# Applying Transformer-XL to Q&A

**Sam Xu***
Department of Electrical Engineering
Stanford University
samx@stanford.edu

## Abstract

In this paper, we first re-implement QANet [1], a architecture highly inspired by the transformer model [2]. Then by making adjustments to incorporate elements of Transformer-XL and other high performing SQuAD models, we were able to achieve a modest performance gain on SQuAD 2.0. Overall, we were able to achieve a final EM and F1 test score of 64.379 and 68.108 [None PCE]. In order to make the features of Transformer-XL compatible with the task of Q&A, we shared memory between question passage pairs.

## 1 Introduction

Question and answering is a machine comprehension task in which a passage and a question are provided to a machine. It is the task of the machine to answer the question using the information of the passage. The recent release of SQuAD 2.0 has expanded the task by including unanswerable questions - which rendered many of the past high performing Q&A models ineffective.

QANet is a machine reading and question answering model released in 2018 that exclusively use convolutions and the self-attention mechanisms based on the Transformer [1]. On SQuAD 1.1, the authors of QANet achieved a F1 score 84.6; however, on the SQuAD 2.0 database, most reimplementations of QANet on the leaderboard were only able to achieved F1 scores around 68. To improve upon the works of QANet, this project looks at Transformer-XL [3], a attentive language model also based on the Transformer architecture.

## 2 Related Works

A notable amount of work has been done since the release of QANet and SQuAd 2.0. The works done in Transformer-XL explores the limits of fixed-length context in the setting of language modeling. To overcome this limitation, the Transformer-XL stores the past two hidden-states of its encoder units in a global cache. The access to these past hidden-states allows the architecture to capture segment-level recurrence over a long period of time. Other works have also looked at the task of detecting an "no answer" probability for unanswerable cases, and verifying whether the produced answer is valid under the context of the passage [4]. Additional works have looked at the capability of fused representation by concatenating both the question and passage, with which attention and fusion are conducted across layers on both questions and paragraphs [5].

## 3 Model Architecture

**Our Approach** I began this probject by implementing QANet. Then I added the state caching mechanism and relative positional encodings from the Transformer-XL paper to the encoder units of

*For this project, I viewed and used code from the following implementations of Transformer-XL: https://github.com/kimiyoung/transformer-xl https://github.com/huggingface/pytorch-pretrained-BERT

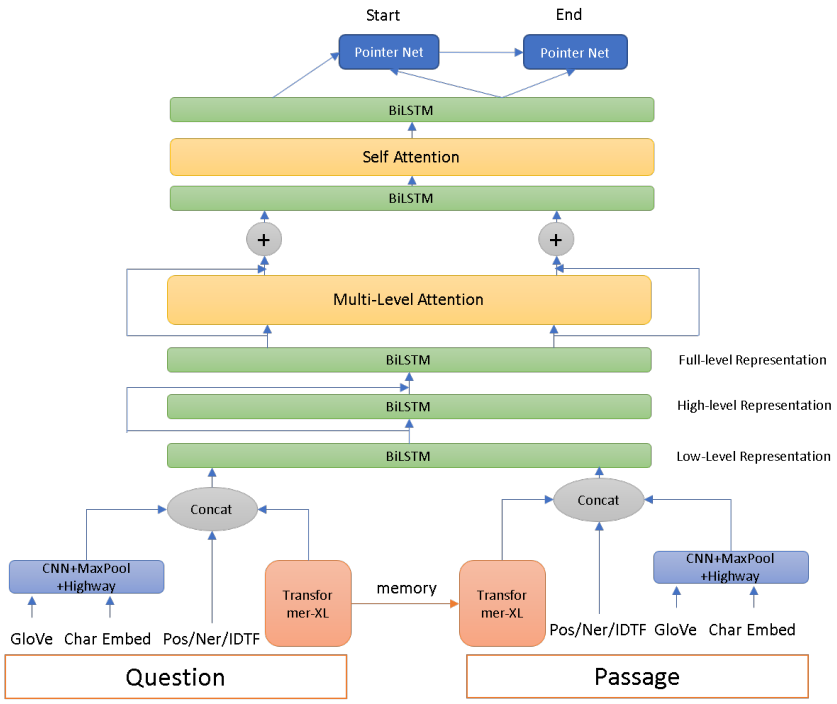


Figure 1: Final Model Architecture.

this architecture; however, after some training and hyperparameter searching, we failed to achieve any improvements in performance with the changes we made. We realized that while transformer-XL is useful where long-term dependency is involved, such as language modeling, applications like question answering doesn't have any tangible benefit from the past hidden states. For example, the hidden-state from the previous question/passage doesn't provide any context for the current question/passage. However, inspired by the demostration of recent developments in NLP that language models can can also serve as a mechanism for downstream tasks [6][7], my final model uses the hidden-state of Transformer-XL to capture the contextual information in the early stages of my pipeline.
Additionally, after having issues with exploding hidden-states and found that QANet to be high susceptible to changes in hyperparameters and the dimensions of our input, we chose to stray from QANet's usage of multiple depthwise separable convolutions, and instead use BiLSTMs in example of more recent models.

### 3.1 Preprocessing

In addition to the provided GLoVE vectors, using spaCy, we extract POS (Parts-of-peech), NER (Nearest entity recognition), exact match, lower-case match, lemma match, and term frequency–inverse document frequency (tf-idf) to be used as additional embedding vectors.

### 3.2 Embedding Layer

We reuse the character-level embedding from QANet, which utilizes the standard character embedding technique in which tokens are turned into vector representation, truncated or padded, sent through a convolution, max-pool, and finally a relu-activation layer. The character embedding is then concatenated with the GLoVE word-vector before sent through a highway-activation layer like in assignment 5 to produce our output vector. This output vector is then concatenated with all the additional embeddings we extracted through preprocessing.
Next, we used hugginface's TransformerXL implementation pretrained on the wikitext103 database from the WikiText Long Term Dependency Language Modeling Dataset. We first obtain the memory output and hidden state of the TransformerXL for the question. Next, we re-use the memory-cell of the question on our subsequent call to the passage so that it can attend the context from the

question. The final hidden state for the question and passage are then concatenated to their respective embeddings to obtain our question representation $Q$ and passage representation $P$.

### 3.3 Contextual Encoding Layer

Let $R = [P, Q]$ be our joint representation of the question and passsage, following examples from recent works [4][5][8], we place a two-layered Bi-LSTM on top of our embeddings and a additional Bi-LSTM on top of their hidden states to yield our contextual representations.

$$\begin{aligned} H_l &= \text{BiLSTM}(R) \\ H_h &= \text{BiLSTM}(H_l) \\ H_f &= \text{BiLSTM}([H_l, H_h]) \end{aligned} \tag{1}$$

Where $H_l, H_h, H_f$ represents the low-level, high-level and full-level semantic information, and $H = [H_l, H_h, H_f]$ their full joint representation.

### 3.4 Multi Attention Layer

Following examples from FusionNet [8], we take the bidirectional attention for each of our three representations with their respective passage and question segment:

$$S = (ReLU(W_p H_p))^T (ReLU(W_q H_q))$$

This will then give our output

$$\begin{aligned} \hat{H_p} &= H_q \times S \\ \hat{H_q} &= H_p \times S^T \end{aligned} \tag{2}$$

Taking the bidirectional attention of all three levels of information and concatenating them will give us our encoded information $\hat{H}$. We will then pass $H$ and $\hat{H}$ through another layers of BiLSTM and Self-Attention to obtain our final output state $O$ where $O_p$ and $O_q$ are the respective passage and question segments.

### 3.5 Prediction Layer

Following examples from Read+Verify [4], we utilize a classic pointer network to obtain our probability of the answer boundaries from the passage when the question is answerable [2016]. We take our final question representation $O_q$ to calculate a fixed-dimension dense vector $t$:

$$\begin{aligned} t &= \sum_j \frac{\exp\left(W_t^T o_j^Q\right)}{\sum_k \exp W^T o_k^Q} o_j^Q \\ \alpha, \beta &= \text{pointer\_network}(O^p, t) \end{aligned} \tag{3}$$

where $\alpha$ and $\beta$ the scores for start and end indexes for the answer. Next we calculate the verify probability of whether or not the question is answerable:

$$\begin{aligned} s_i &= \sum_i \alpha_i * o_i^P \\ s_j &= \sum_i \beta_i * o_i^P \\ F &= [t, s_i, s_j] \end{aligned} \tag{4}$$

We then pass $F$ through a feedforward linear network with a sigmoid activation to obtain our probability.

## 4 Discussion and Error Analysis

In this section, we will provide an overview of the tests we did across our models - in particular, we will discuss several of the extensions that were unsucessful and consequently not included in our final model, and their results.

Table 1: Model Performances (dev)

| **Model (dev)** | **EM** | **F1** |
|---|---|---|
| Baseline | 55.6 | 58.9 |
| QANet | 64.2 | 67.9 |
| QANet with Transformer-XL encoding | 64.4 | 68.0 |
| Final Model with Transformer-XL embedding | 66.3 | 70.2 |

Table 2: Final Model Performance (dev)

| Configuration | EM | F1 |
|---|---|---|
| Only GLoVE vectors | 63.2 | 66.8 |
| + Character Embedding | 64.5 | 68.4 |
| + Increased Dropout + Transformer-XL + Additional Embeddings | 66.3 | 70.2 |

We can see from table 1 that we gained the majority of our performance through our initial implementation of QANet. To adapt the Transformer-XL encoding mechanism, we adapted code from the official Transformer-XL github `https://github.com/kimiyoung/transformer-xl` and inserted caching variables for each block of encoder units in QANet; however, we found that the model performed nearly identical with and without the changes.

For the next step, we experimented by adding additional embeddings to our model using the final hidden-state of the transformer model pretrained on wikitext103 from `https://github.com/huggingface/pytorch-pretrained-BERT`, in addition to various other preprocessed embeddings (POS, NER, tf-idf...). However, we found that the hidden-states in our QANet implementation would blow up given our additional embeddings. After some analysis, we determined the culprit to be the convolutional units within the encoders. Consequently, for our final model, inspired by FusionNet [8], we switched to the BiLSTM units for our encoding. The additional embeddings and the change in architecture finally provided us a 2.0 boost to our F1 score.

However; the removal of the convolutional units came with consequences. On our QANet implementation, one epoch across the training dataset took approximately 20 minutes on a RTX 2080ti. The final model using Transformer-XL embeddings took approxmiately 4 hours for one epoch. This massive reduction in speed is likely caused by the removal of convolutional parallelism and the addition of the Transformer-XL.

For our final model, we used the AdaMax optimizer with a starting learning rate of 0.002. We noticed that all of our models were overfitting and performing much worse on the test-set than our training and dev-set, so we added dropout to our embeddings and increased our dropout probability from 0.1 to 0.3 on our final model. Ultimately, we achieve a EM and F1 test score of 64.379 and 68.108

### 4.1 Analysis on the Impact of Different Components

In this section, we provide the F1 score we obtained in our final model as I added each of the components.

Table 2 shows that the gain in our final performance is almost entirely leveraged from the additional embeddings and the increase in dropout. Due to fear of overfitting and the long training time, I decide to not increase the size of our embedding or hidden size during my hyperparameter search; however, I have tried increasing the embedding size, hidden size, and number of encoders with the vanilla QANet model, and found that it indeed decreases my performance.

### 4.2 Error Analysis on Incorrect Output

Consider this passage from the test set:

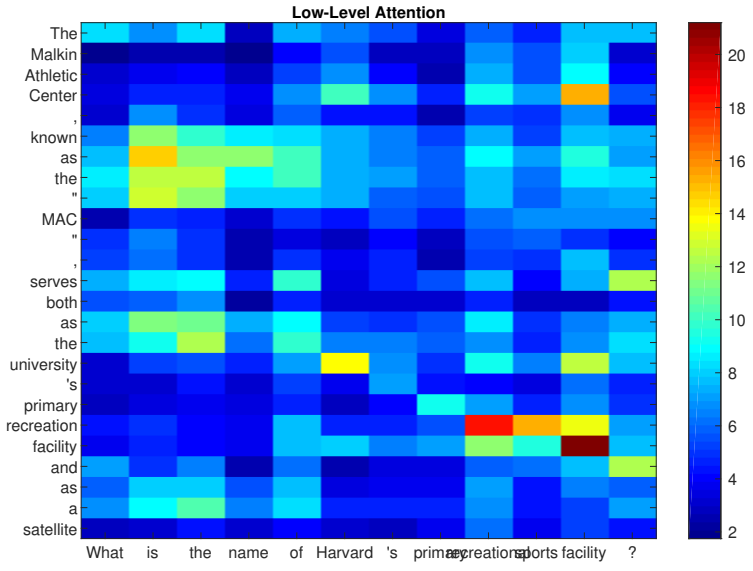


Figure 2: Low-level Attention visualization between the question and passage

"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 ..."

And the question for the following task:

"What is the name of Harvard's primary recreational sport's facility?"

We know directly by simply reading the question that the answer to the question exists, somewhere along the line of "MAC" or "The Malkin Athletic Center". However, my model incorrectly predicts that there was no answer for this passage.

To analyze why my model predicted this, I visualized the similarity matrix in my multi-attention layer for the span of words that I felt like would have been the correct answer. A larger (darker/redder) value in the similarity matrix means a stronger mutual attention between the passage word and question word.

We can see from figure 4.2 that the low-level representation were able capture identical words and words sharing similar context, such as the word facility with facility, and sports is highly connected to recreation. "facility" shares similar word context with "center", and "is" shares similar context with "as". However, as indicated by the fact that its a low level representation, this matrix currently fails to capture any relationship between the question and passage.

From figure 4.2, we see that the high-level similarity matrix remains limited to identical words and similar context words, but we also began to see much higher attention is being paid to the word pair "is known", a relationship possibly captured by our language model embedding.

Finally, from figure 4.2, we see the full level representation between the question and passage. Here, we see something very interesting. The matrix was able to capture the segment of the passage that was the answer to the question, where the segment: "Known as the MAC", is highly attended to by the word "name" from the question, however, the actual answer, **"MAC"**, was entirely left out and

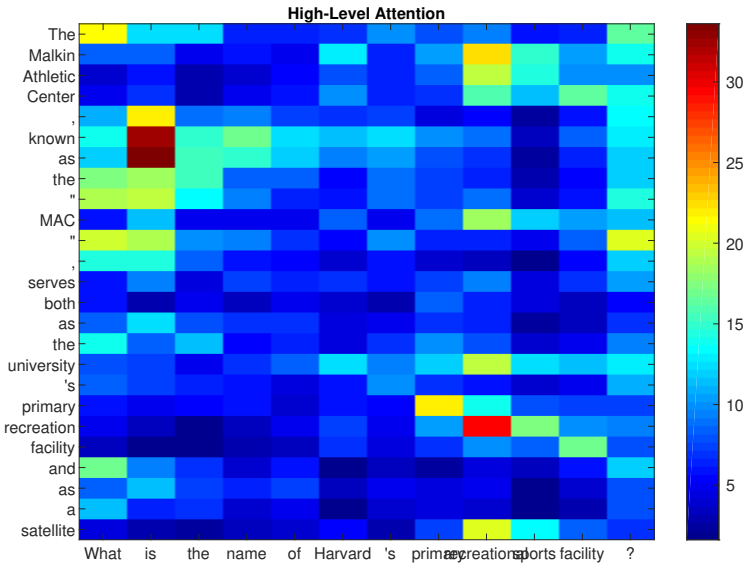


Figure 3: High-level Attention visualization between the question and passage

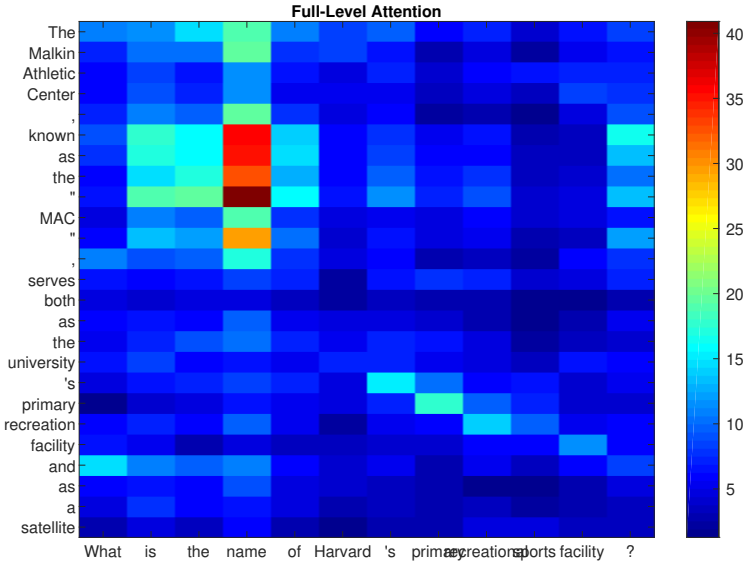


Figure 4: Full-level Attention visualization between the question and passage

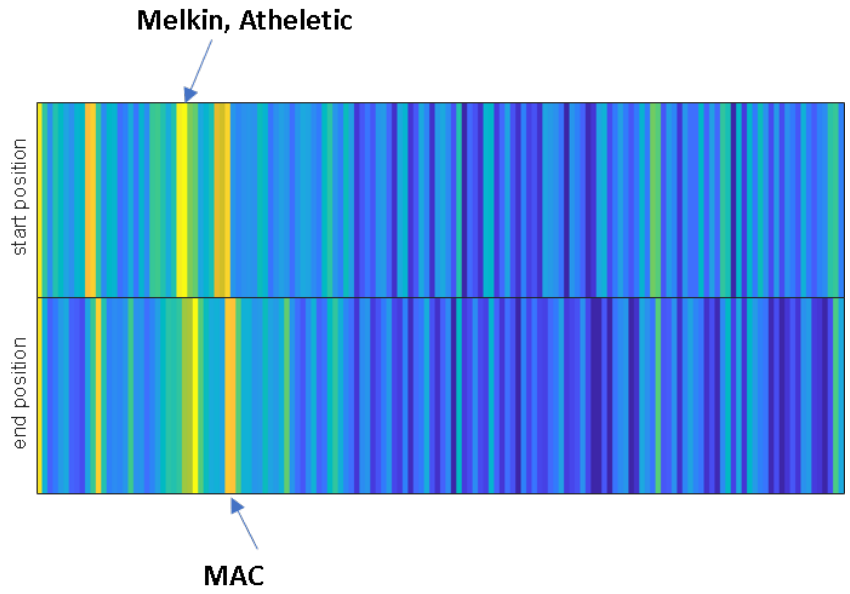


Figure 5: Start/End boundry Probability Distribution for the question: What is the name of Harvard's primary recreational sport's facility? "

unattended.

We can see from the similarity matrix that because the actual answer **"MAC"**, is left out, the model was unable to determine the answer at the end. We hypothesis that this is caused by a few reasons, one, MAC is a unique acryonym that is unlikely to appear anywhere else with the same meaning outside of this excerpt. Consequently, none of the embeddings we use, GLoVe, character, Transformer Xl's language model, Pos/Ner/td-idtf...etc were able to provide any context about the word. Therefore, we see that even though everything else about the problem points to the fact that "MAC" was the answer, the fact that the model has absolultely no idea what a "MAC" is caused it to become unattended.

To confirm our hypothesis, we visualize the final start and end probabilities for our passage 4.2. We can see that from the probability distribution that the task was correctly modeled, and both "Melkin Atheletic Center" and "MAC" has a higher probability than the background logits. However, it was very uncertain about the answer, and we see that other unrelated parts of the passage were also given emphasis. Consequently, the model did not seem to have high confidence on the answer and return that were no answer.

## 5 Conclusion

For this project, I have implemented a final model that incorporates the features of Transformer-XL, QANet, and elements of various other successful SQuAD projects such as FusionNet and Read+Verify to achieve a competitive result on the test set. As error analysis, we visalized the similarity matrix of our multi-level semantic representation. We felt that these visualizations provides an intuitive explanation to the mechanism of this machine learning model. By analyzing these attention mechanisms, we observe that our model was able to capture the relationship between the problem and passage with our embeddings and multi-level representation. Furthur work needs to be done to determine the effects of individual embedding components that was added, and possibly fine tuning the Transformer-XL architecture to the vocabulary of Q&A.

## References


[1] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, Quoc V. Le . (2018) *QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension.* abs/1804.09541.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: *Attention Is All You Need.* http://arxiv.org/abs/1706.03762

[3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, Ruslan Salakhutdinov . (2019) *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context.* http://arxiv.org/abs/1901.02860

[4] Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, Ming Zhou . (2018) *Read + Verify: Machine Reading Comprehension with Unanswerable Questions.* arxiv.org/abs/1808.05759

[5] Wei Wang, Ming Yan, Chen Wu. . (2018) *Multi-Granularity Hierarchical Attention Fusion Networksfor Reading Comprehension and Question Answering.* http://www.aclweb.org/anthology/P18-1158

[6] Peters, M., et. al. (2017) *Deep contextualized word representations*

[7] Devlin, J., et. al. (2018) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

[8] Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, Weizhu Chen: . (2017) *FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension.* arxiv.org/abs/1711.07341

[9] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi: . (2016) *Bidirectional Attention Flow for Machine Comprehension.* arxiv.org/abs/1611.01603

[10] Fu Sun, Linyang Li, Xipeng Qiu, Yang Liu: (2018) *U-Net: Machine Reading Comprehension with Unanswerable Questions.* arxiv.org/abs/1810.06638

[11] Shuohang Wang, Jing Jiang: (2016) *Machine Comprehension Using Match-LSTM and Answer Pointer.* arxiv.org/abs/1608.07905