BERT Is All You Need

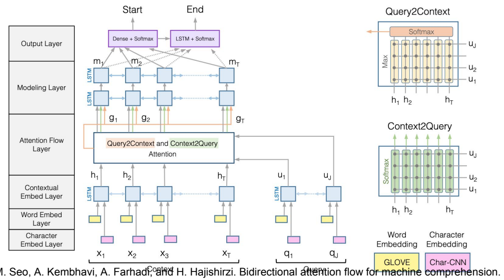
Sam Schwager (sams95) and John Solitario (johnny18)

Introduction

- **Goal:** Use HuggingFace Pytorch BERT implementation for SQuAD 2.0 to create a high-performing model and to develop foundation for carrying out a variety of experiments, leading to more broadly applicable results.
- **Results: F1 score of 76.545** and an **EM score of 73.609.** Impressive results from modifying the loss function, training with dropout, and ensembling across models.

Baseline

- **Bidirectional Attention Flow (BiDAF)** without character level embedding layer.

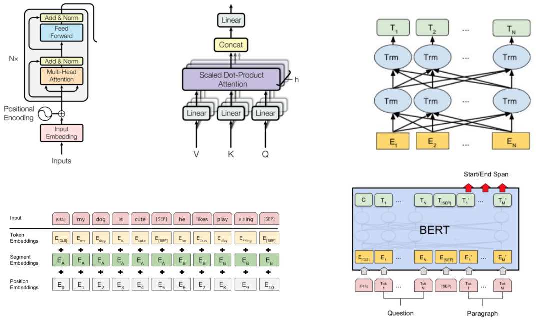


*M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension.

BERT and the Power of Attention

- **BERT Base:** Multi-layer bidirectional Transformer encoder. The Transformer consists of 12 transformer blocks and each block has 12 attention heads. Trained with a hidden size of 768 and a max sequence length of 512. In all, the model has approx. 110 million parameters.

BERT Model Overview



*J. Devlin , M.W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. **A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need.

Modifying the Output Layer

- **Bespoke output layer** replaces the shallow output layer with a single hidden layer with ReLU activation.

$$\mathbf{x}_{output} = ReLU(\mathbf{W}_1\mathbf{T} + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

Adjusting the Loss Function

- **Start vs. End:** Weight the start loss twice as much as the end loss for all types of questions.
- **Answer vs. No Answer:** Weight loss for questions **with** answers twice as much as questions without answers. Weight loss for questions **without** answers twice as much as questions with answers.

Experiments and Results

Model Description	Dropout	Learning Rate	Epochs Trained	No Answer F1 / EM	Has Answer F1	Has Answer EM	F1	EM
BiDAF Baseline	X	0.5	30	-	-	-	59.81	56.7
BERT Small		0.0005	3	53.09	81.96	74.57	66.91	63.38
BERT Small		0.0005	2	53.88	82.89	75.74	67.77	64.35
BERT Small	X	0.0003	3	61.90	81.35	74.74	71.21	68.05
BERT Small	Last	0.0003	3	62.22	82.07	75.33	71.72	68.49
BERT Small	X	0.0003	2	71.31	80.26	73.78	75.59	72.49
BERT Small		0.0003	3	73.58	77.73	71.34	75.57	72.51
BERT Small		0.0003	2	72.63	79.84	73.44	76.09	73.02
BERT - Bespoke Output Layer	X	0.0003	2	66.95	77.34	71.55	71.93	69.15
BERT - Adjusted Loss Function: start loss weighted twice end loss	X	0.0003	2	71.24	78.79	71.65	74.86	71.44
BERT - Adjusted Loss Function: loss weighted twice when no answer		0.0003	2	59.50	81.30	74.12	69.94	66.5
BERT - Adjusted Loss Function: loss weighted twice when has answer		0.0003	2	72.66	80.67	74.47	**76.5**	**73.53**

Intelligent Ensembling

- **Ensembling Method:** Choose answer with the highest probability. However, predict no answer if any of the models in the ensemble predict no answer.
- **Chosen Models:** (1) dropout, (2) weighting questions with answers more heavily than those without, and (3) weighting the answer span start more heavily than the end.

Conclusion

- Achieved competitive scores by leveraging dropout, manipulating the loss function, and ensembling.
- **Future Work:** Apply the most effective strategies to training the large version of BERT. Experiment with additional output layers.