Exploring Attention Mechanisms on SQuAD 2.0

Clara Zou, Yichen Liu, Sibei Zhang
Stanford University

Problem

The goal is to build a question-answering system based on SQuAD 2.0 dataset that could correctly answer a given question based on a given context. The expected answer would be a span of text from the context.

We use BiDAF model as baseline, and aim to improve its performance, explore different attention techniques and compare their performances. More specifically, our explored implementations include character-level embeddings, and different attention layers such as co-attention, self attention and layer-normed scaled dot product attention.

Data

The experiment dataset used in this project is the Stanford Question Answering Dataset (SQuAD 2.0), which contains (context, question, answer) triples. Contexts are excerpts from Wikipedia and the answer is a span of text from the context.

There are:

1. 129,941 examples in training set
2. 6078 samples in dev set (roughly half of the official dev set)
3. 5915 examples in test set (remaining examples in official dev set)

The training set has one answer per question, while dev and test set have three human-provided answers per question.

Reference

1. Natural Language Computing Group, Microsoft Research Asia. R-NET: machine reading comprehension with self-matching networks. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf
2. Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604, 2016.
3. Seo, Minjoon, et al. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603 (2016)..

Methods

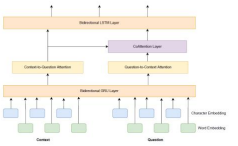
1. Character-level Embeddings

Given $\{c_1, \dots, c_T\}, \{q_1, \dots, q_j\}$ words in the context and the question, respectively:

$$c_{char}, q_{char} = CNN(\{c_1, \dots, c_T\}, \{q_1, \dots, q_j\})$$
$$c = [c_{word}, c_{char}], q = [q_{word}, q_{char}]$$

1. Attention Mechanisms

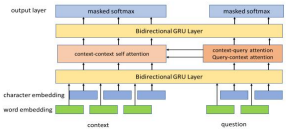
a. Co-Attention Layer



$$\mathbf{q}'_j = \tanh(\mathbf{W}\mathbf{q}_j + \mathbf{b}) \in \mathbb{R}^l \quad \forall j \in \{1, \dots, M\}$$
$$\mathbf{L}_{ij} = \mathbf{c}_i^\top \mathbf{q}'_j \in \mathbb{R}$$
$$\alpha^i = \text{softmax}(\mathbf{L}_{i,:}) \in \mathbb{R}^M$$
$$\mathbf{a}_i = \sum_{j=1}^{M} \alpha_j^i \mathbf{q}'_j \in \mathbb{R}^l$$
$$\beta^j = \text{softmax}(\mathbf{L}_{:,j}) \in \mathbb{R}^N$$
$$\mathbf{b}_j = \sum_{i=1}^{N} \beta_j^i \mathbf{c}_i \in \mathbb{R}^l$$
$$\mathbf{s}_i = \sum_{j=1}^{M} \alpha_j^i \mathbf{b}_j \in \mathbb{R}^l \quad \forall i \in \{1, \dots, N\}$$

b. Self-matching Attention Layer

The self-attention layer attention makes each word attend to all words in the context passage to get better knowledge of the context.



$$s_j^t = v^T tanh(W_v^P v_j^P + W_v^{\tilde{P}} v_t^P)$$
$$a_i^t = exp(s_i^t) / \sum_{j=1}^{n} exp(s_j^t)$$
$$c_t = \sum_{i=1}^{n} a_i^t v_i^P$$

i. Gated attention

$$\sigma(W_g[u_t^P, c_t]) * [u_t^P, c_t]$$

ii. Sigmoid & linear transformation

$$\sigma(W_g[u_t^P, c_t])$$

c. Layer-Normed Scaled Dot Product Attention

normalize the hidden state : $x^{l\prime} = \frac{x^l - \mu^l}{\sigma^l + \epsilon}$

obtain query, key, and value, and perform scaled dot product attention:

$$Output = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Experimentation & Results

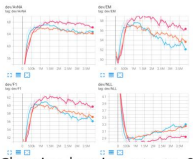
We explored how addition of character level embeddings, attention mechanisms and tuning of hyperparameter would affect **EM (exact match)** and **F1 scores** using **dev** set.

	EM	F1
BiDAF baseline	57.335	60.803
BiDAF + char embeddings	58.847	62.396
BiDAF + char embeddings + Co-attention	52.193	52.193
BiDAF + char embeddings + gated Self-attention	58.427	61.472
BiDAF + char embeddings + Self-attention with transformation (lr = 0.5)	59.435	62.756
BiDAF + char embeddings + Self-attention with transformation (lr = 0.25)	61.822	65.157
BiDAF + char embeddings + Self-attention with transformation (lr = 0.1)	61.838	64.993
BiDAF + char embeddings + Self-attention with transformation (hidden size=150)	58.931	62.107

We adopted BiDAF model with character-level embeddings and self-matching attention layer (with sigmoid & linear transformation) and learning rate = 0.25, hidden size = 100, batch size = 16 as our final model. The final model achieved performance of **EM = 60.118, F1 = 63.866** on **test** set.

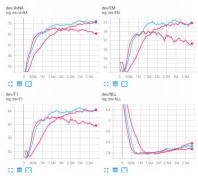
Analysis

1. We found that using sigmoid & linear transformation on attention layer would achieve better EM and F1 scores than gate mentioned in RNET paper [1].



Pink: sigmoid $ linear transformation
Blue: the original gate

2. Changing learning rate to 0.25 and 0.1 greatly improves performance compared to 0.5.



Pink with better performance: learning rate 0.1
Pink with weaker performance: learning rate 0.5
Blue: learning rate 0.25

Conclusion

After training 8 BiDAF-based models with different designs of layers and hyperparameters, we push the dev EM to 61.822 and the dev F1 to 65.157, and subsequently push the test EM to 60.118 and the test F1 to 63.866. In conclusion, the best of our attempted architectural changes with fine-tuned hyperparameters results in better performance than the baseline.

Still, this research entails certain implications for future explorations, including the reasons of the transformation behaving better than the original gate and the rationale behind the unsatisfactory performance of co-attention and layer-normed scaled product self attention, the latter of which we had to terminate during the training process due to its worse training pattern compared with the baseline.