BiEAF: An Bidirectional Enhanced Attention Flow Model for Question Answering Task

Yihan Yang
Student
Ulink College
Guangzhou, China
yihyang2369@ulinkcollege.com

***Abstract*—Question answering is an crucial module to construct an intelligent chatbot. Traditional question answering system utilizes various of deep learning models to answer the queries requested by users. While BiDAF modelled the interaction between context and query for the first time, which is a milestone attempt in this domain. However, BiDAF only takes the correlation across the sentences into account, but the association within the sentence is not considered. In this paper, based on BiDAF, we present a novel method to both utilize the inter-and-intra sentence interaction by deploying the proposed enhanced attention-flow layer. The experimental results on SQUAD dataset show that our method outperforms the baseline models in terms of both EM and F1 evaluation metrics, which proves our model effectiveness.**

Keywords-Deep learning, Neural network, Question answering

I. INTRODUCTION

Artificial intelligence (AI) chatbots are very popular in recent years, of which massive products are emerged both for commercial and personal users, including Apple Siri, Xiao Du from Baidu and Tay from Microsoft. With the development of science and technology, these AI chatbots benefit our lives in some ways and also, at the same time, researchers who try to go deep into this cutting-edge technology have been also involving in this domain in natural language processing community (NLP). In this research field, question answering (QA) is the key but is a very challenging research direction amongst the chatbot systems. Generally, when the user questions chatbot, QA system is activated trying to understand what the question means and find the most appropriate answer span from the given context. To be more specific, the QA system will search the whole contextual knowledge base and find the best fitted one. The system will find the span to the answer using user's questions. If there's not suitable answer in the context, the system will return none to the user or ask for more information to support the answering searching process. As shown in Figure 1, the user is asking that "What laws faced significantly opposition?" The QA system receive the query and retrieve it in the backend knowledge base and select the most matched page as our candidate context. Then the system is required to understanding the retrieved context combining with the given query and find the correct answer span from the context. We primarily pay our attention to step 4, in which the context and query are given, the model needs to understand the underlying semantic information and try to predict the proper span as a final answer that get back to the user.

There are a few previous work attempting to solve this problem. Wang et al. [1] used a gated attention-based recurrent networks to match the question and the context. Yu and Indurthi [2] built a Memory Augmented Neural Networks to analysis and reason over long documents. Yang et al. [4] investigates pre-trained model to boost the QA system capability.

978-1-6654-3829-2/21/$31.00 ©2021 IEEE
DOI 10.1109/ICISE-IE53922.2021.00086

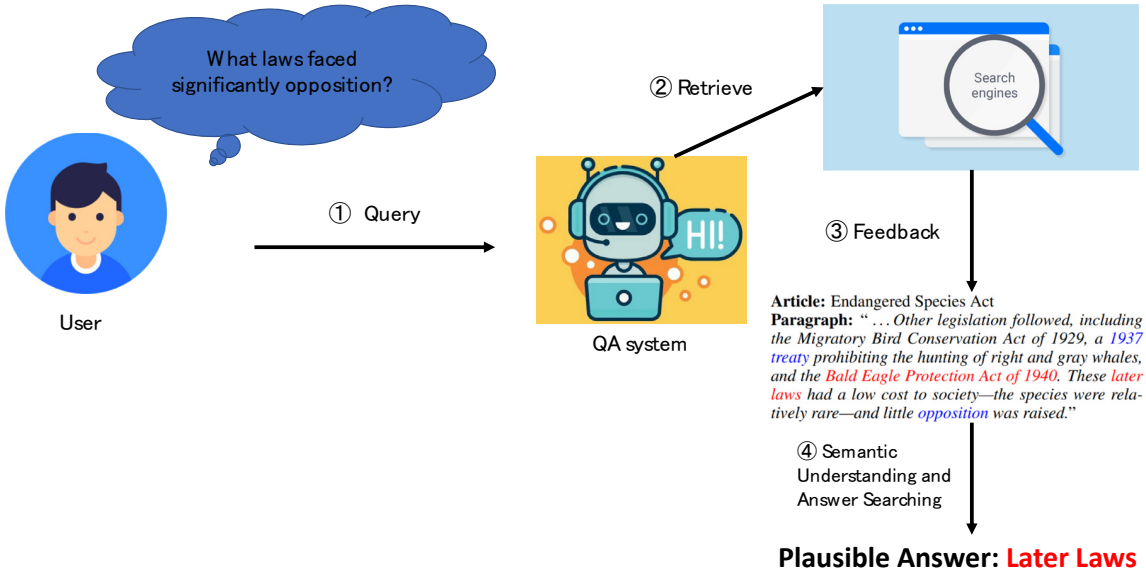

Figure 1. An example of the question answering workflow in SQUAD dataset.

In this paper, inspired by the baseline announced by Seo et al. [5] and Vaswani et al. [6], we proposed a novel QA model, which is based on an encoder-decoder frame and includes 3 modules. The first one, which is encoder layer, is capable of encoding the semantic information from the input text and also the content. Next is enhanced attention flow module. This module is based on attention mechanism. Information from context and query is calculated in different situations with different weights. They are calculated in the self-attention layer. In this way, more important words are assigned higher weights, so they can retain more semantic information in subsequent calculations. Prevent the model from inputting too much noise information, which gives low weight to unimportant words, so that the model can capture useful special vibration. The interaction between query and context is also modeled by enhanced attention flow layer. The final one is the decoder module. It can make use of two multi-classifiers to calculate a start index and an end index. Our answer is the token span between the start index and the end index.

At last, we did an experiment using a public data set SQUAD. Compare to normal models, ours achieve very high score on EM and F1 scores that are higher than BiDAF model, which strongly proves our model effectiveness.

II. Related Work

Question answering is a crucial component for the intelligent chatbot system. There are lots of previous attempts focusing on advancing the research progress in the community. We investigates some of the state-of-the-art methods as listed below.

Wang et al. [1] used a gated attention-based recurrent networks to match the question and the context, thus gaining a question-aware passage representation. By proposing a self-matching attention mechanism, the representation is refined by matching the context with itself. At last, they encoded information of the context and used a pointer network to locate the position of the answers, which lead to an improvement of model performance.

However, as the context becomes longer, which causes a fall in accuracy of reasoning. To solve this problem, Yu and Indurthi [2] built a Memory Augmented Neural Networks (MANNs). This type of neural networks decouples the memory capacity from the number of model parameters, using renowned benchmark datasets such as SQuAD, QUASAR-T, and TriviaQA, which efficiently increase the ability to analysis and reason over long documents.

Many researchers started on a more challenging task in recent years, i.e., multi-hope question answering. One of the most representative work, which involves solid background knowledge to reason, gather and generate information, such as complicated memes like human beings [3]. The researchers first use a multi-attention mechanism to perform multiple-hops of reasoning and a pointer-generator decoder to generate the answer. The two models are presented by strong generative baseline which instantly matches its coordinate span. The model is adjusted using commonsense information introduced by ConceptNet to assist the ratiocination.

Yang et al. [4] investigates a pre-training model which has the encoder based on large scale of corpus with introducing KT-NET, desired knowledge are selected from knowledge bases using an attention mechanism. Thus, selected knowledges can fuse into the input embeddings of models as context predictions and knowledge-aware predictions.

III. Model

Our model is built upon three components: (1) Encoder (2) enhanced attention flow layer (3) decoder. The model structure is shown in Figure 2. We elaborate on these modules below.

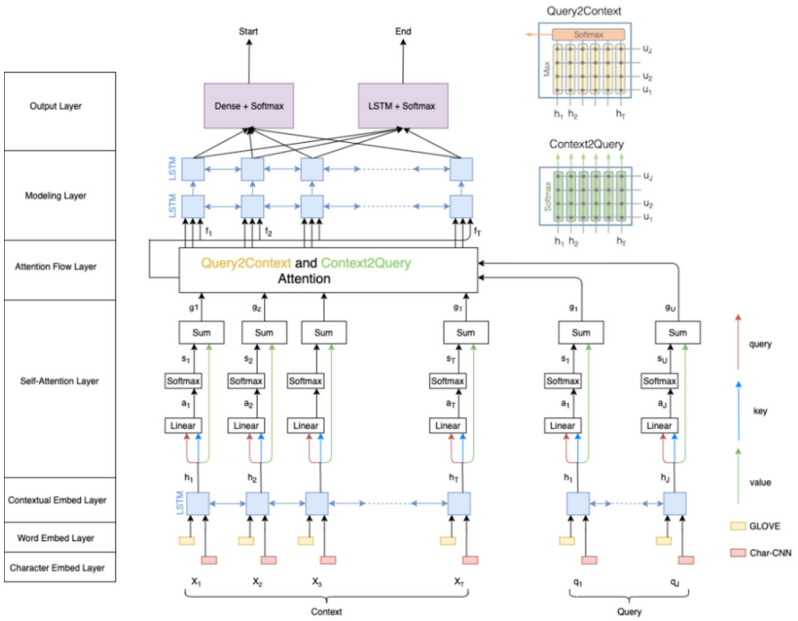

Figure 2. An profile diagram of our model.

A. Encoder

We firstly convert the raw context and query into the semantic matrix by using GloVe word embedding Pennington et al. [7]. Specifically, given the context and query, we gain the word embedding matrix and as below:

$$x_c = GloVe(C);$$
$$x_q = GloVe(Q)$$

Then, we use a bi-directional LSTM [8] to capture the order and contextual features for the word embedding matrix $\mathrm{x_c}$ and $\mathrm{x_q}$.

$$h_c, h_q = BiLSTM(x_c), BiLSTM(x_q)$$

It is worth noting that the word embedding layer and bi-directional LSTM are computing features from the query and context at different levels of granularity, where word embedding contributes more to semantic features, while BiLSTM takes credits to sequential features.

B. Enhanced attention flow

The outputs of the encoder $\mathrm{h_c}, \mathrm{h_q}$ are fed into enhanced attention flow module. Intuitively, the words scattered in the query and context are supposed to act on the model with different weights. Therefore, the more informative words should be highlighted and the useless words should be less involved.

We deploy a self-attention layer to achieve so. In this way, more important words are assigned higher weights, so they can retain more semantic information in subsequent calculations. At the same time, inessential words are assigned lower weights to prevent the model from input too much noise information, so that the model can capture useful features. Formally, given the contextual representation of $\mathrm{h_c}$ we conduct the equations below:

$$\alpha_c = \sigma(W_c \cdot h_c + c_c)$$
$$s_c = softmax(\alpha_c)$$
$$\mathrm{g}_c = \sum \alpha_c \cdot s_c$$

Where σ denotes an activation function, such as ReLU. is served as a normalization function. Similarly, we can gain the same representation for h_q:

$$\alpha_q = \sigma(W_q \cdot h_q + c_q)$$
$$s_q = softmax(\alpha_q)$$
$$\mathrm{g}_q = \sum \alpha_q \cdot s_q$$

Then, we conduct an attention flow layer to link and fuse information from the context and the query words. The attention vector at each time step, along with the embeddings from previous layers, are allowed to flow through to the subsequent modeling layer. This reduces the information loss caused by early summarization and also introduce the mutual information for both context representation as follows:

$$\alpha'_c = \sigma(\mathrm{g}_c, \mathrm{g}_q)$$
$$\alpha'_c = softmax(\alpha'_c)$$
$$\mathrm{f}_c = \sum \alpha'_c \cdot \alpha'_c$$

C. Decoder

The input to the modeling layer is G, which encodes the query-aware representations of context words $\mathbf{f_c}$. The output of the modeling layer captures the interaction among the context words conditioned on the query. This is different from the contextual embedding layer, which captures the interaction among context words independent of the query. We use two

layers of bi-directional LSTM, with the output size of d for each direction. Hence we obtain a matrix G as following equation:

$$\mathrm{G} = BiLSTM_1(BiLSTM_2(\mathrm{f}_c))$$

Finally, we obtain the start and end positional indexes by using two multi-classifiers. Specifically, we calculate the matrix G as:

$$P_{start} = softmax(\sigma(W_s \cdot \mathrm{G} + b_s))$$
$$P_{end} = softmax(\sigma(W_e \cdot \mathrm{G} + b_e))$$

Then, the positions with the highest probability are selected as our prediction Start and End, respectively. Accordingly the word sequence between start index and end index are used as our final answer and will be returned to the user.

IV. Experiment

A. Dataset

Before SQUAD dataset (Rajpurkar et al. [9]) was released by Stanford university in 2016, which is a new question answering dataset consisting of 100,000+ questions posed by crowd workers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage.

We will adopt two metrics which are Exact Match (EM) and F1 score [10], which measures the weighted average of the precision and recall.

B. Main Result

TABLE I. The Primary Experimental Results on SQUAD dataset.

Models	Exact Match	F1 Score
Logistic Regression	40.4	51.0
Dynamic Chunk Reader	62.5	71.0
Fine-Grained Gating	62.5	73.3
Match-LSTM	64.7	73.7
Multi-perspective Matching	65.5	75.1
Dynamic Coattention Networks	66.2	75.9
R-Net	68.4	77.5
BiDAF	68.0	77.3
Our Model	68.5	77.7

We conduct the experiment on SQUAD dataset and the results are listed in Table 1. The machine learning algorithm, i.e., logistics regression unsurprisingly produces the lowest performance due to its simplicity of fitting the complex data. The other baseline models are all constructed by neural network, which all largely surpass the logistics regression baseline and achieve comparable results. BiDAF gains a slightly higher performance since it successfully models the interaction between context and query by using the attention-flow module. Our model is built upon BiDAF but we outperform it as we do not only utilize the attention mechanism across sentence, but also weight the most informative tokens within the context and query, respectively. It enables our model gains a better capability of capturing the fine-grained semantic feature and turn out to be the highest performance with 68.5 EM and 77.7 F1 score.

The plots of training loss for our proposed model and the baseline BiDAF model is presented in Figure 3. It can be seen that compared with BiDAF, our loss drops to a relatively low level while use less epoch, which indicates that our model can converge faster than the baseline in question answering task. Furthermore, the fluctuation in our model is comparatively smoother than BiDAF during the training process, which also proves that our approach can be less likely perturbed and more robust.

C. Ablation study

Since our model contains several sub-modules, we carry out an ablation experiment to test the role of different model components. As we can notice from Table2, the model performance is not affected severely by eliminating the character embedding, which indicates that character embedding actually does not introduce too much additional semantic information. Modelling layer aims to map the semantic features into a higher dimensionality, and model the contextual information between each token. Therefore, the model performance drops largely when we get rid of the modelling layer, since the model is not able to extract the positional features properly. Meanwhile, enhance attention flow is the most crucial component in our model. We can observe from Table 2 that the performance drops by 2% EM and F1 score, approximately, which indicates the interaction between and within sentences play an important role in predicting the correct answer.

TABLE II. Ablation Study: (-) denotes we abandon the corresponding component from the model structure

Models	Exact Match	F1 Score
Our model	68.5	77.7
(-) Char embedding	67.1	75.0
(-) modelling layer	62.7	73.3
(-) enhance attention flow	60.3	69.1

V. Conclusion

In this paper, we focus on the question answering task, which is an essential module amongst the intelligent chatbot. In order to capture the inter and intra information between context and query, we introduce an self-attention layer in the enhanced attention flow module to boost the model performance based on BiDAF. However, our model contains massive of parameters and deteriorate the training cost, which is time-consuming during the training stage. This problem can be explored in the future work.

References

[1] W. Wang, N. Yang, and F. Wei, "Gated self-matching networks for reading comprehension and question answering", Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, vol. 1, pp. 189-198.

[2] S. Yu, S.R. Indurthi, S. Back, "A multi-stage memory augmented neural network for machine reading comprehension" Proceedings of the workshop on machine reading for question answering, 2018, pp. 21-30.

[3] L. Bauer, Y, Wang, M. Bansal, "Commonsense for generative multi-hop question answering tasks", arXiv preprint arXiv:1809.06309, 2018.

[4] A. Yang, Q. Wang, J. Liu, "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension", Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2346-2357.

[5] M. Seo, A. Kembhavi, A. Farhadi, "Bidirectional attention flow for machine comprehension", arXiv preprint arXiv:1611.01603, 2016.

[6] A. Vaswani, N. Shazeer, N. Parmar, “Attention is all you need”, Advances in neural information processing systems, 2017, pp. 5998-6008.

[7] J. Pennington, R. Socher, C.D. Manning, “Glove: Global vectors for word representation”, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.

[8] J. Cao, E.M. van Veen, N. Peek, “EPICURE Ensemble Pretrained Models for Extracting Cancer Mutations from Literature”, arXiv preprint arXiv:2106.07722, 2021.

[9] P. Rajpurkar, J. Zhang, K. Lopyrev, “Squad: 100,000+ questions for machine comprehension of text”, arXiv preprint arXiv:1606.05250, 2016.

[10] J. Cao, C. Wang, “Social media text generation based on neural network model”, Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. 2018, pp. 58-61.