Multi-CLS BERT:
An Efficient Alternative to Traditional Ensembling

Haw-Shiuan Chang*, Ruei-Yao Sun*, Kathryn Ricci*, and Andrew McCallum

Introduction

Background:

- **Traditional ensembles** of **multiple BERT models boost performance** on natural language understanding tasks over single models
- ***However***, traditional ensembles are **expensive**
  - Computational cost, memory, space footprint

Research question:

Can we achieve the benefits of ensembling while minimizing the cost?

→ **Proposed method:**

Ensemble **multiple CLS embeddings** within a **single BERT model**

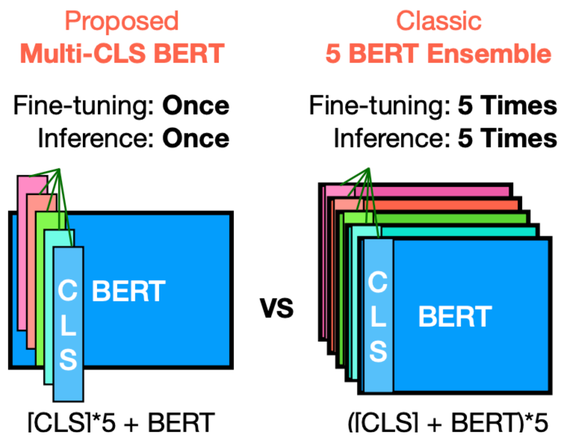


Figure 1: Proposed method.

Pretraining

- **Learn multiple, diversified CLS embeddings:**
  → Adapt state-of-the-art pretraining objectives for BERT *(Aroca-Ouellette and Rudzicz, 2020)*
- **Learn to represent fine-grained semantics:**
  → Incorporate hard negatives into the objective

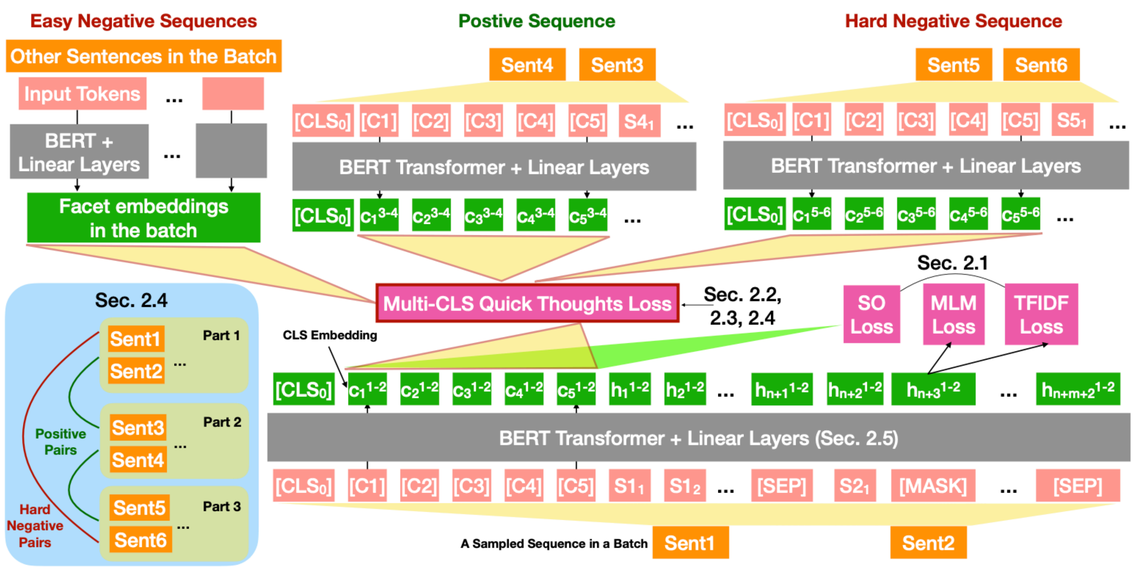


Figure 3: Pretraining method.

Architecture

- **Instead of traditional single BERT CLS embedding, leverage a fixed number of multiple CLS embeddings:**
  → Insert *K* special CLS tokens
  → Use the *K* final-layer CLS hidden states to represent the input text
  → Aggregate the *K* CLS embeddings during fine-tuning/inference
- **Prevent collapse of the multiple CLS embeddings:**
  → Insert linear layers in between selected BERT layers at multi-CLS input positions
  → Add novel parameterization

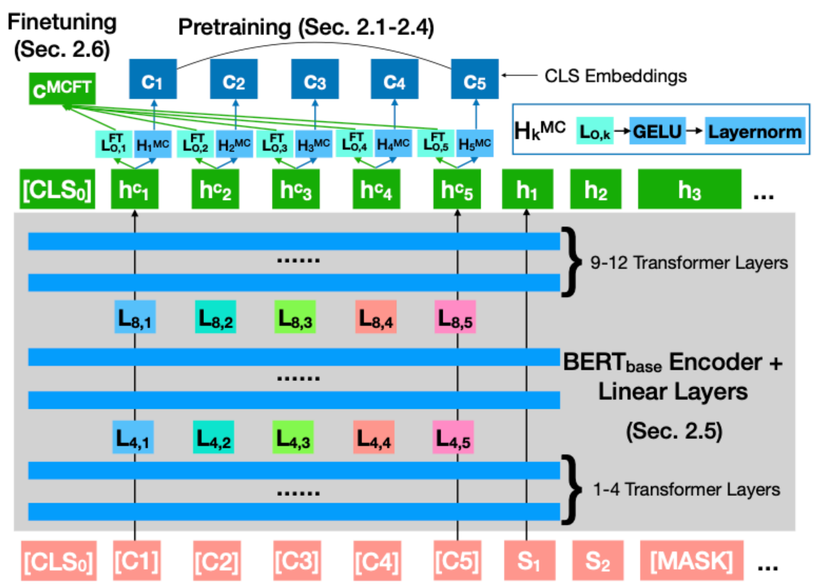


Figure 2: Architecture.

Main Result

Configuration ↓	Model Name ↓	Model Size ↓	GLUE 100	GLUE 1k	GLUE Full	SuperGLUE 100*	SuperGLUE 1k*	SuperGLUE Full
BERT Base	Pretrained	109.5M	55.71 ± 0.62	71.67 ± 0.15	82.05 ± 0.08	57.18 ± 0.43	61.55 ± 0.37	65.04 ± 0.36
	MTL	109.5M	59.29 ± 0.27	73.26 ± 0.13	83.30† ± 0.07	57.50 ± 0.41	62.94 ± 0.36	66.33 ± 0.33
	Ours (K=1)	111.3M	57.84 ± 0.32	73.28 ± 0.13	83.40 ± 0.07	57.31 ± 0.35	63.35 ± 0.18	66.29 ± 0.18
	Ours (K=5, $\lambda = 0$)	118.4M	61.54 ± 0.32	**74.14** ± 0.12	83.41 ± 0.07	58.29 ± 0.33	63.71 ± 0.26	**66.80** ± 0.25
	Ours (K=5, $\lambda = 0.1$)	118.4M	**61.80** ± 0.35	74.10 ± 0.13	**83.47** ± 0.05	58.20 ± 0.31	63.61 ± 0.27	66.74 ± 0.26
	Ours (K=5, $\lambda = 0.5$)	118.4M	60.49 ± 0.35	74.02 ± 0.12	**83.47** ± 0.08	**58.41** ± 0.38	**63.78** ± 0.25	**66.80** ± 0.24
	Ours (K=5, $\lambda = 1$)	118.4M	59.86 ± 0.34	73.75 ± 0.14	83.43 ± 0.07	57.84 ± 0.40	63.56 ± 0.22	66.39 ± 0.22
BERT Large	MTL	335.2M	61.39 ± 0.37	75.30 ± 0.27	84.13 ± 0.11	59.03 ± 0.54	65.21 ± 0.38	69.16 ± 0.37
	Ours (K=1)	338.3M	59.19 ± 0.43	75.35 ± 0.21	84.59 ± 0.07	57.35 ± 0.42	64.67 ± 0.43	69.24 ± 0.41
	Ours (K=5, $\lambda = 0$)	350.9M	63.19 ± 0.49	75.73 ± 0.26	84.51 ± 0.05	59.46 ± 0.44	65.43 ± 0.38	69.56 ± 0.31
	Ours (K=5, $\lambda = 0.1$)	350.9M	**64.24** ± 0.40	**76.27** ± 0.12	**84.61** ± 0.08	**59.88** ± 0.43	65.58 ± 0.26	**70.03** ± 0.25
	Ours (K=5, $\lambda = 0.5$)	350.9M	63.02 ± 0.42	75.95 ± 0.10	84.49 ± 0.08	59.42 ± 0.34	**65.84** ± 0.25	69.79 ± 0.25
	Ours (K=5, $\lambda = 1$)	350.9M	62.07 ± 0.45	75.85 ± 0.17	**84.61** ± 0.07	58.74 ± 0.50	65.00 ± 0.29	69.04 ± 0.27

Table 1: Results on GLUE and SuperGLUE for models derived from BERT Base and BERT Large.

Conclusion 1:
Efficient Multi-CLS BERT improves performance over baseline single BERT model and *K*=1 model with only small increase in model size.

Conclusion 2:
Multi-CLS BERT is especially effective in the few-shot setting.

Analysis

Claim 1: Increased performance is due to ensemble effects.
- *Ensembling Multi-CLS BERT only slightly boosts the performance* (Table 2)
- *Expected calibration error* (Table 4)
- *Overlap of most uncertain dataset examples* (Table 3)
- *Qualitative analysis* of nearest-neighbor embeddings (Paper appendix): Multiple CLS embeddings can learn to contribute in complementary ways to solving a task

Claim 2: Using Multi-CLS BERT vs. traditional ensemble reduces computational costs at the expense of a modest drop in performance.
- *Model size* (Table 1)
- *Inference time* (Table 4)

Model ↓	Model Description ↓	K ↓	GLUE 100	GLUE 1k	SuperGLUE* 100	SuperGLUE* 1k
Baselines (BERT Base)	Pretrained	1	56.85	71.68	57.90	62.14
	MLM only	1	55.38	70.74	57.39	61.77
	CMTL+	1	58.65	72.57	56.88	62.63
	MLM + SO + TFIDF	1	60.35	72.65	57.88	62.60
	MTL	1	59.53	73.12	57.51	62.95
Ours (BERT Base)	No Inserted Layers	1	58.06	73.18	57.97	63.34
		5	60.12	73.35	56.46	62.00
	No Hard Negative	1	58.44	73.30	57.19	63.33
		5	61.77	74.18	**58.89**	**63.86**
	Sum Aggregation	5	58.87	73.94	57.41	63.82
	Default	1	57.76	73.30	57.53	63.22
		3	61.09	73.95	57.85	63.31
		5	**62.62**	**74.49**	58.82	**63.86**
		10	60.99	73.59	58.25	62.82
	SWA	1	57.31	72.91	-	-
	Ensemble on Dropouts	1	58.45	72.86	-	-
	Ensemble on FT Seeds	1	60.07	75.20	-	-
		5	63.34	75.35	-	-
Ours (BERT Large)	No Hard Negative	1	60.36	75.69	58.47	65.04
		5	63.23	75.77	**60.33**	**65.75**
	Default	1	60.01	76.03	57.38	65.10
		5	**64.33**	**76.38**	59.99	65.51

Table 2: Ablation studies for the few-shot setting.
Conclusion: Architecture and pretraining methods improve GLUE scores (and SuperGLUE almost as consistently). Traditional ensembles can achieve somewhat higher performance.

	GLUE* 100	GLUE* 1k
Multi-CLS vs ENS	32.57	41.35
Dropout vs ENS	37.17	45.53
Least vs ENS	39.57	48.85
ENS vs ENS	38.67	50.14

Table 3: Overlap ratio of the 20% most uncertain dataset examples as predicted by the two given models ("ENS" = traditional ensemble).
Conclusion: "Multi-CLS vs. ENS" overlap ratio approaches "ENS vs. ENS".

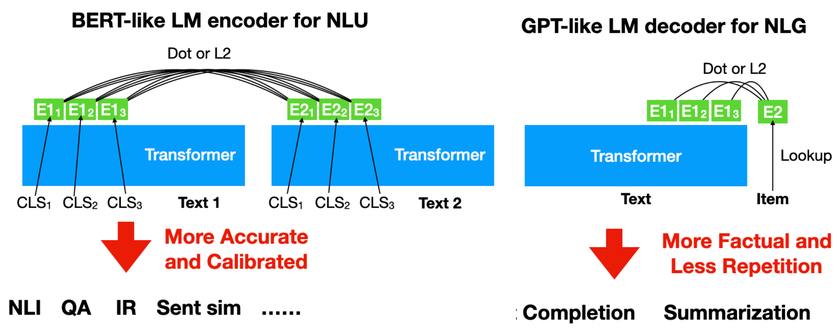
	Inference Time (s)	GLUE* (ECE) 100	GLUE* (ECE) 1k
Ours (K=1)	0.2918 ± 0.0002	25.22 ± 1.99	19.32 ± 1.64
Ours (K=5, $\lambda = 0.1$)	0.3119 ± 0.0004	**15.46** ± 1.79	**17.01** ± 1.64
Ensemble of Ours (K=1)	1.4590 ± 0.0012	13.85 ± 0.97	10.80 ± 0.88

Table 4: Inference time vs. expected calibration error (ECE).
Conclusion: Multi-CLS BERT with *K*=5 achieves significantly lower ECE than the same model with *K*=1, but with very little increase in inference time.

Conclusion

- Using multiple CLS embeddings in a single BERT model with our architecture and pretraining methodologies results in **almost free performance gain**
- Evidence suggests that performance gains are due to **ensemble effects without the cost of a traditional ensemble** of multiple models
- Our methods for successfully implementing diversified multiple CLS embeddings may be **extensible in future studies of efficient ensembling using other types of architectures**

Related Work and References



[1] Aroca-Ouellette, S. and Rudzicz, F., On Losses for Modern Language Models. EMNLP 2020
[2] H.-S. Chang, "Modeling the Multi-mode Distribution in Self-Supervised Language Models," PhD Thesis 2022
[3] H.-S. Chang, and A. McCallum, "Softmax Bottleneck Makes Language Models Unable to Represent Multi-mode Word Distributions," ACL 2022
[4] H.-S. Chang* , Z. Yao*, A. Gon, H. Yu, and A. McCallum, "**Revisiting the Architectures like Pointer Networks to Efficiently Improve the Next Word Distribution, Summarization Factuality, and Beyond**" ACL Findings 2023