Stanford | ENGINEERING
Computer Science

LEAVE IT TO BERT:
Exploring Methods for Robust Multi-Task Performance

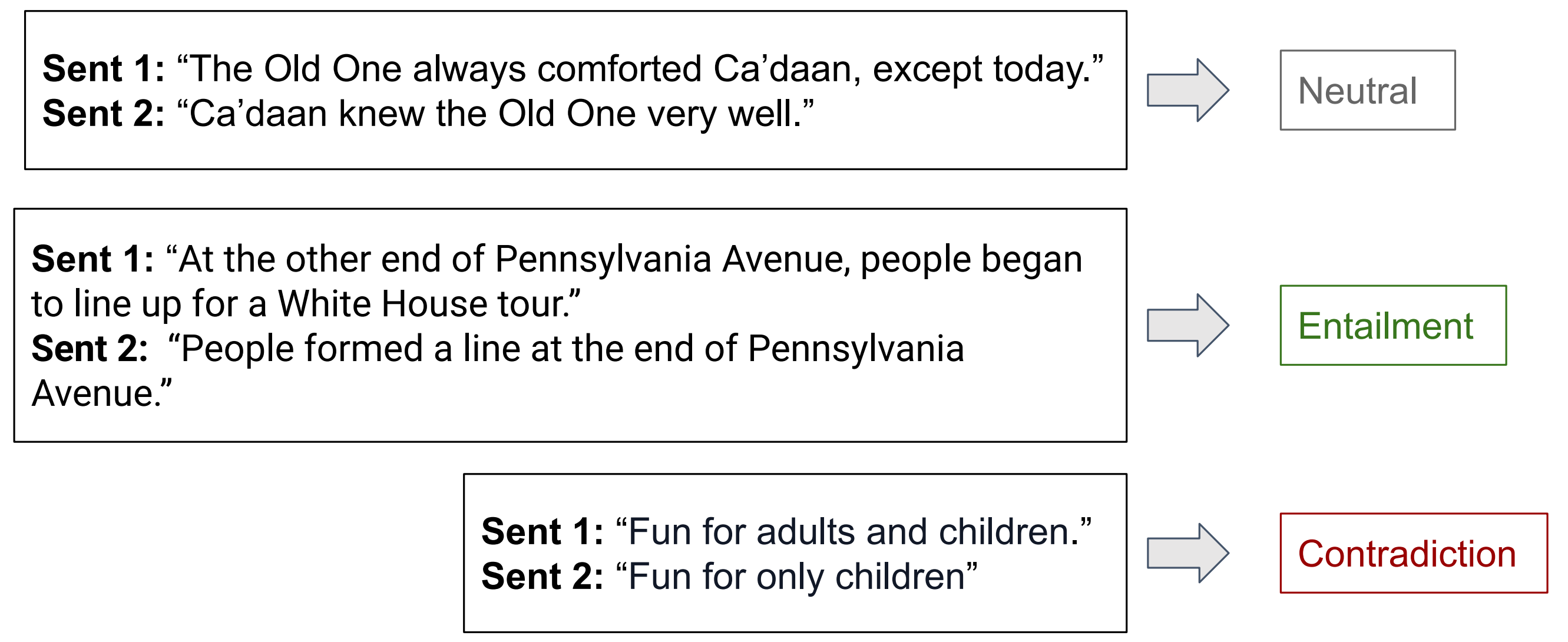
Finn Dayton finndayton@stanford.edu
Abhi Kumar abhi1@stanford.edu
Chris Moffitt cmoffitt@stanford.edu

Background

Deep NLP models have achieved great success on single tasks, but the ability to perform well on multiple tasks is becoming more crucial for many practical applications of NLP. Developing multi-task models is challenging, however, because different tasks have different input representations, output formats, and training data. Moreover, tasks may have conflicting objectives. Transformer-based models such as BERT, have led to significant improvements in the performance of NLP models across a range of tasks. For our project, we train and evaluate a series of models that use shared BERT embeddings to perform well on three tasks simultaneously: sentiment analysis, paraphrase detection, and similarity detection.

Datasets

1. **Multi-Genre NLI Corpus** - 433k sentence pairs from 10 genres (fiction, government, etc) for textual entailment. Each sentence pair is labeled as either "neutral", "entailment", or "contradiction". [1]



Sent 1: "The Old One always comforted Ca'daan, except today."
Sent 2: "Ca'daan knew the Old One very well."
Neutral

Sent 1: "At the other end of Pennsylvania Avenue, people began to line up for a White House tour."
Sent 2: "People formed a line at the end of Pennsylvania Avenue."
Entailment

Sent 1: "Fun for adults and children."
Sent 2: "Fun for only children"
Contradiction

2. **SST** (Stanford Sentiment Treebank) - 11,855 single sentences extracted from movie reviews labeled by sentiment: negative, somewhat negative, neutral, somewhat positive, or positive. [2]
3. **PARA** (Quora Dataset) - 400,000 question pairs and labels indicating whether particular instances are paraphrases of one another. [3]
4. **STS** (SemEval Dataset) - 8,628 different sentence pairs of varying similarity on a scale from 0 (unrelated) to 5 (equivalent meaning). [4]

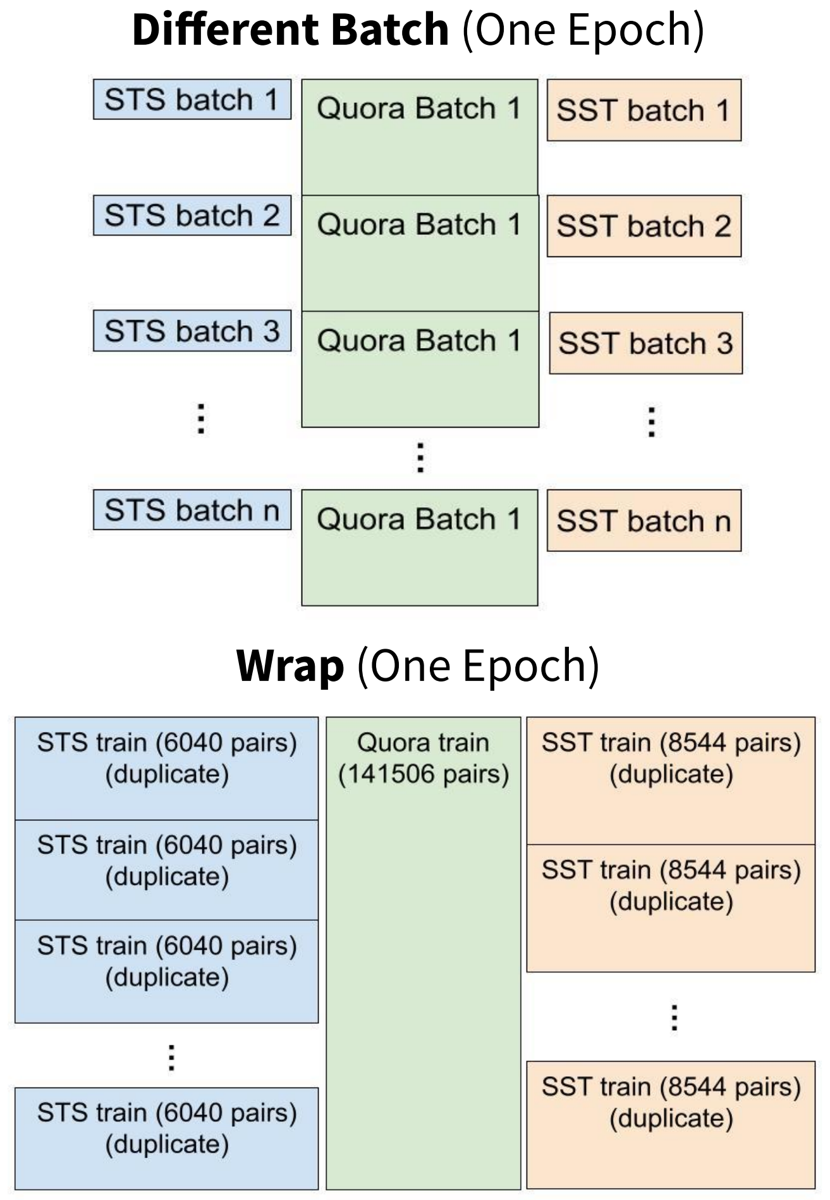
Methods

Gradient surgery (GS):

- Gradient surgery, proposed by Yu et al. [5], projects a task's gradient onto the normal plane of the gradient of any other task that has a conflicting gradient
- Our implementation trains BERT simultaneously on the SST, STS and Quora datasets using GS to reconcile the gradients of each of the three losses. We first calculate and add the losses for each task, take the gradients, project them to resolve conflicts, and finally add them together.
- The model can then make a step in gradient descent that is mutually beneficial for all three tasks.

We attempted two types of round robin training to reconcile the different sizes of SST, STS, and Quora datasets during gradient surgery:

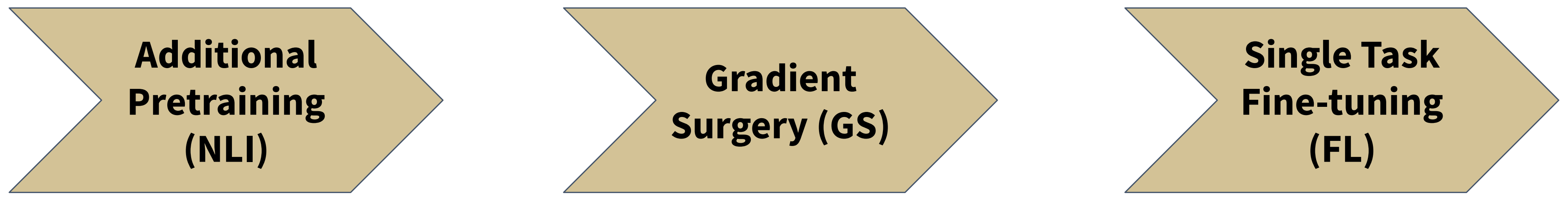
1. **Different Batch Sizes (GS_{bd}):** Each gradient step uses a different number of examples from each dataset
2. **Wrap (GS_{w}):** Each gradient step sees an equal number of examples from each dataset. This means each epoch sees all of Quora (the largest dataset) once and sees many duplicates of STS and SST, which may lead to overfitting.



Different Batch (One Epoch)

Wrap (One Epoch)

STS batch 1
Quora Batch 1
SST batch 1
STS batch 2
Quora Batch 1
SST batch 2
STS batch 3
Quora Batch 1
SST batch 3
STS batch n
Quora Batch 1
SST batch n
STS train (6040 pairs) (duplicate)
STS train (6040 pairs) (duplicate)
STS train (6040 pairs) (duplicate)
STS train (6040 pairs) (duplicate)
Quora train (141506 pairs)
SST train (8544 pairs) (duplicate)
SST train (8544 pairs) (duplicate)
SST train (8544 pairs) (duplicate)



Further pretraining on Multi-Genre data (NLI):

- Model learns to predict textual entailment (adjacent task) and learns general language representations that can be transferred to objective tasks.

Single Task Fine-tuning (FL):

- We freeze shared BERT weights and fine-tune each head individually to learn task-specific features and specialized parameters.

Experiments & Analysis

Baseline:	SST	PARA	STS
Finetune on SST	.310	.380	-.008

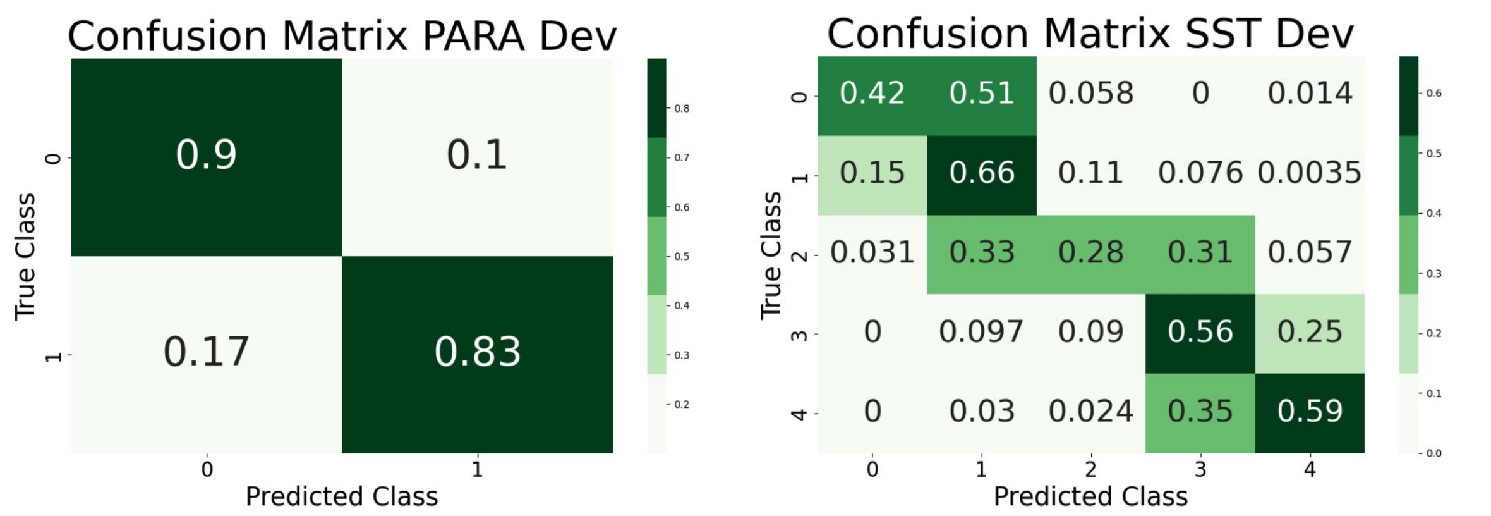
*SST and PARA scores are accuracies and STS score is a pearson correlation coefficient

	Model Architectures								
	ConcatA			ConcatB			ConcatB + AL		
Methods	SST	PARA	STS	SST	PARA	STS	SST	PARA	STS
GS_{bd}	.500	.720	.362	.520	.838	.866	.511	.832	.865
GS_{w}	.528	.732	.352	.511	.843	.850	.520	**.873**	.844
FL	.305	.636	.206	.303	.665	.154	.337	.696	.330
NLI + GS_{bd}	.529	.708	.390	.495	.836	.881	.500	.839	**.892**
NLI + GS_{w}	.514	.740	.326	.503	.850	.868	.516	**.873**	.877
NLI + FL	.342	.661	.104	.343	.753	.635	.330	.751	.711
GS_{bd} + FL	.503	.719	.373	.510	.838	.867	.511	.830	.866
GS_{w} + FL	.527	.742	.356	.508	.847	.851	.518	**.873**	.844
NLI + GS_{bd} + FL	**.531**	.713	.389	.499	.840	.882	.508	.843	.891
NLI + GS_{w} + FL	.508	.748	.332	.499	.853	.868	.514	**.873**	.878

Summary of Results:

- Overall, the model trained with NLI + GS_{w} and the ConcatB + AL architecture (highlighted) performed the best, having the highest average score across all tasks
- The best SST score comes from NLI + GS_{bd} + FL with ConcatA
- The best PARA score comes from multiple experiments using GS_{w} with ConcatB +AL
- The best STS score comes from NLI + GS_{bd} with ConcatB + AL

Confusion matrices of results show that our best model predictions stay close to true labels:



Confusion Matrix PARA Dev

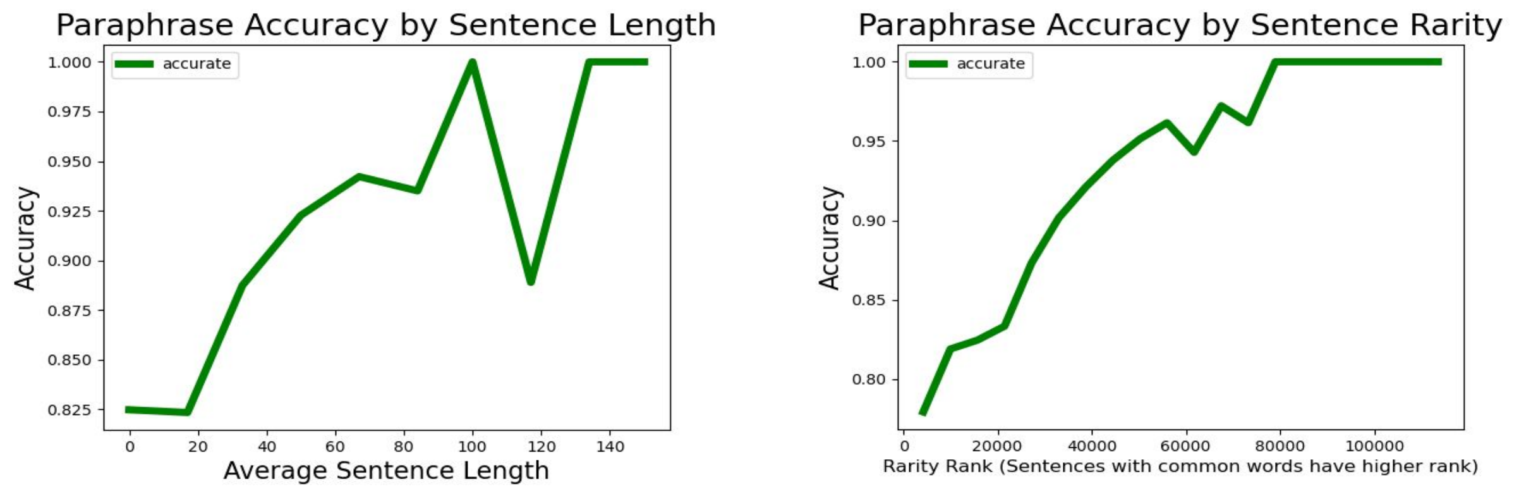
True Class \ Predicted Class	0	1
0	0.9	0.1
1	0.17	0.83

Confusion Matrix SST Dev

True Class \ Predicted Class	0	1	2	3	4
0	0.42	0.51	0.058	0	0.014
1	0.15	0.66	0.11	0.076	0.0035
2	0.031	0.33	0.28	0.31	0.057
3	0	0.097	0.09	0.56	0.25
4	0	0.03	0.024	0.35	0.59

Insights into best model performance against data features:

Longer sentences increase accuracy for paraphrase task. Longer sentences provide more context for the model.



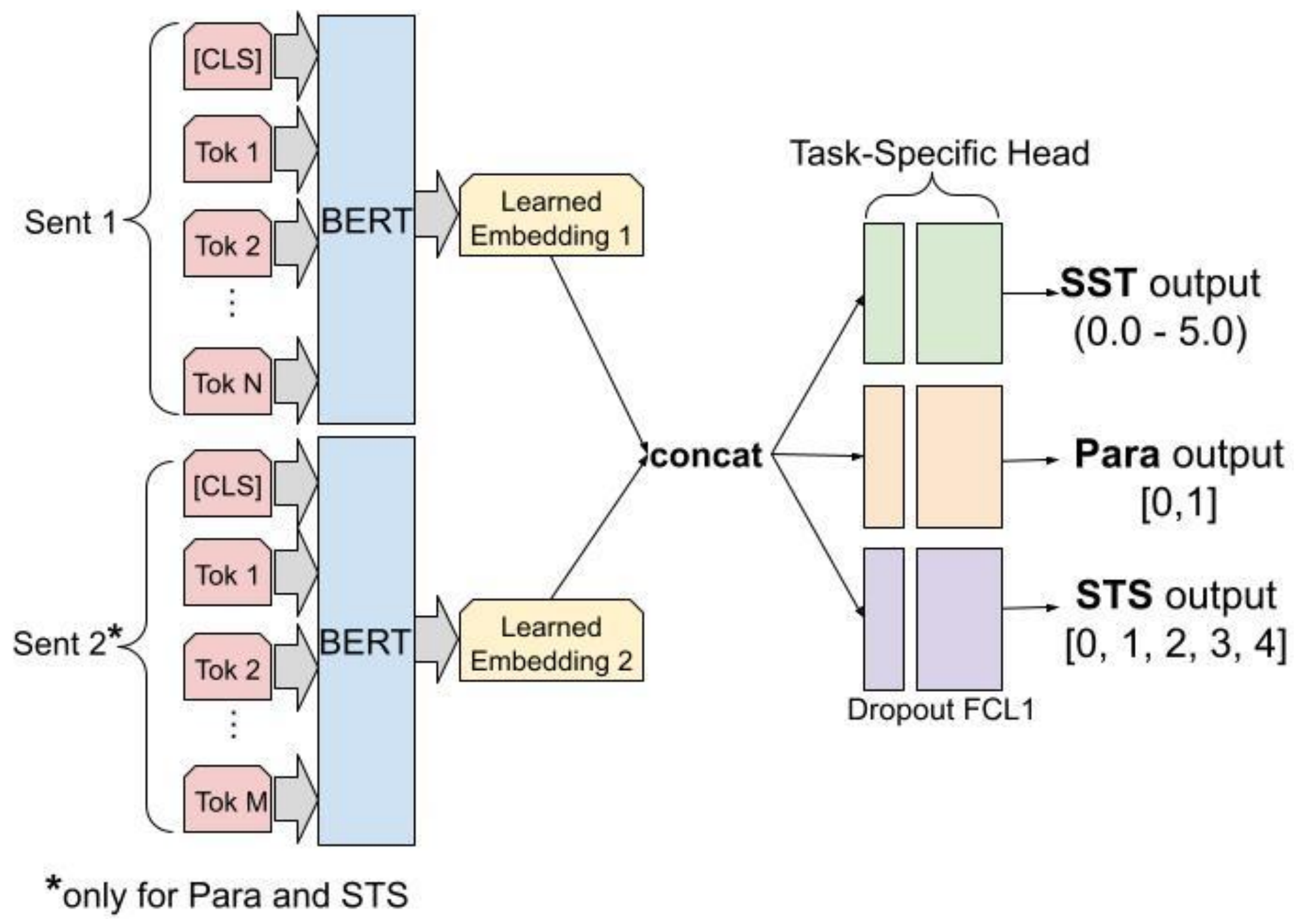
Paraphrase Accuracy by Sentence Length

Paraphrase Accuracy by Sentence Rarity

Sentences with common words (higher Rarity Rank) from the training set have higher accuracy than sentences with rare words.

Model Architectures

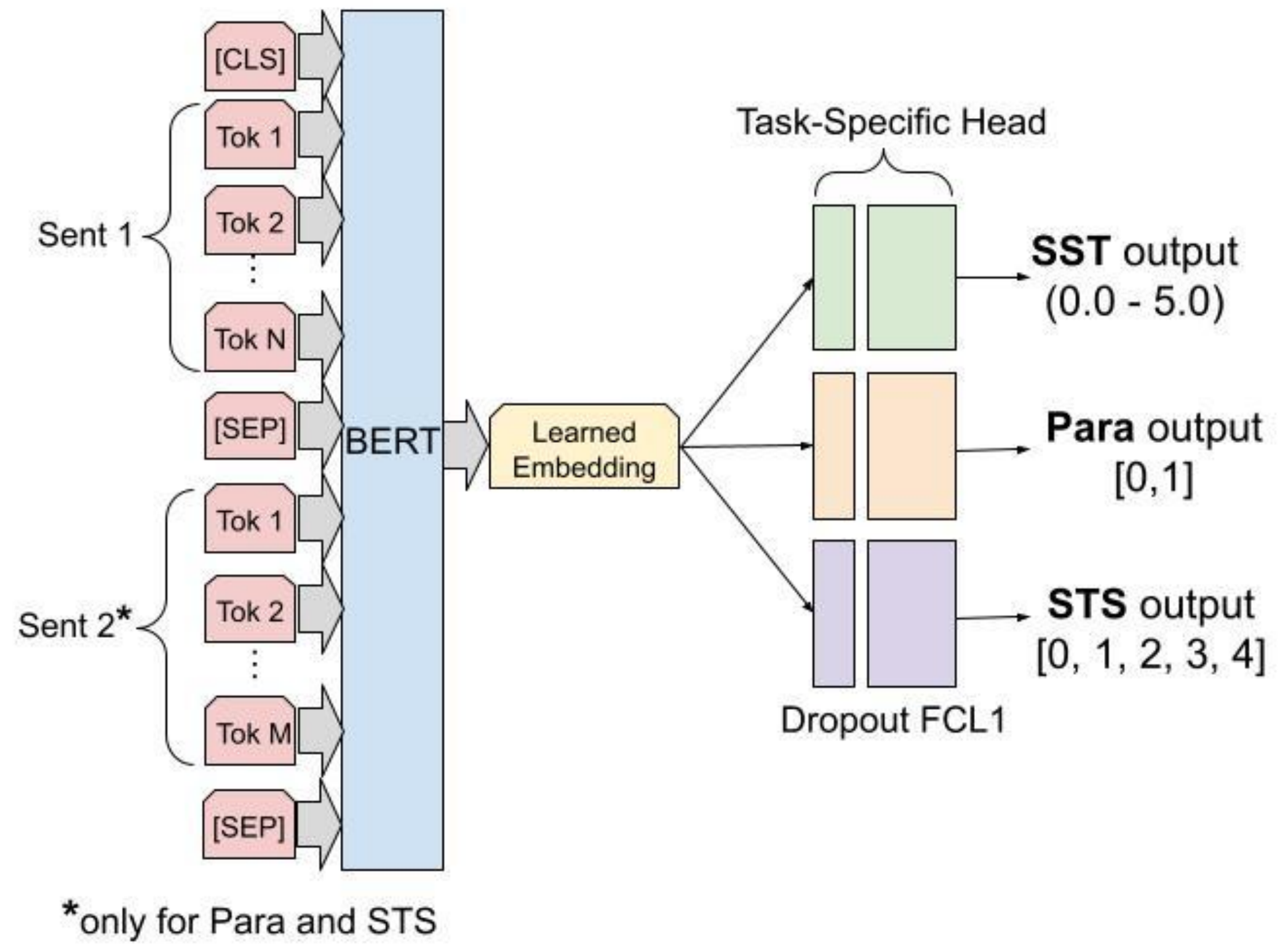
Concat After BERT (ConcatA)



*only for Para and STS

Feed each sentence from a sentence pair separately into BERT then concatenate the resulting two embeddings. One linear + one dropout layer in each task head.

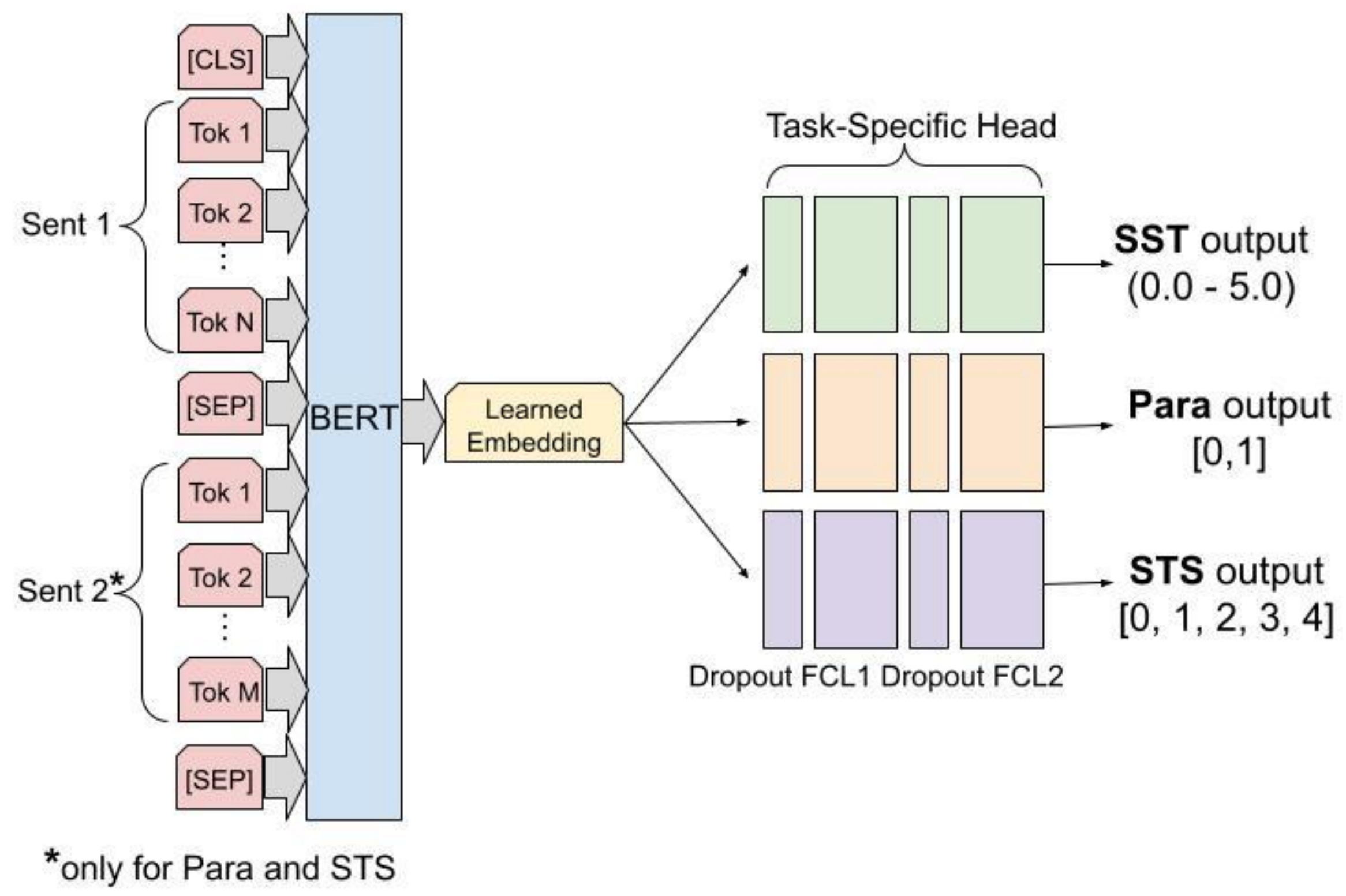
Concat Before BERT (ConcatB)



*only for Para and STS

Concatenate tokenized sentences from a pair and then feed the result into BERT to generate one embedding for both sentences. One linear + one dropout layer in each task head

Concat Before BERT + Added Layer (ConcatB + AL)



*only for Para and STS

Same as ConcatB, but there are two linear + dropout layers in each task head.

Conclusions

- ConcatB provides significant gains for the PARA and STS tasks over Concat A, but hurts SST performance. This is likely because ConcatB causes BERT to specialize in making sentence pair embeddings, but worsen in making single sentence embeddings.
- For most models, ConcatB + AL scored higher than ConcatB on average across the three tasks. This may happen since additional layers on each task head make the model more expressive and allow it to learn more task-specific features.
- NLI pretraining generally provides small bump in performance for all tasks, with the exception of SST performance in the ConcatB models. This is likely because NLI causes the model to overspecialize even further in sentence pair embeddings rather than single sentence embeddings (what SST is).
- Gradient Surgery Wrap consistently underperforms Batch Diff on STS performance. This must be because wrapping causes the model to repeatedly see the same STS examples multiple times which can lead to overfitting.
- The performance of the Final Layer fine-tuning varied. In some models, it provided a small bump in accuracy while in others, the performance decreased, potentially due to overfitting.

References

[1] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122. Association for Computational Linguistics, 2018
[2] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.
[3] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2880–2890, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
[4] Mona Diab, Tim Baldwin, and Marco Baroni, editors. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.
[5] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.