Are you sure of your answer? Think again.

Lakshmi Manoharan
Department of Computer Science
Stanford University
mlakshmi@stanford.edu

Arjun Parthipan
Department of Management Science and Engineering
Stanford University
arjun777@stanford.edu

Abstract

This project implements a deep neural system for question-answering on the Stanford Question Answering Dataset (SQuAD). In particular, we re-implement the Dynamic Coattention Network (DCN) by Xiong et. al.[**1**] that determines the answer span using an iterative reasoning scheme. The DCN uses a two-level attention network to generate coded representations of the passage and the question, and uses a dynamic pointing decoder to iteratively determine the answer. The iterative nature of the DCN was observed to make the model robust to local maxima. Our implementation achieved an F1 score of 71.36% and an EM score of 61.418% on the test set for SQuAD.

1 Introduction

Machine Comprehension remains an interesting challenge in the domain of Natural Language Processing, with ground-breaking models continuously redefining the potential of neural architectures. There have been several significant breakthroughs in the realm of question-answering, with recent models surpassing human performance on the Stanford Question Answering Dataset (SQuAD). The SQuAD, a dataset consisting of 10000+ questions on a set of Wikipedia articles, was first released in 2016. The original SQuAD model[**2**] implemented a logistic regression model, achieving an F1 score of 51.0% against a human performance of 86.6%.

In this project, we explore different neural architectures for a question-answering system. In particular, we re-implement the following: (1) Basic Model with Simple Attention (baseline) (2) Basic Model with Bidirectional Attention Flow (3) Dynamic Coattention Network. We also take efforts to study the properties of the dataset and extensively analyze the architecture to identify its strengths, weaknesses and potential areas of improvement. This report is organized into the following sections: (1) Related Work (2) Model Architecture (3) Experiments Discussion (4) Conclusion.

2 Related Work

Attention mechanisms have been found to contribute significantly to the machine comprehension task, answering a question about a given context paragraph. There are various implementations of attention in different state of the art models for the SQuAD question answering system.

The BiDAF (Bi-Directional Attention Flow) model [**3**] introduced a hierarchical architecture for obtaining a question aware context representation. This is based on the idea that the attention must flow in both directions viz., from the question to the context and from the context to the question. It included character level, word level and contextual embeddings along with bi-directional attention flow. This avoids early summarization and provides a question aware context representation.

Another model that was successful in the machine comprehension task was the R-NET model[**4**]. Here, the question aware context representations are obtained by first building representations of the questions and context separately, using a gated matching layer to match the question and the

context and then implementing a self matching attention mechanism. The self matching attention logic matches the context against itself and thus refines the context representation with information from the entire context. The gated attention based recurrent network allocates different levels of importance to different parts of the context depending on their relevance to the question.

The Dynamic Coattention Network (DCN) model[**1**] is also a successful model for the question answering task. It consists of an encoder module, a coattention unit and a dynamic pointer decoder. This differs from the BiDAF mechanism in that it involves two levels of attention computation. The first level attention output is obtained for each word in the question and for each word in the context. In the second level, the attention is computed over the attention outputs from the first level. The dynamic pointer decoder makes use of the highway maxout network to compute start index and end index probabilities. An iterative procedure is followed to improve predictions by using previous predictions. This method thus helps to recover from local maxima corresponding to incorrect answer indices.

3 Model Architecture

In this section, we provide an overview of the incremental models we have implemented and finally describe the architecture of the modified Dynamic Coattention Network that reimplements [**1**] and couples it with the representative power of the modeling layer featured in [**3**].

We used the default model provided as part of the Stanford Course CS224N as our baseline model and explored several schemes like bi-directional attention flow, co-attention encoding and iterative reasoning, which result in a significant boost in performance, as we will show in Section 4. For all of the architectures described below, we assume that for each SQuAD example (context, question, answer), the context is represented by a sequence of d-dimensional word embeddings $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_M \in R^d$ and the question by a sequence of d-dimensional embeddings $\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N \in R^d$.

3.1 Baseline Model

The baseline model provided had three major components:

1. A bi-directional GRU encoder for encoding the context and question embeddings to obtain the corresponding context and question hidden states, $\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_M \in R^{2h}$ and $\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_N \in R^{2h}$.
2. An attention layer that computes the attention output $\boldsymbol{a}_i$ for each context hidden state $\boldsymbol{c}_i$ as a sum of the question hidden states $\boldsymbol{q}_j$ weighted by the corresponding attention distribution. We then create a blended representation $\boldsymbol{b}_i$ for each context by concatenating each $\boldsymbol{c}_i$ with its corresponding attention output $\boldsymbol{a}_i$.
3. An output layer that uses a fully-connected ReLU layer to downproject the blended representation, followed by two linear downprojecting layers: one used to determine the start position and the other used to determine the end position of the answer span.

3.2 Improvement 1: Bi-Directional Attention Layer

We substituted the simple dot product attention layer in the baseline model with an enhanced bi-directional attention flow model as suggested in [**3**]. Leveraging Question-to-Context (Q2C) attention in addition to the Context-to-Question (C2Q) attention, with no changes to the modeling layer, yielded a 7% boost in F1 score over the baseline model.

3.3 Improvement 2: Co-attention with LSTM Modeling Layer

We implemented a custom model that uses a co-attention layer[**1**] in place of the basic dot product attention layer in the baseline model, coupled with the modeling layer used in [**3**]. We expected a significant increase in performance as this would allow us to capture the interaction among the context words conditioned on the query, rather than a simple word embedding encoding as GloVe. The custom model gained a 22% increase in F1 score over the baseline model.

3.4 Improvement 3: Dynamic Coattention Network with Modeling Layer

The Dynamic Coattention Network [**1**] has three components: (1) an encoder for the context and the question (2) a coattention layer that attends to both context and question simultaneously, and generates a fused representation of the attention contexts (3) a dynamic pointing decoder that iteratively finds the location of the start and end positions of the answer spans. An overall view of the architecture is presented in Figure 1.

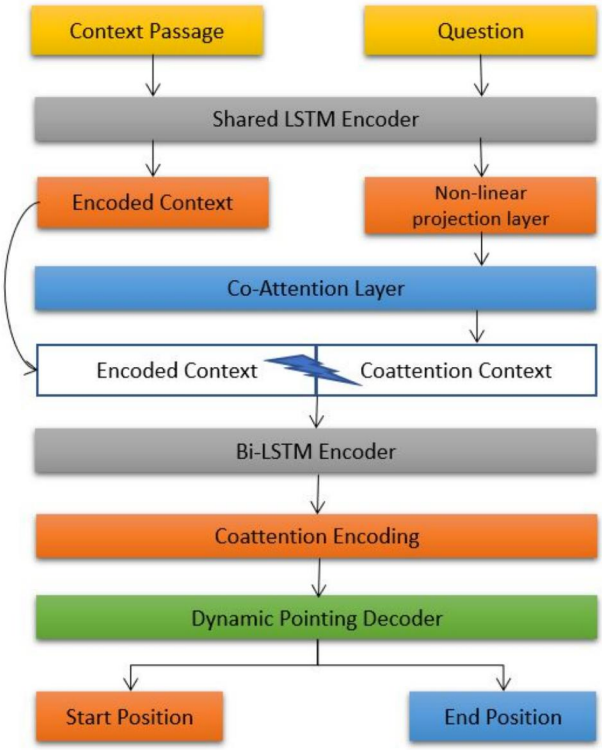


Figure 1: Model Architecture: Dynamic Coattention Network

Encoder Each d-dimensional context vectors $\boldsymbol{x}_i$ and question vector $\boldsymbol{y}_i$ are encoded using an LSTM encoder to obtain context encoding $\boldsymbol{c}_i$ and query encoding $\boldsymbol{q}_i^{'}$. Also, we add the sentinel vectors $\boldsymbol{c}_\Phi$ and $\boldsymbol{q}_\Phi^{'}$ to the context and question matrices $\boldsymbol{C} \in R^{M+1 \times d}$ and $\boldsymbol{Q}^{'} \in R^{N+1 \times d}$. The question matrix $\boldsymbol{Q}^{'}$ is further processed using a tanh non-linearity to obtain $\boldsymbol{Q} \in R^{N+1 \times d}$. This allows for differences in the context and question space.

Coattention Layer We first create the affinity matrix $\boldsymbol{L} = \boldsymbol{C}\boldsymbol{Q}^T \in R^{(M+1)\times(N+1)}$, and use it to generate attention weights $\boldsymbol{A}_Q = softmax(\boldsymbol{L}) \in R^{(M+1)\times(N+1)}$ and $\boldsymbol{A}_D = softmax(\boldsymbol{L}^T) \in R^{(N+1)\times(M+1)}$. We compute the attention summary $\boldsymbol{C}^Q$ and fused summaries $\boldsymbol{A}_D^T\boldsymbol{Q}$ and $\boldsymbol{A}_D^T\boldsymbol{C}^Q$ to obtain $\boldsymbol{C}^D$ as follows:

$$\boldsymbol{C}^Q = \boldsymbol{A}_Q^T\boldsymbol{C} \in R^{(N+1)\times d}$$

$$\boldsymbol{C}^D = \boldsymbol{A}_D^T[\boldsymbol{Q}; \boldsymbol{C}^Q] \in R^{(M+1)\times 2d}$$

Xiong et. al. [**1**] argue that $\boldsymbol{A}_D^T\boldsymbol{C}^Q$ can be thought of as a mapping of the question encoding into the document space. We finally obtain the coattention encoding as given below:

$$\boldsymbol{U}^{''} = BiLSTM([\boldsymbol{D}; \boldsymbol{C}^D])$$

.

Modeling Layer We introduced an additional BiLSTM modeling layer to better capture interaction between the fused context-question encoding obtained from the coattention layer to obtain $\boldsymbol{U'} \in R^{(M+1)\times 4d}$. This encoded representation is then downprojected using a nonlinear layer to obtain the final encoding $\boldsymbol{U'} \in R^{(M+1)\times 2d}$, which forms the input to the Dynamic Pointing Decoder. The introduction of the modeling layer resulted in a considerable boost to the F1 score.

Dynamic Pointing Decoder The architecture of the dynamic pointing decoder is shown in Figure 2. The power of DCNs lies in their ability to iteratively determine answer spans so as to recover from local maxima. Initial start and end positions are determined using a nonlinearity followed by a downprojecting linear layer as in BiDAF [3]. At each iteration, new start and end locations are estimated by exploiting a multi-layer neural architecture coupled with a LSTM decoder.

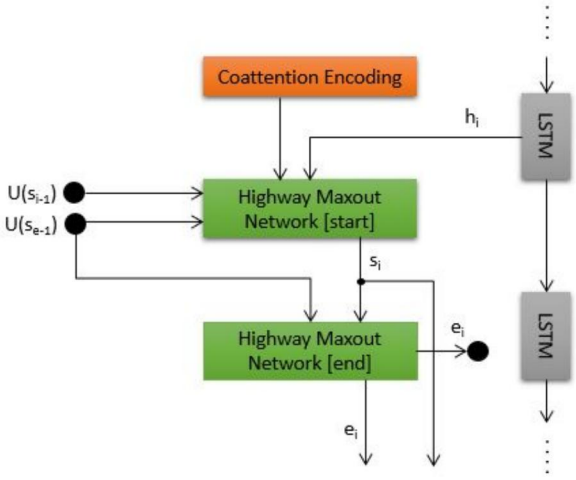


Figure 2: Dynamic Pointing Decoder Unit

Subsequent answer span locations $\boldsymbol{s}_i$ and $\boldsymbol{e}_i$ are found as follows:

$$s_i = \underset{t}{\text{argmax}}(\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_M)$$

$$e_i = \underset{t}{\text{argmax}}(\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_M)$$

We show the computation for $\boldsymbol{\alpha}$ below. $\boldsymbol{\beta}$ is similarly calculated using HMN_{end}.

$$\boldsymbol{\alpha}_t = HMN_{start}(u_t, h_i, u_{s_{i-1}}, u_{e_{i-1}})$$

$$HMN_{start}(u_t, h_i, u_{s_{i-1}}, u_{e_{i-1}}) = max(W_F[m_t^1; m_t^2]) + b^3)$$
$$r = max(W_D[h_i; u_{s_{i-1}}; u_{e_{i-1}}])$$
$$m_t^1 = max(W_1[u_t; r]) + b^1)$$
$$m_t^2 = max(W_2[m_t^1]) + b^1)$$

4 Experiments & Discussion

4.1 Hyperparameter Tuning

4.1.1 Tuning to the dataset

An exploratory analysis on the SQuAD reveals that it contains questions that require complex reasoning techniques to predict the correct answer. The histograms depicting the number of words in the question, the context and the answer within the train set are shown in Figure 3 and Figure 4. It can be seen that the majority of the questions and the contexts are much shorter than their maximum lengths. Also most answers are short (more than 90%) and are less than 10 words long. Hence, we fixed the maximum context length to be 400 and the maximum question length to be 30.

4.1.2 Handling Overfitting

We observed that the model obtained high F1 scores (89%) on the training set, while it performed relatively poor on the dev set. Hence, we tuned the dropout to 0.25 and obtained a 1% increase in the F1 score. We also experimented with different learning rates and identified that a learning rate of 0.001 achieves optimal performance. We use 100-dimensional GloVe embeddings, as longer embeddings did not result in a significant improvement in performance.

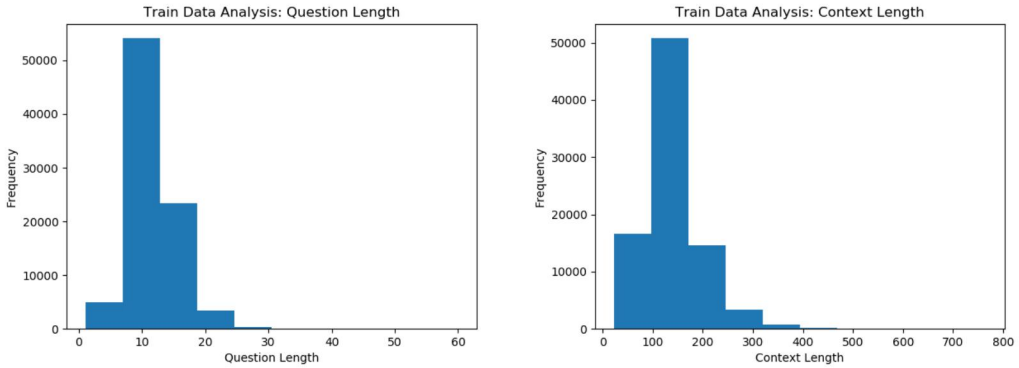
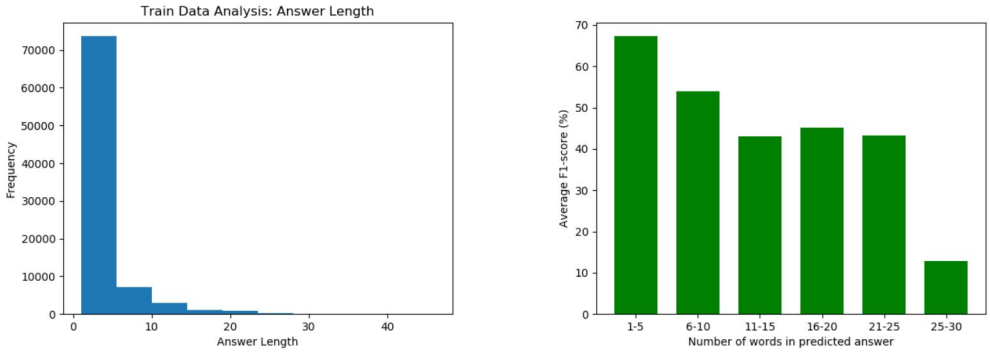


Figure 3: Histograms of the question length and the context length in the train set



(a) (b)

Figure 4: (a) Histogram of answer length in the train set (b) Answer length vs Average F1 score

4.2 Error Analysis

On analyzing a few examples from the dev set, we observed a few common error patterns that repeat themselves. We present the most common ones below.

4.2.1 Answers that require picking from a list of values

Consider the following example.
Context: southern california is home to many major business districts . central business districts (cbd) include downtown los angeles , downtown san diego , downtown san bernardino , downtown bakersfield , south coast metro and downtown riverside .

Question: what is the only district in the cbd to not have " downtown " in it 's name ?
True Answer: south coast metro
Predicted Answer: central business districts

Answers like these requires parsing a list of values and choosing the right one. This is a complex task, not modeled by our current system. We identify this as one of our future tasks.

4.2.2 Answer Length

Figure 4(b) shows the performance of the model for different answer lengths. It is seen that the model performs best for questions that have an answer length in the range of 1 to 5 words. The

”When”, ”Who”, ”What” and ”Where” type of questions had the average answer length to be small. The questions with answer lengths in the range of 11 to 25 words had similar average F1 score.

4.2.3 Ambiguous/Incorrect Answer Boundaries

Consider the following example:
Context:in many parts of the united states , after the 1954 decision in the landmark court case brown v. board of education of topeka that demanded united states schools desegregate ” with all deliberate speed ” , local families organized a wave of private ” christian academies ” . in much of the u.s. south , many white students migrated to the academies , while public schools became in turn more heavily concentrated with african-american students (see list of private schools in mississippi) . the academic content of the academies was usually college preparatory . since the 1970s , many of these ” segregation academies ” have shut down , although some continue to operate . [citation needed]

Question:in what part of the united states did many students migrate to christian academies during the desegregation period ?
True Answer: south
Predicted Answer: u.s. south

There have been several observed instances where there can be multiple correct answer spans. Also, the reading comprehension task in itself is arguably very subjective. It is mainly due to this inherent ambiguity that evaluation on the 3-answer dev-set provided by SQuAD yields a better F1 and EM score than when evaluated against a single gold standard.

4.2.4 Answers involving conjunctions

Consider the passage in Figure 6. The true answer and predicted answer are as follows:
True: flammable cabin and space suit materials
Predicted: space suit

When the answer involves conjunctions like *and*, the model mis-predicts the answer span to exclude one or more of the expressions in the answer.

4.3 Visualization

In an effort to better understand and visualize model performance, we used various tools and techniques to visually capture attention mechanisms and model performance, which we present here. Visualizations pertain to our implementation of the Dynamic Coattention Model.

4.3.1 Question-to-Context Attention

In Figure 5, we observe that the question word ’ad’ attends to the tokens ’30-second’ and ’advertisement’ the most. From the visualization, it is evident that the model learns to attend to words that have most correlation with themselves.

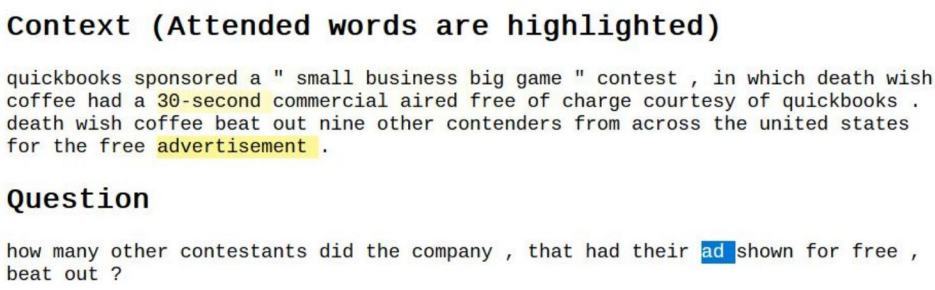


Figure 5: Visualizing Question-to-Context Attention

4.3.2 Start and End Position Probability

Figure 6 shows the visualization of the start index and end index probability distributions for the question answering task. The green highlight indicates the start index and the purple highlight indicates the end index of the answer.

Question

what type of materials inside the cabin were removed to help prevent more fire hazards in the future ?

Context (highlighted = high start probability)

to remedy the causes of the fire , changes were made in the block ii spacecraft and operational procedures , the most important of which were use of a nitrogen/oxygen mixture instead of pure oxygen before and during launch , and removal of flammable cabin and space suit materials . the block ii design already called for replacement of the block i plug-type hatch cover with a quick-release , outward opening door . nasa discontinued the manned block i program , using the block i spacecraft only for unmanned saturn v flights . crew members would also exclusively wear modified , fire-resistant block ii space suits , and would be designated by the block ii titles , regardless of whether a lm was present on the flight or not .

Figure 6: Visualizing Start and End Position Probability

4.3.3 Robustness of DCN to Local Maxima

The DCN model owes its robustness to local maxima to the iterative reasoning scheme of the decoder. Consider the example below:
Context: the crew of apollo 8 sent the first live televised pictures of the earth and the moon back to earth , and read from the creation story in the book of genesis , on christmas eve , 1968 . an estimated one-quarter of the population of the world saweither live or delayedthe christmas eve transmission during the ninth orbit of the moon . the mission and christmas provided an inspiring end to 1968 , which had been a troubled year for the us , marked by vietnam war protests , race riots , and the assassinations of civil rights leader martin luther king , jr. , and senator robert f. kennedy .
True Answer: one-quarter (start_pos:20, end_pos:20)
Predicted Answer: one-quarter (start_pos:20, end_pos:20)

In Figure 7, we observe how the model wrongly predicts the answer to be 'an estimated one-quarter' (start_pos:16, end_pos:20) in the first iteration, but recovers from the local optima to correctly determine the answer span to be (start_pos:20, end_pos:20)

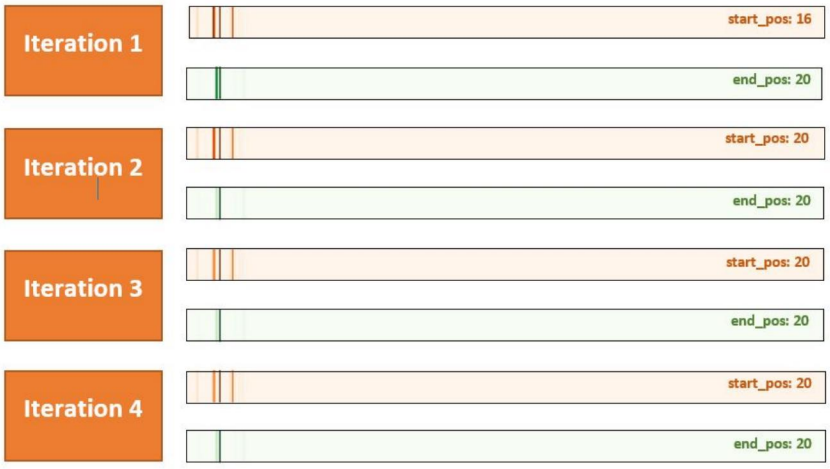
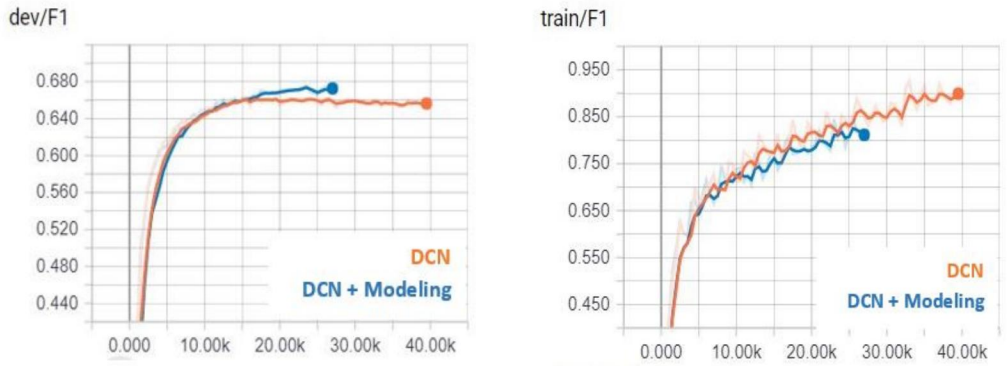


Figure 7: Iterative Reasoning by the Dynamic Pointing Decoder

4.4 Results

The incremental development of the model shed light on the contribution of each of the components to the the overall system. We were able to clearly observe how attention is a powerful tool to solving the complex task of question-answering. In addition to reimplementing the Dynamic Coattention Network [**1**], we were also able to successfully experiment and determine how a modeling layer on top the DCN encoder module can result in a significant increase in performance.



(a) DCN vs [DCN + Modeling] on Dev Set

(b) DCN vs [DCN + Modeling] on Dev Set

Figure 8: Performance Comparison

We present the dev and train F1 scores achieved by our DCN model and DCN + Modeling Layer model in Figure 8. We also notice that the modeling layer alleviates overfitting. We argue that this could be because the modeling layer captures more complex interactions between the query-dependent context with itself. The results of all experiments have been tabulated below:

Performance Comparison		
Our Implementation (Evaluated on Dev Set)		
Model	F1 Score	EM Score
Baseline Model	43.652	34.428
Bidirectional Attention + Baseline	50.375	40.639
Coattention + Baseline	65.994	55.09
Coattention Network + Modeling Layer	67.812	57.569
Dynamic Coattention Network	71.865	61.145
Dynamic Coattention Network + Modelling Layer	TBD	TBD
Reference Implementation		
Dynamic Coattention Network (DCN)	75.6	65.4
Bidirectional Attention Flow (single)	77.3	68
Bidirectional Attention Flow (ensemble)	80.7	72.6
r-Net	83.7	76.7

Table 1: Performance Evaluation

5 Conclusion

Our model achieved an overall improvement of 28% in F1 score over the baseline model. Despite acceptable results, we recognize that question-answering is a challenging task with avenues for improvement. Future work involves improving the model to perform better on questions requiring longer answer lengths as well as generalizing for datasets other than the SQuAD.

References

[1]Xiong, Caiming, Victor Zhong, and Richard Socher. ”Dynamic coattention networks for question answering.” arXiv preprint arXiv:1611.01604 (2016).

[2]Rajpurkar, Pranav, et al. ”Squad: 100,000+ questions for machine comprehension of text.” arXiv preprint arXiv:1606.05250 (2016).

[3]Seo, Minjoon, et al. ”Bidirectional attention flow for machine comprehension.” arXiv preprint arXiv:1611.01603 (2016).

[4]Microsoft Research Asia, ”R-Net: Machine Reading Comprehension with Self-Matching networks”