MOPS: Memory Occupancy and Performance Surveying when using Late-Stage Hard Parameter Sharing for BERT Multitask Learning

Anthony Weng Mark Bechthold Callum Burgess

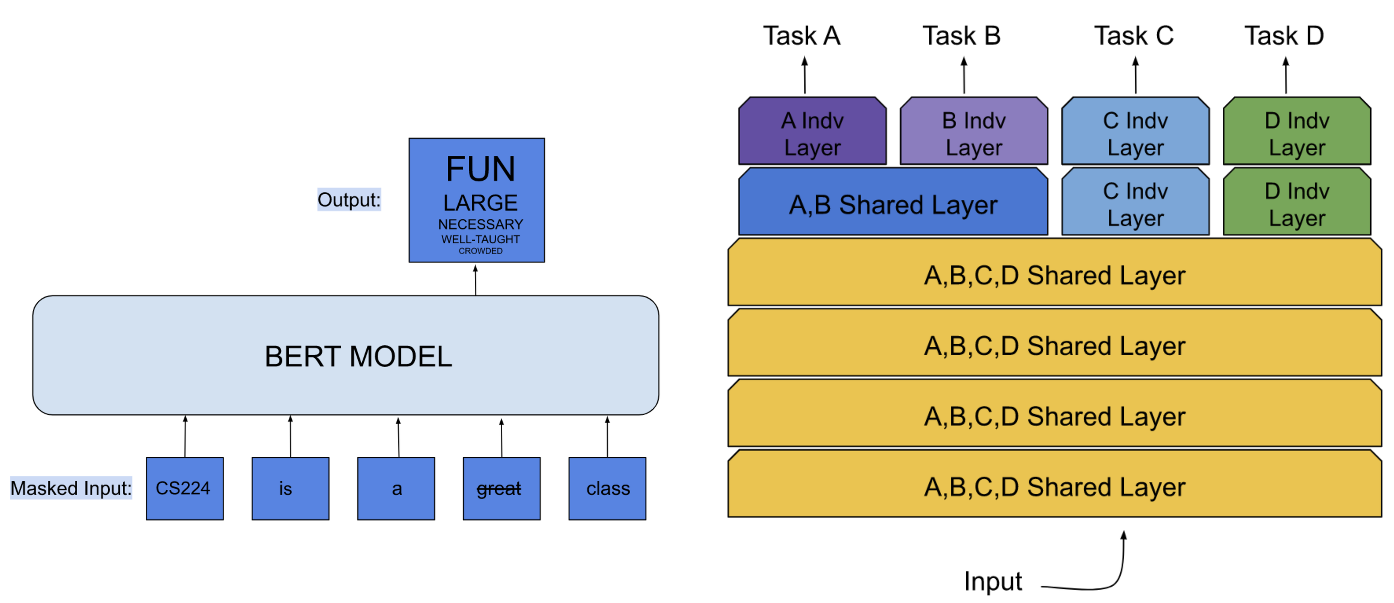
Introduction

Natural language tasks, which include sentiment analysis, question answering, textual summaries, and language translation have multitudes of beneficial applications in a wide variety of topics spanning education and communication. A general model that can understand and analyze a wide variety of tasks, a *multitask model*, can be beneficial for not only consolidation purposes, but from an accuracy standpoint as well:

- We can pre-train our multitask model on a task with a multitude of publicly available data to efficiently allow the model to learn general knowledge and basic intuition for language
- Sharing resources across tasks allows a model to develop intuitions for language as a whole and gives the model a higher variety and more robust set of training data

Background

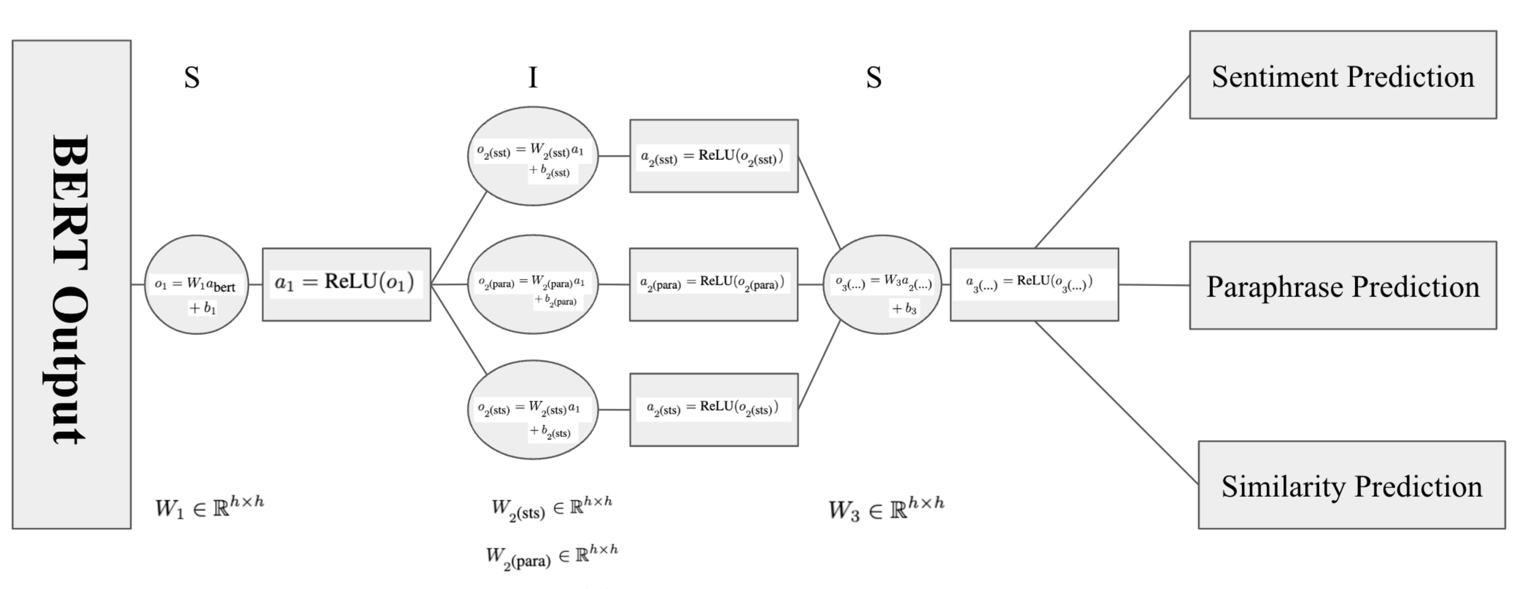
In this project, we use a BERT (Bidirectional Encoder Representations from Transformers) multi-task language model as a foundation for our study, which has cemented itself as a state-of-the-art model that can be effectively applied to almost any natural language processing task! This model keyly uses a "masked language model" objective, pictured below, to effectively build upon its bidirectional transformer architecture and craft a great pretrained model that can be applied to sentence and token level tasks:



Pictured above to the right is an example multitask model exhibiting parameter sharing, where certain tasks go through the same layers when evaluating and training. A multitask BERT model with hard parameter sharing has had past success in a study by [2].

Approach

We decided to explore various downstream parameter sharing regimes to see if we could improve overall performance across tasks. To do this, we insert a three layer network after the BERT and vary whether each the tasks share weights for that layer or have their own weights for the layer.



Results

- * represents baseline implementation I-I-I.
- Report top 4 non-baseline models with fixed learning rate.
- Tune learning rate and weighting of our custom loss.

Table 1. Summary of selected results for model configurations using *pretrained* BERT weights.

Model Configuration			Best Overall Score - Validation Set				
Param. Sharing Regime	Learning rate	**w**	Score	PD Acc.	SC Acc.	STS Corr.	Time- and- memory-relative overall score
I-I-I*	1e−3	[1/3, 1/3, 1/3]	0.328	0.609	0.397	-0.020	0.328
I-I-S			0.354	0.625	0.410	0.027	0.357
S-I-I			0.342	0.625	0.408	-0.008	0.328
I-S-S			0.339	0.625	0.404	-0.013	0.358
S-S-I			0.377	0.587	0.397	0.147	0.375
I-S-S	8e−6	[1/3, 1/3, 1/3]	0.392	0.618	0.386	0.173	0.412
I-S-S	8e−6	[2/5, 1/5, 2/5]	0.313	0.375	0.364	0.199	0.305

- Report best performing model is I-S-S with score of 0.5.
- Tune learning rate across models and report S-I-I with a score of 0.397.
- Tune weighting of loss function.

Table 2. Summary of selected results for model configurations using *finetuned* BERT weights.

Model Configuration			Best Overall Score - Validation Set				
Param. Sharing Regime	Learning rate	**w**	Score	PD Acc.	SC Acc.	STS Corr.	Time- and- memory-relative overall score
I-I-I*	1e−5	[1/3, 1/3, 1/3]	0.492	0.625	0.491	0.359	0.492
S-I-I			0.494	0.627	0.503	0.353	0.556
I-S-S			0.500	0.625	0.524	0.351	0.576
S-I-S			0.469	0.593	0.495	0.320	0.580
S-S-I			0.490	0.625	0.490	0.355	0.587
S-I-I	8.57e-06	[1/3, 1/3, 1/3]	0.397	0.375	0.491	0.325	0.471
I-S-S	1e-05	[2/5, 1/5, 2/5]	0.493	0.625	0.491	0.363	0.602

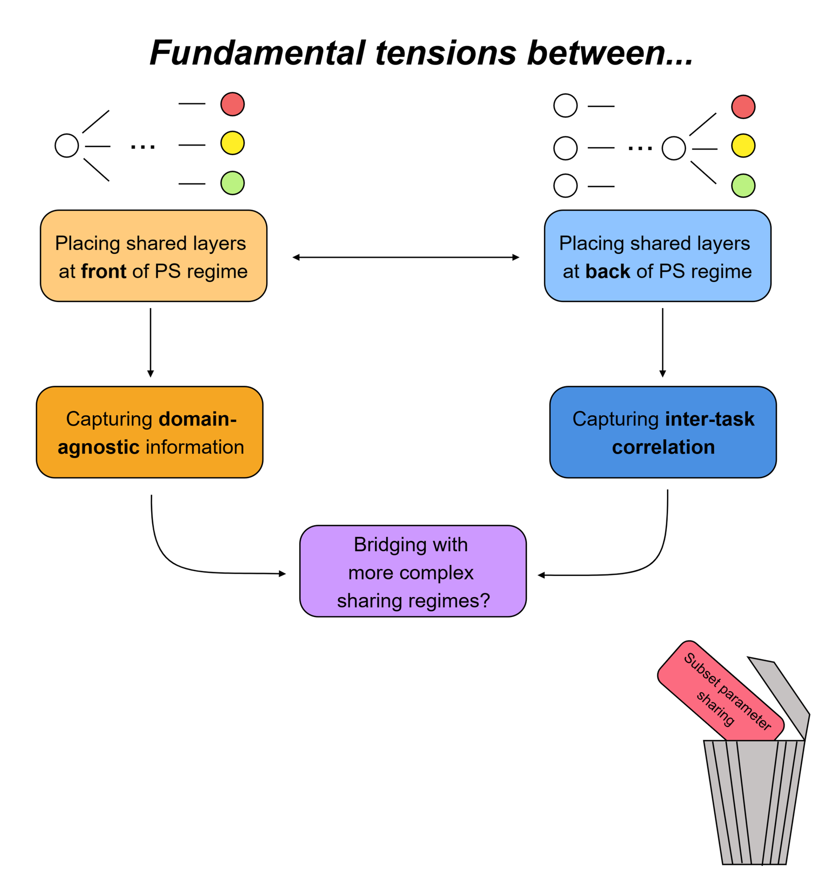
Table 3. Score characteristics of the model configuration used for test set evaluation.

Model Configuration			Score Characteristics				
Param. Sharing Regime	Learning rate	**w**	Data split	Score	PD Acc.	SC Acc.	STS Corr.
I-S-S	1e−5	[1/3, 1/3, 1/3]	Training	0.750	0.639	0.823	0.789
			Validation	0.500	0.625	0.524	0.352
			Test	0.481	0.631	0.523	0.288

- We report the best performing model across training, dev, and test sets.
- We note that the model does not generalize well from training to validation in the sentiment and similarity tasks.

Analysis and Discussion

- Verifying results from Pahari et al. [3], we observe performance gains by placing shared layers at the **front** of a parameter sharing regime to capture **domain-agnostic knowledge**.
  - Signaling the potential transferability of techniques of soft parameter to hard parameter sharing, and vice versa.
- However, performance gains are also yielded by placing shared layers at the **end** of a parameter sharing regime, which helps to capture **inter-task correlation** between outputs of related tasks.
  - But, placing shared layers at the end of a regime can adversely influence outputs for unrelated tasks–especially for **tasks with different output ranges**.
- This trade-off between placing shared layers at the front and end of a parameter sharing regime cannot be resolved by only parameter sharing across a **subset** of correlated tasks–doing so may lead to global performance regression.



Methodology

Four primary divisions of experiments are conducted:

Experiments 1 and 2 evaluate the:

- absolute performance; and
- time- and memory-relative performance

of different parameter sharing regimes.

Experiments 3 and 4 further examine the benefit of tuning the:

- learning rate; and
- task-specific loss weightings:

$$\text{LOSS} = w_{sts}\text{MSE}_{\text{sts}} + w_{sst}\text{CE}_{\text{sst}} + w_{para}\text{MSE}_{\text{para}}$$

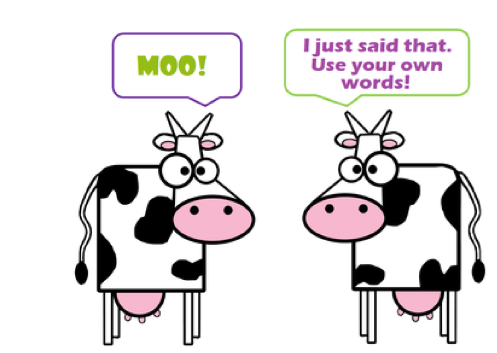
An ongoing experiment, **Experiment 5** investigates parameter sharing across a **subset** of tasks.

Data and Tasks

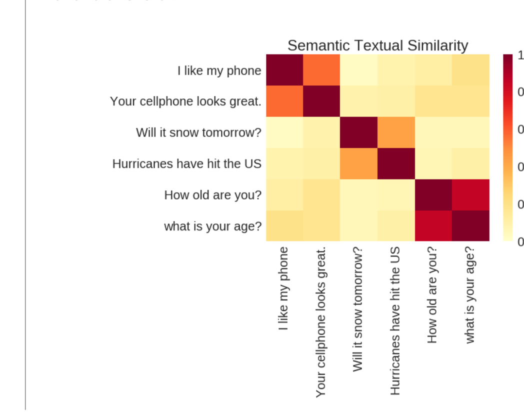
All experiment models are evaluated on the same multitask learning objective. Evaluation tasks include:

Sentiment classification on the Stanford Sentiment Treebank (SST) dataset.

Paraphrase detection on the Quora dataset.



Semantic textual similarity rating on the SemEval STS Benchmark dataset.



Shared Experiment Characteristics and Metrics

- *Two* iterations of each experiment are conducted, one using **pretrained** BERT weights and one using **fine-tuned** BERT weights.
- Models are scored by their **average score** on each learning task:

$$\text{Score}_{\text{model}} = \frac{1}{3} \times (\text{Score}_{ParaphraseDetection} + \text{Score}_{SentimentClassification} + \text{Score}_{SemanticTextualSimilarity})$$

- **Task-specific scores** are measured by: accuracy of predicted labels for paraphrase detection and sentiment classification; correlation of predicted and true labels for semantic textual similarity rating.
- We also compute the time- and memory-relative performance of models:

$$\text{TM relative score} = \begin{array}{c}\text{Overall score of}\\ \text{non-baseline model}\end{array} \times \frac{\begin{array}{c}\text{Training time required for}\\ \text{baseline model}\end{array}}{\begin{array}{c}\text{Training time required for}\\ \text{non-baseline model}\end{array}} \times \frac{\begin{array}{c}\text{Memory required for}\\ \text{baseline model storage}\end{array}}{\begin{array}{c}\text{Memory required for}\\ \text{non-baseline model storage}\end{array}}$$

Figure 1. Equation for computing the TM-relative overall score

Key References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. 2019.

[3] Niraj Pahari and Kazutaka Shimada. Multi-task learning using bert with soft parameter sharing between layers. pages 1–6, 2022.

[4] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. *CoRR*, abs/1903.05987, 2019.