Investigating the Characteristics of a Transformer in a Few-Shot Setup: Does Freezing Layers in RoBERTa Help?

Digvijay Ingle **Rishabh Tripathi** **Ayush Kumar** **Kevin Patel** **Jithendra Vepa**
Observe.AI, India
{digvijay.ingle, rishabh.tripathi, ayush}@observe.ai
{kevin.patel, jithendra}@observe.ai

Abstract

Transformer based language models have been widely adopted by industrial and research organisations in developing machine learning applications in the presence of limited annotated data. While these models show remarkable results, their functioning in few-shot settings is still poorly understood. Hence, we perform an investigative study to understand the characteristics of such models fine-tuned in few-shot setups. Specifically, we compare the intermediate layer representations obtained from a few-shot model and a pre-trained language model. We observe that pre-trained and few-shot models show similar representations over initial layers, whereas the later layers show a stark deviation. Based on these observations, we propose to freeze the initial Transformer layers to fine-tune the model in a constrained text classification setup with K annotated data points per class, where K ranges from 8 to 64. In our experiments across six benchmark sentence classification tasks, we discover that freezing initial 50% Transformer layers not only reduces training time but also surprisingly improves Macro F1 (upto 8%) when compared to fully trainable layers in few-shot setup. We also observe that this idea of layer freezing can very well be generalized to state-of-the-art few-shot text classification techniques, like DNNC and LM-BFF, leading to significant reduction in training time while maintaining comparable performance.

1 Introduction

The immense success of pre-trained language models (PLMs), such as BERT (Devlin et al., 2019) has significantly fueled their adoption to several real world NLP applications. However, the massive parameterization of these models inherently assumes access to large training data to fine-tune them for specific tasks. Collection of such large high quality annotated data is not only time-consuming but also a costly exercise. This gives rise to a research stream specifically focused towards developing techniques that help adoption of these models in a highly constrained setting, where only a small annotated dataset is available, a setup commonly referred to as *low resource setting*. Recent years have witnessed significant advancements in popular low-resource settings like - *Weak Supervised Learning* (Zhang et al., 2022; Wang et al., 2019), *Zero-Shot Learning* (Zhong et al., 2021; Ye et al., 2022) and *Few-Shot Learning* (Brown et al., 2020; Gao et al., 2021). Despite the success of these techniques, their functioning still remains a mystery. There has been a significant amount of work done on interpretability of representations learnt by these language models in presence of large task-specific data (Phang et al., 2021; Fayyaz et al., 2021; Kumar et al., 2021). However, understanding the representations learnt in presence of few-shot examples is relatively less studied. Hence, in this paper, we attempt to compare and contrast the characteristics of representations learnt by a BERT-style language model in presence of large as well as few-shot training examples with the intention to learn better few-shot models.

Our work is primarily motivated from the findings of Phang et al. (2021), where the authors perform a study to investigate the similarities and differences between the representations learned by PLMs and task-tuned models. We replicate a similar analysis on RoBERTa-base model, where we compare the representations obtained from the PLM and those obtained by fine-tuning it on SST-2 task in an oracle setup. We refer to an oracle setup, as an ideal setting where the entire PLM is fine-tuned on a specific task in presence of a large training dataset. We use centered kernel alignment (CKA; (Kornblith et al., 2019)) to measure similarity of representations as this is also the metric used by the authors for comparison. We observe that the representations obtained from initial layers of a fine-tuned model show high degree of similarity with those obtained from a pre-trained RoBERTa-

Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 238 - 248
December 8, 2022

base model (Figure 1a). On the other hand, the representations from later layers highly deviate from the pre-trained model. This is also coherent with the observations reported by Phang et al. (2021) on ALBERT (Lan et al., 2020) model. Additionally, we compare the similarity of representations obtained from PLM to those obtained by fine-tuning it with K-shot examples, where we randomly sample $K = 8$ training examples per class for fine-tuning. We find that a similar nature of observations exists in case of K-shot model (Figure 1b), implying that most of the task-specific information is learnt in later half of the Transformer layers irrespective of the size of training data used for fine-tuning.

Thus, based on the results from Figure 1, we observe that the representations from models fine-tuned in both oracle and few-shot setups capture linguistic properties similar to that of PLMs at the initial layers. Hence, we conjecture that the initial layers can be frozen while training models in few-shot setup. We hypothesize that the reduced parameter space post such layer freezing would help learn better few-shot models.

In this work, we perform a comprehensive study of the impact of freezing specific layers while fine-tuning language models on six popular sequence classification tasks in a constrained setup where we have access to limited dataset of K annotated examples per class, where $K \in \{8, 16, 32, 64\}$. Specifically, our research contributions include:

- We show that initial 50% of the Transformer layers can be safely frozen while maintaining performance equivalent to or better than model fine-tuned with fully trainable layers in a few-shot setup. Our results indicate that this observation not only holds true for vanilla fine-tuning but also can be generalized to state-of-the-art (SOTA) few-shot techniques.

- We observe significant reductions in training time across K values for SOTA few-shot models and specifically lower K values for vanilla fine-tuning. This further helps justify our hypothesis that the reduction in parameter space due to freezing Transformer layers helps in faster convergence of the model in a few-shot setup. Moreover, the reduced training time further leads to broader environmental impact due to reduced carbon footprint (Patterson et al., 2021).

- While simply fine-tuning the classification head (100% of the Transformer layers frozen) might seem to be an intuitive choice for training few-shot models, given the significantly low size of training data, our experiments demonstrate that this strategy never helps and some proportion of Transformer layers are always required to be trainable.

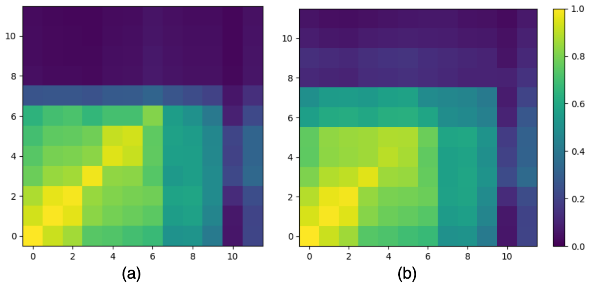

Figure 1: CKA similarity matrix based on <s> representations for: a) Pre-Trained (X-axis) vs Fine-Tuned on full training set (Y-axis), b) Pre-Trained (X-axis) vs Fine-Tuned with 8-shot examples (Y-axis)

- Most notably, while there has been a significant work on studying the representations of PLMs and the impact of freezing specific layers on a variety of NLU tasks, to the best of our knowledge, this is the first work that studies these aspects in a few-shot setup.

2 Problem Setup

2.1 Task Formulation

For the purpose of this study, we assume access to a pre-trained language model, $\mathcal{L}$. The end goal is to utilize $\mathcal{L}$ to learn a text classifier $\mathcal{M}$ for task $\mathcal{D}$ with a label space $\mathcal{C} = \{c_1, c_2, ..., c_n\}$. We further assume a training set $\mathcal{D}_{train}$ for the task $\mathcal{D}$, with only K training examples per class such that the total number of training examples, $K_{total} = K \times |\mathcal{C}|$ and $\mathcal{D}_{train} = \{x_i, y_i\}_{i=1}^{K_{total}}$. For model selection and hyper-parameter tuning, we assume a validation set $\mathcal{D}_{val}$ which is of the same size as that of the training set $\mathcal{D}_{train}$, i.e, $|\mathcal{D}_{val}| = |\mathcal{D}_{train}|$. This constraint is significantly important as it conforms to the goal of learning in a low resource setting. Finally, we assume an access to an unseen test set, $\mathcal{D}_{test} = \{x_i^{test}, y_i^{test}\}$ for evaluation of $\mathcal{M}$ on task $\mathcal{D}$. For all our experiments, unless stated otherwise, we use $\mathcal{L}$ = RoBERTa-base and $K \in \{8, 16, 32, 64\}$.

2.2 Datasets

We conduct a systematic study across three binary classification tasks - CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013) and Subj (Pang and Lee, 2004) and three multi-class classification tasks - AG News (Zhang et al., 2015), SST-5 (Socher et al., 2013) and SNIPS (Coucke et al., 2018). For AG News and Subj tasks, we do not have readily available validation sets, $\mathcal{U}_{val}$. Hence, we randomly sample 20% of the examples from the training set to create a validation set for these tasks. For CoLA, SST-2 and SST-5 tasks, we use their official validation sets. Similarly, for CoLA, SST-2 and SST-5 we do not have annotated test sets, $\mathcal{U}_{test}$. Hence, we randomly sample 10% of the examples from the training set to create unseen test sets, whereas we use the official test sets for AG News and Subj tasks. The remainder of the training set is referred to as $\mathcal{U}_{train}$. For model development, we finally obtain subsamples $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ from $\mathcal{U}_{train}$ and $\mathcal{U}_{val}$ respectively for each $K \in \{8, 16, 32, 64\}$ as described in section 2.1 such that $\mathcal{D}_{train} \subset \mathcal{U}_{train}$ and $\mathcal{D}_{val} \subset \mathcal{U}_{val}$. Note that for each of the tasks, we use a common test set for reporting model performance, i.e. $\mathcal{D}_{test} = \mathcal{U}_{test}$ for each $K \in \{8, 16, 32, 64\}$.

3 Experimental Setup

Based on the findings of Phang et al. (2021) and our experimental results in Figure 1, we investigate the impact of freezing Transformer layers on training a model in a K-shot setup, where $K \in \{8, 16, 32, 64\}$. We hypothesize that freezing particular layers would significantly reduce the parameter space which would in-turn benefit the process of fine-tuning a PLM specifically when we are operating in a constrained setup where we have access to only a limited number of annotated training examples. In order to test this hypothesis, we freeze the first $N\%$ of Transformer layers while fine-tuning $\mathcal{L}$ on task $\mathcal{D}$. Specifically, we start with $N = 0$ which resembles fully trainable Transformer layers and sequentially vary N in steps of 25. We continue this until $N = 100$ where we freeze the entire RoBERTa architecture allowing only the classification head to train. We study this setup on both vanilla fine-tuning and state-of-the-art (SOTA) few-shot techniques for fine-tuning RoBERTa-base model on each of the benchmark datasets described in Section 2.2.

3.1 Vanilla Fine-Tuning

Given a pre-trained language model $\mathcal{L}$ and text sequence x, we first obtain a tokenized sequence $\bar{x}$. Each sequence $\bar{x}$ is prefixed with a start of sentence token `<s>` and suffixed with end of sentence token `</s>`. The language model $\mathcal{L}$ is then used to map $\bar{x}$ to a sequence of hidden states h_p, such that $h_p \in \mathbb{R}^d$, where d = dimensionality of the hidden vector space. For fine-tuning the model on task $\mathcal{D}$, we add a task specific classification head, $softmax(\mathbf{W}h_{<s>})$, which returns a probability distribution over the label space $\mathcal{C}$. Here, $\mathbf{W} \in \mathbb{R}^{|\mathcal{C}| \times d}$ represents the randomly initialized weights at the start of the training, whereas $h_{<s>}$ is the hidden vector representation of `<s>` token. We further freeze $N\%$ of layers as per the approach described in Section 3. Finally, we train the entire network for a maximum of 10 epochs on a T4 GPU to minimize the cross-entropy loss. However, during training, we use early stopping criteria, where we utilize validation loss as the metric to choose the best checkpoint. Specifically, we stop the training, if validation loss does not improve for five consecutive evaluation steps. We perform a hyper-parameter sweep over the range - learning rate $\in \{1e-5, 5e-5, 1e-4\}$, batch size $\in \{4, 8, 16, 32\}$ and choose the best setting as evaluated on $\mathcal{D}_{val}$. Additionally, it is well-known that fine-tuning based on small data suffers from instability and the results may significantly vary based on choice of data split (Zhang et al., 2021). Hence, we report average performance and training times across 5 different $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ splits.

We specifically use vanilla fine-tuning approach for our experiments because it coheres with the standard fine-tuning of language models and is usually quoted as a baseline in SOTA few-shot techniques. Hence, benchmarking our methodology on vanilla fine-tuning allows us an opportunity to test the limits to which the performance of such a simple yet effective system can be pushed to.

3.2 SOTA Few-Shot Classification

In order to investigate the generalizability of our observations, we validate our experimental settings with layer freezing on following SOTA few-shot techniques. We utilize the original code-base open-sourced by the authors for the following techniques and report the results in terms of Macro F1 and training times as obtained from their training pipelines.

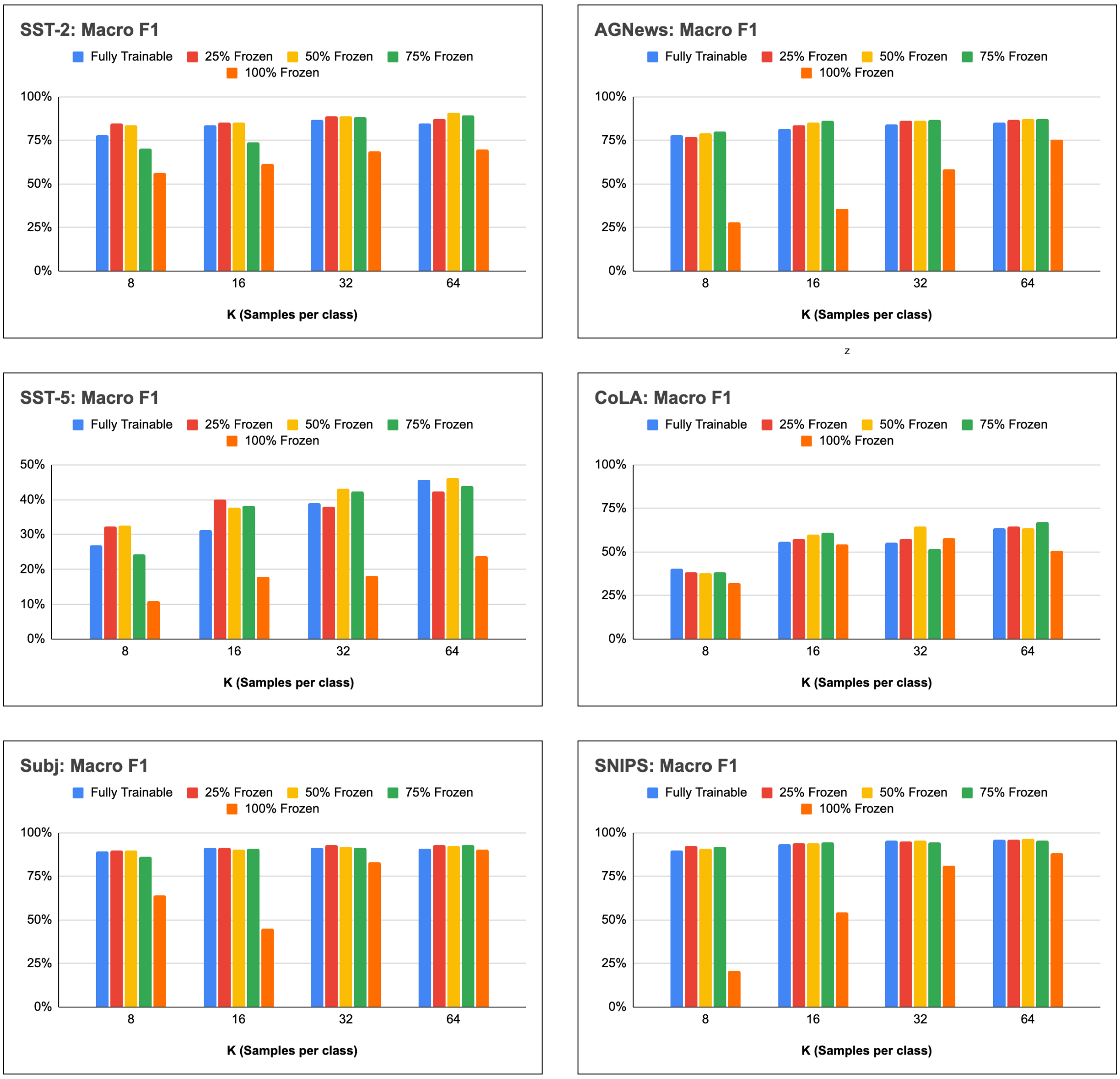

Figure 2: Comparison of Macro F1 (Y-axis) using vanilla fine-tuning post freezing of Transformer layers. In about 77% of the settings with upto 50% layers frozen, we observe an improvement in Macro F1 over fully trainable setup.

3.2.1 DNNC

DNNC (Zhang et al., 2020) is a state-of-the-art model that leverages nearest neighbor classification schema to perform few-shot text classification. Specifically, it uses the training data to create positive and negative examples that include ordered pairs of training data points belonging to the same class and different classes, respectively. It further uses BERT-style model pre-trained on natural language inference (NLI) task to fine-tune a binary classifier to estimate the best matching training example for a user input. The matched training example is then used to infer output class label. We specifically choose this model for benchmarking our setup since it is one of the commonly adopted few-shot techniques that demonstrated comparable performances in few-shot and oracle setups.

3.2.2 LM-BFF

We utilize LM-BFF (Gao et al., 2021), that uses a prompt-based approach to fine-tune a PLM in few-shot setup. A prompt refers to a human or machine generated natural language instruction that is indicative of the underlying task that a PLM is supposed to be fine-tuned on. Specifically, LM-BFF augments the input with a prompt consisting of a `<mask>` token. This re-formulates the text classification task into a masked language modelling (MLM) task, wherein $\mathcal{L}$ can be fine-tuned using MLM objective to maximize the probability

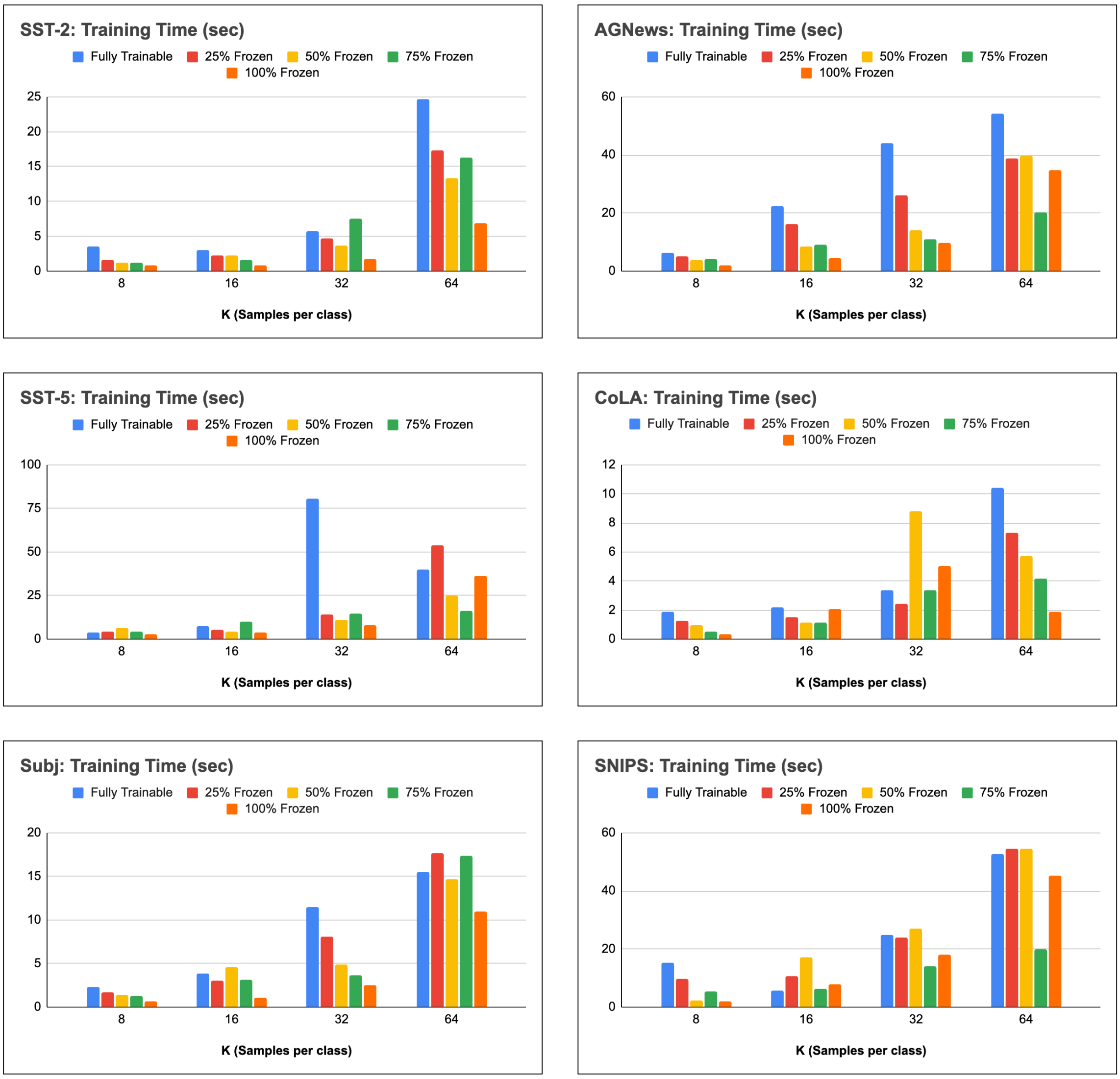

Figure 3: Comparison of training time (Y-axis) using vanilla fine-tuning post freezing of Transformer layers. In 79% of the settings with $K = 8$ and $K = 16$, we observe an improvement in training time over fully trainable setup.

of predicting the word that best resembles the task label corresponding to the input. We choose this approach for our benchmarking since it resulted in state-of-the-art performance over standard few-shot fine-tuning techniques. Moreover, prompt-based setups are becoming increasingly popular in the field of few-shot learning and benchmarking on LM-BFF allows us to validate the generalizability of our proposed method on recent approaches.

4 Results and Discussions

The results obtained from experiments with vanilla fine-tuning and few-shot classification methods have been summarized in Figures 2, 3, 4 and 5.

4.1 Effect of Freezing Layers on Vanilla Fine-Tuning

4.1.1 On Model Performance

For SST-2 task, we observe that freezing 25% and 50% of Transformer layers outperforms fully trainable setup by an absolute margin of 6% and 5% in Macro F1, respectively for $K = 8$ (refer Figure 2). A similar trend is also observed for higher values of K (=16, 32 and 64) where freezing upto 50% of Transformer layers consistently improves Macro F1 over fully trainable setup by a margin of upto 6%. This implies that the reduction in parameter space indeed benefits fine-tuning when we are operating in a few-shot setup. This further co-

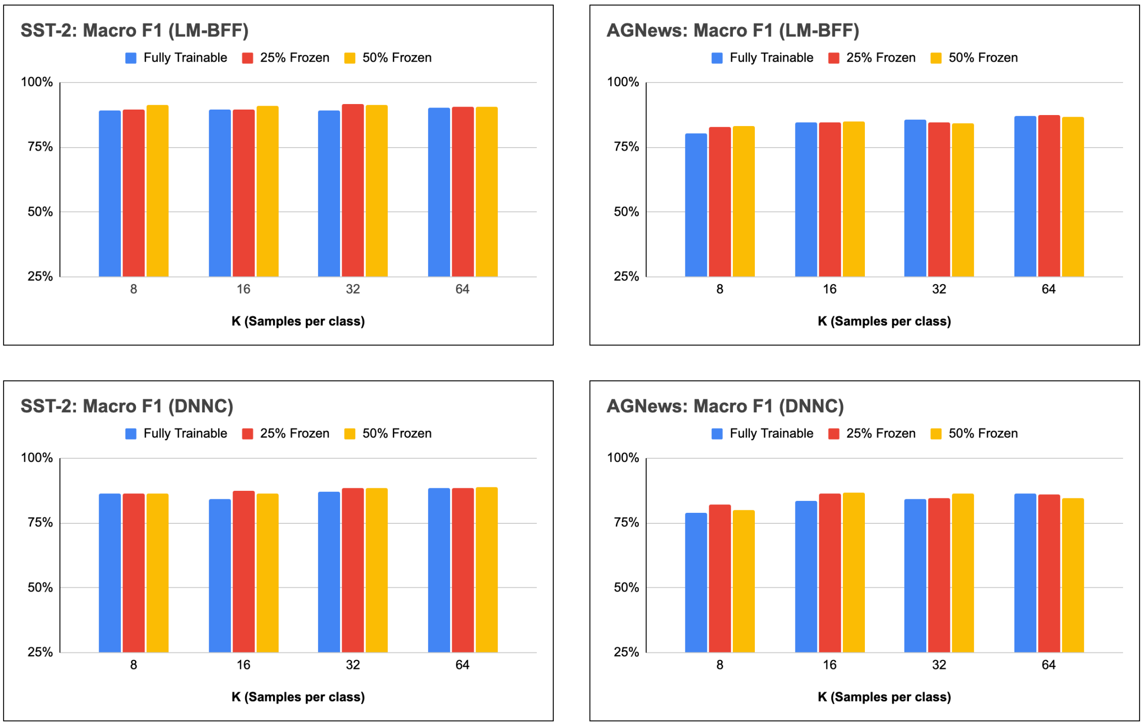

Figure 4: Comparison of Macro F1 (Y-axis) using few-shot methods post freezing of Transformer layers. In 81% of the settings experimented above, we observe an improvement in Macro F1 over fully trainable setup.

heres with our observations in Figure 1, where the representations obtained from initial layers of task-tuned model show a high degree of similarity with those obtained from the pre-trained model. Thus, making these layers trainable does not help learn any additional properties specific to the SST-2 task, instead it hurts the performance when trained in a few-shot setup. Furthermore, we observe a similar trend for other tasks (AG News, SST-5, CoLA, Subj and SNIPS) where freezing upto 50% of the transformer layers generally results in comparable or better performance. This further strengthens our claim that the first 50% of the Transformer layers can be safely frozen while fine-tuning the model in a few-shot setup.

Interestingly, further freezing of layers (>50%) starts showing a downward trend in Macro F1 over SST-2 task for $K \in \{8, 16, 32, 64\}$ implying that freezing these layers prevents the model from learning information useful for the task, which it was able to learn when only 50% of the layers were frozen. These observations are consistent with our findings in Figure 1, where the representations from later half of the Transformer layers show stark dissimilarity with those from the PLM implying that these layers are primarily responsible for learning task-specific information. Specifically, when we freeze 100% of the Transformer layers, we observe that the results show strong alignment with the above findings where it consistently leads to lower performance compared to other setups. It is a common practice to freeze the entire encoder while allowing only the classification head to be trainable while working with low resource setups. Surprisingly, our results suggest that this approach leads to sub-optimal results on all our datasets and one can achieve significantly better results with partial or no layer freezing.

On the other hand, when we freeze 75% of the layers, we see some uncertainty in the trend across tasks and K-values. We hypothesize that this could either be due to proximity to the inflection point where the behavior between similarities of representations between task-tuned and pre-trained model changes or due to certain characteristics in the similarity pattern that are peculiar to specific tasks. However, we leave this idea for further exploration as a part of future work.

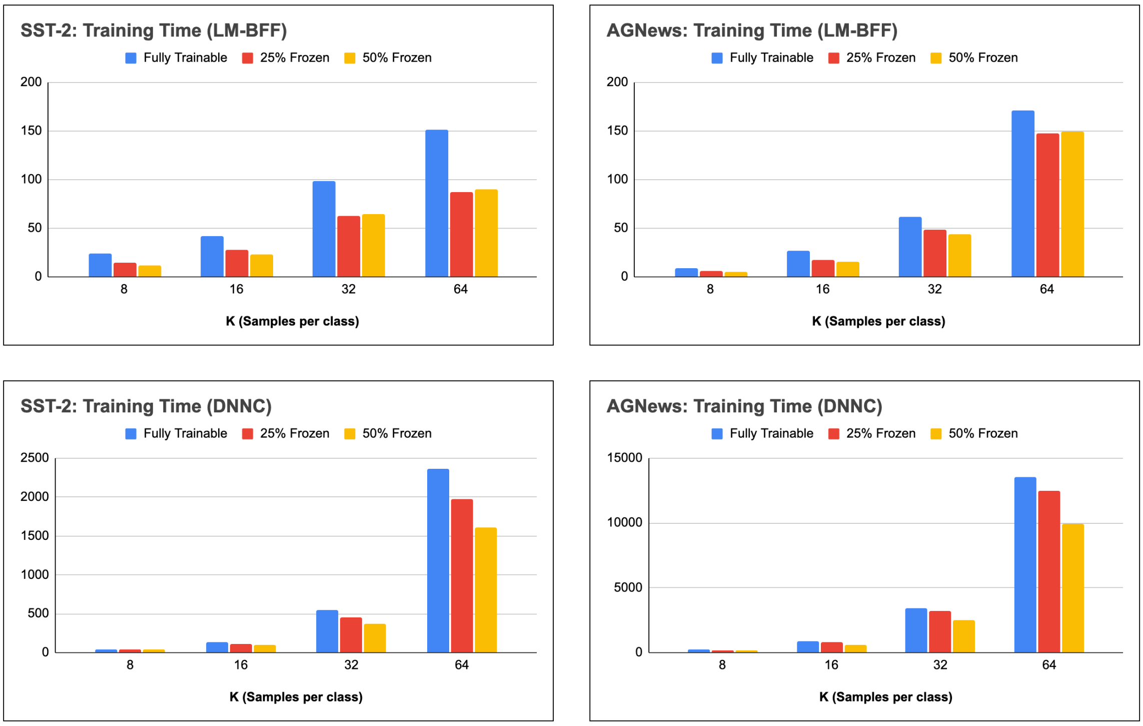

Figure 5: Comparison of training time (Y-axis) using few-shot methods post freezing of Transformer layers. In 100% of the settings experimented above, we observe an improvement in training time over fully trainable setup.

4.1.2 On Model Convergence

For SST-2 task, we observe that increasing the number of frozen layers from $0-100\%$ leads to a decreasing trend in training times for lower values of $K(=8,16)$. Since we are using early stopping criteria, this ensures that we are specifically looking for convergence of the evaluation loss. Thus, the reduction in training time is not only due to reduced computations due to layer freezing but also an effect of faster convergence due to the reduced parameter space. Moreover, we observe a similar trend for other datasets which further strengthens our claim that layer freezing results in reduced training time for vanilla fine-tuning. We do observe certain exceptions, for example higher training time on CoLA for $K=16$ with 100% frozen layers which could be due to the general instability of few-shot setups (Zhang et al., 2021; Dodge et al., 2020).

Furthermore, for higher values of K, we observe a mixed trend in training time with increasing number of frozen layers. This is primarily because higher K leads to more gradient updates leading to higher possibility of deviations from local optima thus affecting the model convergence. We consistently observe this uncertainty in training times for higher values of K across tasks.

4.2 Effects of Freezing Layers on SOTA Few-Shot Classification Techniques

We further investigate the generalizability of proposed layer freezing approach on SOTA few-shot techniques. Based on our experimental results on vanilla fine-tuning, we observe that freezing beyond 50% of the layers generally degrades performance across our experiments (Figure 2). Hence, we only experiment with freezing upto 50% of the Transformer layers in case of SOTA few-shot models. We observe a similar trend in Macro F1 where freezing upto 50% of the layers generally leads to comparable performance. Moreover, freezing layers leads to significant drop in training time across tasks and K values implying that reduced parameter space consistently helps in faster convergence even in case of SOTA few-shot techniques. Refer to figures 4 and 5 for detailed results [1].

4.3 Meta Analysis and Key Takeaways

Table 1 consolidates a summary of absolute gains in Macro F1 and training time with layer freezing for

[1] We also extend the study to CoLA and Subj tasks and observe directionally similar results

	25% Layers Frozen						50% Layers Frozen					
	SST-2											
K	Vanilla Fine-Tuning		DNNC		LM-BFF		Vanilla Fine-Tuning		DNNC		LM-BFF	
	Δ_{F1}	Δ_{Time}	Δ_{F1}	Δ_{Time}	Δ_{F1}	Δ_{Time}	Δ_{F1}	Δ_{Time}	Δ_{F1}	Δ_{Time}	Δ_{F1}	Δ_{Time}
8	8.25	56	0.01	12	0.34	39	6.89	66	0.03	21	2.33	49
16	1.60	27	3.69	16	0.11	34	1.92	26	2.18	30	1.82	45
32	2.37	19	1.47	17	2.77	36	2.62	37	1.64	32	2.40	35
64	3.57	30	-0.11	16	0.06	42	7.49	46	0.44	32	0.35	40
	AG News											
8	-1.15	18	4.00	10	3.26	29	1.93	42	1.34	27	3.57	42
16	2.56	28	3.38	8	0.07	33	4.12	62	3.75	28	0.37	42
32	2.22	41	0.17	7	-0.94	22	2.28	68	2.33	27	-1.63	30
64	1.91	28	-0.64	8	0.49	14	2.32	27	-2.04	27	-0.38	13

Table 1: Comparison of layer freezing strategy across modelling setups averaged across 5 different data splits. **How to read the table?:** Let M_N be a K-shot model fine-tuned over PLM, $\mathcal{L}$, with $N\%$ of the Transformer layers frozen. (Note that, M_0 implies a model with all Transformer layers trainable.) Say, M_0 achieves a Macro F1 of $S_0\%$ with a training time of T_0 seconds and M_N achieves a Macro F1 of $S_N\%$ with a training time of T_N seconds, where $N \in \{25, 50\}$. We quote the improvement in Macro F1 with $N\%$ layer freezing over fully trainable setup as $\Delta_{F1} = 100 \times \frac{S_N - S_0}{S_0}\%$. Additionally, we quote the improvement in training time as $\Delta_{Time} = 100 \times \frac{T_0 - T_N}{T_0}\%$. Also, a negative value of Δ_{F1} implies layer freezing degrades the performance as compared to fully trainable setup.

SST-2 and AG News tasks across our experimental setups. Following are some of the macro-level insights and takeaways from the analysis:

- In 85% (41 out of 48) of the settings we experimented with, we observe that freezing upto 50% of the layers results in performance better than fully trainable setup. Specifically, we observe that 81% (26 out of 32) of the setups with DNNC and LM-BFF outperform fully trainable setup. This further reinforces that proposed layer freezing can very well be generalized to SOTA few-shot models.

- We obtain upto 56% and 68% reduction in training time with vanilla fine-tuning and SOTA few-shot methods respectively, which reinforces that freezing transformer layers leads to faster convergence. Resulting improvement in training efficiency leads to a significant reduction in carbon footprint (Patterson et al., 2021).

- Finally, we note that the reduced parameter space due to freezing of Transformer layers prevents the representations from losing out on the universal linguistic properties learnt by the pre-trained language model due to overfitting on few-shot examples, while allowing more freedom for later layers to learn task-specific features from few-shot examples. While, we observe degradation in performance with layer freezing in 6 settings using DNNC and LM-BFF, we note that 4 of these settings deviate marginally (less than 1%). We hypothesize that this could be due to use of default hyperparameters in the training pipelines released by the authors. We believe, an exhaustive hyperparameter sweep can help eliminate these inconsistencies.

5 Prior Work

Recent years have seen significant development in the field of language modelling using Transformer (Vaswani et al., 2017) based models like GPT (Radford and Narasimhan, 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), etc. A number of studies have been conducted to identify better techniques to fine-tune these models on NLU tasks in both oracle and few-shot settings. Dodge et al. (2020), Lee et al. (2020), Zhang et al. (2021) focus on regularization techniques that help stabilize the fine-tuning of BERT-style models. Specifically, Zhang et al. (2021) demonstrate that standard process of fine-tuning for fixed epochs is sub-optimal for BERT-like models especially in few-shot setting and hence training for large epochs is required. Further, Dodge et al. (2020) show that fine-tuning on small datasets often leads to divergence during training and a simple yet effective approach like early stopping can lead to a better model selection.

Due to the tedious and time-consuming nature of data collection process, few-shot learning has recently started gaining popularity. Zhang et al. (2020) propose DNNC, a nearest neighbor classification approach that uses NLI-style training to predict if two inputs belong to the same class. Additionally, Gao et al. (2021) use a prompting based approach to fine-tune a pre-trained language model in few-shot setup leading to state-of-the-art performance without introducing any new parameters.

Parallelly, there has been a surge in works on interpretability of language models and understanding the patterns in representations learnt by them. Li et al. (2020) probe attention heads to understand certain linguistic patterns learnt by BERT. Kumar et al. (2021) design probing tasks to investigate the ability of BERT-based language models in understanding properties in spoken language. Fayyaz et al. (2021) discover different localizations of linguistic properties learnt by ELECTRA (Clark et al., 2020) and XLNET (Yang et al., 2019). On the other hand, Phang et al. (2021) perform a layer wise comparison of representations learnt by pre-trained and task-tuned models. While they perform an extensive analysis to compare the models in a setup where a large training data is available, the validation of these findings in the few-shot setup is largely unexplored.

6 Conclusions

In this work, we compare the representations obtained from intermediate Transformer layers of RoBERTa-base model and task-tuned models in few-shot setup and discover that the linguistic properties learnt by pre-trained and task-tuned models at the initial layers are very similar and hence can potentially be frozen for training models in few-shot settings. We further study the impact of such freezing of Transformer layers in a few-shot setting. Our experimental results indicate that freezing upto initial 50% of the Transformer layers surprisingly leads to performance either comparable to or better than fully trainable layers in both vanilla fine-tuning as well as SOTA few-shot models. We also observe that the reduced parameter space due to layer freezing leads to faster convergence which in turn leads to reduction in training time for 8-shot and 16-shot setups on both vanilla fine-tuning and SOTA few-shot models across tasks. Specifically, for few-shot models, this observation can even be extended to 32-shot and 64-shot setups. Moreover, layer freezing can be viewed as a medium to foster sustainable NLP research by reducing carbon footprint due to improvement in training efficiency. Finally, our results also establish that a commonly followed practice of completely frozen encoder (100% Transformer layers frozen) never helps in a few-shot setup and a proportion of Transformer layers are always required to be trainable.

7 Future Work

As discussed in Section 4.1.1, we observe an uncertainty in performance trends with 75% of the Transformer layers frozen. In future, we would like to dive deeper into understanding the potential reasons for such an uncertainty. Additionally, in this paper, we primarily focused on studying a K-shot setups with $K \in \{8, 16, 32, 64\}$, however, we believe that the idea of partially freezing Transformer layers can very well be extended to other classes of low-resource settings, like weak supervision and hence we would like to further our experiments in this direction.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2021. Not all models localize linguistic knowledge in the same place: A layer-wise probing on bertoids' representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021*, pages 375–388. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Ayush Kumar, Mukuntha Narayanan Sundararaman, and Jithendra Vepa. 2021. What BERT based language models learn in spoken transcripts: An empirical study. *CoRR*, abs/2109.09105.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020. What does BERT with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5265–5275. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, pages 271–278. ACL.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

Jason Phang, Haokun Liu, and Samuel R. Bowman. 2021. Fine-tuned transformers show clusters of similar representations across layers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021*, pages 529–538. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019. Learning with noisy labels for sentence-level sentiment classification. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 6285–6291. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguistics*, 7:625–641.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *CoRR*, abs/2202.07922.

Jian-Guo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5064–5082. Association for Computational Linguistics.

Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. 2022. A survey on programmatic weak supervision. *CoRR*, abs/2202.05433.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2856–2878. Association for Computational Linguistics.