The Unanswerable Gap: An Exploration of Approaches for Question Answering on SQuAD 2.0

Stanford CS224N Default Project (IID SQuAD track)

Ruth-Ann Armstrong
Department of Computer Science
Stanford University
ruthanna@stanford.edu

Eghosa Amadin
Department of Computer Science
Stanford University
eamadin@stanford.edu

Abstract

In this project, we implemented models that were trained and evaluated using the Stanford Question Answering Dataset (SQuAD) 2.0. For a majority of our models, we incorporated character-level embeddings to strengthen the system's understanding of the semantics and syntax of each context and question. Our implementations were based on two main architectures: the baseline Bidirectional Attention Flow (BiDAF) model and the Dynamic Coattention Network (DCN), which we implemented in full. We found that the baseline BiDAF model with character-level embeddings performed the best and received an EM/F1 score of **61.771/65.089** on the test set.

1 Key Information to include

- Mentor: Zihan Wang, Sharing project: N/A

2 Introduction

Question Answering (QA) is a benchmark for measuring computers' understanding of human language. The primary goal of this task is to allow researchers to construct systems and models that are able to understand human language on both a semantic and syntactic level. Such deep knowledge of language is difficult for computers to achieve. This is why QA remains a useful way to evaluate the strength of an NLP model. Historically, a bottleneck for robust QA models was the lack of structured data since the available datasets were annotated by humans. With the rise of deep learning, the need for a large, high quality dataset to evaluate QA tasks on grew.

In 2016, the Stanford NLP Group released the Stanford Question Answering Dataset (SQuAD) which consists of over 100,000 question-answer pairs. Two years later, SQuAD 2.0 – which includes 50,000 unanswerable questions in addition to the pairs from the first version – was released [1]. These datasets enable the standardized training and evaluation of large deep learning models on QA.

In our implementation, we improved upon the performance of the Bidirectional Attention Flow (BiDAF) model on the SQuAD 2.0 dataset primarily by incorporating character embeddings into the model's embedding layer [2]. This was our most successful implementation and yielded an F1 of 65.089 and an EM of 61.771.

Additionally, we implemented two versions of a Dynamic Coattention Network (DCN) proposed by Xiong et al.: one with a highway maxout (HMN) layer and one with a multi-layer perceptron (MLP) [3]. These models, which were designed for SQuAD 1.0, did not yield optimal results on SQuAD 2.0 even when we attempted to boost their performance by adding character level embeddings, because of their inability to properly distinguish between answerable and unanswerable questions. We also experimented with composing different elements from the BiDAF and DCN models, and with varying

our hyperparameters to achieve optimal results. Among our experiments, we found that the BiDAF + character-level embeddings implementation created the most robust model that both distinguished between answerable and unanswerable questions and predicted answers spans well.

3 Related Work

Xiong et al. introduced the Dynamic Coattention Network (DCN) in 2016. The most innovative contribution in the paper was the iterative decoder. A model that only makes a single pass through the estimations when predicting answer spans could incorrectly choose one of said plausible spans. In contrast, the DCN uses a dynamic pointer decoder to iterate over various potential answer spans. By iterating over a number of potential answer spans, this model makes getting stuck in local maxima that correspond to incorrect answer spans less likely, thus increasing the accuracy of the model. While the iterative decoder was an important contribution, the coattention mechanism described in the DCN paper is relatively weak because it is simply matrix multiplication and does not incorporate any nonlinearities. DCN was designed for SQuAD 1.0 and achieved an EM/F1 of 66.2/75.9 on that dataset.

BiDAF, the baseline model used in this project, implements a rich attention mechanism by incorporating a learnable weight vector into its matrix multiplication scheme that is applied to both the context and query hidden states. It computes both query-to-context and context-to-query attention which are fed into bidirectional LSTMs to yield probabilities for how likely it is that a given word is the start/end word of the answer span. The original BiDAF model was also designed for SQuAD 1.0 and achieved an EM/F1 of 68.0/77.3.

Our goal in conducting this project was to explore effective ways to apply and adapt these models for QA on the updated SQuAD 2.0 dataset.

4 Approaches

For this project, we explored multiple approaches to improve on the scores achieved by the baseline. The approach that produced the best result consisted of adding character-level embeddings to the baseline model. The main papers that our exploration was based on were BiDAF [2] and DCN [3].

4.1 Baseline Model

The baseline model is a modified implementation of (BiDAF) described in [2] with character-level embeddings removed. The various approaches described below use code from the baseline model (https://github.com/minggg/squad.git) as a framework within which we added the layers relevant to the different implementations.

4.2 Character-Level Embeddings

The most successful of our approaches involved augmenting BiDAF with character level embeddings, using the model described in the original BiDAF paper (see Appendix, Fig. 1 for a diagram of the full model from the original paper) [2]. Here, embeddings are produced by loading pretrained character-level vector representations, then passing them through a 1D Convolutional Neural Network (CNN). These are then appended to the word embeddings, and the concatenated result is then passed to the encoder layer.

Since the original paper did not specify the CNN configuration, we ran sub-experiments to obtain the optimum size of the convolving kernel, and the optimum setting for the number of output channels (or the dimensionality of the character-level embeddings). We did this by training the model with different settings of these parameters for an epoch then selecting the setting which led to the steepest decrease in training loss. In our model, we use a kernel size of 3 and 140 output channels.

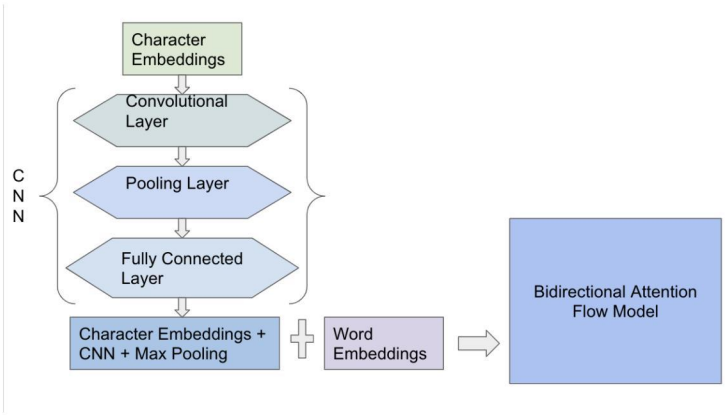

Fig. 1. Character Embedding Layer Diagram.

4.3 Character Level Embeddings + Coattention

The second model that we trained involved adding the coattention mechanism described in the DCN paper to the baseline BiDAF implementation enhanced with character-level embeddings described above.

In the baseline, the encoder produces context hidden states $c_1, ..., c_N \in \mathbb{R}^l$ and question hidden states $q_1, ..., q_M \in \mathbb{R}^l$.

Our modification, based on the parts of the encoder from the DCN paper, works as follows. Both a linear layer and tanh nonlinearity are applied to the question hidden states by the equation q'_j =tanh$(Wq_j + b) \in \mathbb{R}^l$. This is done to introduce variation between the context encoding space and the document encoding space. Trainable sentinel vectors initialized to zero are then appended to the q' and d hidden states to allow coattention layer to not attend to any particular word in the context. These states are then passed to the coattention layer from the DCN paper, the details of which are described in Section 4.5.

4.4 Coattention + BiDAF Output

A third approach that we explored – our earliest attempt at improving on the baseline – was composing the baseline BiDAF implementation (without character-level embeddings) with the coattention layer described in the DCN paper [3]. The architecture of DCN approach is described in further detail in the following section. Specifically, we combined the embedding, encoder, and coattention layers described in the DCN paper with the modeling and output layers in the BiDAF paper.

4.5 Dynamic Coattention + Character Level Embeddings

The fourth model that we implemented was the full Dynamic Coattention Network [3]. Our best results using this model came from concatenating character level embeddings to the word level embeddings of documents and questions.

DCN has four main layers: embedding, encoder, coattention, and decoder. The coattention layer consists of a series of matrix multiplication between different concatenations and products of the document encodings, D, and the question encodings, Q. These values are then put into a bidirectional LSTM to compute U – new updates for the model's representations of each document word. U is later used to select new estimates of the start and end words of the answer span. An iteration of the dynamic pointing decoder is as follows: The DCN model chooses its prediction for the start and end index of the answer span by using an LSTM and a Highway Maxout Network (HMN) with 3 separate maxout layers and a tanh layer to compute scores for each word in the document as the start index and as the end index, providing a measure of how strong of an estimate it is. Our next predictions for the start/end indices are the argmax values across the scores for each word in the document [4] [5]. We repeat this "predict, recompute, update" mechanism until either our estimates converge or we carry out 4 iterations. To train the model, we use the average of the cumulative cross entropy loss

across all of the iterations of our dynamic decoder. See Figures 2 and 3 in the Appendix for DCN architecture diagrams.

Since the original paper was written prior to the introduction of SQuAD 2.0, there is no specification for the handling of questions with no answer. We handle these in a similar manner to the baseline: we prepend an out of vocabulary token to the beginning of each context. When the DCN decoder selects this token as either the start or ending position for an answer, we predict no answer.

4.6 Coattention + Multi-layer Perceptron

Our final model is identical to the DCN model described above, except for the final layer. Here, the HMN in the decoder layer is replaced with a multi-layer perceptron (MLP). This version was an ablation implemented by Xiong et al. in the DCN paper. However, they used a 2-layer MLP while we implemented a 3-layer MLP.

5 Experiments

5.1 Data

We used the updated version of the Stanford Question Answering Dataset, SQuAD 2.0 with unanswerable questions [1]. The task that our model aims to achieve with this dataset is question answering, and the identification of unanswerable questions. Given a context and question, the model outputs an answer or classifies the question as having no answer.

5.2 Evaluation method

Our evaluation metrics are EM, F1 scores, and AvNA. Our goal was to improve on the metrics of the baseline model on the dev set which were AvNA: 67.38, F1: 60.96 and EM: 57.8. In addition to this, we analyzed the specific types of errors produced by our models by observing sample predictions produced on evaluation steps while training on TensorFlow.

5.3 Experimental details

5.3.1 Baseline + Character Embeddings

We ran three experiments training our BiDAF model enhanced with character embeddings. In all experiments, the size of the convolving kernel was set to 3 and the dimensionality for the character embedding was set to 140. For each experiment, we trained our model for 30 epochs and used hidden layers with 100 features.

Experiment	Optimizer	Learning Rate	Dropout
1	Adam	0.01	0.2
2	Adadelta	0.5	0.2
3	Adadelta	0.5	0.5

5.3.2 Other Approaches

Below we present our additional approaches. Some of the approaches were run for a lower number of epochs because we terminated training upon observing suboptimal scores relative to the baseline or to the approaches above.

Model	Dropout Rate	Hidden Layer Size	Optimizer	Learning Rate	# Epochs
Dynamic Coattention Network + Character Embeddings	0.15	200	Adam	0.001	30
DCN Encoder + Coattention + BiDAF Modelling and Output	0.2	200	Adam	0.001	15
Coattention + Multi-layer Perceptron	0.2	200	Adam	0.001	15
Character Embeddings + Coattention	0.2	100	Adadelta	0.5	30

5.4 Results

5.4.1 Character Level Embeddings

The experiments with the Baseline + Character Embedding model that performed best were Experiments 1 and 2 which both used a dropout rate of 0.2. Our best performing model achieved an **EM score of 61.771 on the test leaderboard and an F1 score of 65.089** for the IID SQuAD track.

We present a summary of the results in comparison with those achieved by the baseline below.

	AvNA	EM	F1
Baseline	67.38	57.8	60.96
Character Embeddings: Adam	70.26	60.1351	64.211
Character Embeddings: Adadelta	71.18	61.771	65.089

The performance boost that resulted from adding character embeddings to the baseline were higher than we expected. These results show that richer embedding data helped the BiDAF mechanism perform far better on SQuAD 2.0 than just word embeddings.

5.4.2 Other Approaches

Our other models performed relatively poorly on SQuAD 2.0. In order to optimize our search for a model that beat the baseline, we terminated their training at earlier epochs once we observed that they significantly underperformed. However, it is still useful to compare the relative effectiveness of our different models based on our evaluation metrics. We compare the scores they achieve at 15 epochs – halfway through training time – in the table below, and include the metrics for the baseline and our character-level embedding models for the same timestep. The graphical version of the table is presented in Appendix, Fig 4.

	AvNA	EM	F1
Character Embeddings: Adadelta	68.86	60.73	63.85
Character Embeddings: Adam	68.69	59.43	63.36
Dynamic Coattention Network + Character Embeddings	66.8	57.2	60.99
Baseline	65.57	56.39	59.62
Character Embeddings: Dropout rate 0.5	62.58	53.5	56.23
DCN Encoder + Coattention + BiDAF Modelling and Output	63.62	51.47	54.82
Coattention + Multi-layer Perceptron	61.3	50.01	53.21
Character Embeddings + Coattention	62.14	50.21	53.11

We expected our approaches that used elements from DCN to perform better on the updated SQuAD 2.0 than they did. This indicates that elements from DCN do not translate well to the more difficult task of handling a dataset with unanswerable questions.

6 Analysis

Before we begin our sectioned discussion on the different model architectures that we explored, we note an interesting trend across all models. For many of our incorrect predictions that did not involve N/A as the true or predicted output, our models predicted one named entity in place of another (see Appendix, Example 1). In some cases, the models predicted a defining clause in place of its corresponding noun (see Appendix, Example 2). The first observation shows us that, though the models recognized when named entities were required as answers, they were sometimes unable to differentiate between the correctness of different named entities. The second observation highlights a limitation of our evaluation metric: though in some instances predicting a defining clause was technically correct, this resulted in a score of 0 on both EM and F1.

6.1 Baseline + Character Embeddings

Our most successful approach involved augmenting the the baseline BiDAF implementation by concatenating the word embeddings with character embeddings. Character level embeddings allow the model to learn from the morphemes that make up the words in the context and question which allows for richer data to be passed to the encoding layer. Further, since these embeddings are done at a subword level, our augmented model is better able to handle out-of-vocabulary words than the baseline.

It is likely that this experiment worked the best, because it boosted the components of the baseline model – which already performed relatively well on SQuAD 2.0 – by providing the encoder with richer input. The DCN model is the basis of much of our exploration of the alternate approaches which performed comparatively poorly. Though DCN performs well on SQuAD 1.0, our results show that the components of the BiDAF model translate much better to the more difficult task of selecting answers from the context *in addition* to classifying questions as unanswerable than the components of DCN.

We ran three main experiments involving the BiDAF model with character embeddings. Our first pair of experiments involved varying the optimizers that we used during training. In the graph below, we see that the two optimizers produced relatively similar EM, F1 and AvNA scores. A particularly interesting difference between the graphs is the faster convergence of the Adam optimizer on earlier epochs. The Adam optimizer makes use of momentum, which is likely what accounts for faster convergence. In spite of this, the model plateaus more abruptly than the one trained using Adadelta and finishes with a value slightly lower than the Adadelta optimizer across all three evaluation metrics. This tells us that though using Adam leads to faster convergence, it can perform slightly worse than Adadelta if it gets stuck in poor local optima.

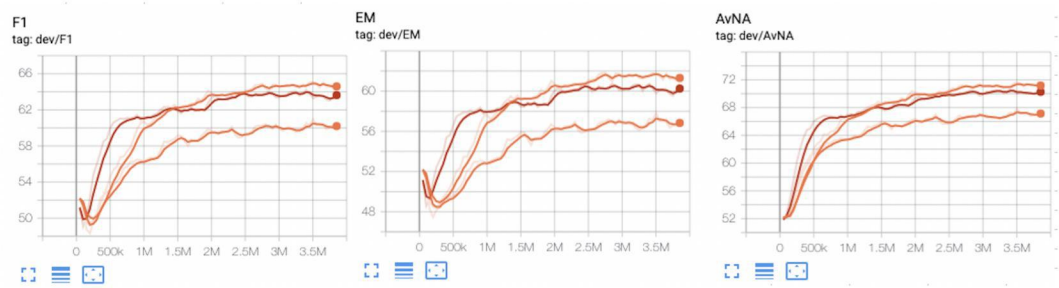

Fig. 2. F1, EM and AvNA scores for the Character Embedding model trained with Adadelta (higher orange), the Character Embedding model trained with Adam (brown), and the Baseline model with only Word Embeddings (lower orange).

Our third experiment involved using a higher dropout rate of 0.5. This model performed poorly in comparison to experiments which used a dropout rate of 0.2. This is likely because the high dropout rate led to the model predicting answers nearly at random. Though dropout helps with making models more generalizable, a high dropout rate can hurt a model since the nodes in the network are zeroed out with high frequency. This causes the model to lose too much of the information it learns during each iteration of training.

To determine the limitations of our strongest model, we conducted an error analysis on the types of questions for which it produced incorrect predictions based on the text produced in Tensorboard on our evaluation steps. Of the 100 examples that we observed, 29 resulted in predictions that would have received 0 on both EM and F1. A vast majority - 62% were as a result of incorrectly predicting a span when there was no answer (N/A). 17% involved predicting N/A when there was a correct span, and 21% involved choosing the wrong span.

The most interesting error that we saw involved a correct prediction – the model predicted the Chinese translation of a noun (Answer: Yuán Cháo Prediction: 元朝) rather than its English translation. Though the example answer had the same noun in English, here the model would have achieved EM and F1 scores of 0. This example highlights the limitations of evaluating using F1 and EM when there is an insufficient range of references and there are multiple correct answers.

6.2 Baseline + Character Embeddings + Coattention

After running this implementation for 15 epochs, we found that it underperformed greatly in comparison to the baseline. To perform an in depth error analysis, we tallied the number of completely incorrect predictions (those that would achieve an F1 score of 0) and classified the types of incorrect predictions.

At the time of our analysis, we observed 100 predictions of which 39 were completely incorrect. In an overwhelming majority of the incorrect predictions, our model incorrectly guessed N/A though there was an answer present. This result is unsurprising since 33.3% of the questions in the SQuAD 2.0 dataset have no answer based on their context: indeed, a model might perform better if it biases towards predicting N/A in cases of uncertainty. This reveals a very interesting trend. In the previous section, we saw that many of the incorrect answers involved incorrectly predicting a span even in the cases where there was no answer. Here, most of the incorrect answers involve incorrectly predicting no answer even though there is a correct answer span in the context. This supports the hypothesis that either the coattention mechanism from the DCN paper on its own, or its composition with BiDAF results in a model that relies too heavily on defaulting to no answer predictions. This reliance might have caused the model to be less likely to *attempt* to guess a span in the cases where there was some probability that N/A was the correct output. Perhaps this behavior is what resulted in the discrepancy in scores between the BiDAF model with character embeddings and the models which used components from the DCN paper.

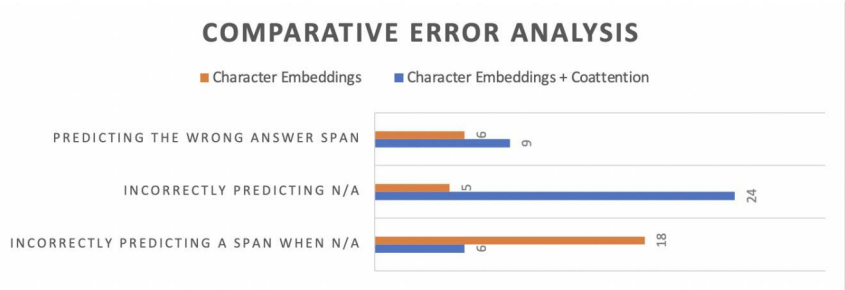

Fig. 3 Error Analysis Chart

6.3 Coattention and BiDAF Ensemble

We originally expected that replacing the BiDAF model's attention layer with the DCN's coattention mechanism would strengthen the model. This hybrid model actually performed worse than the baseline.

An examination of our incorrect predictions reveals a similar problem to the one described in the section above – the incorrect predictions tended to result from predicting N/A even though there was a span in the context that correctly answered the question. This bolsters our hypothesis that the coattention mechanism combined with layers from BiDAF might have resulted in a model that relied too heavily on predicting N/A in uncertain cases.

6.4 Dynamic Coattention with Highway Maxout Network + Character Embeddings

The architecture of the DCN was designed for SQuAD 1.0, not SQuAD 2.0. Because we applied the model to a task that it was not originally intended to handle, it performed relatively poorly.

A possible explanation for this poor performance is the dynamic nature of the decoder. Shifting predictions when solely predicting spans may affect the EM, but the model can still achieve a high F1. However, when both predicting spans and determining whether or not a question is answerable, having an iterative decoder may lead the model to have a higher chance of incorrectly shifting original span predictions to N/A classifications or incorrectly shifting original N/A classifications to span predictions. Example 3 in Appendix demonstrates an instance where the DCN predicts an

answer span, but the question is unanswerable.

This hypothesis is supported our analysis of the predictions in TensorFlow. Nearly all the errors that we observed involved incorrectly classifying questions as unanswerable or predicting answers when none was present. We observed 34 incorrect predictions out of 100 examples. 29% involved predicting a span when there was no answer, 58% involved predicting no answer when there was an answer span and 18% involved picking the wrong span.

Another possible explanation for the DCN's poor performance is that the embeddings and encodings required for accurately determining whether a question is unanswerable are not the same as those useful for predicting spans. Therefore, the encoder and coattention layers would not translate well into this new, more complex task and so the decoder also performs poorly.

6.5 Coattention Network with Multi-layer Perceptron

This implementation was an ablation of the original DCN, so our reasoning for why it performed poorly on SQuAD 2.0 is consistent with Sections 6.3 and 6.4. Example 4 in Appendix is a demonstration of this model outputting N/A when the question was answerable. The similarities between the erroneous predictions in Examples 3 and 4 – making mistakes with N/A classifications – highlight both the similarities between the HMN and MLP implementations of the DCN and its general inability to discern answerable questions from unanswerable questions.

7 Conclusion

In this project, we explored multiple approaches for improving upon the baseline model's performance on the question answering task on SQuAD 2.0. We improved upon the baseline on the QA task by a difference of 3.8 on AvNA, 3.97 on EM and 4.13 on F1 by adding character-level embeddings. This shows that enriching the embedding layer of an NLP model can significantly boost its performance. We also learned that though models like DCN performed well on QA for SQuAD 1.0, their components did not translate well to SQuAD 2.0. A primary limitation of our work is that because of the limited time frame of the project, we were unable to do a highly exhaustive parameter search. Though we explored different dropout rates, learning rates, hidden layer sizes and optimizers, it might have been interesting to do this exploration on an even broader scale. Another limitation is that our models were trained and optimized primarily on English text data. One avenue for future work is implementing a new method for identifying unanswerable question that composes well with DCN, which allows it to perform well on both classifying potentially unanswerable questions, and predicting correct spans when an answer is present.

References

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

[2] A. Farhadi1 H. Hajishirzi Minjoon Seo, Aniruddha Kembhavi. Bi-directional attention flow for machine for machine comprehension. 2018.

[3] Richard Socher Caiming Xiong, Victor Zhong. Dynamic coattention networks for question answering. 2017.

[4] Jurgen Schmidhuber Rupesh Kumar Srivastava, Klaus Greff. Highway networks. 2015.

[5] Mehdi Mirza Aaron Courville Yoshua Bengio Ian J. Goodfellow, David Warde-Farley. Maxout networks. 2013.

A Appendix

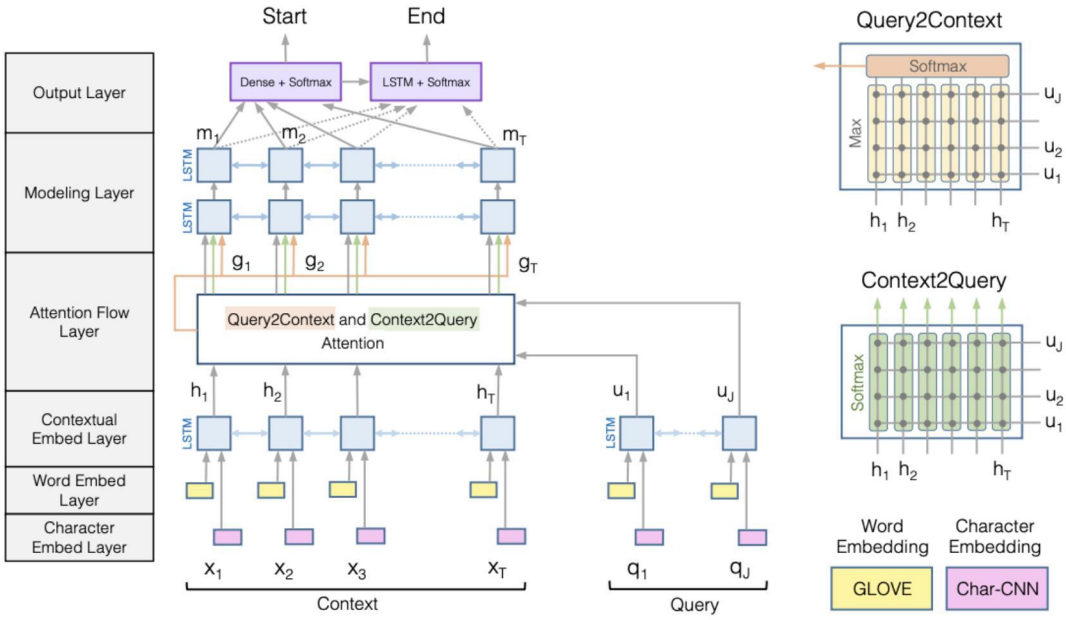

Figure 1: BiDAF Architecture Including Character Level Embeddings [2]

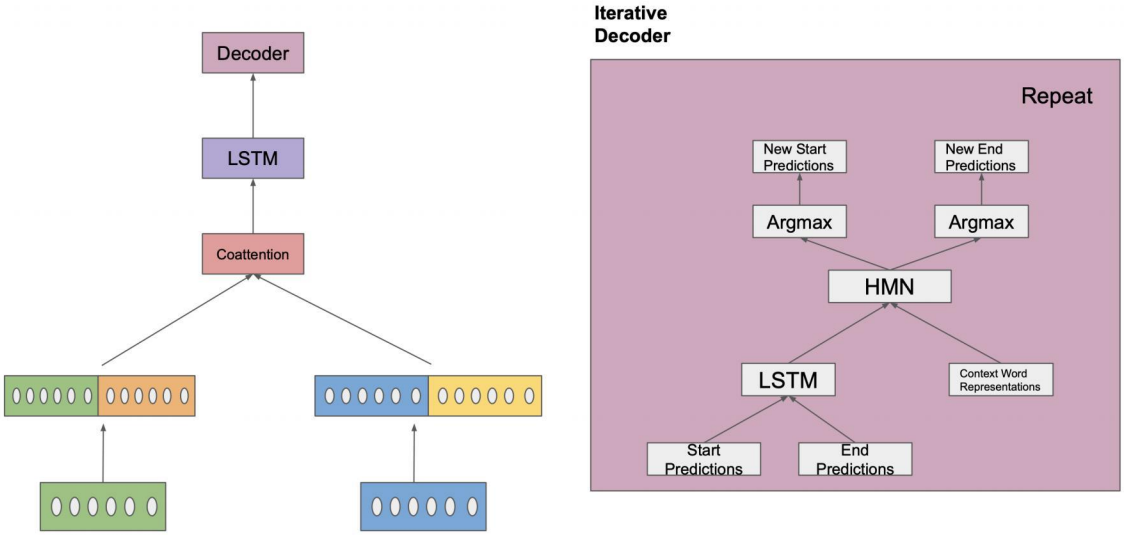

Figure 2: Dynamic Coattention Network Architecture

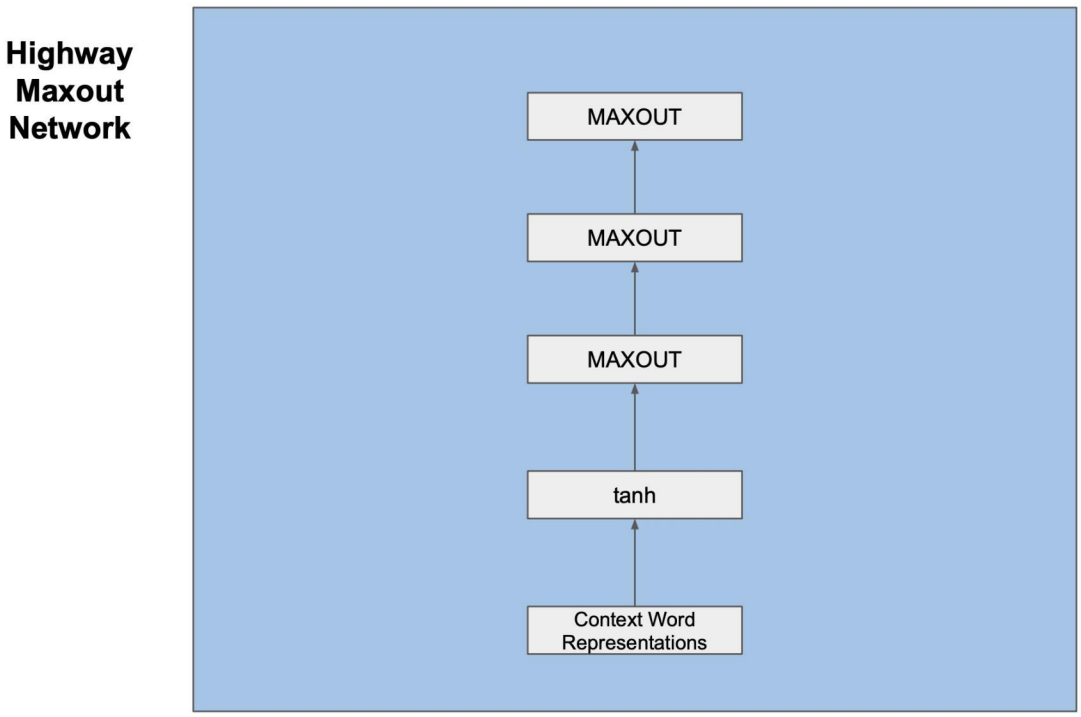

Figure 3: Highway Maxout Network Architecture

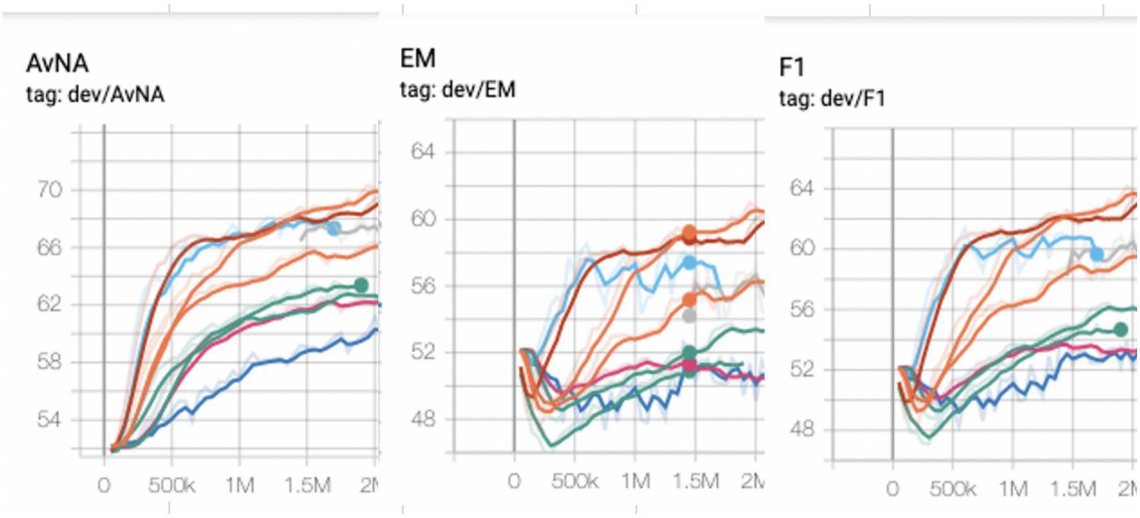

Figure 4: Graph of results of all models at 15 epochs. From highest to lowest on the F1 graph, we have Character Embeddings: Adadelta, Character Embeddings: Adam, Dynamic Coattention Network + Character Embeddings (blue and grey combined), Baseline, Character Embeddings: Dropout rate 0.5, DCN Encoder + Coattention + BiDAF Modelling and Output, Coattention + Multi-layer Perceptron and Character Embeddings + Coattention.

Question: Who designed the garden for the University Library?
Context: Another important library – the University Library, founded in 1816, is home to over two million items. The building was designed by architects Marek Budzyński and Zbigniew Badowski and opened on 15 December 1999. It is surrounded by green. The University Library garden, designed by Irena Bajerska, was opened on 12 June 2002. It is one of the largest and most beautiful roof gardens in Europe with an area of more than 10,000 m2 (107,639.10 sq ft), and plants covering 5,111 m2 (55,014.35 sq ft). As the university garden it is open to the public every day.
Answer: Irena Bajerska
Prediction: Marek Budzyński and Zbigniew Badowski

Example 1: An example where a model (here, baseline + character embedding) incorrectly predicts one named entity in place of another.

Question: What German ruler invited Huguenot immigration?
Context: Frederick William, Elector of Brandenburg, invited Huguenots to settle in his realms, and a number of their descendants rose to positions of prominence in Prussia. Several prominent German military, cultural, and political figures were ethnic Huguenot, including poet Theodor Fontane, General Hermann von François, the hero of the First World War Battle of Tannenberg, Luftwaffe General and fighter ace Adolf Galland, Luftwaffe flying ace Hans-Joachim Marseille, and famed U-boat captain Lothar von Arnauld de la Perière. The last Prime Minister of the (East) German Democratic Republic, Lothar de Maizière, is also a descendant of a Huguenot family, as is the German Federal Minister of the Interior, Thomas de Maizière.
Answer: Frederick William
Prediction: Elector of Brandenburg

Example 2: An example where a model predicts a defining clause corresponding to a noun.

Question: Optional Committees are committees which are set down under what?
Context: Committees comprise a small number of MSPs, with membership reflecting the balance of parties across Parliament. There are different committees with their functions set out in different ways. Mandatory Committees are committees which are set down under the Scottish Parliament's standing orders, which govern their remits and proceedings. The current Mandatory Committees in the fourth Session of the Scottish Parliament are: Public Audit; Equal Opportunities; European and External Relations; Finance; Public Petitions; Standards, Procedures and Public Appointments; and Delegated Powers and Law Reform.
Answer: N/A
Prediction: the Scottish Parliament's standing orders

Example 3: Incorrect Prediction from DCN + Character Embeddings

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

Example 4: Incorrect Prediction from MLP + Character Embeddings