Reasoning on Multi-Hop Questions with HotpotQA

Jamil Dhanani
Stanford University
jdhanani@stanford.edu

Suma Kasa
Stanford University
sumakasa@stanford.edu

Suprita Shankar
Stanford University
suprita@stanford.edu

Abstract

Question Answering (QA) is a highly researched topic in Natural Language Processing, but so far, the focus has mainly been on “factoid" questions, with questions that focus on a single sentence in the corpus, without additional reasoning. The challenging problem of identifying answers with multiple hops of reasoning in text has been relatively unexplored. To address this, the HotpotQA dataset contains questions that require more complex reasoning, which may require information across several paragraphs of text. The paper also introduces a baseline model with some performance metrics. In this paper, we propose improvements on this baseline model. We conducted error analysis on the predictions by the baseline model, identified improvements, and trained models based on those improvements. On several of the models we trained, we showed an improvement in the F1 score and EM score, across both the Answer and Supplementary Fact categories. Our best model obtained an F1 score of 65.72% (7.4 points more than the baseline) on answers and 78.99% (12.3 points above the baseline) on the supporting facts on the dev eval set in the distractor setting.

1 Introduction

Question Answering is an integral part of intelligent systems today. It is applied in a wide variety of applications, from voice assistants, like Alexa and Siri to more complicated tasks such as Visual QA. To build a robust QA System, an important ability is to be able to answer multi hop questions. A simple example of multi-hop question is *"Where was the Governor of California born?"*. Depending on the document, we may not be able to infer the answer using one sentence. We may have to first identify the Governor of California (currently, Gavin Newsom), and then, once that is resolved, get his place of birth. Previously, knowledge bases and knowledge graphs were used to answer these type of queries. The main downside of the approach is that we are constrained by the schema and ontology of the knowledge base.

2 Related Work

2.1 Dataset

There have been many popular datasets such as SQuAD (1), which aim to provide a dataset to train models on reading comprehension. SQuAD has questions that are designed to be answered by analyzing a single paragraph, but it is not as representative of more complex questions we might see, that require additional reasoning.

To overcome the above limitations, HotpotQA provides questions that require reasoning over multiple documents, in natural language, without constraining itself to an existing knowledge base (2). It also provides the system with strong supervision in the form of *supporting facts* to understand how the answer was derived, which will help guide systems to perform meaningful, explainable reasoning. HotpotQA stands out among other QA datasets as it provides a more realistic and difficult dataset, representing more closely how QA systems should be built in reality.

There are two settings for the HotpotQA dataset; the *distractor* and *fullwiki* setting. To challenge the model to find the true supporting facts in the presence of noise, the authors use bigram TF-IDF to

retrieve eight paragraphs as “distractors" given the question query. These eight paragraphs, along with the two gold paragraphs, are then fed into the system to test its robustness. In the *fullwiki* setting, the first paragraph of all the Wikipedia articles is fed into the system. According to the authors, this "tests the performance of the system ability at multi-hop reasoning in the wild" (2).

2.2 QA Systems

Although the number of papers on HotpotQA is very limited, there are other works that have attempted to address other question answering tasks, such as SQuAD and TriviaQA (3).

The baseline model in HotpotQA is based on a model proposed by Clark and Gardner in *Simple and Effective Multi-Paragraph Reading Comprehension* (4). The model uses a combination of GloVe-based word embeddings, and character embeddings combined using a CNN. These concatenated word embeddings are passed through GRUs, and then through a bi-attention layer to combine information from both the context and query. Please refer to the diagram of the model in A.

The Clark and Gardner model mentioned above is inspired in part by Birectional Attention Flow (BiDAF), a concept introduced in *Bidirectional Attention Flow for Machine Comprehension* (5). Bi-Attention is a crucial part of the baseline model as it enables the model to learn the relationship between the query and the context, thus helping the model to learn a query-aware context representation.

Another important work which influenced architectures for common NLP tasks, including QA, is *Attention Is All You Need* (6). In this paper, the authors propose to replace the traditional recurrent architecture with "transformers", a attention-based unit. The authors replace the traditional RNN architecture with a purely attention-based architecture, which assists with parallelization during training, and improves model performance on several different tasks, including SQuAD.

Another paper, *Language Modeling with Gated Convolutional Networks* (7) establishes the success of non-recurrent approaches for language tasks. This paper takes advantage of Gated CNNs instead of RNNs for language modelling tasks.

To address the issue of reasoning in NLP, Weston, Chopra, and Bordes proposed an architecture called Memory Networks (8), which aims to reason with inference components combined with a long-term memory store, which can be written to and read from. Reasoning about relationships between entities and spans in the text is an important part of the question answering task, which is why the authors chose QA as one of the key applications of memory networks.

3 Approach

3.1 Baseline Model

The baseline model is adopted from the Clark and Gardner paper “Simple and Effective Multi-Paragraph Reading Comphrehension” (4) published in ACL 2018. It achieves a 15 point increase in F1 over the prior work on TriviaQA (3), a dataset which includes questions and multi-paragraph documents from which they are retrieved. The only addition to the model described in this paper to the baseline model is the 3-way classifier after the last recurrent layer to produce the probabilities of “yes”, “no”, and span-based answers. Please refer to appendix A for a detailed explanation.

The baseline model achieves the following performance on the dataset, on the dev set in the distractor setting. Definitions of the metrics are given in Section 4.2 on Evaluation.

Table 1: Baseline Model Metrics

Ans EM	Ans F1	Sup EM	Sup F1	Joint EM	Joint F1
44.44	58.28	21.95	66.66	11.56	40.86

3.2 Baseline Model Analysis and Modifications

The following table summarizes the errors encountered in the evaluation set, split by both the answer (*Ans*) and supporting facts (*Sup*). These numbers are given for the Exact Match metric.

Table 2: Baseline Model Errors

Total Dev set	Total correct answers	Total correct ans + correct sp	Incorrect ans+ correct sp	Incorrect ans+ incorrect sp
7405	2193 (29.61%)	108 (4.93% of correct ans)	1167 (15.75%)	3224 (4.35%)

One of the most evident issues with the baseline model is that it has poor EM metrics on *Sup*, even when the model has the correct answer. The model used in the HotpotQA paper adds an auxiliary output to generate a binary classification on whether or not the phrase is a supporting fact, with the use of an additional RNN (not included in the Clark paper) after the self-attention layer. Modifications to this output and the means of generating it will be a focus in the project.

We identified different issues with the baseline model that we aimed to address in this project.

3.2.1 Learning and Optimization

We noticed that the learning rate decay in the baseline model is quite aggressive; at each epoch after the patience is exceeded, the learning rate is cut in half, until it reaches 1% of the initial learning rate. As we show in Section 4.5, after a certain number of training steps, the training loss very quickly plateaus and does not decrease significantly further. Adjustments to the learning rate decay, or different optimization algorithms, may assist during training.

3.2.2 Attention

The baseline model has bi-attention on the context and query, which is useful for finding related words between the two texts, but there are some issues of *importance* in both the query and context that are not captured in the baseline. Here are examples which demonstrates this deficiency:

Table 3: Selected Errors Due to Attention

	Issue in Ans	**Issue in Sup**
Question	Which magazine began publication first, Reunions magazine or Money?	Who is older, Annie Morton or Terry Richardson?
Gold	**Ans:** Money	**Ans:** Terry Richardson **Sup:** Annie Morton, Terry Richardson
Baseline	**Ans:** magazine	**Ans:** Annie Morton **Sup:** Annie Morton, Kenton Richardson
Explanation	Not attending enough to the words *Reunions* and *Money* in the query	The baseline did not attend to the correct entities in the question

Some form of self-attention can help in migitigating these kinds of errors.

3.2.3 Reasoning

HotpotQA has many questions that require "multi-hop" reasoning, but the baseline model was designed for datasets without this type of question, where the answer is explicitly answered in a single part of the paragraph. Additional layers of bi-attention may help here, so that the model may reason past just a single "hop"; it can take the output of the first attention between context and query, and it may learn more sophisticated relationships between them. Adding additional bi-attention layers has the added effect of increasing the representational power of the model by increasing the number of parameters and layers.

3.2.4 Representation

Smaller, less expressive models may not have the representational power to model the complex relationships required to solve the task. We explore different methods to ensure enough information is passed to each layer, and enough weights are in the model to perform well on this task.

4 Experiments

4.1 Dataset

We use HotpotQA, a question answering (QA) dataset containing complex, multi-hop questions, for our experiments. The dataset provides 112,779 questions which require reasoning over multiple hops to answer the question.The dataset has two kinds of benchmark settings : distractor mode where the model has to identify the supporting facts to retrieve the answers in presence of noise paragraphs and fullwiki mode where the model has to identify the supporting facts from full wikipedia text. The scope of this project is strictly limited to distractor setting. Refer back to Section 1 and 2 for more information on this dataset.

4.2 Evaluation Method

In order to show improvement to the baseline model, we will use two primary metrics, Exact Match (EM) and Macro-Averaged F1, across three different categories, the Answer (*Ans*), Supplementary Facts (*Sup*), and the joint between *Ans* and *Sup*. The EM score is defined as the percentage of predictions that match one of the ground truth answers or supplementary facts exactly (first defined in (1)), and the macro-averaged F1 measures the average overlap between the prediction and ground truth answer. The joint F1 is calculated as

$$P^{(joint)} = P^{(ans)}P^{(sup)}, R^{(joint)} = R^{(ans)}R^{(sup)}$$

$$\texttt{Joint F1} = \frac{2P^{(joint)}R^{(joint)}}{P^{(joint)} + R^{(joint)}}$$

The joint EM is 1 only if both tasks achieve an exact match and otherwise 0. All metrics are calculated on each question-answer pair individually, and averaged over the entire set. This is as defined in (2). We will report the EM and F1 scores for each of our experiments, for all of *Ans*, *Sup*, and *Joint*.

4.3 Experimental Details

We organized our experiments and architectural changes based on the issues identified with the baseline model in Section 3.2, specifically around **learning and optimization**, **attention**, **reasoning**, and **representation**. We have divided this section based on those categories.

4.3.1 Attention

As elaborated earlier, we observed that the baseline model had errors which may have been caused by a lack of attention on important spans in both the query and the context. We trained several models with different architectural changes to help mitigate this issue.

- **Model Att-A** (self-attention, query only): We pass the query through a self-attention layer, followed by a linear layer to reduce the size of the activation, followed by the existing bi-attention layer in the model. The rest of the architecture is the same. The motivation for this change is to add attend to the question, because evidence that the model was not attending to relevant parts of the question was found in error analysis.
- **Model Att-B** (self-attention, query+context with separate weights): We pass both the query and paragraphs through their own respective self-attention layers followed by linear layers, before combining them in the existing bi-attention layer in the model.
- **Model Att-C** (self-attention, query+context with shared weights): This model is similar to **C**, except that the weights in the self-attention layer and the following linear layer are shared between the context and the query.

Diagrams of selected self-attention architectures (**Att-A**, **Att-B**, and **Att-C**) are in Appendix D.

4.3.2 Learning and Optimization

- **Model Opt-A** (lr /= 1.5): Since we noticed that the learning rate decay was perhaps happening too quickly, the first thing we tried was simply adjusting the update rule to divide the learning rate by 1.5 instead of 2.

- **Model Opt-B** (lr /= 1.5 + Dropout): As shown in Section 4.5, simply changing the learning rate causes the model to overfit, so we added regularization to mitigate that issue.
- **Model Opt-C** (Adam optimizer + Dropout): Adam (9) is one of the most commonly used learning algorithms, and has shown to be robust to different initializations; we decided to try using Adam with some regularization to prevent overfitting.

4.3.3 Representation

- **Model Rep-A** (no linear, +hidden): In this model, we removed the linear layer between the bi-attention and self-attention and the linear layer between self-attention and the supporting facts RNN. Additionally, also increased the number of hidden layers. The motivation for this change was, we did not want to attenuate the output of the self-attention and the bi-attention before passing to the RNN.
- **Model GCNN-A** (GatedCNN replacing GRU): To assist with representation, we also tried using a Gated CNN; CNNs may perform better in some NLP tasks by being able to represent local patterns, and combine them to form position-independent features. Models in which we used a gated CNN are prefixed with **GCNN**. In this model, we replaced each GRU RNN in the model with a Gated CNN, for the reasons mentioned above.
- **Model GCNN-B** (GatedCNN + adam + dropout): We extended GCNN-A using our findings from the Learning and Optimization section, by using the Adam optimization algorithm, as well as dropout for regularization.
- **Model GCNN-C** (+ bi-attention after self-attention): We extended the Gated CNN model using some of what we found by trying the Reasoning models, where adding additional bi-attention layers proved to be beneficial.

4.3.4 Reasoning

- **Model Rsn-A** (based on model Rep-A, with bi-attention before start-token): In addition to Model Rep-A, to improve the reasoning capability of the model, we added a bi-attention layer before the RNN of the start token after the RNN of the supporting fact. For optimization, we used Adam with an initial learning rate of 0.00005, and dropout with `keep_prob = 0.9`.
- **Model Rsn-B** (+2 bi-att with ques_output, Adam, dropout): In the baseline model, the reasoning engine is the bi-attention layer between the ques_output and context_output. To achieve better reasoning, we added another bi-attention layer.

4.3.5 Selected Architectural Diagrams

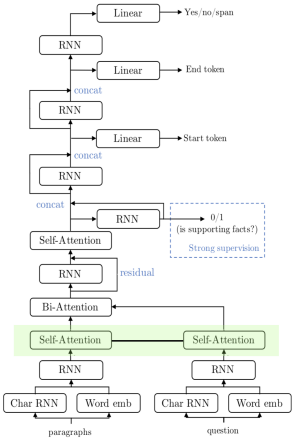

Figure 1: Model Att-C

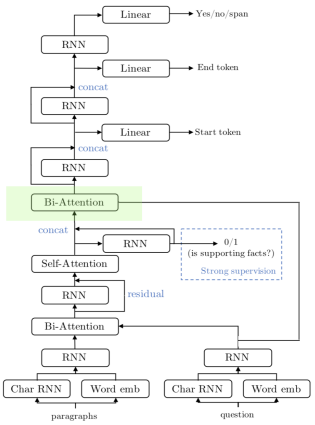

Figure 2: Model Rsn-A

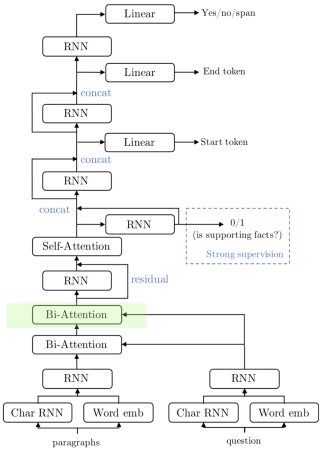

Figure 3: Model Rsn-B

4.4 Results

The following is a table of results of our best-performing or notable models in each category. The full table of results for all models is located in Appendix B.

Table 4: Results from Best-Performing Models

	Model	Ans F1	Ans EM	Sup F1	Sup EM	SP Prec	SP Rec	Joint F1	Joint EM
Base	Baseline	58.28	44.44	66.66	21.95	65.55	70.00	40.86	11.56
Opt-C	adam +dropout	60.25	45.66	66.12	20.42	66.83	71.11	42.31	10.88
Att-C	self-att,q+c	60.03	46.33	69.11	23.52	68.28	76.25	44.09	12.80
Rep-A	no linear, +hidden	64.47	49.95	75.41	33.12	75.22	81.04	50.92	19.00
GCNN-C	+bi-att post-self-att	65.52	49.36	70.78	31.22	73.29	81.94	50.41	20.43
Rsn-A	Rep-A +bi-att start-token	**65.72**	**50.91**	**78.99**	**37.20**	**77.20**	85.27	**53.77**	**21.24**
Rsn-B	+2 bi-att ques_output	64.88	50.37	78.47	35.62	75.87	**85.80**	52.93	19.89

4.5 Analysis of Results

4.5.1 Learning and Optimization

The first category of experiments we tried was purely in learning and optimization, in order to identify if there were any improvements that could be made to the learning algorithm or the learning rates. We visualized the loss curves of some of the modifications we made. Figure 4 shows the baseline model loss curve; as mentioned before, it plateaus quickly and the training loss does not decrease further.

We also trained a model which changed the update rule to divide the learning rate by `1.5` instead of 2, and the loss curve is shown in Figure 5. Although this update rule for the learning rate resulted in better convergence in the training loss, the model overfit and the validation loss increased toward the end of training. This indicated to us that in future experiments, we should consider using Dropout (10), and changing the baseline model's default parameter for `keep_prob` from 1.0 (no dropout) so that some form of regularization is used in training.

We then trained another model with the same learning rate decay rule (`lr /= 1.5`), and Dropout with `keep_prob = 0.9`, and found that the overfitting problem was largely mitigated, as seen in Figure 6.

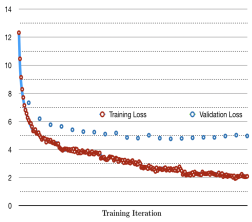

Figure 4: **Baseline** `lr /= 2.0`

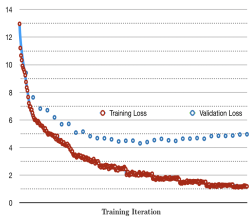

Figure 5: **Opt-A** `lr /= 1.5`

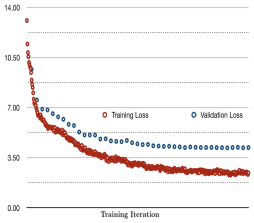

Figure 6: **Opt-B** `lr /= 1.5` + Dropout

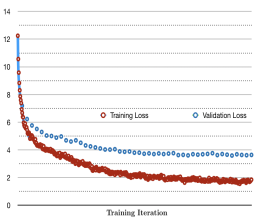

Figure 7: **Opt-C** Adam + Dropout

For our next experiment, we used Adam (9) and kept Dropout with (`keep_prob` = 0.9); the loss curve is shown in Figure 7. The model does not overfit as much as the other learning rate updates (the train-validation gap is less than the baseline), and the model converges on a set of parameters with a lower training and validation loss. This remains an area for further exploration; due to computational resource and time constraints, we were unable to explore this area further, but we believe there still exists areas for improvement in the baseline model regarding learning and optimization.

4.5.2 Architectural Changes

Our goal was to not only increase the F1 score for answers but also increase the precision and recall on the supporting facts. As seen in the table below, we observe significant improvements in the number of correct supporting facts that the model predicts. In the baseline model, we observed that

among all the correct answers, only 108 of 2193 (roughly 5%) had the correct supporting facts. With our best-performing model, that metric is now 19.75%. This increase in the metrics of the supporting facts was also supplemented by the model's performance on correct answers, which increased by 84.5% (see Appendix B).

Table 5: Breakdown of Errors in Models

	Total Dev set	Total correct answers	Total correct ans + correct sp	Incorrect ans+ correct sp	Incorrect ans+ incorrect sp
Baseline	7405	2193 (29.61%)	108 (4.93% of correct ans)	1167 (15.75%)	3224 (43.53%)
Model Rsn-A	7405	3549 (47.92%)	701 (19.75% of correct ans)	2751 (37.15%)	481 (6.49%)

Our new model increased the number of correct answers by 18.31%. Previously, given the correct answers only 5% of them had correct supporting facts. In our current model, this number is up to 20%.

Each of the models listed above have their own strength and outperformed the baseline model on many of the questions. Our motivation to add the self-attention was because we consistently observed that the model was not sufficiently attending to relevant parts of the question. Implementing this model gave us significant gains on the supporting facts.

Our initial goal was to increase the precision and recall on the supporting facts. Adding self-attention, removing the linear layers, helped us drastically increase the supporting facts scores. But the model struggled to give the correct answers especially where reasoning was involved. Since we had a decent coverage on supporting facts, our intuition was to give model another chance to do reasoning. For this, we added a bi-attention layer before the start token RNN. Adding this extra bi-attention would give the model another chance to look at the correlation between the outputs so far and the question. This change, helped us gain approximately 1.5 points in the F1 *Ans* score. In addition to F1 scores, we identified that the Model consistently did better than the baseline model when it came to identifying the correct attribute in the question. It also achieved better reasoning on some of the multi-hop reasoning questions, examples of which are given in Appendix E. Additionally, when we analyzed the supporting facts and noticed that, as compared to the baseline model, our model retrieved more instances of the entity mentions in the paragraph, as shown in in Appendix G.

Our best performing model gives wrong answers to approximately 52% of the questions. Many questions the model gets wrong are complex and require very strong reasoning across multiple entities. One area that the model continues to makes mistake is dates, as shown in Appendix F, under Pattern 2. Given multiple dates in the given paragraph the model cannot often disambiguate between the two.

The table below summarizes a comparison of our best-performing model (**Rsn-A**) with the baseline. The model improves on the baseline EM metric considerably, though there still remain some examples on which the baseline performs better.

Table 6: Comparison of Baseline and Best-Performing Model EM

	Baseline Correct	**Baseline Wrong**
Model Correct	1808	1781
Model Wrong	385	3402

We analyzed our EM scores, examples and realized that many of the answers were close by but were not an exact match, as shown in Appendix F and Appendix H. Below are the F1 score comparison of Model-Rsn A against the Baseline model. Though there are a significant number of examples where our best model performs better than the baseline, there are still some questions in which the baseline achieves a higher F1 than our model.

Table 7: Comparison of Baseline and Best-Performing Model F1

Baseline F1 > Model F1	**Model F1 > Baseline F1**	**Baseline F1 = Model F1**
606	2786	3984

Visualizing Attention

To understand the architectural changes better we decided to visualize the attention activations.

For example, let us look at the query *Who is older, Annie Morton or Terry Richardson?"*, referenced earlier in Section 3.2. The following charts show the attention activations for each word in the context paragraphs. The chart on the left hand side is the self-attention activations on the baseline model and the one on the right hand side is the bi-attention(before the start token) activations on the Rsn-A model. In the baseline model we observe the activations are high for several words, such as *Morton*, *Richardson*, *Aldo*, *Sally*, and *publishing*, among others. However, this attention distribution is not as "focused" as it should be, since the query asks for comparing only *Annie Morton* and *Terry Richardson*. This model is in contrast with Model **Rsn-A** (bi-attention before start-token) that has a high activation for "Terry Richardson" and is able to answer the question correctly.

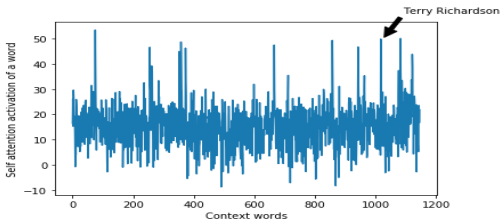

Figure 8: Self-attention Activation, Baseline

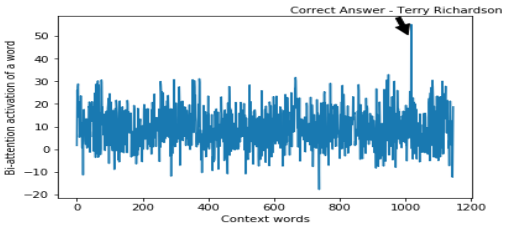

Figure 9: Bi-attention Activation, **Rsn-A**

Additionally, we analyzed the heatmap (shown below) to understand the attention emphasis in the query. In the heatmap for the baseline model, we observe that the baseline model is not attending to *"Terry Richardson"* in the query, and produces *Annie Morton* as the answer. On the other hand, the Model **Rsn-A**, with an extra bi-attention layer, is able to retrieve its focus back to the question, and attends more to the correct answer in the query.

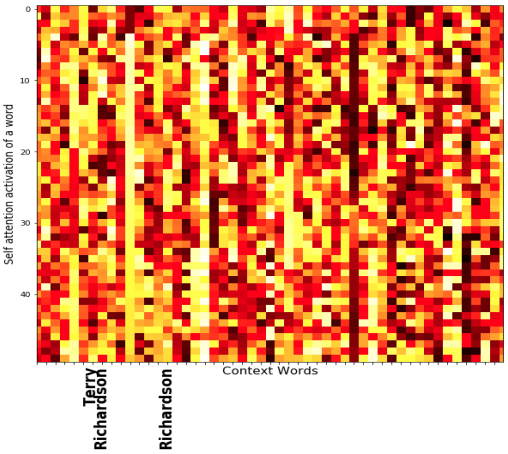

Figure 10: Self-attention Activation, Baseline

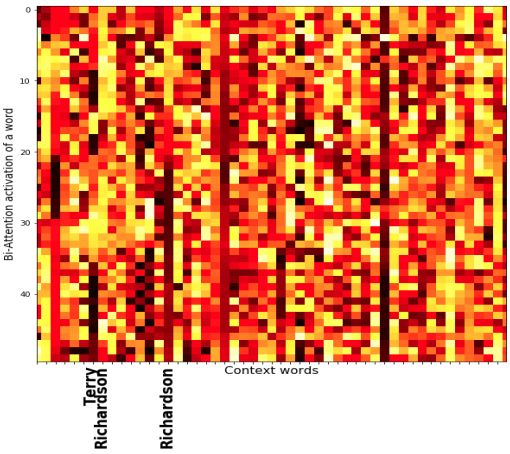

Figure 11: Bi-attention Activation, **Rsn-A**

5 Conclusion and Future Work

In this project, we identified some issues with the baseline model in the HotpotQA paper, and then proposed and implemented some learning and architectural changes to that model. These changes increased the Answer F1 score from 58.28 to 65.98, which is much closer to the unpublished state of the art results on the HotpotQA leaderboard.

Although we explored a variety of improvements to the baseline architecture, we did not explore additional hyperparameter tuning beyond learning and optimization, and we believe that there is still room to do more in this area. In addition, using some concepts from memory networks (8) may help with questions that require multiple hops of reasoning; this is an area in which we did not get time to explore.

6 Acknowledgements

We would like to thank our mentors **Peng Qi** (pengqi@cs.stanford.edu) for his time and valuable insights throughout the project and **Amita A. Kamath** (kamatha@stanford.edu) for her feedback on the writing quality. We would also like to thank the other authors on the HotpotQA paper, including Zhilin Yang and Saizheng Zhang, for providing the baseline model implementation on which we worked (at https://github.com/hotpotqa/hotpot). In addition, we would also like to thank Junki Ohmura (at https://github.com/jojonki/Gated-Convolutional-Networks) from whose PyTorch implementation of GatedCNN, a part of our model is adapted.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 2383–2392. [Online]. Available: http://aclweb.org/anthology/D16-1264

[2] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," 2018.

[3] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," 2017.

[4] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," 2017.

[5] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[7] Y. N. Dauphin, A. Fan, M. Auli, and D. G. 1, "Language modeling with gated convolutional networks," 2018.

[8] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

A Baseline Model Architecture

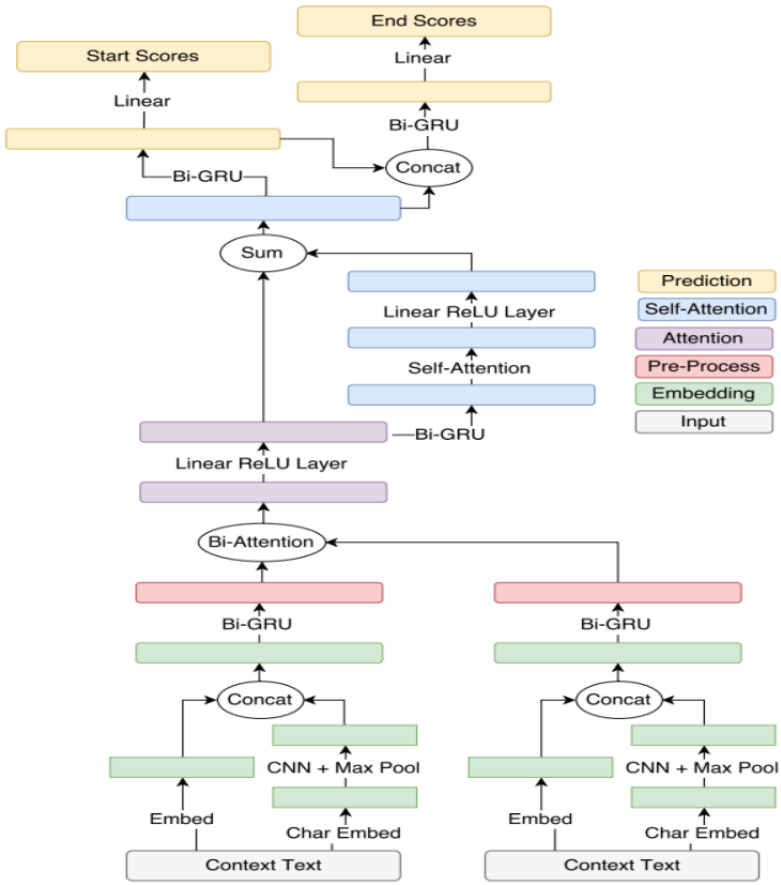

Figure 12: The baseline architecture, from Clark and Gardner (4)

The baseline model is adopted from the Clark and Gardner paper "Simple and Effective Multi-Paragraph Reading Comphrehension" (4) published in ACL 2018. It achieves a 15 point increase in F1 over the prior work on TriviaQA (3), a dataset which includes questions and multi-paragraph documents from which they are retrieved. The only addition to the model described in this paper to the baseline model is the 3-way classifier after the last recurrent layer to produce the probabilities of "yes", "no", and span-based answers.

Word embeddings are generated using pre-trained GloVe word embeddings concatenated with character-based embeddings trained using a convolutional network. A bi-directional GRU is then used to map the document and question embeddings to context-aware embeddings. A BiDAF model is used to build a query-aware context representation by calculating attention scores of a context word with respect to question word, computing an attended vector for each context token. Query-to-context vectors are computed from the previous vectors which are passed through a layer of self-attention followed by another bidirectional GRU. The same attention mechanism is then applied to the passage and the last layer of the model is passed through a prediction layer with a layer of bi-directional GRU and a linear layer. Answer end scores are predicted by passing the hidden states of the earlier layer are concatenated with the input and fed into a bidirectional GRU and linear layer. Softmax is applied to start and end scores to obtain start and end probabilities and optimize the cross-entropy loss.

B Results

	Model	Ans F1	Ans EM	Sup F1	Sup EM	SP Prec	SP Rec	Joint F1	Joint EM
Base	Baseline	58.28	44.44	66.66	21.95	65.55	70.00	40.86	11.56
Opt-A	lr /= 1.5	59.34	45.83	64.88	20.89	65.84	70.75	41.55	11.96
Opt-B	lr /= 1.5 +dropout	58.88	45.26	63.96	20.27	64.07	70.50	40.94	11.13
Opt-C	adam +dropout	60.25	45.66	66.12	20.42	66.83	71.11	42.31	10.88
Att-A	self-att, query only	55.81	42.39	63.98	19.23	65.29	67.88	37.47	10.02
Att-B	self-att, query+context (separate w)	61.98	47.68	64.41	21.50	67.36	68.81	42.03	11.99
Att-C	self-att, query+context (shared w)	60.03	46.33	69.11	23.52	68.28	76.25	44.09	12.80
Rsn-A	bi-att pre start-token, adam, dropout	**65.72**	**50.91**	**78.99**	**37.20**	**77.20**	85.27	**53.77**	**21.24**
Rsn-B	+2 bi-att ques_output, adam, dropout	64.88	50.37	78.47	35.62	75.87	**85.80**	52.93	19.89
Rep-A	no linear, +hidden	64.47	49.95	75.41	33.12	75.22	81.04	50.92	19.00
GCNN-A	GatedCNN	56.31	45.03	65.35	26.69	67.02	69.35	38.84	13.35
GCNN-B	adam +dropout	60.62	48.21	66.92	28.47	69.43	72.78	42.03	14.57
GCNN-C	+bi-att after self-att	65.52	49.36	70.78	31.22	73.29	81.94	50.41	20.43

	Baseline	**Model Rsn-A** no linear, +hidden	**Model Att-C** self-att, query+context (shared)
Total Samples	7405	7405	7405
Correct Ans	2193	3499 (+60%)	3256
Correct *Ans* and Correct *Sup*	108	657 (+508%)	457 (+323%)
Correct *Ans* and *Sup* has extra entities	1167	2404 (+106%)	2172 (+86%)
Incorrect *Ans* and Correct *Sup*	1959	543 (-72%)	627 (-68%)
Incorrect *Ans* and Missing *Sup*	3224	785 (-76%)	734 (-76%)

C Comparing Selected Examples from Baseline and Rep-A Model

Question	Max Hoffmann along with Hindenburg and Ludendorff, masterminded the devastating defeat of the Russian armies in a battle fought when?
Gold Ans	**26-30 August 1914**
Gold Sup	[Max Hoffmann, Battle of Tannenberg]
Baseline Ans	21 March 1918
Baseline Sup	[Max Hoffmann, Max Hoffmann]
Model Ans	**26-30 August 1914**
Model Sup	[Max Hoffmann, Max Hoffmann, Battle of Tannenberg]

Question	Which band is from England, Fireflight or Dirty Pretty Things?
Gold Ans	**Dirty Pretty Things**
Gold Sup	[Fireflight, Dirty Pretty Things (band)]
Baseline Ans	Fireflight
Baseline Sup	[Fireflight, Dirty Pretty Things (band)]
Model Ans	**Dirty Pretty Things**
Model Sup	[Fireflight, Fireflight, Dirty Pretty Things (band)]

Question	Which band is from England, Fireflight or Dirty Pretty Things?
Gold Ans	**Dirty Pretty Things**
Gold Sup	[Fireflight, Dirty Pretty Things (band)]
Baseline Ans	Fireflight
Baseline Sup	[Fireflight, Dirty Pretty Things (band)]
Model Ans	**Dirty Pretty Things**
Model Sup	[Fireflight, Fireflight, Dirty Pretty Things (band)]

D Comparison of Self-Attention Model Architectures

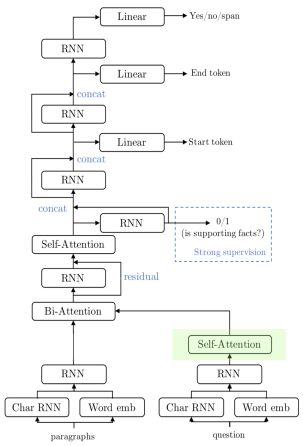

Figure 13: Model Att-A

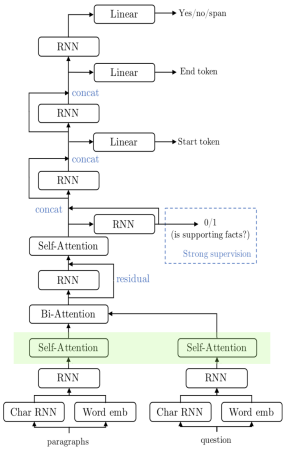

Figure 14: Model Att-B

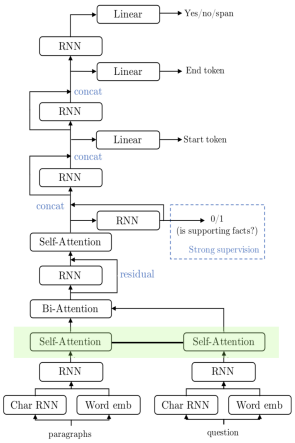

Figure 15: Model Att-C

E Model Rsn-A Correct Answer Analysis

Pattern 1: Multi-hop Reasoning

Question	What WikiLeaks using whistleblower is notable for having a hacking organization with a user base of over 1,800,000?
Gold Ans	Jeremy Hammond
Baseline Ans	Friends
Model Ans	Jeremy Hammond
Hypothesis	Model parses Jeremy Hommond's paragraph and makes note that he is the founder of HackThisSite.org and this site has over 1.8M followers.

Question	What netflix series, produced by Joe Swanberg, had an actress best known for her role as Vanessa on "Atlanta"?
Gold Ans	Easy
Baseline Ans	Arrested Development
Model Ans	Easy
Hypothesis	Model parses Easy as a Netflix TV series produced by Joe Swanberg and then under Zazie Beetz's paragraph it understands that she is an American actress best known for the role of Vanessa on "Atlanta" and she also appeared in the Netflix anthology series "Easy".

Question	Milo Parker starred in which 2014 movie alongside Gillian Anderson?
Gold Ans	Robot Overlords
Baseline Ans	Straightheads
Model Ans	Robot Overlords
Hypothesis	Model parses Milo Parker to understand that he acted in Robot Overlords and other movies. From Robot overlords it understands that it is a British fiction film starring Gillian Anderson.

Pattern 2: Attribute Resolution

Question	What time did the show, in which Gordon Burns was the host, usually air on Monday?
Gold Ans	7pm
Baseline Ans	20 November 1995
Model Ans	7pm
Hypothesis	Model looks for the right attribute, "time" and resolves it

Question	Tao Okamoto made her debut as the lead in the 2013 film featuring the character from what Comic company's line?
Gold Ans	Marvel Comics
Baseline Ans	The Wolverine
Model Ans	Marvel Comics
Hypothesis	the baseline model is not able to infer that the desired attribute is the "company line"

F Model Rsn-A Error Analysis

Pattern 1: Date issues

Question	Which band was formed first, Wavves or Social Code?
Gold Ans	Social Code
Baseline Ans	Social Code
Model Ans	Wavves
Hypothesis	Wavves has two dates. One of which when it was formed (2008) and the birth-date of the lead singer (1986). The model seems to compare 1986; the baseline probably performs better because 2008 appears earlier in the text.

Question	Who was born first, Aleksandr Ivanovsky or Arthur Lubin?
Gold Ans	Aleksandr Ivanovsky
Baseline Ans	Aleksandr Ivanovsky
Model Ans	Arthur Lubin
Hypothesis	There are multiple dates in the given paragraph and the Model seems to get confused on which one to use. Baseline model seems to be looking at usually first appearing date and gives it the most priority. Both the models need to be able handle dates better

Pattern 2: Close Answers

Question	During the Miss USA 2015, Olivia Jordan of Oklahoma was crowned by which American television host, model, taekwondo coach, and beauty queen who won Miss USA 2014?
Gold Ans	Nia Sanchez
Baseline Ans	Nia Sanchez
Model Ans	Nia Temple Sanchez

Question	The movie Chariots of Fire was based on the true story of which Scottish Christian athlete?
Gold Ans	Eric Liddell
Baseline Ans	Eric Liddell
Model Ans	Eric Henry Liddell

Question	The boxer that defeated Oliver Lavigilante in the 2012 Summer Olympics is of what nationality?
Gold Ans	a Ghanaian boxer
Baseline Ans	Beninese
Model Ans	Ghanaian

G Extra Supporting Facts

Model outputs extra supporting facts of the entity occurrences which helps it answer the question better

Question	What food does one of Daniel Greene's subjects' restaurant chain specialize in?
Gold Ans	hamburgers
Baseline Ans	fried chicken
Model Ans	hamburgers
Golden Sup	[Daniel Greene (artist), 4] [Dave Thomas (businessman), 1]
Model Sup	[Daniel Greene (artist), 4] [Dave Thomas (businessman), 0] [Dave Thomas (businessman), 1]

Question	Which 50th Congressional District representative was Brent Roger Wilkes connected to in a scandal?
Gold Ans	Duke Cunningham
Baseline Ans	United States House of Representatives from California
Model Ans	Duke Cunningham
Golden Sup	[Brent R. Wilkes, 1] [Duke Cunningham, 1]
Model Sup	[Brent R. Wilkes, 0] [Brent R. Wilkes, 1] [Duke Cunningham, 0] [Duke Cunningham, 1]

H Gold Answer Issues

Answer Contains Extraneous Words

Question	What location under Charing Cross railway station did G-A-Y move to?
Gold Ans	venue Haven
Baseline Ans	Haven

Question	In what city does a Christian minister who won the singles title with Jan Kodes in the 1970 French Open currently reside?
Gold Ans	Perth, Western Australia
Baseline Ans	Perth

Question	How many times was the writer, who invited Hu Lanqi to meet him in Moscow, a nominee for the Nobel Prize in Literature ?
Gold Ans	a five-time nominee
Baseline Ans	five-time

Question	The boxer that defeated Oliver Lavigilante in the 2012 Summer Olympics is of what nationality?
Gold Ans	a five-time nominee
Baseline Ans	five-time