

QaN I have Your Attention? Exploring Attention in Question-Answering Model Architectures

Stanford CS224N Default Project - Mentor Yuyan Wang

Paula Kusumaputri
Stanford University
paulaks1@stanford.edu

Abstract

In this project, we build non-pre-trained models for the question-answering task on the Stanford Question Answering (SQuAD) 2.0 dataset, exploring on the effect of attention on the result. We explore the performance of deep learning model architectures that utilize attention: BiDAF (context-query attention), Dynamic Co-Attention (second-level attention) and QANet (self-attention). We explored the baseline BiDAF model, and improved it through character embeddings and co-attention, as well as re-implemented QANet. We ensembled results, and obtained highest performance of F1/EM of 67.96/64.41 for single model dev, F1/EM of 71.02/68.39 for ensemble dev, and F1/EM of 68.39/65.83 for ensemble test. We performed analysis on the single model and ensembles to better understand the model mechanisms and performance.

1 Introduction

Understanding a passage of text well enough to answer questions about it requires competence at tasks that traditional machine learning approaches tend to struggle, including ascribing meaning to a word based on other words in its context—nearby and distant. Attention-based techniques have advanced state of the art performance on a variety of machine learning tasks in recent years, including question answering, offering a more effective “understanding” of natural language.

Performance on the Stanford Question Answering Dataset (SQuAD) is commonly used to determine the efficacy of novel techniques in natural language processing. The dataset consists of examples with a context that contains some information, a question about that information, and the answer to that question.

In this work, we explore the performance of three deep learning model architectures that utilize attention: Bidirectional Attention Flow (BiDAF), which employs context-query attention, Dynamic Co-Attention (DCN), which uses a second-level co-attention, and QANet, which leverages self-attention. In addition, we experiment with various methods to extend the basic model architectures, including model ensembling.

2 Related Work

Attention has been utilized in several of the models developed for the SQuAD challenge.

BiDAF [1] achieved a high position on the SQuAD leaderboard (77.3 F1, 68.0 EM) when it introduced context-query attention to existing sequence-aligned RNNs. Since then, the Transformer [2] proposed relying entirely on self-attention to encode representations of inputs without the use of sequence-aligned RNNs. QANet [3] achieved an even higher position on the SQuAD leaderboard (84.6 F1, 76.2 EM) by combining ideas from both BiDAF and the Transformer. While it retained context-query attention to translate meaning between the context and question, it replaced the recurrence-based encoder employed in BiDAF with a transformer-based encoder. What resulted was a model with a higher performance and more efficient training time, as compared to other recurrence-based models.

Dynamic coattention network [4] added a second-level attention to the first-level context-query attention used by BiDAF.

Bidirectional Encoder Representations from Transformers (BERT) [5] introduced pre-trained models that encode bidirectional representations of unlabeled text, so that just one additional output layer is needed to create state-of-the-art models for a wide range of tasks, such as question answer and language inference, without substantial task-specific architecture modifications. Several BERT-derived models have achieved state-of-the-art performance on the SQuAD leaderboard.

3 Approach

3.1 Bi-Directional Attention Flow (BiDAF)

The default starter-code [6] provided us with a baseline version of Bi-Directional Attention Flow (BiDAF) model, which consists of 5 layers: embedding layer, encoder layer, attention layer, modeling layer, and output layer.

We implemented character-level embeddings to the **embedding layer** in order to improve morphology and out-of-vocabulary words. An initial embedding, padded and truncated to a length of 16 characters per word, is passed through a 1d-convolution filter and max-pool to give a fixed-size embedding with length $d_c = 200$. The character-level embedding is then concatenated with a GloVe [7] pre-trained word embedding with length $d_w = 300$. The resulting concatenated $d_w + d_c$ word representation is passed through a projection and 2-highway network [8].

The original BiDAF **context-query attention layer** consists of the weights α_i, β_j and outputs a_i, b_j .

$$\begin{aligned} \alpha^i &= \text{softmax}(L_{i,:}) \in \mathbb{R}^{M+1} \text{ and } a_i = \sum_{j=1}^{M+1} \alpha_j^i q'_j \in \mathbb{R}^l \\ \beta^j &= \text{softmax}(L_{:,j}) \in \mathbb{R}^{N+1} \text{ and } b_j = \sum_{i=1}^{N+1} \beta_j^i c_i \in \mathbb{R}^x \end{aligned} \quad (1)$$

Inspired by the Dynamic Coattention Network [4], we added a second-level attention to improve the representations of the context-to-query and query-to-context attentions. The second-level attention output combines α_j weights with b_j outputs.

$$s_i = \sum_{j=1}^{M+1} \alpha_j b_j \in \mathbb{R}^l \quad (2)$$

3.2 QANet

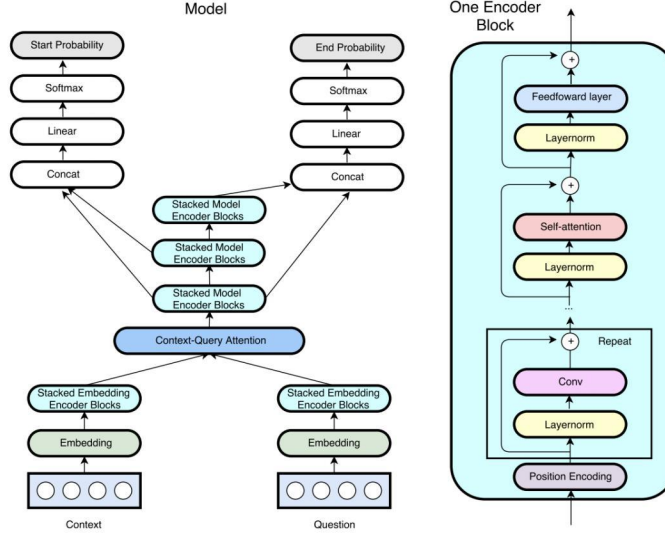
Inspired by the high-performance of transformer-based models for question-answering, we implemented QANet, which consists of the same 5 layers as BiDAF: embedding layer, encoder layer, attention layer, modeling layer, and output layer (Figure 1).

The QANet **embedding layer** consists of word-level and character-level embeddings. We used our implementation of character-level embeddings from BiDAF.

Instead of the LSTM-based encoder used in BiDAF, the QANet **embedding encoder layer** consists of a positional-encoding and 3 types of residual sub-blocks: convolutional, self-attention, and fully-connected. Without sequence-based RNNs, QANet encodes word sequence with sine/cosine-wave frequency super-positions [2]. The convolutional sub-block consists of depthwise separable convolutions [9][10]. The self-attention sub-block consists of a variation of causal self-attention from Assignment 5 [11]. Unlike in the assignment, the QANet encoder requires masking of padding words but not masking future words. The fully-connected sub-block consists of a standard fully-connected layer. All residual blocks use layer-normalization [12] on the embedding dimension. Weights are shared between the context and question embedding encoders.

The d_x -output of embedding encoder for both the context and question is then passed through the **context-query attention layer**. For this layer, we reused the baseline model's BiDAF Attention layer. The d_{4h} output of context-query attention layer is passed through a series of 3 stacked model encoder

Figure 1: QANet Model Architecture diagram



blocks in the **model encoder layer**. Note that the model encoder block has the same structure as embedding encoder block, but with different numbers of convolution-blocks and number of repeated model blocks.

After the series of 3 stacked model encoder blocks finish their calculations, these outputs are passed to **output layer**. Similar to BiDAF, the QANet calculation is done through a start answer pointer and end answer pointer. The start answer pointer is calculated from M_0, M_1 model output, and end answer pointer is calculated from M_1, M_2 model output. They calculate the probability for each of the context word to be the start or end of the answer span.

4 Experiments

4.1 Data

Our dataset is derived from SQuAD 2.0 dataset [13], where instead of directly accessing the official test sets, we use a split version of the official dev data. This split dataset consists of 129941 training, 6078 dev, and 5915 test examples, which are in the form of examples e , a triple of $e = (c, q, a)$, containing context, question, and answer (a span from the context). In SQuAD 2.0, it is possible for a given (c, q) , that there exists no span that answers the question, as the question is unanswerable.

4.2 Evaluation method

We evaluate primarily on the F1 and EM (Exact Match) scores, also used in the official SQuAD 2.0 evaluation. We also include the AvNA (Answer vs No Answer) to better understand the model's classification accuracy considering only the answer (any span predicted) vs no-answer predictions.

4.3 Experimental details

4.3.1 BiDAF Model

We implemented the character-level embedding, and replicated parameters from BiDAF paper: with max 16 number of characters in a word, character embedding size 64 passed through 100 1d-convolutions, each with kernel width 5. For BiDAF- model hyper-parameters, we used hidden size of 100, learning rate of 0.5, batch size 64, dropout rate 0.2, and trained for 30 epochs with an AdaDelta [14] optimizer. Early on, we had observed the importance of using exponential moving average (ema) throughout our experiments, since BiDAF2(-ema) performance reduced by $-1.43\text{F1}/-1.41\text{EM}$.

We implemented second-level co-attention into our baseline context-query attention, but did not observe large performance improvement, even after hyperparameter-tuning of using $p_{drop} \in \{0.1, 0.2\}$. After training these models, we ensembled four of our high-performing BiDAF models, which showed a significant improvement in F1 and EM over any BiDAF single model.

4.3.2 QANet Model

We trained QANet model for 30 epochs and followed hyper-parameters from the paper, such as using character embedding dimension 200, hidden size 128. We also used the Adam optimizer [15] with inverse-exponential learning rate in first 1000 steps, constant at 0.001 thereafter, and with $\beta_1 = 0.8, \beta_2 = 0.999, \epsilon = 10^{-7}$.

We observed model’s over-fitting tendency during experimentation, and discovered the importance of sufficient regularization. We applied dropout on word- ($p_w = 0.1$) and character- ($p_c = 0.05$) embedding, dropout rate of 0.1 elsewhere and a dropout every 2 layers in the convolution sub-block of the encoder block, as well as dropout every 2 model encoder block stacks. On the residual block, we applied stochastic-depth layer dropout [16], with $p_l = 1 - \frac{l}{L}(1 - p_L)$ where $p_L = 0.9$.

Due to GPU memory limitations, we reduced the batch-size from 64 to either 32 or 16. We also used smaller number of attention-heads and convolution blocks in the model encoder layer, while keeping the rest of parameters same as the paper (embedding encoder block 1, number model conv layer 2, number embedding conv layer 4, hidden size 128). When memory errors persist, we switched from Azure NC6 to NC12 GPU.

4.3.3 Data Augmentation

We added data-augmentation by back-translation, as per QANet [3] paper to reduce overfitting. Instead of translating the context as it was done by the paper, we translated the questions to add diversity to the way a type of question is asked. We implemented this using Google Translate API [17] with English/French back-translation. With time-constraints and API limits, we only translated 5% of all 129k training examples, and this lack of additional diversity through data-augmentation seemed insufficient to improve model.

4.3.4 Active Learning

In our initial analysis, we observed the difficulty it was for the single QANet models to learn "hard" examples, where "hard" is measured by higher loss value. To alleviate this, we experimented by using a naive version of active learning. We trained an additional 10% of data at the end of each epoch, starting from epoch 15, by sampling with probability proportional to the example’s loss value. The idea is that we have the model re-learn the "hard" examples once they are halfway through the epochs, such that they are able to learn those examples better.

When sampling with replacement, the model continues to try (and fail) learning the same set of hundreds of examples multiple times during training, plummeting the total F1/EM scores. Once we switch to sampling without replacement, the model is able to improve per-epoch on these harder examples.

4.3.5 Ensembling

Ensemble-methods has proven to be effective in increasing performance over single models, as it is able to combine multiple 'weaker'-learners to build a stronger learner for the task. We experimented with using two different ensemble methods: averaging probability provided by each model to produce the start/end-pointers, as well as direct majority-voting on the CSV files produced by each model for the answer spans.

4.4 Results

The results for subset of the models run can be seen in the table 1, broken down by the dev and test set. As we had limited test-submissions, not all of the results includes the test F1/EM scores. Based on these results, we saw that each QANet- models performed higher than the BiDAF- models. In addition, the ensemble methods performed higher than each individual single-model. We also observed a slight (approximately -2F1/EM) decrease of the test set evaluation.

Model Name	Description	Dev F1	Dev EM	Dev AvNA	Test F1	Test EM
Single Models						
BiDAF1	Baseline	60.43	57.08	67.53	-	-
BiDAF2	Base+CharEmb	62.89	59.50	69.53	-	-
BiDAF3	Base+CharEmb+CoAtt1	62.34	58.76	68.66	-	-
BiDAF4	Base+CharEmb+CoAtt2	63.34	59.30	70.29	-	-
QANet0	3Conv+2Head (batch size=64)	66.90	63.33	73.10	-	-
QANet1	3Conv+4Head (batch size=32)	67.96	64.41	74.63	-	-
QANet2a	5Conv+4Head (batch size=32)	68.26	65.00	73.89	-	-
QANet2b	5Conv+4Head (batch size=32)	68.60	65.15	75.01	-	-
QANet3	5Conv+8Head (batch size=16)	68.38	65.06	74.19	-	-
Ensemble Models						
Ensemble1	BiDAF (Majority, 4 models)	66.15	63.38	70.96	-	-
Ensemble2	QANet (Majority, 3 models)	69.92	66.93	75.06	-	-
Ensemble3	QANet (Majority, 5 models)	70.66	67.87	75.75	68.39	65.44
Ensemble4	QANet (Ave Prob, 5 models)	70.53	67.45	75.77	-	-
Ensemble5	QANet+BiDAF (Majority)	71.02	68.39	75.55	68.39	65.83

Table 1: Table to show F1/EM results of some of our trained SQuAD models

5 Analysis

5.1 Data Slice Analysis

We examined the performance of a singleton QANet from our ensemble to better understand the strengths and limitations of the approach, independent of model ensembling.

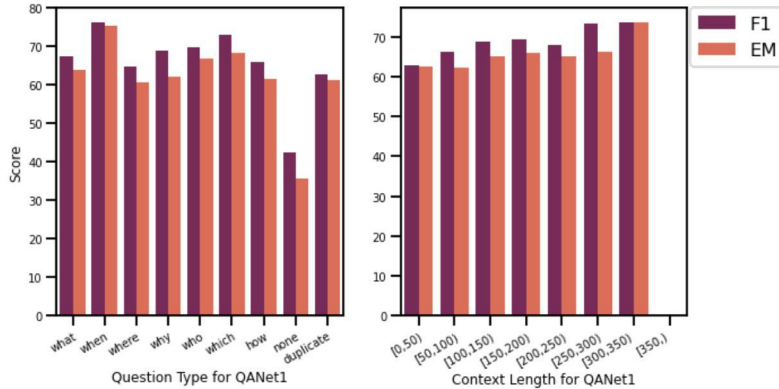


Figure 2: **Different question categories (left).** Performance was consistent for questions containing one or more occurrences of exactly one common question word: what (n=3,490), when (n=426), where (n=246), why (n=87), who (n=664), which (n=208), or how (n=555). Performance was worse for questions in the none category (n=59). **Different context lengths (right).** Performance was consistent for contexts of different lengths, but improved for longer contexts compared to shorter contexts: 0-49, 50-99, 100-149, 150-199, 200-249, 250-299, 300-349. Context lengths of 350+ were not evaluated.

First, we examined the performance of the model on different question categories (Figure 2). Questions containing one or more occurrences of exactly one common “question word” were assigned to their own categories: what, when, where, why, who, which, or how. Questions containing zero occurrences of any of these words were assigned to the none category, and questions containing occurrences of at least two of these words were assigned to the duplicate category. We observed consistent performance for questions containing exactly one occurrence of a question word (64.8-76.2

F1, 60.6-75.4 EM). Both metrics were lowest for *where* question words and highest for *when* question words. We observed significantly worse performance for questions in the *none* category (43.3 F1, 35.6 EM). Manual inspection of these questions revealed them to be exceptionally difficult for reasons such as inverted sentence structure and ambiguous wording (Appendix A.1).

We also examined the performance of the model on different context lengths (Figure 2). Contexts were discretized into lengths of 0-49, 50-99, 100-149, 150-199, 200-249, 250-299, 300-349, and 350+. We observed consistent performance for contexts of different lengths (63.0-73.7 F1, 62.4-73.7 EM). Both metrics were lower for the shorter context lengths and higher for the longer context lengths. These results indicate that although the model needs to deliberate between a larger number of possible answer spans in longer context lengths than shorter context lengths, longer context lengths may offer more useful information to contextualize words in the context itself as well as the question.

Lastly, we examined the error types of the answer produced by a single-model: whether the predicted answer was a substring, an overlap, or completely missed the ground truth answer spans in the paragraph (Figure 3). We can see that the most common error occurs when the model incorrectly predicted some answer, even though the question is unanswerable. As some of the errors were due to positional issues in the span (either prediction was too long or too short) modifications that might still need to be made to the model architecture to obtain a more precise answer span.

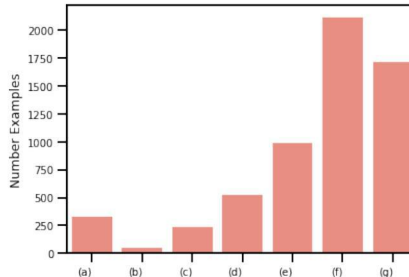


Figure 3: **Different answer types.** (a) prediction is substring of the ground-truth or vice-versa (b) some overlap between them, (c) no overlap between them, (d) prediction is incorrectly NA, (e) prediction is incorrectly an answer, (f) prediction is correctly NA, (g) prediction and ground-truth are exact match

5.2 Attention Analysis

Given the reliance of both BiDAF and QANet on context-query attention and self-attention, we performed additional analysis to characterize the behaviors of these attention mechanisms. To demonstrate the difference, we randomly chose an example that was incorrectly classified by BiDAF but correctly classified by all of our QANet variations (Table 2).

Question	The word imperialism has it's origins in which ancient language?
Context	Imperialism is a type of advocacy of empire. Its name originated from the Latin word "imperium", which means to rule over large territories. Imperialism is "a policy of extending a country's power and influence through colonization, use of military force, or other means". Imperialism has greatly shaped the contemporary world. It has also allowed for the rapid spread of technologies and ideas. The term imperialism has been applied to Western (and Japanese) political and economic dominance especially in Asia and Africa in the 19th and 20th centuries. Its precise meaning continues to be debated by scholars. Some writers, such as Edward Said, use the term more broadly to describe any system of domination and subordination organised with an imperial center and a periphery.
BiDAF Answer	Western
QANet Answer	Latin
UUID	1ceb01ddf0ba4c93fb95e6b40

Table 2: Example 1ceb01ddf0ba4c93fb95e6b40 is incorrectly classified by BiDAF and correctly classified by QANet.

Inspecting the context-query attentions (Figure 4) shows subtle but insignificant differences in the context-query attention weights output by the two models. However, primarily due to differences in the modeling layers, BiDAF outputs uncertain probabilities for No Answer and *Western*, whereas QANet outputs a stronger probability for *Latin*, with *Western* as a potential second-best answer (Figure 5). These results indicate that use of self-attention residual blocks in the modeling layer of QANet improves upon the use of recurrence-based RNNs in the modeling layer of BiDAF.

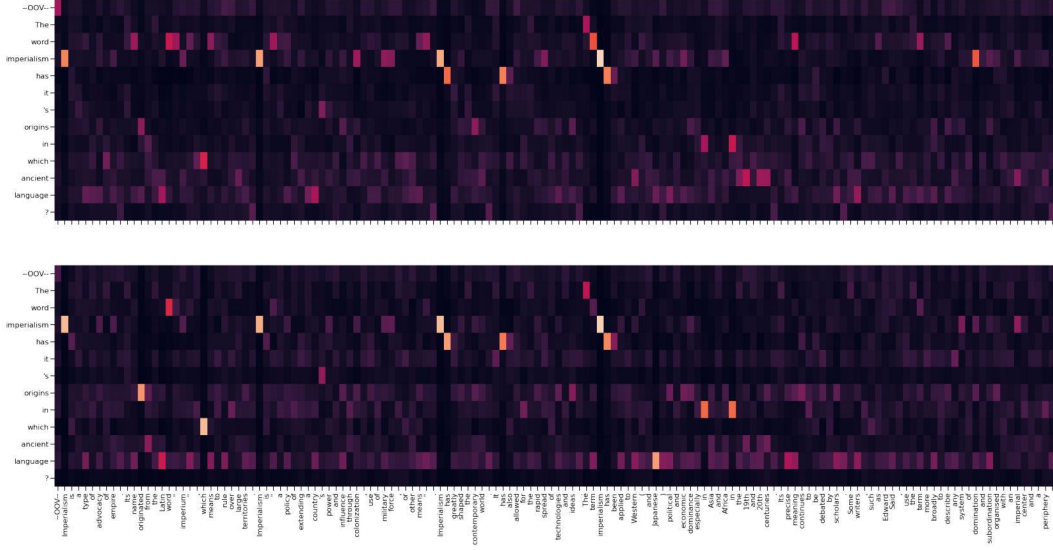


Figure 4: The context-query attention for the context of example 1ceb01ddf0ba4c93fb95e6b40 for **BiDAF** (top) vs. **QANet** (bottom) shows subtle but insignificant differences.

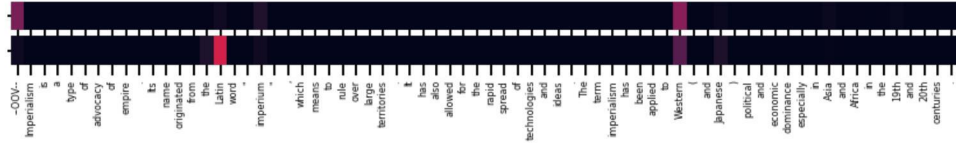


Figure 5: The start pointer probabilities on example 1ceb01ddf0ba4c93fb95e6b40 show uncertainty between two incorrect answers by **BiDAF** (top) vs. high confidence in the correct answer by **QANet** (bottom). Words with approximately 0 predicted probability by both models are truncated for readability purposes.

5.3 Ensemble Analysis

Our model ensembles outperform our singleton models (Table 1) due to model diversity and improved confidence.

To determine the level of agreement between the models and how they contribute to the ensemble final vote, we calculated the inter-rater reliability with Fleiss kappa scores on AvNA metric. We also conditioned the score conditioned on the ensemble producing the correct vs. incorrect answer, and found that disagreements exist much more often for the incorrectly classified examples (0.41 Fleiss) than correctly classified ones (0.74 Fleiss). Together, these results suggest that the level agreement of the ensemble can be used as a proxy for confidence in our score—when all of the models in the ensemble agree, we have stronger belief that our answer is correct. The ability to produce a confidence in a prediction is a powerful feature in potential real-world applications.

The trend in agreement scores are similar for all of our Ensemble3 and Ensemble4, showing that the different methods of ensembling produces similar agreement patterns among the models.

Based on these values claiming disagreement, we sample some examples (Table 3) incorrectly classified by the ensemble which had some disagreements between the models.

Error Type: Models correctly chooses entity type with wrong value.	
Question	What was the population of the Dutch Republic before this emigration?
Context	After the revocation of the Edict of Nantes, the Dutch Republic received the largest group of Huguenot refugees, an estimated total of 75,000 to 100,000 people. Amongst them were 200 clergy. Many came from the region of the Cévennes, for instance, the village of Fraissinet-de-Lozère. This was a huge influx as the entire population of the Dutch Republic amounted to ca. 2 million at that time. Around 1700, it is estimated that nearly 25% of the Amsterdam population was Huguenot.[citation needed] In 1705, Amsterdam and the area of West Frisia were the first areas to provide full citizens rights to Huguenot immigrants, followed by the Dutch Republic in 1715. Huguenots intermarried with Dutch from the outset.
Model Answers	'2 million', '75,000 to 100,000', '75,000 to 100,000', '75,000 to 100,000', '75,000 to 100,000'
Ensemble Answer	"75,000 to 100,000"
Ground-Truth	'ca. 2 million', '2 million', '2 million'
UUID	028fba9bb26fdb32ee8a0a47
Error Type: Error in Cause and effect where the answer provided by model is what caused the unequal treatment instead of how it made dynasty seem.	
Question	How did the unequal treatment of Chinese versus Mongols in the Yuan make the dynasty seem?
Context	Politically, the system of government created by Kublai Khan was the product of a compromise between Mongolian patrimonial feudalism and the traditional Chinese autocratic-bureaucratic system. Nevertheless, socially the educated Chinese elite were in general not given the degree of esteem that they had been accorded previously under native Chinese dynasties. Although the traditional Chinese elite were not given their share of power, the Mongols and the Semuren (various allied groups from Central Asia and the western end of the empire) largely remained strangers to the mainstream Chinese culture, and this dichotomy gave the Yuan regime a somewhat strong "colonial" coloration . The unequal treatment is possibly due to the fear of transferring power to the ethnic Chinese under their rule . The Mongols and Semuren were given certain advantages in the dynasty, and this would last even after the restoration of the imperial examination in the early 14th century. In general there were very few North Chinese or Southerners reaching the highest-post in the government compared with the possibility that Persians did so in the Ilkhanate. Later the Yongle Emperor of the Ming dynasty also mentioned the discrimination that existed during the Yuan dynasty. In response to an objection against the use of "barbarians" in his government, the Yongle Emperor answered: "... Discrimination was used by the Mongols during the Yuan dynasty, who employed only "Mongols and Tartars" and discarded northern and southern Chinese and this was precisely the cause that brought disaster upon them
Model Answers	fear of transferring power to the ethnic Chinese under their rule
Ensemble Answer	fear of transferring power to the ethnic Chinese under their rule
Ground-Truth	colonial, somewhat strong "colonial" coloration, colonial
UUID	11cbcdf80fae8444eb3c95ce
Error Type: Model cannot differentiate between two similar types (ie new vs old oil).	
Question	What happened when new oil was withdrawn from the market?
Context	Price controls exacerbated the crisis in the US. The system limited the price of "old oil" (that which had already been discovered) while allowing newly discovered oil to be sold at a higher price to encourage investment. Predictably, old oil was withdrawn from the market, creating greater scarcity . The rule also discouraged development of alternative energies. The rule had been intended to promote oil exploration. Scarcity was addressed by rationing (as in many countries). Motorists faced long lines at gas stations beginning in summer 1972 and increasing by summer 1973.
Model Answers	NA, NA, 'creating greater scarcity', 'creating greater scarcity', 'creating greater scarcity'
Ensemble Answer	creating greater scarcity
Ground-Truth	No Answer
UUID	03ffeabaf7f55deef26b0ae15

Table 3: Sample data classified by the ensemble.

6 Conclusion

In this paper, we have implemented and explored different attention-based models, and ensembled them for SQuAD 2.0 question-answering task. We achieved performance of F1/EM 71.02/68.39 on Dev Leaderboard, and 68.39/65.83 on Test Leaderboard.

We had created models, and analyzed the role attention played in them. In our experimentation efforts to improve performance, we found a couple of approaches (such as data augmentation through back-translation of questions, or naive active-learning of the sampled train data) to not improve much. Although this could be attributed to many different reasons, we hypothesize that QANet required a lot more data, or some inherent modification to the architecture if these approaches are to work completely. In addition, we created an ensemble of our models and analyzed some issues with the way our ensemble is choosing the answer.

We believe there is a lot of future work to both improve on attention mechanism to evaluate performance, as well as designing better ways to evaluate the efficacy of these methods.

References

- [1] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.
- [2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [3] 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. *arXiv preprint arXiv:1804.09541*, 2018.
- [4] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering, 2018.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [6] CS 224N Default Final Project: Building a QA system (IID SQuAD track). <http://web.stanford.edu/class/cs224n/project/default-final-project-handout-squad-track.pdf>, 2021. [Online; accessed 2-Mar-2021].
- [7] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [8] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks, 2015.
- [9] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.
- [10] Lukasz Kaiser, Aidan N. Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation, 2017.
- [11] Assignment 5: Self-Attention, Transformers, and Pretraining). <http://web.stanford.edu/class/cs224n/assignments/a5.pdf>, 2021. [Online; accessed 18-Mar-2021].
- [12] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.
- [13] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don’t know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.
- [14] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.
- [15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [16] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016.
- [17] Google Translate API. <https://cloud.google.com/translate>, 2021. [Online; accessed 01-Mar-2021].

A Appendix

A.1 Illustrative examples of the exceptional difficulty of the none question type category

Question	The term Huguenot was originally meant to confer?
Context	A term used originally in derision , Huguenot has unclear origins . Various hypotheses have been promoted. The nickname may have been a combined reference to the Swiss politician Besançon Hugues (died 1532) and the religiously conflicted nature of Swiss republicanism in his time, using a clever derogatory pun on the name Hugues by way of the Dutch word Huisgenoten (literally housemates), referring to the connotations of a somewhat related word in German Eidgenosse (Confederates as in "a citizen of one of the states of the Swiss Confederacy"). Geneva was John Calvin's adopted home and the centre of the Calvinist movement. In Geneva, Hugues, though Catholic, was a leader of the "Confederate Party", so called because it favoured independence from the Duke of Savoy through an alliance between the city-state of Geneva and the Swiss Confederation. The label Huguenot was purportedly first applied in France to those conspirators (all of them aristocratic members of the Reformed Church) involved in the Amboise plot of 1560: a foiled attempt to wrest power in France from the influential House of Guise. The move would have had the side effect of fostering relations with the Swiss. Thus, Hugues plus Eidgenosse by way of Huisgenoten supposedly became Huguenot, a nickname associating the Protestant cause with politics unpopular in France.
Model Answers	unclear origins
Ground-Truth	derision
UUID	00bafbca5f0d7f61e00a41cb5

Question	If this was developed for the Air Force, then does the Air Force still technically own the Intellectual Property?
Context	Baran developed the concept of distributed adaptive message block switching during his research at the RAND Corporation for the US Air Force into survivable communications networks, first presented to the Air Force in the summer of 1961 as briefing B-265 , later published as RAND report P-2626 in 1962, and finally in report RM 3420 in 1964. Report P-2626 described a general architecture for a large-scale, distributed, survivable communications network. The work focuses on three key ideas: use of a decentralized network with multiple paths between any two points, dividing user messages into message blocks, later called packets, and delivery of these messages by store and forward switching.
Model Answers	briefing B-265
Ground-Truth	No answer
UUID	22820cdaed0afd0fab4d0ab3c

A.2 Training Tensorboard Curves

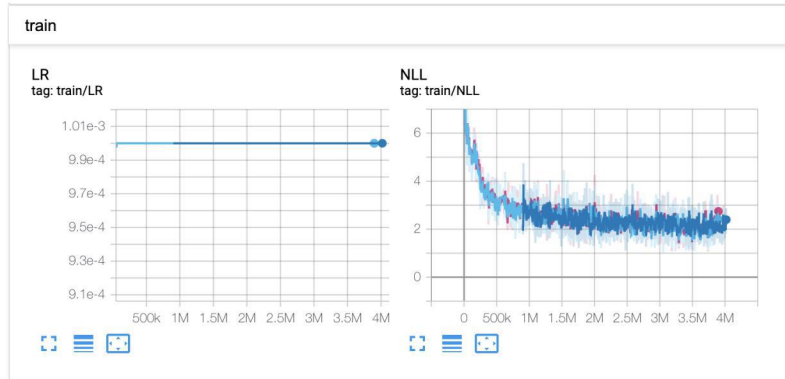


Figure 6: Tensorboard-displayed train LR and NLL on some of the QANet models done during training. Note that I forgot to run tmux on one of y training, so the dark blue line started from the middle as it was loaded from a previous checkpoint.

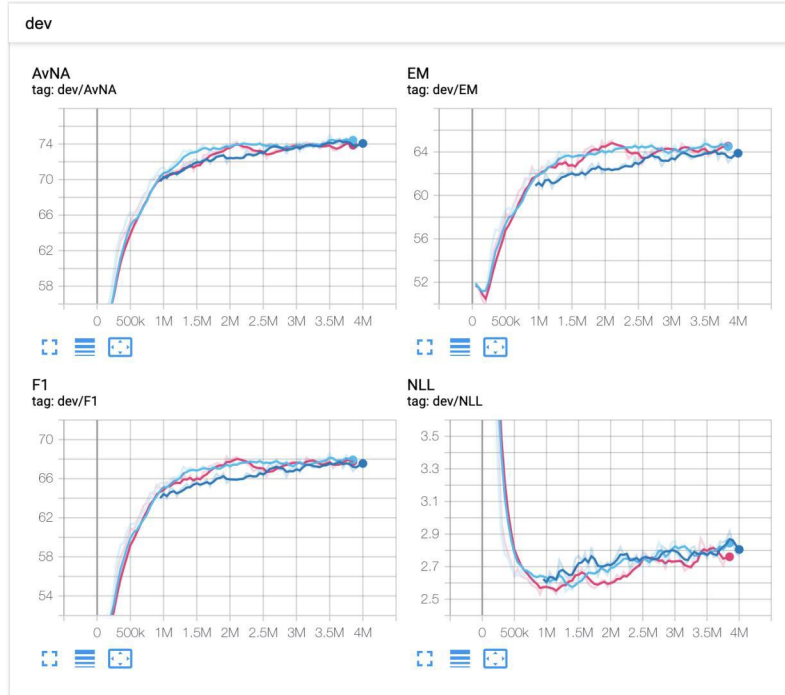


Figure 7: Tensorboard-displayed dev evaluation on some of the QANet models done during training. Note that I forgot to run tmux on one of y training, so the dark blue line started from the middle as it was loaded from a previous checkpoint.

A.3 Remarks

During implementation and analysis, we used the Azure GPUs provided by the course, as well NVIDIA Titan-X GPU locally. QANet implementation was done from scratch, but due to some major overfitting issues in the dev, we refer to QANet implementation (<https://github.com/NLPLearn/QANet>) for debugging purposes, as was recommended to by my mentor Yuyan and other TAs in Office Hours. From this, we derived: a PyTorch-version of their positional-encoding implementation, their method of He/Xavier weight initialization, and modified where we applied our dropout locations. For example, we added dropout before the first and third model encoder block that previously did not exist. Graphs, diagrams and most analysis was done with a combination of python scripts and some Jupyter notebooks.