Question Answering with Hierarchical Attention Fusion Layers & Multi-Task Learning

Tyler Yep
tyep@stanford.edu

Woody Wang
wwang153@stanford.edu

Abstract

Machine reading comprehension combined with unanswerable questions is a novel, challenging task in the field of natural language processing, which motivated the creation of the SQuAD 2.0 dataset. We experiment with implementing a hierarchical attention fusion network to capture complex relationships between the context and question. We also adopt a multi-task learning approach to the problem by adding an answer verifier to the output of our model which predicts whether or not a question is answerable, and further experiment with plausible answer pointers as well. We built the architecture from scratch, replacing components of the generously provided starter code framework of a BiDAF model. After model tuning, we obtain an F1 score of 65.48 and an EM score of 63.06 on the held-out test set, which ranks competitively on the class leaderboard. In our work, we use ablative analysis to show the benefits of each component of our network, as well as qualitative analysis of model predictions.

1 Introduction

Machine reading comprension (MRC) is a difficult task in the field of natural language processing. A model must be able to process and understand a context and question in order to provide a reasonable answer. After the success of the SQuAD challenge, with the most promising systems exceeding human level performance, Rajpurkar et al. [4] developed the SQuAD 2.0 dataset, which incorporates unanswerable questions in order to make the task of question answering more difficult.

In order to be successful in question answering with unanswerable questions, a model must be able to distinguish whether or not a question is answerable while attempting to locate a plausible answer span. Thus, we decompose the problem into the two main tasks of an answer pointer and an answer verifier, which we refer to as a multi-task learning problem, inspired by Sun et. al [7].

We also experiment with hierarchical attention fusion networks using techniques proposed by Wang et al [8]. The hierarchical attention layers leverage the benefits of both co-attention and self-attention in order to gradually focus the attention on the optimal answer span with refined granularity.

On a high level, we can define our three main contributions as follows: (1) We use a mixture of word, character, and manual features as input to improve performance. (2) We implement hierarchical attention fusion layers from scratch to capture the complex relationship between the question and context. (3) We adapt a modified answer verifier from scratch to formulate a multi-task learning problem where we attempt to predict if a question is answerable whilst locating the answer span.

We show that with the aforementioned modifications, we achieve competitive metrics on the class leaderboard based on F1 and EM scores, and we critically analyze our model's performance through quantitative and qualitative analysis.

2 Related Work

Most of the top models on the current SQuAD 2.0 leaderboard emphasize using unique attention mechanisms to improve performance on question answering. Our provided baseline contains an implementation of the Bi-Directional Attention Flow (BiDAF) [6], which produces a question-aware context and a context-aware question from its attention layer, which has shown to be successful on the SQuAD leaderboard.

Another promising approach to attention is the multi-granularity hierarchical attention fusion networks proposed by Wang et al. [8]. The authors propose a hierarchical attention network to focus on the answer span progressively with multi-level soft-alignment. On a high level, the attention approach draws inspiration from typical human reading patterns. The authors claim that people first read through the whole passage to capture a snapshot of the main body of the passage. Then, while considering the question, people try to understand the main intent of the question related with the passage's theme. Next, a rough answer span is located in the passage and attention becomes focused on the local context of that span. Finally, people will remind themselves of the question and select the best answer according to the previously located answer span.

The high-level intuition seemed reasonable, and the proposed methods of using a hierarchical attention network that can gradually focus the attention on the desired answer span seemed very promising, so we chose to implement a version of the hierarchical attention network from scratch for our final model. Although the original authors of the hierarchical attention fusion networks use pretrained contextual embeddings in their final model, we show that with some modifications, the hierarchical attention network performs well, even without pretrained contextual embeddings.

Beyond focusing on attention, we analyzed work done specifically to tackle the challenge of unanswerable questions. Sun. et al.'s U-Net [7] looked particularly promising, as the authors propose to reframe the machine reading comprehension problem as a multi-task learning problem with an answer pointer, plausible answer pointer, and answer verifier. Since the hierarchical attention network used a simpler output layer than BiDAF, we reasoned that perhaps using a more expressive layer would yield better results. We found the answer verifier to be the most beneficial, both empirically and intuitively, as it seeks to predict whether or not a question is answerable, which influences the optimization of the overall model.

3 Approach

3.1 Main Approach and Architecture

In Figure 1, we display our final model architecture, which can be summarized as three main components. First, we have an embedding layer, which is detailed in Section 3.2. Second, we have our hierarchical attention layer, described in Section 3.3. Finally, we have our output layer, which includes an answer verifier and is described in Section 3.4.

We can formally define our problem as follows: given a set of tuples (Q, P, A), where $Q = (q_1, q_2, ...q_m)$ denotes a question of length m words, $P = (p_0, p_1, p_2, ...p_n)$ denotes a context passage with length n words, and $A = (r_s, r_e)$ denotes the true start and end indices of the answer in the passage, our model must learn a function $f : (Q, P) \rightarrow (\hat{r}_s, \hat{r}_e)$, where $(\hat{r}_s, \hat{r}_e)$ represents the prediction of the start and end index of the answer span (note that we append a p_0 token at the beginning of each passage to represent a no-answer token). The constraints on our answer span are that $0 \leq \hat{r}_s \leq \hat{r}_e$, and $\hat{r}_e - \hat{r}_s + 1 \leq L_{max}$, where $L_{max} = 15$ represents the maximum length of an answer span prediction.

3.2 Embedding Layer

The embedding layer takes input representations of a question Q and context passage P and produces embedded representations of the question $\tilde{Q}_e = (\tilde{q}_1, \tilde{q}_2, ...\tilde{q}_m)$ and context passage $\tilde{P}_e = (\tilde{p}_0, \tilde{p}_1, \tilde{p}_2, ...\tilde{p}_n)$. For each word in the question and context, we look up a word embedding $w_i \in \mathbb{R}^{D_w}$ and character-level embedding $c_i \in \mathbb{R}^{len(word) \times D_c}$, where D_w and D_c are the word and character embedding sizes, respectively. We then feed each c_i through a 1D convolution to produce $\tilde{c}_i \in \mathbb{R}^{D_w}$. We then concatenate the ExactMatch features denoted as f_{EM} described in

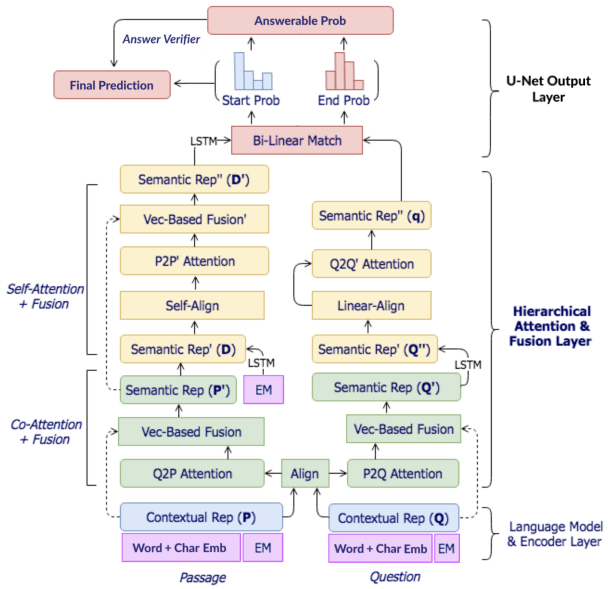

Figure 1: Final model architecture (attention layers borrowed from Wang et al. [8])

Section 3.2.1 with the character and word embeddings to form $e = [w_i, \tilde{c}_i, f_{EM}]$. Finally, we pass e through a projection layer and a highway network [5] to obtain our final embeddings $\tilde{e}_i \in \mathbb{R}^H$, where H is a hyperparameter representing the hidden size.

3.2.1 ExactMatch Features

We define the ExactMatch features as three binary indicator features proposed by Chen et al. [1]. Let each question and context be denoted as q and c, respectively. In particular, for each word in the question q_i, we define a binary feature $\mathbb{1}(q_i \in c)$. Likewise, for each word in the context c_i, we define a binary feature $\mathbb{1}(c_i \in q)$. We add a binary feature for each of the words in the original, lowercase, and lemmatized form. While Chen et al. used this for context encodings, we use these features for both the question and context embeddings to emphasize the importance of exact matches of words from question to context as well as context to question.

At the start of our self-attention context layer, we merge these features into our model a second time as a way of guiding the attention to focus on these features. Similar to the concept of spaced repetition for human learning, we found that having the network revisit these important features again later in the network improves our performance on the EM/F1 metrics.

3.3 Attention Layers

We will now summarize the hierarchical attention and fusion layer that Wang et al. propose. Using $\tilde{Q}_e$ and $\tilde{P}_e$ from the embedding layer, we can construct a soft-alignment matrix S to measure the semantic similarity between question and passage as:

$$S_{ij} = Att(\tilde{q}_i, \tilde{p}_i) = ReLU(W_{lin}^T \tilde{q}_i)^T \cdot ReLU(W_{lin}^T \tilde{p}_i), \tag{1}$$

where W_{lin} is a trainable weight matrix. We then use S_{ij} to compute attention between question and passage, which is used to compute attended vectors in the passage to question (P2Q) and question to passage (Q2P) direction, respectively. The P2Q direction can be summarized by computing $\alpha_j = softmax(S_{:j})$. We then compute an aligned passage representation from the question as:

$$\tilde{Q}_{:t} = \sum_j \alpha_{tj} \cdot \tilde{Q}_{e:j} \forall j \in [1, ...m]. \tag{2}$$

The Q2P direction equations are similar and result in obtaining $\tilde{P}_{k:}$, where $\tilde{P}$ is the weighted sum of the most important words in the passage with respect to the question. With $\tilde{P}$ and $\tilde{Q}$, the authors propose a vector-based fusion kernel to compute $P' = Fuse(P, \tilde{Q})$ and $Q' = Fuse(Q, \tilde{P})$.

The next step is to use self-attention in order to further align the question and passage representation against itself. Using P', let $D = BiLSTM([P'; f_{EM}])$, where f_{EM} are the ExactMatch features mentioned earlier that we use to guide the attention. We then use a bilinear self-alignment function summarized as:

$$\tilde{D} = softmax(D \cdot W_L \cdot D^T) \cdot D \tag{3}$$

We then have a question-aware passage representation D and a self-aware representation $\tilde{D}$ that are fused to form D'. Finally, we run D' through a bidirectional LSTM to form D'', the final contextual passage representation.

Since the question representation is shorter in length, the authors propose using a linear transformation to encode the question representation. First, we compute $Q'' = BiLSTM(Q')$. The final single question vector can then be computed with a linear self-alignment as:

$$\gamma = softmax(w_q^T \cdot Q'') , \tag{4}$$

$$q = \sum_j \gamma \cdot Q''_{:j}, \forall j \in [1, ..., m]. \tag{5}$$

3.4 Output Layers

Finally, we implemented the U-Net model's answer pointer, plausible answer pointer, and answer verifier proposed by Sun et al. [7] to better handle the challenge of unanswerable questions without adding too much overhead to the model. We modified the bilinear match layer in hierarchical attention to use additional layers for the pointers and verifier. However, we found that our model achieved the best performance with simply the answer pointer and answer verifier, which we describe below.

Given our summarized question vector q and final contextual passage representation D'', we compute a vector of probabilities corresponding to each position in the context:

$$p_{start} = softmax(q \cdot W_s^T \cdot D'') \tag{6}$$

$$p_{end} = softmax(q \cdot W_e^T \cdot D'') \tag{7}$$

At test time, we choose the values of $\hat{r}_s$ and $\hat{r}_e$ that maximize $p_{start}(\hat{r}_s) \cdot p_{end}(\hat{r}_e)$.

Additionally, we compute the probability the question is answerable using the U-Net answer verifier. We first create a vector F containing the question representation, the no-answer token's attended context representation, and the two answer spans, each combined with the passage representation. We then pass F through a sigmoid and a linear layer to obtain a probability of the question's answerability.

$$F = [q, D''_0, c_s, c_e], \qquad c_s = p_{start} \cdot D'', \qquad c_e = p_{end} \cdot D'' \tag{8}$$

$$p_{answerable} = \sigma(W_f^T F) \tag{9}$$

Having a separate predictor for question answerability improves our model's performance significantly, as we are effectively employing multi-task learning by semi-supervising the model's answer span prediction with its ability to predict whether or not a question is answerable. With our answer pointer, we compute a negative log likelihood loss component:

$$\mathcal{L}_A = -\big(\log(p_{start}(i)) + \log(p_{end}(j))\big), \tag{10}$$

where i and j are the gold start and end locations in the context passage, respectively. We then compute a separate loss function for our answer verifier:

$$\mathcal{L}_{AV} = -\big(\delta \cdot \log(p_{answerable}) + (1 - \delta) \cdot \log(1 - p_{answerable})\big), \tag{11}$$

where $\delta \in \{0, 1\}$ depending on if the question is answerable. Summing the two equations, we obtain the final loss:

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_{AV} \tag{12}$$

4 Experiments

4.1 Dataset

We use the provided SQuAD 2.0 dataset for all experiments. We incorporate 300-dim GloVe embeddings, character embeddings, and three additional binary ExactMatch features described in Section 3.2. The ExactMatch features are computed during the data preprocessing phase, so they incur a minimal cost on the overall runtime of training the model.

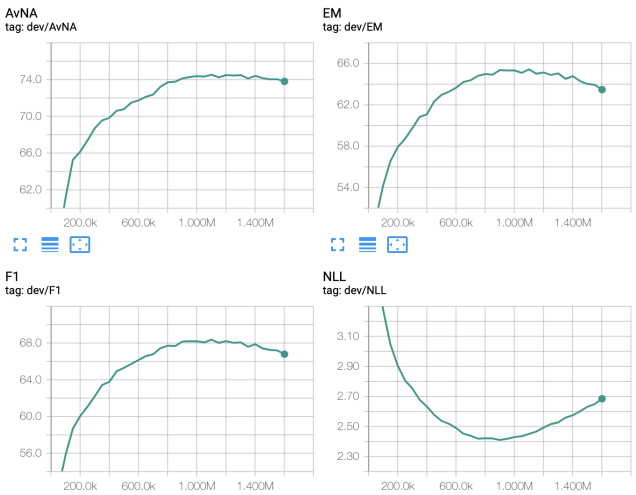

Figure 2: Final Model Architecture Development Curves

4.2 Evaluation Methods

We evaluated our methods using the SQuAD 2.0 leaderboard and EM/F1 scores. We focused on comparisons between incremental modifications to our model, beginning with the baseline BiDAF model given in the starter code. We also referenced the class leaderboard for inspiration of models to consider. To see how important each component of our model architecture was, we conducted an ablation study on our final model.

For qualitative analysis, we filtered and manually inspected output to understand the type of questions our model answered incorrectly. We discuss qualitative results further in Section 6.2.

4.3 Experimental Details

We began by building off of the provided implementation of a BiDAF [6] model. For the embedding layer, we modified the layer to take a concatenation of word embeddings, character embeddings, and ExactMatch manual features. Then, we implemented the hierarchical attention network method for creating the passage to question (P2Q) and question to passage (Q2P) representations explained by Wang et al [8]. We replaced the BiDAF model's attention layers with our new attention network.

Finally, we substituted the BiDAF output layer with the bilinear match layer given by Wang et al. combined with the answer verifier in the U-Net output layer. We experimented with different answer pointers, and we modified our loss function to be the objective described in Section 3.4.

Note that since we do not have access to an official implementation of the hierarchical attention network, and we are not using pretrained contextual embeddings like ELMo or BERT, we are not using the provided EM/F1 score by Wang et al. as a baseline benchmark.

We ran each model for roughly 2 million iterations using a dropout of 0.23 for each as a starting point, which we found through empirical testing. We stopped some models early when it was clear that they were overfitting the training data. We experimented with different optimizers (AdaMax, AdaDelta, Adam), different learning rates for each optimizer, and different dropout probabilities (0.1, 0.3, 0.5), and we found training stability to be significantly impacted by the choice of hyperparameters. For example, with too small of a learning rate, in some cases the model would become stuck in a local optima and always predict no-answer.

5 Results (Non-PCE)

After performing a hyperparameter grid search, we settled on the following final list of tuned hyperparameters: batch size = 128, optimizer = Adam, initial learning rate = 0.001, hidden size = 100, dropout = 0.23, learning rate decay = 0.99.

Model Metrics on Dev Set	EM	F1	AvNA	NLL
Baseline BiDAF	55.99	59.29	67.90	3.08
Hierarchical	56.11	58.92	66.14	2.98
BiDAF+EM+Char	60.85	64.36	70.49	2.76
Hierarchical+EM+Char	63.99	67.19	73.37	2.55
Hierarch+EM+Char+UNet	64.78	67.27	71.47	9.27
Final Model	**65.43**	**68.37**	**74.53**	**2.45**

Figure 3: EM, F1, AvNA, and NLL scores of different models tested on the development set

Our final model consists of word + character embeddings, ExactMatch features, hierarchical attention, and the U-Net answer verifier. With our final model, we obtain our best results on the development set as seen in Figure 3. We list only the models with major architectural changes for clarity.

After experimenting with an initial implementation of the hierarchical attention network, we found that substituting hierarchical attention for BiDAF attention led to a minor improvement in EM and a minor reduction in F1 compared to the given official baseline metrics. We suspect that this discrepancy in performance could be because hierarchical attention networks rely on pre-trained embeddings more than BiDAF does, as the original authors use ELMo in their final model. This is corroborated by our observation that after adding ExactMatch features and character embeddings to both the hierarchical attention network and BiDAF models, our hierarchical attention network obtained far superior performance on the EM and F1 metrics. After modifying the training objective to include the binary cross entropy loss of the answer verifier's predictions, we noticed a further improvement of our final model, supporting the idea that the hierarchical attention network relies on more expressive input and output layers to perform well.

Surprisingly, when we implemented the full U-Net output layer (answer pointer, plausible answer pointer, and answer verifier), we obtained slightly lower metrics than when we simply use the answer verifier (note that the large difference in NLL is due to a modified objective function). We suspect that the plausible answer pointer did worse because we effectively rewarded the model for predicting a plausible answer, even when it should predict no-answer. Furthermore, since the full U-Net objective function involved splitting the loss into three components, the additional complexity seemed to make training slower with diminishing returns on performance. With just the answer verifier, we found that the model converged more quickly and performed better on discerning a question's answerability, as desired and as seen in the improved metrics.

Our final test leaderboard submission obtained scores of 63.06 EM and 65.48 F1. We can see that we are to some extent overfitting to the development set, which makes sense as we tracked the F1 and EM scores throughout training to inform our hyperparameter tuning. Also, the held-out test set contains some hand-labeled examples, which could make the distribution of the test set data slightly different than the distribution of the train and development data, which might partially explain the difference in development and test F1 and EM scores.

6 Analysis

6.1 Ablation Analysis

Model Metrics on Dev Set	EM	F1
Final Model	**65.43**	**68.37**
- Hierarchical Attention (BiDAF instead)	62.84	65.28
- ExactMatch features	63.16	66.73
- Character Embeddings	61.09	64.55
- U-Net Answer Verifier	63.99	67.19

Figure 4: Ablation tests of our final model on the development set

In order to measure the individual contribution of each model component, we ran an ablation study. Figure 4 summarizes the performance of our model with individual components removed on the development set. We can see that the character embeddings are the most crucial to the performance

on both EM and F1, which makes sense as character embeddings aid in providing representations for OOV words, whereas word embeddings would simply use a fixed <UNK> token representation. The next most important component is the hierarchical attention network layer. When replaced with BiDAF attention, we observe a significant drop of about 5% on the metrics. We hypothesize that the hierarchical attention layer is more effective with more expressive inputs, as we have seen empirically in our experiments. To evaluate the effectiveness of the ExactMatch features, we see that by removing the manual features, EM decreases by about 3.5% and the F1 decreases by about 2.5%. We notice that the drop in the EM score is more significant than the F1 decrease. We posit that since the ExactMatch features mark words that match in the question and the context passage, the ExactMatch features influence the EM score more than the F1 score. Finally, we see that removing U-Net answer verifier degraded performance least.

6.2 Qualitative Error Analysis

During development, we assessed our model's predictions qualitatively to see where improvements could be made. In Sections 6.2.1 and 6.2.2, we show two examples of issues that were alleviated by architectural changes in our final model. In Sections 6.2.3 and 6.2.4, we highlight two examples our model struggles with, as well as possible ways to remedy the problems.

6.2.1 OOV words in the question and context

- **Question:** What is polish for "mermaid"?
- **Context:** The mermaid (syrenka) is Warsaw's symbol and can be found on statues throughout the city and on the city's coat of arms. This imagery has been in use since at least the mid-14th century. The oldest existing armed seal of Warsaw is from the year 1390, consisting of a round seal bordered with the Latin inscription Sigilium Civitatis Varsoviensis (Seal of the city of Warsaw). City records as far back as 1609 document the use of a crude form of a sea monster with a female upper body and holding a sword in its claws. In 1653 the poet Zygmunt Laukowski asks the question:
- **Answer:** syrenka
- **Prediction:** syrenka

Although our model initially struggled with predicting OOV words, our final model successfully identified many solutions that include out-of-vocabulary words. In the above example, the model is able to identify the relationship between "syrenka" and "mermaid", even though "syrenka" is not in the trained vocabulary. The character-level embeddings likely helped solve this problem, as the model can potentially infer that the parentheses around "syrenka" signaled a definition or alternate meaning of the word. Moreover, the character-level embeddings provide OOV words with a unique representation instead of grouping them all as a single <UNK> representation.

6.2.2 Exact Match

- **Question:** What is the name of Harvard's primary recreational sports facility?
- **Context:** Harvard has several athletic facilities, such as the Lavietes Pavilion, a multi-purpose arena and home to the Harvard basketball teams. The Malkin Athletic Center, known as the "MAC", serves both as the university's primary recreation facility and as a satellite location for several varsity sports. The five-story building includes two cardio rooms, an Olympic-size swimming pool, a smaller pool for aquaerobics and other activities, a mezzanine, where all types of classes are held, an indoor cycling studio, three weight rooms, and a three-court gym floor to play basketball. The MAC offers personal trainers and specialty classes. It is home to Harvard volleyball, fencing and wrestling. The offices of several of the school's varsity coaches are also in the MAC.
- **Answer:** Malkin Athletic Center
- **Prediction:** Malkin Athletic Center

Adding ExactMatch features helped our model greatly, particularly by lemmatizing all question words and searching for them in the context. In the above example, since our model gives more weight to lemmatized context words that are also present in the question, the sentence containing the answer was given significantly more attention, since "primary", "recreation", "sports", and "facility" were in the question text. Notice that the lemmatization helps with marking "recreational" and "recreation" as matches in the ExactMatch features. Compared to the class leaderboard, we have a relatively higher EM than F1 score, which we can potentially attribute to our ExactMatch features.

6.2.3 Improper recognition of question type

- **Question:** Interest groups and government agencies that were concerned with energy were no match for who?
- **Context:** In the United States, scholars argue that there already existed a negotiated settlement based on equality between both parties prior to 1973. The possibility that the Middle East could become another superpower confrontation with the USSR was of more concern to the US than oil. Further, interest groups and government agencies more worried about energy were no match for Kissinger's dominance. In the US production, distribution and price disruptions "have been held responsible for recessions, periods of excessive inflation, reduced productivity, and lower economic growth."
- **Answer:** Kissinger
- **Prediction:** Kissinger's dominance

Our model occasionally struggled to identify the asked question type. In the above example, the question clearly asks for the name of a person (Kissinger), signaled by "for who?" at the end of the sentence, but our model predicted "Kissinger's dominance" instead. However, it is important to note that our model only predicts answer spans in the passage, and in this case, the standalone name "Kissinger" is not present in the context. We would likely need train a separate model to match our final output with the correct question type in order to handle these questions. Moreover, additional named-entity recognition features may help in determining that the answer is intended to be a noun.

6.2.4 Predicting plausible answer instead of no-answer

- **Question:** NSF was engineered and operated by who?
- **Context:** The Very high-speed Backbone Network Service (vBNS) came on line in April 1995 as part of a National Science Foundation (NSF) sponsored project to provide high-speed interconnection between NSF-sponsored supercomputing centers and select access points in the United States. The network was engineered and operated by MCI Telecommunications under a cooperative agreement with the NSF. By 1998, the vBNS had grown to connect more than 100 universities and research and engineering institutions via 12 national points of presence with DS-3 (45 Mbit/s), OC-3c (155 Mbit/s), and OC-12c (622 Mbit/s) links on an all OC-12c backbone, a substantial engineering feat for that time. The vBNS installed one of the first ever production OC-48c (2.5 Gbit/s) IP links in February 1999 and went on to upgrade the entire backbone to OC-48c.
- **Answer:** N/A
- **Prediction:** MCI Telecommunications

Our model also occasionally predicted a plausible answer instead of no-answer. In the above example, the answer to the question is not present in the text, however our model predicts the plausible answer. This shows that our model is susceptible to pattern matching, instead of actual comprehension. We attempted to solve this problem by adding the U-Net plausible answer pointer, however we found that adding the plausible answers ultimately degraded overall model performance. To mitigate this issue, we should explore different methods of encoding confidence of the answerability of a question in order to predict no-answer properly beyond our attempts of using U-Net's modified objective.

7 Conclusion & Future Work

In this paper, we implement and modify the hierarchical attention network proposed by Wang et al [8] as well as the U-Net answer verifier proposed by Sun et al [7]. We found that our model's performance is highly dependent on having sufficiently expressive input embeddings, as seen by our model's performance improvement after adding ExactMatch features and character-level embeddings. Our final model achieves competitive EM and F1 scores relative to the class leaderboard, but there is still significant room for improvement with respect to the state of the art. Through ablation analysis, we see that our model relies most on character-level embeddings, likely because they help with handling OOV words in the context and question. Through qualitative inspection of predictions, we see that our model is prone to choosing "plausible" answers when there is truly no answer. We attempted to remedy this by using an answer verifier in the output layer, which improved performance. We also experimented with a plausible answer pointer, which did not improve performance.

For future work, we have seen that expressive and informative inputs contribute a significant impact on the model's performance. Thus, an immediate next step could be to try integrating state of the art pretrained contextual embeddings such as BERT [2] to our model's embedding layer. It would also be interesting to examining different ways of regularizing the model to help with generalization, perhaps with techniques such as zoneout [3]. Future work could also experiment with different attempts to divide the problem definition into more sub-tasks, in order to not only focus on predicting a reasonable answer span, but also to be confident in the answerability of a question.

References

[1] Danqi Chen and Adam Fisch et al. Reading wikipedia to answer open-domain questions. *Association for Computational Linguistics*, 2017.

[2] Jacob Devlin and Ming-Wei Chang et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018.

[3] David Krueger and Tegan Maharaj et al. Zoneout: Regularizing rnns by randomly preserving hidden activations. *Neural and Evolutionary Computing*, 2017.

[4] Robin Jia Pranav Rajpurkar and Percy Liang. Know what you don't know: Unanswerable questions for squad. *Association for Computational Linguistics*, 2018.

[5] Klaus Greff Rupesh Kumar Srivastava and Jürgen Schmidhuber. Highway networks. *arXiv*, 2015.

[6] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. Bi-directional attention flow for machine comprehension. 2018.

[7] Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. U-net: Machine reading comprehension with unanswerable questions. 2018.

[8] Wei Wang, Ming Yan, and Chen W. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, page 1705–1714, 2018.