Adapter-Based Contextualized Meta Embeddings

Anonymous ACL submission

Abstract

This paper introduces MetaLoRA and MetaUniPELT, two meta-embedding approaches that extends Low Rank Adaptation (LoRA) and adapters for fine-tuning and combining multiple pretrained models. We find that both models improve performance across a range of monolingual and multilingual tasks, outperforming baselines such as fully fine-tuned single models, simple concatenation of pretrained embeddings with classification layer fine-tuning and soft-voting ensembles. On the XGLUE benchmark, we find a 1.7 test score increase over the best fully-fine tuned model and a 0.24 increase over the best fully-fine tuned ensemble on sentence classification tasks. Our results underscore the potential of parameter-efficient fine-tuning of ensembles as efficient and effective alternatives to full fine-tuning and standard ensemble methods.

1 Introduction

Traditionally, the two primary approaches to utilizing pretrained language models (PLMs) (Devlin et al., 2018; Radford et al., 2018; Conneau et al., 2019; Chi et al., 2020) for downstream tasks has been either fully fine-tuning (FT) the entire model or only fine-tuning a task-specific classification layer (CFT) while the remaining layers are frozen. However, both of these approaches have their respective limitations, such as 1) significant computational costs and catastrophic forgetting (Kirkpatrick et al., 2017) of pretrained knowledge during full FT and 2) suboptimal performance in the case of CFT. Parameter efficient tuning (PEFT) methods (Rebuffi et al., 2017; Houlsby et al., 2019; Hu et al., 2021; Mao et al., 2021; Li and Liang, 2021; Pfeiffer et al., 2020a) have led to a good trade-off between FT and CFT by only tuning a small fraction of the total number of parameters. However, when the downstream task is semantically far from the self-supervised pretraining task (e.g masked language modeling), adapters struggle to meet similar performance to FT. Hence, this work aims to bridge that gap by extending PEFT methods to ensembled PLMs where the total number of parameters being tuned is still significantly less than full FT of a single model, while having the benefit of being highly parallelizable at inference time. In this paper we propose a meta-embedding method that harnesses the advantages of multiple pretrained models while mitigating the shortcomings of conventional fine-tuning strategies. We propose to integrate features from multiple pretrained models and only fine-tune a small fraction (<12.5% for UniPELT and <15% for LoRA) parameters in each of the models within the ensemble using PEFT methods. This encourages the models to adapt their embeddings to be more task-specific while benefiting from interdependent features, while still retaining the valuable knowledge gained during pretraining. These fine-tuned meta-embeddings are concatenated and processed through a classifier for both sentence-level and token-level tasks. We find that our proposed PEFT-based ensemble approach outperforms ensemble baselines and single fully-finetuned models with a relatively small number of tunable parameters for sentence-level tasks and outperforms standard ensembling on token-level tasks in both monolingual and multilingual settings

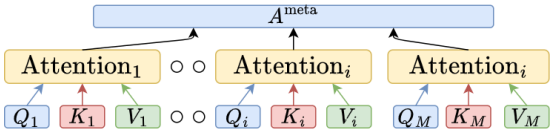

Figure 1: MetaLoRA Meta-Embeddings

2 Related Research

In this section, we revise related research on the two main aspects of this work, 1) PEFT methods that improve efficiency and scalability of PLMs

by reducing the number of trainable parameters and 2) Contextualized Meta-Embeddings (CMEs) methods that combine multiple embedding representations for improved generalization.

2.1 Parameter Efficient Fine-Tuning

Adapters Adapter tuning is a popular PEFT approach that adds a small number of additional parameters to the PLMs that are tunable and task-specific while the underlying pre-trained PLM parameters are kept frozen. The original **bottleneck adapters** (Houlsby et al., 2019) places another tunable feedforward layer in front of the pretrained feedforward layer output of each self-attention block. Recently there has been variants that experiment with the placement of adapters within the self-attention block (Pfeiffer et al., 2020c; Karimi Mahabadi et al., 2021; Pfeiffer et al., 2020b).
Low-Rank Adaptation (**LoRA**) (Hu et al., 2021) fine-tuning is another popular PEFT method that involves factorizing the weight matrices in the PLM into low-rank matrices. The low-rank matrices are then trained on the downstream task and merged with the original weight matrices after training, thus inference speed is not increased after training. Generalized LoRA (**GLoRA**) (Chavan et al., 2023) extends LoRA by using a modular layer-wise structure search that incorporates a more generalized prompt module design per layer, allowing for flexibility in fine-tuning. Other methods such as **Prefix tuning** (PT) (Li and Liang, 2021) instead fine-tunes a prefix of continuous input vectors that is prepended to the input prompt for large PLMs. Thus, unlike typical adapters, the fine-tuning is happening at the very *input* to the model while the model parameters are frozen. Lastly, **UniPELT** combines prefix tuning, bottleneck adapters and LoRA submodules in a unified setup, also using a gating mechanism that activates which submodule to use throughout training.

2.2 Contextualized Meta-Embeddings

CMEs combine multiple contextualized embeddings to produce a stronger representation for downstream tasks. Kiela et al. (2018) pioneered meta-embeddings by training an LSTM sentence encoder on top of a set of dynamically combined word embeddings. Poerner and Schütze (2019) propose a GCCA based multi-view sentence encoder that combines domain-specific and generic sentence embeddings for unsupervised Duplicate Question Detection. Neill and Bollegala (2018) have used multi-task learning via a meta-embedding reconstruction loss that is used as a secondary loss to the main supervised task loss. Patel et al. (2021) propose sentence meta-encoders in the multilingual setting and show that such frozen generic sentence embeddings provides significant benefits in supervised sentence classification when using CFT. Bollegala and O'Neill (2022) have also surveyed the SoTA in meta-embeddings for various tasks. However, these works do not explore the use of PEFT for ensemble-based models, specifically meta-embeddings.

3 Methodology

This section provides a description of our proposed approach, highlighting the effectiveness of LoRA or adapters in fine-tuning multiple PLMs for meta-embeddings, referred to as *MetaLoRA* and *MetaUniPELT*. However, in our description we assume the use of MetaLoRA for simplicity. Before describing MetaLoRA, we introduce how LoRA freezes pretrained model weights and injects trainable low-rank decomposition (LRD) matrices into each layer for a *single* model and learns the residual from the fixed pretrained weight. Assume $\mathbf{W}_0, \boldsymbol{b}_0$ are the weights and bias pretrained weights, let f be a linear layer, thus $f(\boldsymbol{x}) = \mathbf{W}_0\boldsymbol{x} + \boldsymbol{b}_0$ and $\boldsymbol{x} \in \mathbb{R}^d$ is the layer input and during training, $\mathbf{W}_0$ and $\boldsymbol{b}_0$ are frozen. The learning process is then given as Equation 1 where $\Delta\mathbf{W}$ is the learnable LRD weights.

$$f(x) = \mathbf{W}_0\boldsymbol{x} + \Delta\mathbf{W}\boldsymbol{x} + \boldsymbol{b}_0 = \mathbf{W}_{\text{LoRA}}\boldsymbol{x} + b_0 \quad (1)$$

3.1 MetaLoRA

We now describe how LoRA can be applied to an ensemble of PLMs for simulatenous LoRA fine-tuning of meta-embeddings. To begin, the pre-trained model weights of each model are frozen in the ensemble, preventing the pretrained knowledge from being destructed during fine-tuning. Then, trainable LRD matrices are inserted into each layer of each model in the ensemble. During fine-tuning, only the LRD matrices of each model are trained to minimize the loss on the downstream task, while the remaining pretrained weight matrix remains frozen. In the forward-pass the outputs of each fine-tuned model are concatenated to produce the meta-embedding, before applying the classifier on top of this embedding. Equation 2 shows the formulation of MetaLoRA for n models in the ensemble, where Concat is the concatenation operator and $g(\cdot)$ applies a concatenation of projected LoRA linear outputs for the final meta-embedding. Here, $\mathbf{W}^i_{\text{LoRA}}$

and b_0^i are the LRD weights and bias of the i-th pre-trained model in the ensemble and $\boldsymbol{x}^1, \boldsymbol{x}^2 \ldots \boldsymbol{x}^n$ are the input embeddings corresponding to n different tokenizers that are associated with each n models (the number of tokenizers is less than the number of models if more than one model shares the same tokenizer).

$$g(\boldsymbol{x}^1, \boldsymbol{x}^2 \ldots \boldsymbol{x}^n) = \text{Concat}(\mathbf{W}^1_{\text{LoRA}}\boldsymbol{x}^1 + \boldsymbol{b}^1_0, \mathbf{W}^2_{\text{LoRA}}\boldsymbol{x}^2 + \boldsymbol{b}^2_0, \ldots, \mathbf{W}^n_{\text{LoRA}}\boldsymbol{x}^n + \boldsymbol{b}^n_0) \quad (2)$$

Once the PLMs have been MetaLoRA fine-tuned, the final meta-embedding can be used for downstream tasks. In preliminary experiments, UniPELT produced the strongest results of PEFT methods (not including LoRA), hence we used MetaUniPELT in our experiments.

We note that when using meta-embeddings for token-level tasks, as in MetaLoRA, we must assume that the tokenizers for each model within the ensemble are the same. This is because having different sub-tokenization vocabularies via the same or varying tokenization schemes can lead to misaligned predictions. Thus, the models we experiment with for token-level tasks share the same tokenizer. We focus our token-level experiments on a shared tokenizer across all models, although Myer's algorithm (Myers, 1986) can be used to align different tokenizers[1]. We now describe our baseline models.

Model Baselines Below we list a description of the baselines we use in our experiments. **Single Models** - we use XLM-R$_{\text{Large}}$ (Conneau et al., 2020), InfoXLM$_{\text{Large}}$ (Chi et al., 2020) and Multilingual-E5$_{\text{Large}}$ (Wang et al., 2022) as the single model FT baselines. These are also the underlying models that we use for all other ensembles, including MetaLoRA and MetaUniPELT. **Ensemble-FT:** a fully fine-tuned ensemble that outputs averaged logits across each n models logit output, using n classification layers. **BagEnsemble-FT:** the same as Ensemble-FT except each model learns with on a unique subset of the training data using random sampling without replacement (Breiman, 1996) e.g 3 models that each only observe one third of the training data each. **Ensemble-CFT:** this is Ensemble-FT but we only fine-tune the classification layer and the remaining weights of each model are frozen. **Sentence Meta-Encoder** (SME-CFT) (Patel et al., 2021)**:** this is a meta-embedding of PLM embeddings with a single classification layer for CFT that takes the meta-embedding as input. It is similar to Ensemble-CFT except the concatenation of embeddings is done prior to passing to a *single* classification layer. **Meta-FT:** a fully fine-tuned meta-embedding i.e no PEFT methods, but also use concatenation of ensemble embeddings to a single classification layer.

[1]see https://github.com/explosion/tokenizations

Model	Mean	1:-1	1,2,3:-1,-2,-3	1	-1
Single Model					
XLM-R$_{\text{Large}}$	79.25	77.25	78.13	72.15	75.81
InfoXLM$_{\text{Large}}$	79.58	78.13	79.45	69.41	77.81
Multilingual-E5$_{\text{Large}}$	79.68	78.59	79.40	73.52	78.19
Ensemble Full-FT					
Ensemble-FT	82.60	79.53	81.93	73.90	80.41
BagEnsemble-FT	82.08	79.15	81.37	73.91	80.93
Ensemble Classification-FT					
Ensemble-CFT	75.45	70.41	72.51	68.45	71.89
SME-CFT	78.19	72.51	74.91	70.58	77.64
Proposed					
Meta-FT	83.37	79.14	81.58	75.81	81.80
MetaLoRA	79.72	78.14	79.53	73.32	78.83
MetaUniPELT	79.50	78.43	79.71	74.04	79.11

Table 1: **Varying Layer Mean Pooling on XNLI.**

	de	en	es	fr	ru	Avg.
XLM-R$_{\text{Large}}$	83.82	92.71	83.01	78.00	78.53	83.21
InfoXLM$_{\text{Large}}$	83.89	92.65	83.28	78.12	78.51	83.28
Multilingual-E5$_{\text{Large}}$	80.51	90.89	79.52	75.47	70.52	79.37
Ensemble-FT	87.09	93.08	85.71	80.69	80.67	85.45
BagEnsemble-FT	86.92	92.90	85.56	80.57	80.82	85.35
Ensemble-CFT	83.22	92.48	82.68	77.45	78.13	82.79
SME-CFT	82.33	91.90	82.10	77.14	77.52	82.19
Meta-FT	87.22	93.21	85.84	80.64	80.63	85.51
MetaLoRA	87.31	93.24	85.98	80.72	80.65	**85.57**
MetaUniPELT	84.69	92.81	84.981	81.49	82.87	85.37

Table 2: **Fine-Tuning XLM-R$_{\text{Large}}$ Results on News Classification.** Test Accuracy on English and Zero-Shot Results for German, Spanish, Russian and French.

4 Experimental Results

Sentence-level Tasks Here we analyse two of the sentence-level tasks from the XGLUE benchmark, namely XNLI and News Classification. Table 1 shows the zero-shot test results for varying sentence representations used as apart of the concatenated meta-embedding. Consistenly we find that concatenating "Mean" pooled embeddings outperforms different combinations. For example "1:-1" is the mean between the first and last hidden layers. Similarly for the remaining columns the indices include represent which subset of layers are mean pooled. We find performance varies as a function of which layers are chosen for meta-embeddings but mean pooling all layers consistently outperforms subsets of layers for XNLI. Lastly, we find MetaUniPelt outperforms single model FT baselines, SME-CFT

Model	XNLI	NC	NER	PAWSX	POS	QAM	QADSM	WPR	Sent. Av.	Token Av.	Average
Single Model											
XLM-R$_{\text{Large}}$	79.25	83.21	80.61	89.23	80.38	69.82	71.05	73.27	77.64	80.50	78.35
InfoXLM$_{\text{Large}}$	79.58	83.28	80.08	90.83	80.33	69.82	71.6	73.68	78.13	80.21	78.65
Multilingual-E5$_{\text{Large}}$	79.68	79.37	74.45	89.37	76.19	70.53	71.71	74.47	77.70	75.32	77.11
Ensemble Full-FT											
Ensemble-FT	82.6	85.45	81.15	90.50	80.93	70.28	72.94	74.88	79.46 (↑)	81.05 (↑)	79.86 (↑)
BagEnsemble-FT	82.08	85.35	**81.49**	91.52	80.86	70.06	71.33	74.09	79.07 (↑)	**81.18** (↑)	79.60 (↑)
Ensemble Classification-FT											
Ensemble-CFT	75.45	82.79	62.25	88.43	72.40	69.02	70.25	73.77	76.62 (↓)	67.34 (↓)	74.30 (↓)
SME-CFT	78.19	82.19	65.68	88.31	74.51	71.39	72.56	75.37	78.00 (↓)	70.10 (↓)	76.02 (↓)
Proposed											
Meta-FT	**83.37**	85.51	80.99	90.54	**81.21**	70.28	**73.57**	74.88	**79.70** (↑)	81.10 (↑)	**80.05** (↑)
MetaLoRA	79.72	**85.57**	72.84	**92.62**	75.35	70.41	72.68	**76.62**	79.60 (↑)	74.10 (↓)	78.73 (↑)
MetaUniPELT	79.5	85.37	72.63	92.42	75.52	**70.81**	72.43	76.42	79.49 (↑)	74.07 (↓)	78.14 (↓)

Table 3: **XGLUE benchmark zero-shot results.** Sent Av. is the average XGLUE test-set score on sentence-level tasks and Token Av. is for token-level tasks.

	de	en	es	nl	Avg.
XLM-R$_{\text{Large}}$	72.27	92.74	76.44	81.00	80.61
InfoXLM$_{\text{Large}}$	71.89	91.49	75.12	81.82	80.08
Multilingual-E5$_{\text{Large}}$	68.13	86.13	72.18	74.38	74.45
Ensemble-FT	72.83	92.91	77.01	81.82	81.15
BagEnsemble-FT	71.91	93.04	78.84	82.15	**81.49**
Ensemble-CFT	49.12	84.31	58.13	57.45	62.25
SME-CFT	40.39	81.33	55.35	55.52	58.15
Meta-FT	72.42	91.84	77.80	81.89	80.99
MetaLoRA	67.30	85.18	71.82	67.05	72.84
MetaUniPELT	67.14	86.62	71.03	65.74	72.63

Table 4: **Fine-Tuning Results on Named Entity Recognition.** Test F1 score on English and Zero-Shot Results in German, Spanish and Dutch.

is a few percentage points away from ensemble-based FT models. From Table 2 we find that both zero-shot (non-english) and standard (english) test results improve over the best single FT model baseline by 2.29% and 1.53% respectively using MetaLoRA, while MetaUniPELT achieves similar performance. In fact, MetaLoRA achieves the best performance and outperforms baselines.

Token-level Tasks Our main finding for token-level tasks (POS and NER), based on the average Token Av. in Table 3 and the Avg. in Table 4, is that since the input-output compression ratio (i.e forcing the model to compact sentences into a fixed size vector representation) is less when compared to sentence-level tasks, the benefit of meta-embeddings are diminished somewhat for token predictions. We posit, this is a reason why not fully fine-tuning ensemble approaches for token-level tasks leads to performance degradation. Moreover, POS and NER tasks are semantically far from the PLM pretraining objective. Hence, PEFT methods that do not directly update attention matrices suffer in performance. From Table 4 we find that largest gains in performance over baselines are made when evaluating on english, the language each model is trained on. This suggests that while PEFT methods result in models that generalize better than baselines in the supervised learning case for token-level tasks, they may lack some parameters to preserve zero-shot performance on the remaining languages it has not been trained on.

Overall Results Finally, Table 3 shows the average sentence-level, token-level and the overall average task *understanding* scores on the XGLUE benchmark for our proposed for our proposed MetaLoRA and MetaUniPELT and baselines. While the average token-level task score is lower, we find that based on the zero-shot cross-lingual performance that both MetaLoRA and MetaUniPELT improve over all baselines for sentence-level tasks and slightly improve performance over all tasks when comapared to fully-finetuned single models.

5 Conclusion

This work presented parameter-efficient meta-embedding models, namely MetaLoRA and MetaUniPELT. We find that these models outperform baselines on sentence embedding tasks and also outperforms classification only fine-tuned ensembles for token-level tasks. To our knowledge, this is the first focused study on how PEFT can be used in the context of ensembles. In future work, we look to improve the alignment between these models input tokenization and corresponding token embeddings for token level tasks, including generative models, not only discriminatively pretrained models. In conclusion, we find that contextualized meta-embeddings can outperform alternative ensemble approaches and full fine-tuned single models with only few tunable parameters being used.

References

Danushka Bollegala and James O'Neill. 2022. A survey on word meta-embedding learning. *arXiv preprint arXiv:2204.11660*.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24:123–140.

Arnav Chavan, Zhuang Liu, Deepak Gupta, Eric Xing, and Zhiqiang Shen. 2023. One-for-all: Generalized lora for parameter-efficient fine-tuning. *arXiv preprint arXiv:2306.07967*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Conference of the Association for Computational Linguistics, ACL 2020, Virtual Conference, July 6-8, 2020*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. *arXiv preprint arXiv:1804.07983*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2021. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*.

Eugene W Myers. 1986. An o (nd) difference algorithm and its variations. *Algorithmica*, 1(1-4):251–266.

James O' Neill and Danushka Bollegala. 2018. Meta-embedding as auxiliary task regularization. *arXiv preprint arXiv:1809.05886*.

Raj Nath Patel, Edward Burgin, Haytham Assem, and Sourav Dutta. 2021. Efficient multi-lingual sentence classification framework with sentence meta encoders. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1889–1899. IEEE.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020c. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Nina Poerner and Hinrich Schütze. 2019. Multi-view domain adapted sentence embeddings for low-resource unsupervised duplicate question detection. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1630–1641.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.