Supplementary Material for "PoWER-BERT: Accelerating BERT Inference via Progressive Word-vector Elimination"

1. Additional Results

1.1. Validation of the Scoring Function Using Mutual Information.

`PoWER-BERT` uses a scoring mechanism for estimating the significance of the word-vectors satisfying the following criterion: the score of a word-vector must be positively correlated with its influence on the final classification output (namely, word-vectors of higher influence get higher score).

For a word-vector w and a head h, we define the significance score of w for h as $\texttt{Sig}_h(w) = \sum_{w'} \mathbf{A}_h[w', w]$ where A_h is the *attention matrix* for head h and w' signifies other words in the input sequence. The overall significance score of w is then defined as the aggregate over the heads: $\texttt{Sig}(w) = \sum_h \texttt{Sig}_h(w)$.

We validate the scoring function using the well-known concept of mutual information and show that the significance score of a word is positively correlated with the classification output. Let $\mathbf{X}$ and $\mathbf{Y}$ be two random variables. Recall that the *mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ is defined as $\mathbf{MI}(\mathbf{X};\mathbf{Y}) = \mathbf{H}(\mathbf{X}) - \mathbf{H}(\mathbf{X}|\mathbf{Y})$, where $\mathbf{H}(\mathbf{X})$ is the entropy of $\mathbf{X}$ and $\mathbf{H}(\mathbf{X}|\mathbf{Y})$ is the conditional entropy of $\mathbf{X}$ given $\mathbf{Y}$. The quantity is symmetric with respect to the two variables. Intuitively, Mutual information measure how much $\mathbf{X}$ and $\mathbf{Y}$ agree with each other. It quantifies the information that can be gained about $\mathbf{X}$ from information about $\mathbf{Y}$. If $\mathbf{X}$ and $\mathbf{Y}$ are independent random variables, then $\mathbf{MI}(\mathbf{X};\mathbf{Y}) = 0$. On the other hand, if the value of one variable can be determined with certainty given the value of the other, then $\mathbf{MI}(\mathbf{X};\mathbf{Y}) = \mathbf{H}(\mathbf{X}) = \mathbf{H}(\mathbf{Y})$.

For this demonstration, we consider the SST-2 dataset from our experimental study with input length $N = 128$ and number of classification categories $C = 2$ (binary classification). Consider an encoder j and we shall measure the mutual information between the classification output of the original model and a modified model that eliminates a single word at encoder j. Consider the trained `BERT` model without any word elimination and let $\mathbf{X}$ denote the classification label output by the model on a randomly chosen input from the training data. Fix an integer $k \leq \ell_{j-1}$ and let w be the word with the k^{th} highest significance score. Consider a modified model that does not eliminate any words on encoders $1, 2, \ldots, j-1$, eliminates w at encoder j, and does not eliminate any further words at encoders $j+1, j+2, \ldots, 12$. Let $\mathbf{Y}_k$ be the random variable that denotes the classification output of the above model on a randomly chosen input. The mutual information between $\mathbf{X}$ and $\mathbf{Y}_k$ can be computed using the formula:

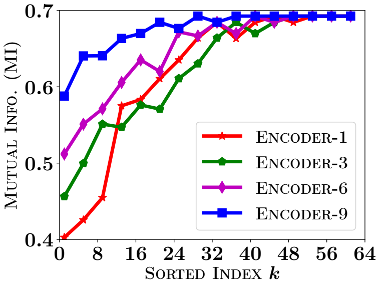

Figure 1: Demonstration of mutual information

$$\sum_{b,b' \in \{0,1\}} \Pr(\mathbf{X} = b, \mathbf{Y} = b') \cdot \ln \left[\frac{\Pr(\mathbf{X}=b, \mathbf{Y}=b')}{\Pr(\mathbf{X}=b) \cdot \Pr(\mathbf{Y}=b')} \right]$$

We measured the mutual information between $\mathbf{X}$ and $\mathbf{Y}_k$ for all encoders j and for all $k \in [1, 128]$. Figure 1 shows the above data, wherein for the simplicity of presentation, we have restricted to encoders $j = 1, 3, 6, 9$. In this case, since the model predictions are approximately balanced between positive and negative, the baseline entropy is $\mathbf{H}(\mathbf{X}) \sim \ln(2) = 0.69$. We can make two observations from the figure. First is that as k increases, the mutual information increases. This implies that deleting words with higher score results in higher loss of mutual information. Alternatively, deleting words with lower score results in higher mutual information, meaning the modified model gets in better agreement with the original model. Secondly, as the encoder number j increases, the mutual information approaches the baseline entropy faster, confirming our hypothesis that words of higher significance scores (or more words) can be eliminated from the later encoders. The figure demonstrates that the score $\texttt{Sig}(\cdot)$ captures the significance of the words in an effective manner.

1.2. Comparison to Prior Methods.

Figure 2 presents the pareto curves for the three remaining datasets: SST-2, MNLI-mm and STS-B from the GLUE benchmark. These curves shows that `PoWER-BERT` achieves better trade-off between accuracy and inference

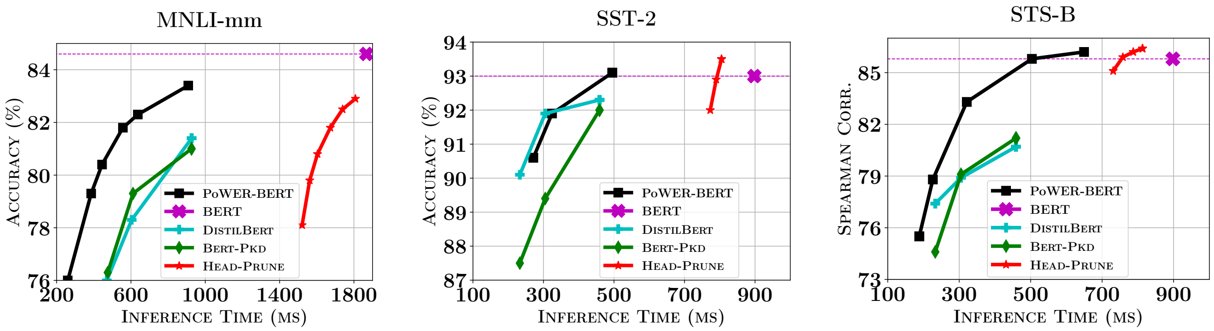

Figure 2: Comparison to prior methods. Pareto curves showing accuracy vs. inference time for three GLUE benchmark datasets: SST-2, MNLI-mm and STS-B. Rest of the datasets have been shown in the main paper.

Table 1: Hyper-parameters to reproduce the results in Table 2 in the main paper. Each experiment was repeated 3 times and average of these runs were reported in the paper. LR signifies the Learning Rate.

Dataset	BERT params LR	Soft-Extract LR	Lambda λ	Batch Size
CoLA	0.00002	0.0015	0.007	32
RTE	0.00003	0.003	0.001	16
QQP	0.00003	0.0001	0.0003	64
MRPC	0.00006	0.003	0.003	64
SST-2	0.00003	0.0005	0.0002	64
MNLI-m	0.00003	0.0002	0.0001	64
MNLI-mm	0.00003	0.0001	0.0001	64
QNLI	0.00003	0.0002	0.00015	16
STS-B	0.00003	0.003	0.001	64
IMDB	0.00003	0.0002	0.0007	8
RACE	0.00001	0.0002	0.0001	4

time compared to other state-of-the-art inference time reduction methods.

2. Hyper-parameter details.

Dataset specific Hyper-parameters for PoWER-BERT. Training PoWER-BERT primarily involves four hyper-parameters, which we select from the ranges listed below: a) learning rate for the newly introduced soft-extract layers - $[10^{-4}, 10^{-2}]$; b) learning rate for the parameters from the original BERT model - $[2 \times 10^{-5}, 6 \times 10^{-5}]$; c) regularization parameter λ that controls the trade-off between accuracy and inference time - $[10^{-4}, 10^{-3}]$; d) batch size - $\{4, 8, 16, 32, 64\}$. Table 1 presents the dataset specific hyper-parameters used to obtain the results in Table 2 of the main paper. The code for PoWER-BERT is publicly available at https://github.com/IBM/PoWER-BERT.

Hyper-parameters for Baseline Methods. We compare PoWER-BERT with the state-of-the-art inference time reduction methods: DistilBERT, BERT-PKD and Head-Prune. For DistilBERT two hyper-parameters were tuned: learning rate - $[2\times10^{-5}, 6\times10^{-5}]$ and batch size - $\{16, 32, 64\}$.

For BERT-PKD four hyper-parameters were tuned: Temperature (T) that controls the extend to which the student rely on the teacher's soft predictions, α that balances the cross-entropy and the distillation loss, β that weights the feature importance for distillation in the intermediate layers and learning rate for the teacher model. The ranges used for each of these hyper-parameters was: T - $\{5, 10, 20\}$, α - $\{0.2, 0.5, 0.7\}$, β - $\{10, 100, 500, 1000\}$ and learning rate - $[2 \times 10^{-5}, 6 \times 10^{-5}]$.

For Head-Prune the only tunable hyper-parameter is the learning rate and we set the range to be: $[2 \times 10^{-5}, 6 \times 10^{-5}]$.

For all the baseline methods, the hyper-parameters were tuned using the Dev set and the accuracy were reported on the publicly available Test set