Machine Reading Comprehension on SQuAD

Don Mai
Stanford University
Stanford, CA 94305
donmai@stanford.edu

Tian Tan
Stanford University
Stanford, CA 94305
tiantan@stanford.edu

Abstract

For this project, we explored different attention-based RNN architectures (DCN, DCN+, and FusionNet), which tackle a machine reading comprehension (MRC) task that consists of trying to construct a model that performs well on the Stanford Question Answering Dataset (SQuAD). We pick certain components from those RNN architectures to implement on top of the provided baseline model, and also try some additional optimization and model tuning strategies. As a result of our various additions and changes, on SQuAD, we are able to achieve a dev F1 score 75.3, EM 65.1 for our single model, and a dev F1 score 78.3, EM 68.4 for our ensemble model consisting of 4 significantly different model architectures.

1 Introduction

In recent years, there has been a substantially increased amount of interest and advancement in the area of machine reading comprehension (MRC). In particular, the publicly released Standard Question Answering dataset (SQuAD) has spurred the development of many prediction models for on the reading comprehension problem of Question Answering (QA). This dataset consists of 100,000+ question-answer pairs on 500+ articles, with a roughly 80% train-10% dev-10% test split[1]. One way models can make predictions for this dataset is by learning from training examples in this dataset, and then, given a context paragraph and question pair, being asked to come up with an answer that is guaranteed to be a "span" of text in the given context paragraph. For some context paragraph and question pairs, there may be multiple valid answers, since the answers are crowdsourced, and so both a F1 and Exact Match (EM) metric are used to assess the performance of your model. One class of models that has performed particularly well on this dataset are attention-based models, which encapsulate a mechanism that helps the model only focus on the relevant portions of the context paragraph (e.g., by making the context aware of the query) when selecting an answer span.

For this report, we explore and analyze existing attention-based architectures that perform well on the SQuAD dataset. We discuss our approach towards constructing our own model for prediction on this dataset. We then move on to discussing the components from various architectures that we added on top of the baseline, and also describe some extensions and modifications that we make. We then discuss and analyze in detail some results that were obtained from prediction on the dev set.

2 Approach

Because our approach towards constructing a performant model involves re-implementing certain components of recently published attention architectures for SQuAD, in this section, we will also discuss the relevant literature and existing work corresponding to those models in some detail.

We start by introducing our neural architectures for machine comprehension on SQuAD. These include: the full model of Dynamic Coattention Network (DCN) [2], the Deep Residual Coattention encoder as discussed in DCN+ [3], and the FusionNet architecture in [4]. Extra extensions include:

using better and higher dimensional pretrained GloVe word embeddings [5], a smarter span selection mechanism, a policy gradient finetuning scheme (which is applied to FusionNet in our experiments, as FusionNet is our best performing single model) and a distributional ensemble strategy.

For subsequent discussion, let's assume we are given a document/context (C) of m words and a question (Q) of n words. Each word is represented by using pretrained GloVe word embeddings: we use g_i^C to represent the GloVe embedding for the i-th word in the context.

2.1 Deep Residual Coattention (DRCoattn) Encoder

The Deep Residual Coattention (DRCoattn) [3] is a multi-level information fusion module that builds upon the coattention encoder in [2]. It modifies the coattention encoder in two ways: (1) the DRCoattn incorporates self-attention by stacking coattention layers, and (2) the outputs from coattention layers are merged with residual connections. *We implemented this DRCoattn encoder with the provided baseline decoder*. In order to keep the main report compact, please refer to Appendix A in the *separate Supplementary Material* for additional details for this encoder.

2.2 Dynamic Coattention Network (DCN)

We choose to implement the full model of DCN in [2], which includes a coattention encoder and a dynamic pointing decoder. For compactness, we refer the reader to the *Appendix D in the supplementary material* for all our implementation details of the DCN decoder (some details are not explicitly included in the original paper).

2.3 FusionNet

The final attention-based architecture we implemented is FusionNet in [4].It achieved the first position for both single and ensemble model on the official SQuAD leaderboard as of Oct. 4th, 2017, which motivates us to replicate their attention-based mechanism so we could achieve similar levels of performance.

2.3.1 Fully-aware attention on history of word

Huang et al.[4] defined the history of the i-th word as the concatenation of all representations generated for this word. To better exploit the rich information embedded in history-of-word for neural structure models, they proposed the idea of fully-aware attention, which can be used to fuse various levels of information from one body (e.g. question) to another (e.g. context). Consider two set of history-of-words for words in text A and B: $\{HoW_1^A, \dots, HoW_m^A\}, \{HoW_1^B, \dots, HoW_n^B\} \subset \mathbb{R}^{d_h}$. Fusing text body B to body A via fully-aware attention is conducted by computing the summarized information vector for every word in body A. For the i-th word in A, the summarized fusion is computed as,

- Generate an attention score $S_{ij} = S(HoW_i^A, HoW_j^B) \in \mathbb{R}$ for $j = 1, \dots, n$.
- Compute the normalized attention weights: $\alpha_{ij} = exp(S_{ij}) / \sum_k exp(S_{ik})$.
- The summarized fusion information for the i-th word in A is, $\sum_j \alpha_{ij} HoW_j^B$, and it is often concatenated to the current history of the i-th word.

A symmetric nonlinear scoring function is proposed and used in FusionNet, which is of the following form,

$$S_{ij} = f(W(HoW_i^A))Df(W(HoW_j^B)) \tag{1}$$

where $W \in \mathbb{R}^{k \times d_h}, D \in \mathbb{R}^{k \times k}$ are trainable matrices, and D is diagonal, k is the attention hidden size. $f(\cdot)$ is an element-wise activation function, and $f(\mu) = max(0, \mu)$ is used in FusionNet.

2.3.2 Architecture

An illustration of FusionNet figure is included in Appendix B of the supplementary materials. The end-to-end neural architecture consists of the following components.

Input vectors. For each word in context (C) and question (Q), we use the 300-dim GloVe embeddings pretrained on the 840B Common Crawl corpus [5] as input vectors. *Note: this is different from the original FusionNet, which uses significantly higher dimensional input vectors with more features (e.g., concatenated CoVe embeddings, NER and POS tags).*

Fully-aware multi-level fusion: word-level and reading. Now, we will introduce the multi-level fusion in FusionNet starting with the word-level fusion, which is an attention based fusion on GloVe embedding g_i,

$$\hat{g}_i^C = \sum_j \alpha_{ij} g_j^Q, \quad \alpha_{ij} \propto exp(S(g_i^C, g_j^Q)), \quad S(v_x, v_y) = ReLU(Wv_x)^T ReLU(Wv_y).$$

where $W \in \mathbb{R}^{300\times300}$. The enhanced input vectors for context is then $\tilde{g}_i^C = [g_i^C; \hat{g}_i^C]$. Next, in the reading component, a separate BiLSTM is used to form low-level and high level concepts for C and Q,

$$h_1^{Cl}, \ldots, h_m^{Cl} = BiLSTM(\tilde{g}_1^C, \ldots, \tilde{g}_m^C), \quad h_1^{Ql}, \ldots, h_n^{Ql} = BiLSTM(g_1^Q, \ldots, g_n^Q)$$
$$h_1^{Ch}, \ldots, h_m^{Ch} = BiLSTM(h_1^{Cl}, \ldots, h_m^{Cl}), \quad h_1^{Qh}, \ldots, h_n^{Qh} = BiLSTM(h_1^{Ql}, \ldots, h_n^{Ql})$$

The hidden size of BiLSTM is set to be 125 (the same as in the original paper).

Question Understanding. we then apply a new BiLSTM to form the *final question representation* U_Q,

$$U_Q = \{u_1^Q, \ldots, u_n^Q\} = BiLSTM([h_1^{Ql}; h_1^{Qh}], \ldots, [h_n^{Ql}; h_n^{Qh}])$$

where each $u_j^Q \in \mathbb{R}^{250}$ is the understanding vector for the j-th word in question (Q).

Fully-aware multi-level fusion: higher-level. For higher level fusion, the information in the question (Q) is fused to the context (C) via multi-level fully-aware attention on history-of-word. At the current stage, the history-of-words for C and Q are $HoW_i^C = [g_i^C; h_i^{Cl}; h_i^{Ch}], \quad HoW_i^Q = [g_i^Q; h_i^{Ql}; h_i^{Qh}]$. Then three different levels of information from Q are fused to C through fully-aware attention described above,

$$\text{Low-level: } \hat{h}_i^{Cl} = \sum_j \alpha_{ij}^l h_j^{Ql}, \quad \alpha_{ij}^l \propto exp(S^l(HoW_i^C, HoW_j^Q)).$$
$$\text{High-level: } \hat{h}_i^{Ch} = \sum_j \alpha_{ij}^h h_j^{Qh}, \quad \alpha_{ij}^h \propto exp(S^h(HoW_i^C, HoW_j^Q)).$$
$$\text{Understanding-level: } \hat{u}_i^C = \sum_j \alpha_{ij}^u u_j^Q, \quad \alpha_{ij}^u \propto exp(S^u(HoW_i^C, HoW_j^Q)).$$

where all the scoring functions S^l, S^h, S^u are the fully-aware attention score function defined in Equation (1), but with different and independent parameters to learn different levels of fusion. Attention hidden size is set to be $k = 250$. Next, a new BiLSTM is used to obtain a context representation with fused information from the question,

$$\{v_1^C, \ldots, v_m^C\} = BiLSTM([h_1^{Cl}; h_1^{Ch}; \hat{h}_1^{Cl}; \hat{h}_1^{Ch}; \hat{u}_1^C], \ldots, [h_m^{Cl}; h_m^{Ch}; \hat{h}_m^{Cl}; \hat{h}_m^{Ch}; \hat{u}_m^C])$$

Fully-Aware Self-Boosted Fusion. Finally, the self attention/fusion is applied on the history-of-word for the context via fully-aware attention. The authors believe that the self-boosted fusion can help the model consider distant parts in the context. Now, the history-of-word is updated to be $HoW_i^C = [g_i^C; h_i^{Cl}; h_i^{Ch}; \hat{h}_i^{Cl}; \hat{h}_i^{Ch}; \hat{u}_i^C; v_i^C]$, and the fully-aware attention is applied, $\hat{v}_i^C = \sum_j \alpha_{ij}^s v_j^C, \quad \alpha_{ij}^s \propto exp(S^s(HoW_i^C, HoW_j^C))$. *The final context representation* is generated by (the LSTM hidden size is still 125),

$$U_C = \{u_1^C, \ldots, u_m^C\} = BiLSTM([v_1^C; \hat{v}_1^C], \ldots, [v_m^C; \hat{v}_m^C])$$

Decoder. The decoder takes the question understanding U_Q and context understanding U_C as inputs and generates predictions for the answer span, more precisely, the start and end probability distributions P_i^S, P_i^E for each word i in the context. We implement the same decoder as in the FusionNet paper.

2.3.3 Training

For FusionNet training, we first conduct a supervised learning with cross entropy objective that maximize $\sum_k (log(P^S_{s^k}) + log(P^E_{e^k})$, where s^k, e^k are the ground truth start and end positions for the k-th example. Then, we deploy a second phrase fine tuning with policy gradients [6] and simple reinforcement learning (RL) rewards *after the supervised training*. The procedure is similar to the one used in the Reinforced Mnemonic Reader [7]. The RL state x can be viewed as the GloVe embeddings of the question and the context, $x = \{g^C_1, \dots, g^C_m, g^Q_1, \dots, g^Q_n\}$, and the FusionNet can be viewed as a policy network $\pi(a|x)$ parameterized by θ which maps the state x to a probability distribution over the joint discrete action space $A = \{1, \dots, m\}^2$, where the superscript 2 indicates we need to select one action position for the start a_s and one for the end a_e. The actions are selected by sampling from distributions $a_s \sim P^S, a_e \sim P^E$, and each RL trajectory only lasts for one-time step here (one-shot decision-making). We adopt a simple RL reward, for simplicity, as follows,

$$R(x, [a_s, a_e]) = -(|a_s - s| + |a_e - e|), \quad [a_s, a_e] \in A \tag{2}$$

Then, the REINFORCE policy gradient can be computed and used to update θ by minimizing the following loss function,

$$L_{rl}(\theta) = -\frac{1}{k}\sum_k log(\pi_\theta(a|x))R(x,a) = -\frac{1}{k}\sum_k (logP^S_{a_s} + logP^E_{a_e})R(x, [a_s, a_e]) \tag{3}$$

where k indicates the k-th instance in a mini-batch. Since each trajectory only has one example, we currently have not applied any variance reduction technique.

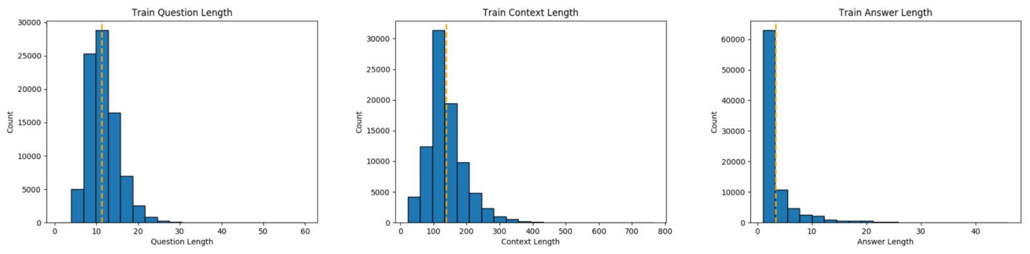
2.4 Smarter Span

For all our models (except the RL finetuning model), we predict the answer span $\hat{s}$, $\hat{e}$ that has the maximum $P^S_{\hat{s}} P^E_{\hat{e}}$ subject to the constraint $0 \leq \hat{e} - \hat{s} \leq 15$.

2.5 Ensemble Strategy

We deploy a distributional ensemble strategy when combining different models. First, we average the start P^S and end P^E distributions generated from different models. Then, we use the smarter span extension discussed above to select answer spans from the averaged distribution.

3 Experiments



(a) Training question length (b) Training context length (c) Training answer length

Figure 1: Histogram of training data. Dashed vertical orange line represents mean value.

3.1 Experimental details

Figure 1 shows the histogram of the training dataset. It clearly shows that the upper bound of question length is about 30, most of the context paragraphs are below 400 words, and most of the correct answers are less than 15 words. Therefore, we pick our question length $n = 30$, context length $m = 400$, and smarter span limit of 15 for all models. The first two choices help reduce training time, whereas the last choice helps with prediction performance. Also, we use 300d GloVe word vectors

Table 1: Evaluation results of various models on **Dev** set (including ablation study)

	Our Dev		Original Score	
Model	**F1**	**EM**	**F1**	**EM**
Coattn (only)[1]	75.01	63.72	-	-
DCN	75.15	64.06	75.6	65.4
DRCoattn (only)[1]	73.13	62.81	-	-
FusionNet	**75.26**	**65.08**	83.6	75.3
FusionNet (w/ RL)	75.02	64.58	-	-
Ensemble[2]	**78.34**	**68.41**	-	-
Ensemble[2](w/o smart span)	76.92	67.09	-	-
Baseline[3](100d)	43.59	34.63	-	-
Baseline[3](300d)	47.25	37.80	-	-

[1] indicate using encoder only (with baseline decoder).
[2] ensemble model includes Coattn, DCN, DRCoattn, and FusionNet. [3] ablation study on GloVe embeddings, either 100d or 300d vectors.

pretrained on the 840B Common Crawl corpus [5] for all model inputs (no more extra features), and use Adam [8] optimizer with a learning rate of $1e-3$ (and fine tuning with $1e-4$).

For DCN and coattention layers, we use a hidden size of 200 for all neural units. Sentinel vectors are randomly initialized and optimized during training. For the decoder, we set the number of fixed iterations to 4 and use a $maxout$ pool size of 16, dropout rate of 0.2, and batch size of 32.

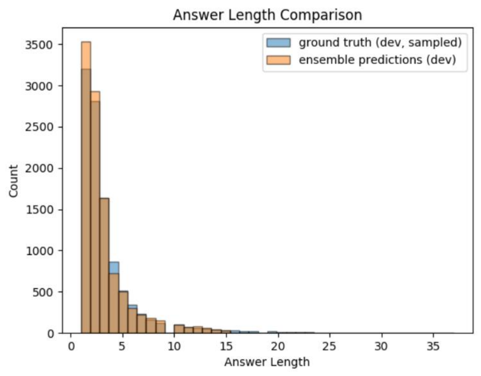
For FusionNet, we use a dropout rate of 0.3 (fine tuning with 0.4), and for LSTMs we use the variational dropout scheme in [9]. The model is trained with batch size of 128. We also use a different hidden size of 125 as specified by the paper.

3.2 Main results

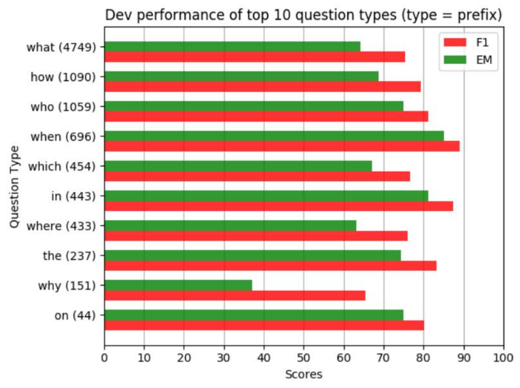
Table 1 summarizes our main results (F1/EM scores) on the official dev set. *The test set scores are included in the additional materials PDF that will accompany this report.* Our best single model is FusionNet which achieved $F1 = 75.3, EM = 65.1$, followed by full DCN (encoder + decoder) with $F1 = 75.2, EM = 64.1$,. Our ensemble model combines the outputs from the following 4 models: the coattention (only) model, the DCN, the DRCoattn (only) model and the FusionNet, and it achieved our best scores of **F1 = 78.3** and **EM = 68.4** on the Dev set. Since here we are combining 4 different neural architecture outputs and these outputs may reflect different internal distributional representations, we argue that the distributional ensemble strategy would outperform the majority vote and it indeed improves our Dev performance by more than 3%. As we will see later in error analysis (Section 3.3), however, the averaged distribution can lose certain long range dependency and information already learned in the single FusionNet. Additional reasons for why ensembling boosted performance significantly in our case include the fact that our ensemble consists of rather different architectures: alternative ways of using ensembling to achieve similar performance gains (e.g., original FusionNet ensemble) involve training a significantly larger number of models with the same architecture but changing up their initialization.

Unfortunately, our proposed policy gradient fine tuning for FusionNet ends up overfitting too much to the training set (our best observed training score: $F1 = 92\%, EM = 80\%$, and the FusionNet with RL tuning does not outperform the model using cross-entropy (CE) loss. We believe that the simple reward function in Equation (2) make not be sufficient to help the model generalize well, and we will leave the various ways of designing a better reward function (such as using $F1$ score as in [3, 7]) as a direction for future work. We can also see that smarter span and higher dimensional word embeddinsg do help improve the performance.

Compared with the original model performance of DCN and FusionNet in the reference paper, our implementation of DCN achieved similar scores on Dev set, but there is still a gap in performance for our FusionNet. We extrapolate that the additional (621-dimensional) features besides GloVe play an important role in FusionNet. i.e., since our embeddings were low-dimensional in relation to theirs, we were not able to get as much out of the FusionNet architecture as we would have liked; we feel as though this makes sense since the FusionNet architecture relies on having good history-of-word representations for words, which require the input word vectors to have lots of features to begin with. We will consider adding those extra features in our future work.



(a) Ensemble: Dev answer length



(b) Ensemble: Dev F1/EM

Figure 2: (a) Dev answer length comparison between our best ensemble model and the ground truth For dev, we sampled one of the valid answers for ea. question to ensure that the counts are comparable. (b) Dev F1/EM scores of the ensemble for the top 10 question types.

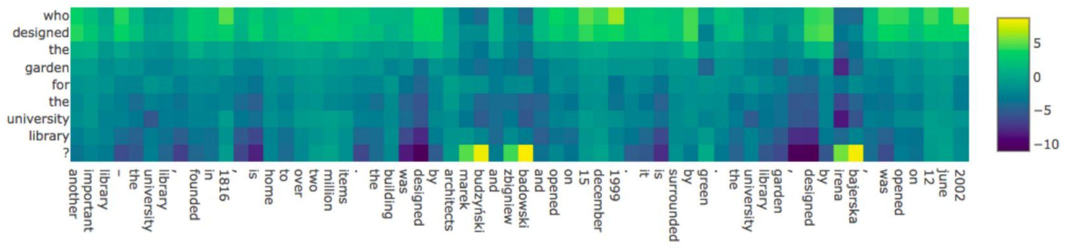
Figure 2 shows the performance analysis of our ensemble model. By utilizing the smarter span mechanism, the answer length histogram closely matches that of the ground truth on the Dev set except that our ensemble model tends to predict shorter answers. Figure 2 (b) summarizes the performance F1/EM for the top 10 question types. The model performs the best for the "When" type question, which mostly requires only a short answer, while it performs the worst for "Why" questions. Typically, "Why" type questions require longer answers and broader range of dependencies needs to be captured by the model; these are potential reasons for why this type of question is hard for our model to answer.

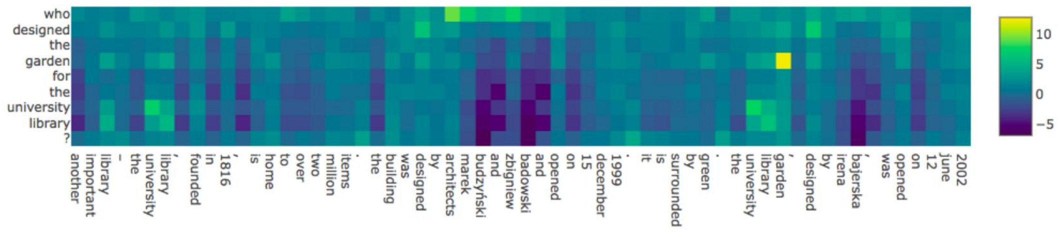
3.3 Dev Set Examples and Error Analysis (DCN, FusionNet and Ensemble)

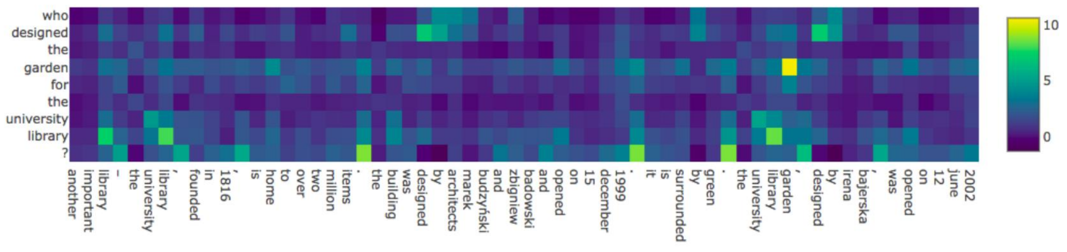
3.3.1 Long range dependency

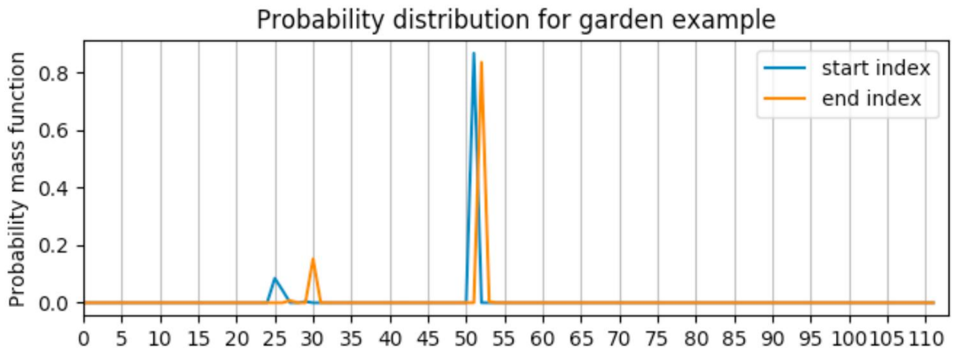
Context: kenya ’ s armed forces , like many government institutions in the country , have been tainted by corruption allegations . because the operations of the armed forces have been traditionally cloaked by the ubiquitous blanket of “ state security ” , the corruption has been less in public view , and thus less subject to public scrutiny and notoriety . this has changed recently . in what are by kenyan standards unprecedented revelations , in 2010 , credible claims of corruption were made with regard to recruitment and procurement of armoured personnel carriers . further , the wisdom and prudence of certain decisions of procurement have been publicly questioned .
Question: why has the corruption not be in the public view?
Answers: *DCN:* because the operations of the armed forces; *FusionNet:* because the operations of the armed forces have been traditionally cloaked; *Ensemble:* because the operations of the armed forces; *Ground Truth:* [’Because the operations of the armed forces have been traditionally cloaked by the ubiquitous blanket of “state security”’, ’state security’, ’state security’]

Analysis: This "Why" type question requires a fairly long answer and a long range dependency. As pointed out by the authors in [4], the self-boosted attention mechanism indeed helps here for capturing long range information, and FusionNet generates a better answer (higher F1 score) than the DCN model. Interestingly, after combining the distributions, the ensemble model loses some of the long range information and tends to predict a shorter answer. Although this may be an issue for answering "Why" type questions, the distributional averaged ensemble can help reduce variance and in general improve the overall performance.


(a) Low-level attention: S^l scoring matrix


(b) High-level attention: S^h scoring matrix


(c) Understanding-level attention: S^u scoring matrix



(d) Start and End distributions from decoder

Figure 3: **FusionNet multilevel attention visualization.** (a)-(c): multilevel summarization/similarity matrices from encoder (Section 2.3.2). (d) Final output start and end position distributions from decoder. Note that even though the pmf is displayed as continuous for nice visualization, even though the distribution is actually discrete.

3.3.2 Inaccurate attention position

Context: the broncos ' defense ranked first in the nfl yards allowed (4,530) for the first time in franchise history , and fourth in points allowed (296) . defensive ends derek wolfe and malik jackson each had 5.5 sacks . pro bowl linebacker von miller led the team with 11 sacks , forced four fumbles , and recovered three . linebacker demarcus ware was selected to play in the pro bowl for the ninth time in his career , ranking second on the team with 7.5 sacks . linebacker brandon marshall led the team in total tackles with 109 , while danny trevathan ranked second with 102 . cornerbacks aqib talib (three interceptions) and chris harris , jr. (two interceptions) were the other two pro bowl selections from the defense .

Questions: how many yards did the broncos ' defense give up ?
Answers: *DCN:* nfl yards; *FusionNet:* 4,530; *Ensemble:* 4,530; *Ground Truth:* '4,530'.

Analysis: For this question, the DCN model attends to a wrong position (shifted) in the context, while FusionNet is able to pinpoint the correct answer possibly owning to its consideration of multi-level information. When averaging as part of our ensemble strategy, the FusionNet contributes more than DCN here (FusionNet is fairly confident in its answer) and the ensemble model can still find the correct answer. (*Note:A couple of more Dev set examples are included in Appendix C in the supplementary material.*)

3.4 FusionNet: Multilevel Attention Visualization

We now analyze the predictive ability of the best-performing model in our ensemble, the FusionNet, by examining an example in significant detail and reporting what we observe with respect to the various levels of attention employed in the FusionNet model. Consider the following example in our Dev set: **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 . (rest of tokens omitted for brevity)
Question: who designed the garden for the university library ?

The correct answer is the single answer span "irena bajerska" (all crowdworkers picked this span), which appears in positions $51, 52$ of the context (0-indexed). Our FusionNet is able to correctly predict this answer span (and, as an aside, the ensemble is as well).

The steps it takes in order to come up with this prediction is quite impressive: recall from the discussion in Section 2.3.2 that the prediction of FusionNet architecture depends on the output of multiple attention layers that focus on different levels of understanding: from low-level to high-level understanding. Each layer has a corresponding affinity/similarity matrix (S^l, S^h, S^u) that is indicative of the relationship between context word representations and query word representations.

In Figure 3(a), we can see that the low-level attention layer places high attention scores on the tokens "marek budzyński", "zbigniew badowski", and "irena bajerska". This is interesting because all of these tokens correspond to names, and these tokens together cover all of the names within the context. Furthermore, all of these names correspond to architect designers, and we can see that we are not able to use the information from the associated S^l scoring matrix alone to decide among which of the three names to pick to answer the question. Very frequently questions that start with 'who' are answered with the name of a person, and this low-level layer is able to pick up on that relationship.

In Figure 3(b), the high-level attention gives a really high score for 'garden' in the context with 'garden' in the question. Because we know that this model is able to output the correct name out of the three names, we can see that the nearby position of 'garden' to the correct name is a factor that could help the network make the correct prediction. We also observe that this layer in general reserves giving very high scores (indicated by a 'yellow' cell) to words in the context that also appear in the question.

Lastly, Figure 3(d) shows that how the attention output from the encoder is used to construct the probability distribution for the predicted start and end indices. It clearly shows that there is a lot of probability mass for start and end indices 51 and 52, and it also means that the model is very confident in predicting the correct answer "irena bajerska". We can also observe some non-zero probability mass assigned to context indices corresponding to the alternative names, likely as a result of the high attention scores from the low-level attention. Combining with higher level fusion, the model is able to assign probability masses correctly. This is consistent with the results-supported claim (that the FusionNet authors made) about the necessity to capture full information in the context or the question in order to have a complete information comprehension when answering a question.

4 Conclusion

In this paper, we re-implement several recent attention-based architectures for machine reading comprehension on SQuAD. We are able to match the original reference Dev score for our implementation of DCN, but with less features, our version of FusionNet does not perform as well as the reference model. Thus, adding more features, e.g. CoVe embedding and POS, NER tags, will be our next step to improving our model. Our distributional ensemble strategy boosts the Dev performance by more than 3% and our best ensemble model achieved a score of $F1 = 78.3, EM = 68.4$. Another future direction would be extending our model that incorporates RL, by designing better reward functions for policy-gradient fine tuning and experimenting with more advanced policy gradient methods.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[2] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

[3] Caiming Xiong, Victor Zhong, and Richard Socher. Dcn+: Mixed objective and deep residual coattention for question answering. *arXiv preprint arXiv:1711.00106*, 2017.

[4] Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *arXiv preprint arXiv:1711.07341*, 2017.

[5] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[6] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[7] Minghao Hu, Yuxing Peng, and Xipeng Qiu. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*, 2017.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.