and** Jürgen Schmidhuber. "Highway Networks". **in***CoRR*: abs/1505.00387 (2015). arXiv: `1505.00387`. URL: `http://arxiv.org/abs/1505.00387`.

[9] Ashish Vaswani **andothers**. "Attention Is All You Need". **in***CoRR*: abs/1706.03762 (2017). arXiv: `1706.03762`. URL: `http://arxiv.org/abs/1706.03762`.

[10] Adams Wei Yu **andothers**. "QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension". **in***CoRR*: abs/1804.09541 (2018). arXiv: `1804.09541`. URL: `http://arxiv.org/abs/1804.09541`.

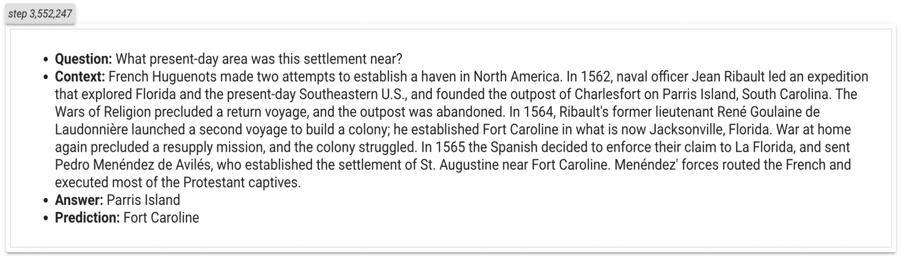
A Appendix



step 3,252,045

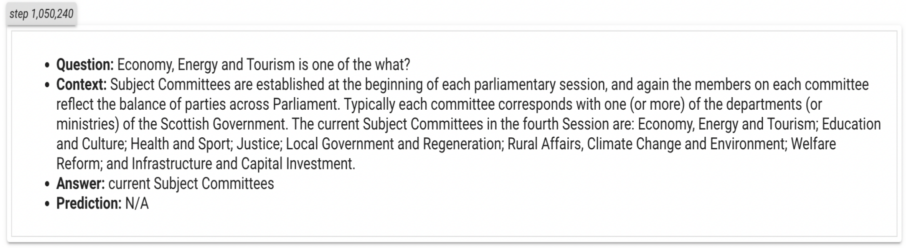
- **Question:** How do centripetal forces act in relation to vectors of velocity?
- **Context:** where is the mass of the object, is the velocity of the object and is the distance to the center of the circular path and is the unit vector pointing in the radial direction outwards from the center. This means that the unbalanced centripetal force felt by any object is always directed toward the center of the curving path. Such forces act perpendicular to the velocity vector associated with the motion of an object, and therefore do not change the speed of the object (magnitude of the velocity), but only the direction of the velocity vector. The unbalanced force that accelerates an object can be resolved into a component that is perpendicular to the path, and one that is tangential to the path. This yields both the tangential force, which accelerates the object by either slowing it down or speeding it up, and the radial (centripetal) force, which changes its direction.
- **Answer:** perpendicular
- **Prediction:** N/A

Figure 8: Baseline model predicts "no answer" when an answer exists: The model may not have learnt to infer that phrases such as "vectors of velocity" and "velocity vector" have the same meaning.


step 3,552,247

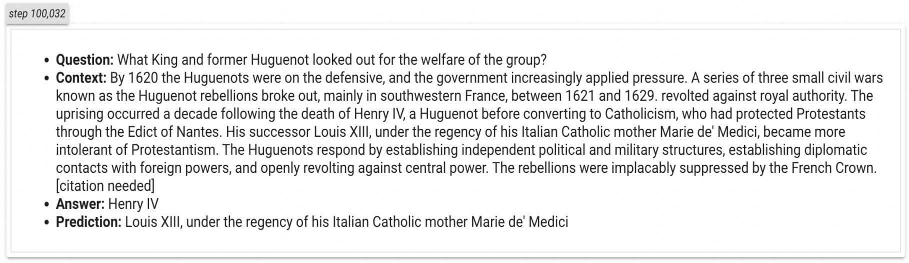
- **Question:** What present-day area was this settlement near?
- **Context:** French Huguenots made two attempts to establish a haven in North America. In 1562, naval officer Jean Ribault led an expedition that explored Florida and the present-day Southeastern U.S., and founded the outpost of Charlesfort on Parris Island, South Carolina. The Wars of Religion precluded a return voyage, and the outpost was abandoned. In 1564, Ribault's former lieutenant René Goulaine de Laudonnière launched a second voyage to build a colony; he established Fort Caroline in what is now Jacksonville, Florida. War at home again precluded a resupply mission, and the colony struggled. In 1565 the Spanish decided to enforce their claim to La Florida, and sent Pedro Menéndez de Avilés, who established the settlement of St. Augustine near Fort Caroline. Menéndez' forces routed the French and executed most of the Protestant captives.
- **Answer:** Parris Island
- **Prediction:** Fort Caroline

Figure 9: Baseline model predicts answer different from the ground-truth: Granted that the question is ambiguous, this example shows that the model has developed some semantic understanding of phrases such as "in what is now." This question is quite difficult compared to some of the other questions we analyse.


step 1,050,240

- **Question:** Economy, Energy and Tourism is one of the what?
- **Context:** Subject Committees are established at the beginning of each parliamentary session, and again the members on each committee reflect the balance of parties across Parliament. Typically each committee corresponds with one (or more) of the departments (or ministries) of the Scottish Government. The current Subject Committees in the fourth Session are: Economy, Energy and Tourism; Education and Culture; Health and Sport; Justice; Local Government and Regeneration; Rural Affairs, Climate Change and Environment; Welfare Reform; and Infrastructure and Capital Investment.
- **Answer:** current Subject Committees
- **Prediction:** N/A

Figure 10: QANet predicts "no answer’ when an answer exists: Once again QANet fails to understand the meaning of a sentence because of the use of complicated syntax, commas and semi-colons.


step 100,032

- **Question:** What King and former Huguenot looked out for the welfare of the group?
- **Context:** By 1620 the Huguenots were on the defensive, and the government increasingly applied pressure. A series of three small civil wars known as the Huguenot rebellions broke out, mainly in southwestern France, between 1621 and 1629. revolted against royal authority. The uprising occurred a decade following the death of Henry IV, a Huguenot before converting to Catholicism, who had protected Protestants through the Edict of Nantes. His successor Louis XIII, under the regency of his Italian Catholic mother Marie de' Medici, became more intolerant of Protestantism. The Huguenots respond by establishing independent political and military structures, establishing diplomatic contacts with foreign powers, and openly revolting against central power. The rebellions were implacably suppressed by the French Crown. [citation needed]
- **Answer:** Henry IV
- **Prediction:** Louis XIII, under the regency of his Italian Catholic mother Marie de' Medici

Figure 11: QANet predicts wrong answer: The model makes a very severe mistake. It does not seem to have a good understanding of what phrases in the question such as "the group" is referring to or the meaning of the phrase "former Huguenot". Instead it focuses on finding the name of a name, probably based on the "what" phrase in the question.

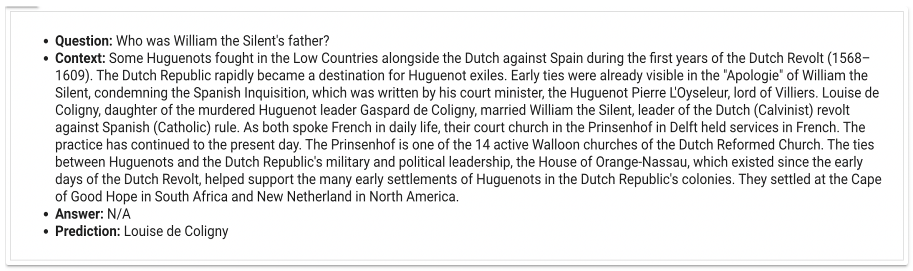
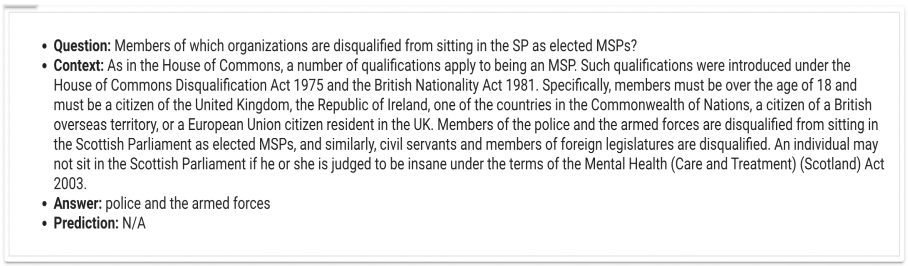

- **Question:** Who was William the Silent's father?
- **Context:** Some Huguenots fought in the Low Countries alongside the Dutch against Spain during the first years of the Dutch Revolt (1568–1609). The Dutch Republic rapidly became a destination for Huguenot exiles. Early ties were already visible in the "Apologie" of William the Silent, condemning the Spanish Inquisition, which was written by his court minister, the Huguenot Pierre L'Oyseleur, lord of Villiers. Louise de Coligny, daughter of the murdered Huguenot leader Gaspard de Coligny, married William the Silent, leader of the Dutch (Calvinist) revolt against Spanish (Catholic) rule. As both spoke French in daily life, their court church in the Prinsenhof in Delft held services in French. The practice has continued to the present day. The Prinsenhof is one of the 14 active Walloon churches of the Dutch Reformed Church. The ties between Huguenots and the Dutch Republic's military and political leadership, the House of Orange-Nassau, which existed since the early days of the Dutch Revolt, helped support the many early settlements of Huguenots in the Dutch Republic's colonies. They settled at the Cape of Good Hope in South Africa and New Netherland in North America.
- **Answer:** N/A
- **Prediction:** Louise de Coligny

Figure 12: QANet ensemble predicts and answer when no answer exists: The model is able to identify that there William the Silent is related to Louise de Colgny, but is unable to decipher that Louise is not William’s father. Here we see the model fail to analyse the model semantically under the stress of complicated syntax and information.



- **Question:** Members of which organizations are disqualified from sitting in the SP as elected MSPs?
- **Context:** As in the House of Commons, a number of qualifications apply to being an MSP. Such qualifications were introduced under the House of Commons Disqualification Act 1975 and the British Nationality Act 1981. Specifically, members must be over the age of 18 and must be a citizen of the United Kingdom, the Republic of Ireland, one of the countries in the Commonwealth of Nations, a citizen of a British overseas territory, or a European Union citizen resident in the UK. Members of the police and the armed forces are disqualified from sitting in the Scottish Parliament as elected MSPs, and similarly, civil servants and members of foreign legislatures are disqualified. An individual may not sit in the Scottish Parliament if he or she is judged to be insane under the terms of the Mental Health (Care and Treatment) (Scotland) Act 2003.
- **Answer:** police and the armed forces
- **Prediction:** N/A

Figure 13: QANet ensemble predict "no answer" when an answer exists: The model is confused because "Scottish Parliament" is abbreviated to "SP" in the question.

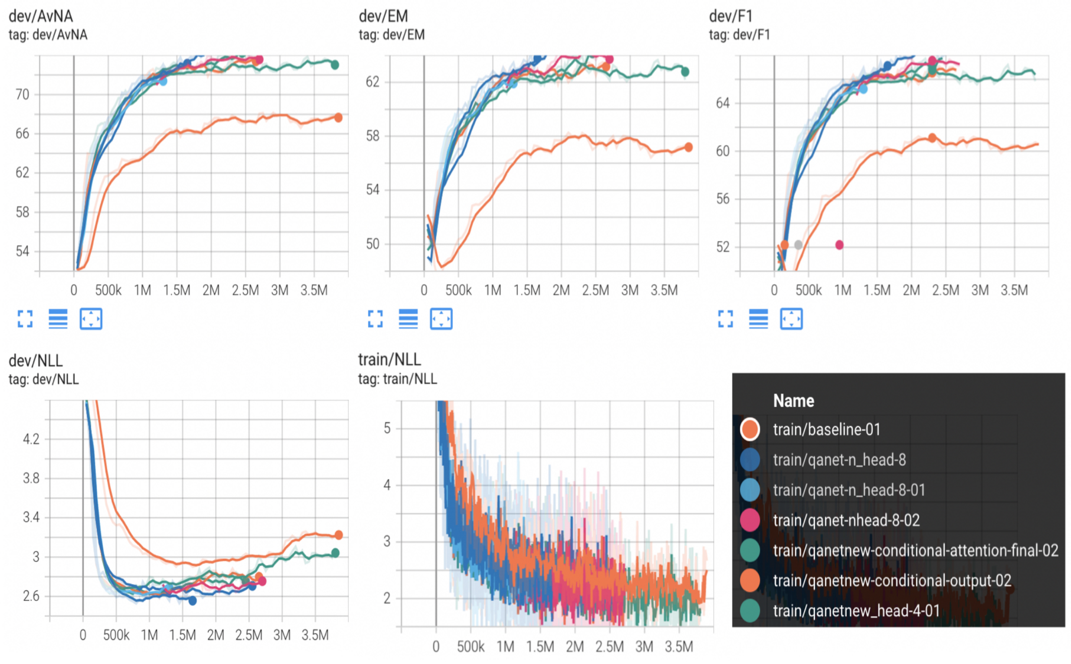


Figure 14: Plot of the dev and training loss, and EM and F1 scores for our models