Conditioning External and Internal Context Awareness through Attention Overload and Character-level Embeddings

CS 224N Final Project by Umar Patel

Problem

Problem Statement: Common pre-trained word embeddings do not capture key information on contextual relationships which are essential for encapsulating accurate word and phrase meaning. This leads some NLP models to have a difficult time picking up on contextual nuances, especially for long sequences or sequences with uncommon words. I aim to transform the GLoVE input embeddings into more context-aware and positionally attuned inputs, as well as add a trained character-level embedding layer to enhance conditioning on the internal structure of words, ultimately enhancing the BiDAF Q&A system.

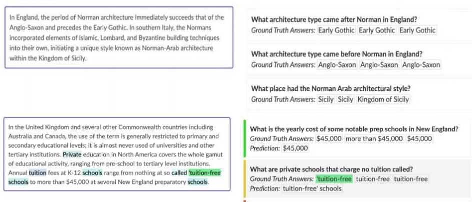
Goal: In this project, I implemented a learned positional encoding layer, a pre-model scaled DP self-attention scheme, as well as a character-level embedding layer for both the queries and context to make up for the lack of context awareness in the default model.

Existing Approaches Q&A approaches that inspired my project: BERT applies the transformer model architecture to Q&A; QANet makes use of stacked encoder blocks with convolution, attention, and feedforward layers and a special context-query attention layer.

Data/Task

My project involves improving upon the BiDAF Q&A model using the techniques described above. I will train on the SQuAD training set of over 130,000 examples of sample question-context pairs like the one below, and test on a condensed version of the official dev set. The official test will be conducted on the complete SQuAD test set, which is hidden.

Evaluation method: I will use two metrics to evaluate my model's performance: the Exact Match (EM) score, which is a binary measure of whether the output matches the ground truth answer exactly, and the F1 score, a less strict measure that is the harmonic mean of precision and recall.



Stanford SQuAD Dataset 2.0

Key References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv*: 1706.03762

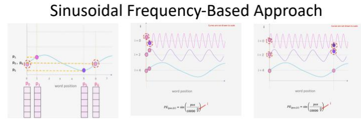
[2] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv preprint arXiv*: 1804.09541

[3] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv*: 1611.01603

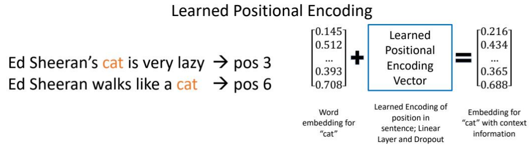
Approach

The first part of my implementation involves a positional encoding layer. I implemented a sinusoidal frequency-based positional encoder as described in Vaswani et al., 2017 [1], and a learned positional encoder layer, the ladder of which performed much better.

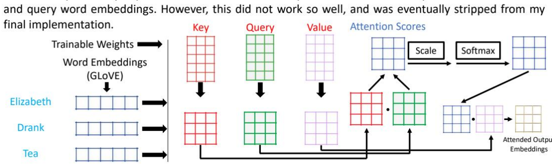
Sinusoidal Frequency-Based Approach



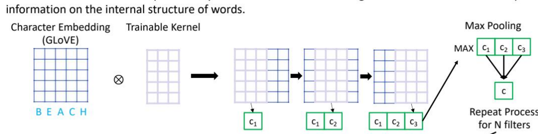
Learned Positional Encoding

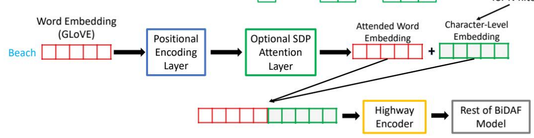


The second part of my implementation was a scaled dot product self-attention layer for both the context and query word embeddings. However, this did not work so well, and was eventually stripped from my final implementation.



Finally, I implemented a character-level embedding layer using 2D-covolution and max pooling and concatenated it with the positionally encoded word embeddings in order to better incorporate information on the internal structure of words.





Results

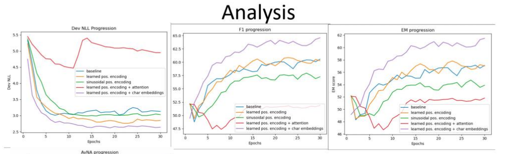
I conducted ~8-9 distinct experiments but these 5 listed in the table sum up my project's overall conclusions (Explanation of F1/EM scores explained in Data/Task section).

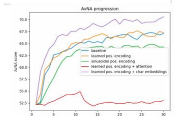
	F1	EM	AvNA
Baseline BiDAF model	60.55	57.10	67.05
Sinusoidal Positional Encoding Implementation Scores:	58.20	54.80	65.18
Learned Positional Encoding Scores:	60.91	57.92	67.33
Learned Positional Encoding + SDP Attention Scores:	52.19	52.19	52.14
Learned Positional Encoding + Character Level Embeddings:	**64.57**	**61.49**	**70.56**

Bolded shows best result

Overall, it's clear that the best results stem from the learned positional encoding and character level embedding implementation, outscoring the baseline model by over 4 points each in both the F1 and EM scores.

Analysis





From the F1 and EM graphs, it is clear that the final learned positional encoding and character level embedding implementation performs better than the baseline and all other implementations at all levels of training.

The learned positional encoding + character level embedding implementation is the one I experimented with which received the highest scores (I ran multiple implementations with different kernel sizes for the 2D convolution, explained in detail in my paper).

Conclusions

We see that by adding implementations and layers that emphasize contextual and positional awareness to the context/query embeddings, as well as incorporating key information on internal word structures through character-level embeddings, we are able to significantly improve upon the baseline BiDAF model. I also found that too much attention can be detrimental and confuse our model, as the SDP attention layer after the positional encoding actually worsened overall performance.

I didn't have time to get to it, but future work may involve stopping training earlier, as many of my experiments appeared to have peaked within epoch 20-25 range. This would allow me to see if preventing overfitting makes slight differences in overall performance. Furthermore, I also would like to delve deeper into the odd behavior of the positional encoding/self attention case and why its progression during training and overall worse performance was such an outlier from the rest of the experiments.