SQUAD Challenge : A Hybrid Model for Question Answering

Hacer Umay Geyikci
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
umay@stanford.edu

Onur Cezmi Mutlu
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
cezmi@stanford.edu

Abstract

In this paper, as a part of CS224n course project, we present an architecture that can be used in the SQUAD (Stanford Question Answering Dataset) question answering challenge. Considering several state-of-the-art approaches we propose a hybrid structure, with the purpose of achieving a better whole than individuals. We achieve 75/62% F1/EM score using this approach.

1 Introduction

Reading comprehension, as a subproblem of Natural Language Processing, has always been a critical challenge in the advancement of artificial intelligence. It is one of the main elements to achieve AI singularity since it defines the ability of a machine to communicate with the outer world without a prescribed pattern. In search for improvements in the field, SQUAD (Stanford Question Answering Dataset) dataset [1] is a popular environment to test the new approaches.

As the final project to the CS224n - Natural Language Processing with Deep Learning course, we chose to tackle with the SQUAD question answering challenge. In [2] state-of-the-art achieves F1/EM score of 89.28/82.48% whereas the human performance in the same challenge is 91.22/82.3% [1]. Being a beginner in such complicated structures and due to lack of time, we did not consider re-implementing a very complicated state-of-art structure. Instead, in this project we combine two previously successful methods and try to create a better model by identifying missing part of one and trying to overcome that difficulty with the ideas from the other.

The models that we selected for implementation are the architectures mentioned in "Bidirectional Attention Flow for Machine Comprehension" [2] and "Reading Wikipedia to Answer Open-Domain Questions" [3]. In those papers the scores obtained in same challenge were 81/73% and 79/69%, respectively.

2 Dataset & Problem Definition

In the project we are using SQUAD given in [1]. This large dataset consists of 100.000+ question-answer pairs from 500+ articles from Wikipedia, all created by human effort. The data is in the form of context-question-answer triplets where context is a couple of sentences, question is a sentence and answer is the start and ending locations of the answer in the context. In Figure 1 we provide some analysis on data. As can be examined in the figure the answer length tend to be short whereas the context lengths are gathered around length 150.

As the name of the challenge suggests, our task is to find the correct answer to a question given the context. What is meant by "answer" is the starting and ending locations of the answer phrase within the context. To write down this formally, we say that we have the context words represented by the

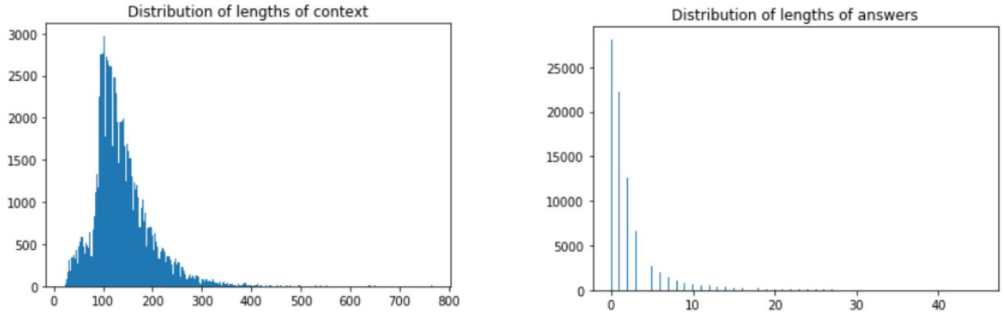

Figure 1: Context(left) and answer(right) length distributions

sequence $\mathbf{w} = \{w_1, w_2, ...w_N\}$ and question words are by $\mathbf{v} = \{v_1, v_2, ...v_M\}$. If we denote the start and end locations of the answer in the context by $\mathbf{a} = (a_{start}, a_{end})$ then the problem reduces to finding a relation between these identities, i.e. $f(\mathbf{w}, \mathbf{v}) = \mathbf{a}$. With the purpose of finding such a function people have been developing complex neural network architectures and the hybrid model that we implemented is one such approach.

3 Our Model

3.1 Architecture

Our model architecture can be investigated as several subsystems. In order to maintain a comprehensive understanding of the structure, we first provide the overall picture below in Figure 2.

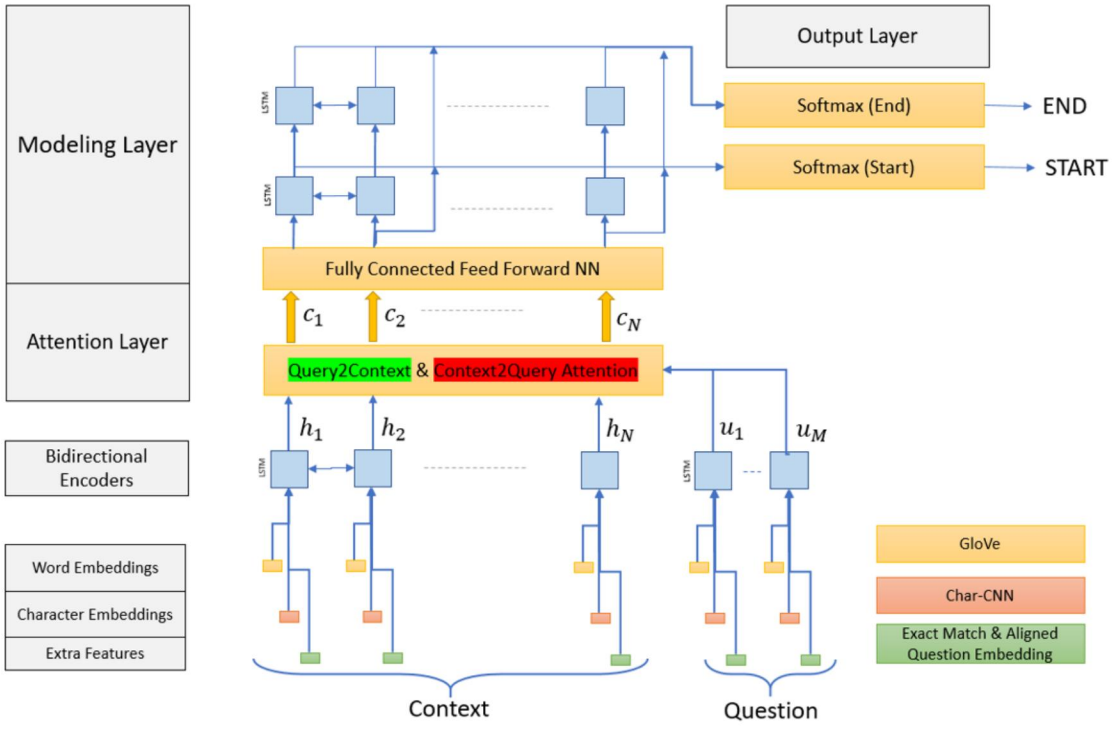

Figure 2: Model Architecture

3.1.1 Embeddings

We have three different types of representation for words in both context and question. We concatenate the following three to represent a word:

- *GloVe word embeddings*: We used 100 dimentional embeddings. We represent word embeddings of context words by $x_1, x_2...x_N \in \mathbb{R}^d$ and the question words are by $y_1, y_2...y_M \in \mathbb{R}^d$.
- *Character embeddings*: Obtained from 1-D character level CNN. We created 1-D trainable character vectors and fed them to CNN. In CNN we have a number of filters followed by ReLU non-linearity, output of which is max-pooled to obtain a fixed size vector. This structure is directly inherited from BiDAF[2]
- *Exact match (EM) and Aligned Question Embedding (AE) features*: These extra features are inherited from DrQA[3]. Ablative analysis in the paper states that usage of only these two features is enough to obtain a near optimal operation (optimal with respect to their maximum). In order to create the binary EM feature, we simply check whether a context word is present at the question. For AE feature we first calculate calculate weights $a_{i,j} = softmax(\alpha(x_i)\alpha(Q)]_j$ where $\alpha(.)$ is a single dense layer with ReLU non-linearity. Then the AE feature for the i^{th} context word becomes $AE_i = \sum_j a_{i,j} y_j$

We therefore create two embedding matrices $\tilde{X} \in \mathbb{R}^{201 \times N}$ and $\tilde{Y} \in \mathbb{R}^{201 \times M}$ where N and M are maximum acceptable context and question lengths.

3.1.2 Encoders

In order to capture temporal interactions between words we use single layer bidirectional LSTM networks [4]. With the hidden layer size of d in each direction, we obtain matrices $H \in \mathbb{R}^{2h \times N}$ and $U \in \mathbb{R}^{2h \times M}$, by concatenating vectors in both directions.

3.1.3 Attention

At this layer, we inherit the innovative bidirectional attention idea from [2]. In this layer a similarity matrix is calculated first which is defined as

$$S_{i,j} = {w_{sim}}^T [h_i; u_j; h_i \circ u_j]$$

where $w_{sim} \in \mathbb{R}^{6h}$ is a vector to be trained. Then the Context-to-Question(C2Q) attention outputs a_i are calculated as follows:

$$a_i = \sum_{j=1}^{M} softmax(S_{i,:})_j q_j \in \mathbb{R}^{2h}$$

Finally the Question-to-Context(Q2C) attention outputs are calculated as follows:

$$\beta = softmax(\max_j S_{:,j}) \in \mathbb{R}^N$$

$$c' = \sum_{j=1}^{N} \beta_i c_i \in \mathbb{R}^{2h}$$

After creating Q2C and C2Q outputs the following vectors are passed to the next level:

$$\mathbf{c_i} = [h_i; a_i; h_i \circ a_i; h_i \circ c'_i]$$

3.1.4 Modeling Layer

Up until this part of the model we created features that summarizes the relations between question and context words and now the modeling layer will use this ”information” and pass it to the final layer.

We start this layer with a fully connected feed forward NN. Our purpose here is to compress the information being passed from attention layer. The motivation for compression is two-fold. First, through compression we get rid of redundancies as furthermore through a slightly lossy compression we manage to prevent over-fitting. So, basically this network acts as a regularizer. We do not provide an information theoretic background to support our choice of dimensionality reduction and it depends mainly on heuristics. Second motivation for this layer is to reduce the dimensionality to obtain a faster training system. These vectors will define dimensions of matrices in the following layers and a reduction in size of this vector reduces the number of parameters to be trained significantly. We reduce the vector dimension to 2h from 8h and denote the resultant matrix by $P \in \mathbb{R}^{2h \times N}$.

Now the length 2h vectors are fed to a bidirectional LSTM network. Choosing the hidden layer size of the network to be h, we denote its outputs by the matrix $M \in \mathbb{R}^{2h}$, again both directions are concatenated. The matrix M is then fed to another bidirectional LSTM network with the same hidden layer size and we obtain the matrix $M^2 \in \mathbb{R}^{2h \times N}$

3.1.5 Output Layer

This is the final layer in our design. We have two pipelines in parallel here, one for start location and the other for end location. We first create the matrices $M_start = [M; C']$ and $M_end = [M^2; C']$ using the previous outputs. These matrices are fed to a linear projection layer to convert them into length N vectors. This projection layer is to be trained. The resultant vectors are then passed through softmax functions and argmax'ed to obtain the predictions.

3.2 Training

We used standard cross-entropy loss for both locations and summed them up to obtain main loss function to be used. We used AdamOptimizer with learning rate 0.001. Dropout parameter for all neurons in the architecture were chosen to be 0.25. In the light of the data analysis we have done in section 2, We examined the number of lengths larger than a value cases and found that we can use question_len of 25 and context_len of 400 without harming too many examples. That way we shrunk our model.

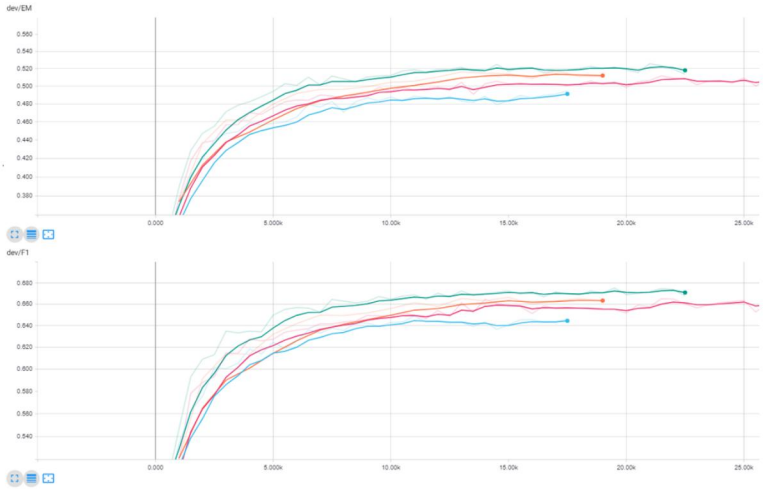

Figure 3: dev-EM(top) and dev-F1(bottom) scores of several architectures (Cyan : BiDAF Attention layer + output RNN layers - Pink : Cyan + Extra features - Orange : Parameter tuning on Pink - Green : Orange + CharCNN)

We used a hidden_size of 100 for all LSTM networks. The character embeddings we used were of length 20 which resulted from 100 filters inside the CNN.

Training sessions took around 10 hours on Nvidia GTX1060 6GB GPU. In Figure 3 we provide the training curves for several architecture types. As can be seen, each additional structure that we introduced, increased the score.

3.3 Evaluation & Analysis

When we performed the official evaluation on entire dev (dev1.1.json) dataset, our best model (Char-CNN) has resulted in 75/62% F1/EM score, However, due to high number of modifications we made on the batch generating code, we were not able to run it successfully on Codalab. We therefore published our second best model and obtained the test score of 72.19/61.77%.

In order to see the ups and downs of our final model we also investigated actual examples to see the behavior of the system. We provide some examples that we found interesting:

- – CONTEXT: on 7 january 1900 , tesla left colorado springs . [citation needed] his lab was torn down in 1904 , and its contents were sold two years later to satisfy a debt .
  – QUESTION: when did tesla depart from colorado springs ?
  – TRUE ANSWER: 1900
  – PREDICTED ANSWER: 7 january 1900
  – F1 SCORE ANSWER: 0.500
  – EM SCORE: False
  In this example we see that there are some flaws in the dataset itself. The answer that was predicted is actually better than the true answer in the dataset. Of course this will always be a challenge since the labeling depends on human effort.
- – CONTEXT: orientalism , as theorized by edward said , refers to how the west developed an imaginative geography of the east . this imaginative geography relies on an _essentializing_ discourse that represents neither the diversity nor the social reality of the east . rather , by _essentializing_ the east , this discourse uses the idea of place-based identities to create difference and distance between " we " the west and " them " the east , or " here " in the west and " there " in the east . this difference was particularly apparent in textual and visual works of early european studies of the orient that positioned the east as irrational and backward in opposition to the rational and progressive west . defining the east as a negative vision of itself , as its inferior , not only increased the west s sense of self , but also was a way of ordering the east and making it known to the west so that it could be dominated and controlled . the discourse of orientalism therefore served as an ideological justification of early western imperialism , as it formed a body of knowledge and ideas that rationalized social , cultural , political , and economic control of other territories .
  – QUESTION: the west saw the east as what ?
  – TRUE ANSWER: inferior
  – PREDICTED ANSWER: a negative vision of itself
  – F1 SCORE ANSWER: 0.000
  – EM SCORE: False
  This is a similar example, where the answer both the true answer and the prediction is correct. Since they are not the same the answer is marked incorrect.
- – CONTEXT: terra preta (black earth) , which is distributed over large areas in the amazon forest , is now widely accepted as a product of indigenous soil management . the development of this fertile soil allowed agriculture and silviculture in the previously hostile environment ; meaning that large portions of the amazon rainforest are probably the result of centuries of human management , rather than naturally occurring as has previously been supposed . in the region of the xingu tribe , remains of some of these large settlements in the middle of the amazon forest were found in 2003 by michael _heckenberger_ and colleagues of the university of florida . among those were evidence of roads , bridges and large plazas .

  - QUESTION: what type of soil is considered a product of soil management by indigenous peoples in the amazon forest ?
  - TRUE ANSWER: terra preta (black earth)
  - PREDICTED ANSWER: fertile soil
  - F1 SCORE ANSWER: 0.000
  - EM SCORE: False
    In this example we see that our model is confused by the question "what type?". It searches for a noun accompanied by an adjective however the actual thing that was meant to be asked was the name of the soil. We see that a deeper understanding is necessary.

-
  - CONTEXT: there were many religions practiced during the yuan dynasty , such as buddhism , islam , and christianity . the establishment of the yuan dynasty had dramatically increased the number of muslims in china . however , unlike the western khanates , the yuan dynasty never converted to islam . instead , kublai khan , the founder of the yuan dynasty , favored buddhism , especially the tibetan variants . as a result , tibetan buddhism was established as the de facto state religion . the top-level department and government agency known as the bureau of buddhist and tibetan affairs (_xuanzheng_ yuan) was set up in _khanbaliq_ (modern beijing) to supervise buddhist monks throughout the empire . since kublai khan only esteemed the sakya sect of tibetan buddhism , other religions became less important . he and his successors kept a sakya imperial preceptor (_dishi_) at court . before the end of the yuan dynasty , 14 leaders of the sakya sect had held the post of imperial preceptor , thereby enjoying special power . furthermore , mongol patronage of buddhism resulted in a number of monuments of buddhist art . mongolian buddhist translations , almost all from tibetan originals , began on a large scale after 1300 . many mongols of the upper class such as the jalayir and the _oronar_ nobles as well as the emperors also patronized confucian scholars and institutions . a considerable number of confucian and chinese historical works were translated into the mongolian language .
  - QUESTION: what was the yuan 's unofficial state religion ?
  - TRUE ANSWER: tibetan buddhism
  - PREDICTED ANSWER: buddhism , islam , and christianity . the establishment of the yuan dynasty had dramatically increased the number of muslims in china . however , unlike the western khanates , the yuan dynasty never converted to islam . instead , kublai khan , the founder of the yuan dynasty , favored buddhism , especially the tibetan variants . as a result , tibetan buddhism
  - F1 SCORE ANSWER: 0.091
  - EM SCORE: False
    This is a particularly interesting example. Model feels that "buddhism" is an important word. However, bot start and end pipelines focus on the same object and the system end up marking two seemingly most important points as answer and we end up in a very long prediction.

3.4 Conclusion

The experiments on the SQUAD database have shown that our model was successfully implemented. Our initial motivation was to combine two distinct ideas to melt them in a bowl and obtain a stronger one but we saw that a more detailed approach might be better since the combination was not better than either one. A better hyperparameter tuning might end up in better results for our model but in the limited project time these results were the best we could obtain.

We have a couple of future improvements in our model:

- Span representations at the output might result in a better outcome, since it both gets rid of end-before-start issue and brings a more comprehensive approach to the system. This idea is proposed in [5].
- Deeper networks might come in handy. The encoder level, modeling level and also char-level CNN might benefit from this upgrade since we apparently miss some indirect connections among the words.

Overall, this project has been a very informative research experience for us.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, SQuAD: 100,000 Questions for Machine Comprehension of Text, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.

[2] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," ICLR, 2017.

[3] D. Chen, A. Fisch, J. Weston, and A. Bordes, Reading Wikipedia to Answer Open-Domain Questions, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017.

[4] S. Hochreiter and J. Schmidhuber. "Long short-term memory," Neural Computation, 1997.

[5] Y. Yu, W. Zhang, K. Hasan, M. Yu, B. Xiang, and B. Zhou, "End-to-end answer chunk extraction and ranking for reading comprehension," arXiv preprint arXiv:1610.09996 ,2016.