Enhancing Transformers with Gradient Boosted Decision Trees for NLI Fine-Tuning

Benjamin Minixhofer[1]*, **Milan Gritta**[2] and **Ignacio Iacobacci**[2]
[1]Johannes Kepler University, Linz, Austria
[2]Huawei Noah's Ark Lab, London, UK
bminixhofer@gmail.com
{milan.gritta, ignacio.iacobacci}@huawei.com

Abstract

Transfer learning has become the dominant paradigm for many natural language processing tasks. In addition to models being pretrained on large datasets, they can be further trained on intermediate (supervised) tasks that are similar to the target task. For small Natural Language Inference (NLI) datasets, language modelling is typically followed by pretraining on a large (labelled) NLI dataset before fine-tuning with each NLI subtask. In this work, we explore Gradient Boosted Decision Trees (GBDTs) as an alternative to the commonly used Multi-Layer Perceptron (MLP) classification head. GBDTs have desirable properties such as good performance on dense, numerical features and are effective where the ratio of the number of samples w.r.t the number of features is low. We then introduce FreeGBDT, a method of fitting a GBDT head on the features computed during fine-tuning to increase performance without additional computation by the neural network. We demonstrate the effectiveness of our method on several NLI datasets using a strong baseline model (RoBERTa-large with MNLI pretraining). The FreeGBDT shows a consistent improvement over the MLP classification head.

1 Introduction

Recent breakthroughs in transfer learning ranging from semi-supervised sequence learning (Dai and Le, 2015) to ULMFiT (Howard and Ruder, 2018), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have brought significant improvements to many natural language processing (NLP) tasks. Transfer learning involves pretraining neural networks, often based on the Transformer (Vaswani et al., 2017), on large amounts of text in a self-supervised manner in order to learn transferable language features useful for many NLP tasks. Pretraining is followed by fine-tuning the model on the target task. Pretrained models can also be further trained on intermediate labelled datasets which are similar to the target task before the final fine-tuning stage (Pruksachatkun et al., 2020). We refer to this as *intermediate supervised pretraining*. In this manner, the network learns more meaningful internal representations of the input text that are better aligned with the target task. In order to fine-tune the pretrained network, some latent representation of the text (e.g. the hidden state corresponding to the `[CLS]` token of BERT-like models) is used as the input to a *classification head*, usually a randomly initialised Multi-Layer Perceptron (MLP) (Wolf et al., 2019). The output of the classification head can then be interpreted as probability distribution over classes. The input to the classification head is referred to as *features* throughout the paper. It is a high-dimensional vector that serves as a rich, distributed representation of the input text.

We investigate whether replacing the commonly used MLP classification head with a GBDT (Friedman, 2001) can provide a consistent improvement, using NLI tasks as our use case. GBDTs are known for strong performance on dense, numerical features (Ke et al., 2019), which includes the hidden states in a neural network. The number of input features p, i.e. the dimension of the hidden state corresponding to the `[CLS]` token is not necessarily much larger than the number of samples n (it may even be smaller). GBDTs have proven effective for tasks where $n < p$ (Kong and Yu, 2018) and can be more effective compared to logistic regression if $n \ngtr p$ (Couronné et al., 2018). Therefore, for a language model that was trained on an intermediate supervised task before fine-tuning, we hypothesise that a GBDT may be able to outperform an MLP

[*] Work conducted as Research Intern at Huawei Noah's Ark Lab.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 303–313
August 1–6, 2021. ©2021 Association for Computational Linguistics

classification head as the hidden states already encode information relevant to the target task at the start of fine-tuning. The head must learn to exploit this information exclusively during the fine-tuning stage in which the training data may consist of only a few samples. Our contributions are as follows:

- We integrate the GBDT into a near state-of-the-art (SOTA) language model as an alternative to an MLP classification head and train on the features extracted from the model *after* fine-tuning. We refer to it as standard GBDT.
- We introduce a method to train a GBDT on the features computed *during* fine-tuning, at no extra computational cost by the neural network, showing a consistent improvement over the baseline. We refer to it as **FreeGBDT**.

In the following, we recap different approaches to integrating tree-based methods with neural networks (Section 2). We introduce our FreeGBDT method in Section 3. We present our experimental setup in Section 4 and results on standard NLI benchmarks in Section 5. To conclude, we discuss improvements and limitations of our method in Section 6.

We release our code[1], implemented with LightGBM (Ke et al., 2017) and Huggingface's Transformers (Wolf et al., 2019), to the NLP community.

2 Related Work

Recent work on transfer learning in NLP has often been based on pretrained transformers, e.g. BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), T5 (Raffel et al., 2020) and RoBERTa (Liu et al., 2019). These models are pretrained on large datasets using self-supervised learning, typically a variation of language modelling such as Masked Language Modelling (MLM). MLM consists of masking some tokens as in the Cloze task (Taylor, 1953). The objective of the model is to predict the masked tokens. Recently, approaches using alternatives to MLM such as Electra (Clark et al., 2020) and Marge (Lewis et al., 2020) have also been proposed. Pretraining transformers on large datasets aims to acquire the semantic and syntactic properties of language, which can then be used in downstream tasks. The models can additionally be trained in a supervised manner on larger datasets before being fine-tuned on the target task.

Natural Language Inference is one of the most canonical tasks in Natural Language Understanding (NLU) (Nie et al., 2020; Bowman et al., 2015). NLI focuses on measuring *commonsense reasoning* ability (Davis and Marcus, 2015) and can be seen as a proxy task that estimates the amount of transferred knowledge from the self-supervised phase of training. The task involves providing a *premise* (also called *context*) and a *hypothesis* that a model has to classify as:

- *Entailment*. Given the context, the hypothesis is correct.
- *Contradiction*. Given the context, the hypothesis is incorrect.
- *Neutral*. The context neither confirms nor disconfirms the hypothesis.

The task can also be formulated as binary classification between *entailment* and *not_entailment* (*contradiction* or *neutral*). We focus on NLI as a challenging and broadly applicable NLP task, with multiple smaller evaluation datasets being available as well as the large Multi-Genre Natural Language Inference corpus (Williams et al., 2018, MNLI), which is often used for effective intermediate pretraining (Liu et al., 2019). As such, it provides a testing ground for the GBDT classification head with intermediate supervised pretraining.

Tree-based methods Models based on decision trees have a long history of applications to various machine learning problems (Breiman et al., 1984). Ensembling multiple decision trees via bagging (Breiman, 1996) or boosting (Freund et al., 1999) further improves their effectiveness and remains a popular method for modelling dense numerical data (Feng et al., 2018). Ensemble methods such as Random Forests (Breiman, 2001) and GBDTs combine predictions from many weak learners, which can result in a more expressive model compared to an MLP. There have been several approaches to combining neural networks with tree-based models, approximately divided into two groups.

1. **Heterogeneous ensembling**: The tree-based model and the neural network are trained independently, then combined via ensembling techniques. *Ensembling* refers to any method to combine the predictions of multiple models such as *stacking* (Wolpert, 1992) or an arithmetic mean of the base models' predictions.

[1] Code will be available at `https://github.com/huawei-noah/free-gbdt`.

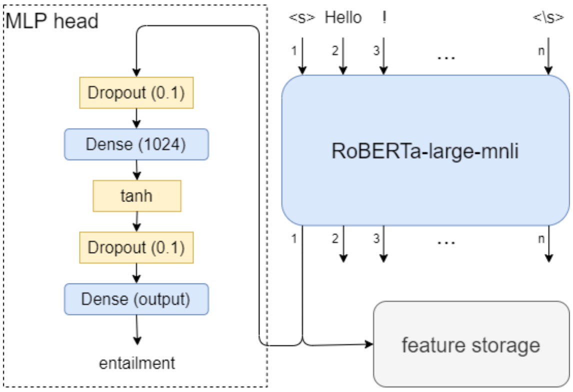

Figure 1: The baseline model architecture. Feature storage is populated *during* fine-tuning for the FreeGBDT but *after* fine-tuning for the standard GBDT method.

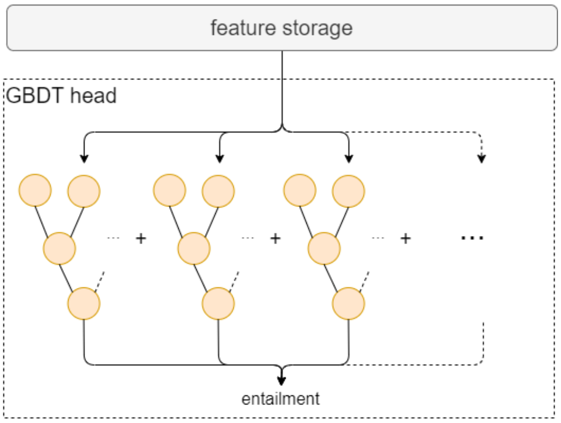

Figure 2: The GBDT classification head.

2. **Direct integration**: The tree-based model is jointly optimised with the neural network.

Heterogeneous ensembling (Li et al., 2019) has proven effective for many applications such as Online Prediction (Ke et al., 2019), Learning-to-Rank for Personal Search (Lazri and Ameur, 2018) and Credit Scoring (Xia et al., 2018). It is also suitable for multimodal inputs, e.g. text, images and/or sparse categorical features as some input types are better exploited by a neural network while others are amenable to tree-based models (Ke et al., 2019).

Direct integration makes the tree-based model compatible with back-propagation thus trainable with the neural network in an end-to-end manner. Examples include the Tree Ensemble Layer (Hazimeh et al., 2020), Deep Neural Decision Forests (Kontschieder et al., 2015) and Deep Neural Decision Trees (Yang et al., 2018). Deep Forests (Zhou and Feng, 2017) are also related although they aim to create deep non-differentiable models instead. Other examples include driving neural network fine-tuning through input perturbation (Bruch et al., 2020), which focuses specifically on using a tree ensemble to fine-tune the neural network representations.

As pretrained transformer-based models have recently achieved strong performance on various NLP tasks (Devlin et al., 2019), we see an opportunity to take advantage of their distributed representations by the means of using a tree-based model as the classification head. Our methods differ from direct integration in that they are not end-to-end differentiable. The training procedure is a sequence, i.e. the transformer-based model is fine-tuned first, then a GBDT is trained with features extracted from the model. We do not interfere with the model updates during training. Finally, the GDBT replaces the MLP classification head. Our approach is invariant to the method with which the neural network is fine-tuned as long as there exists a forward pass in which the features are computed. Recent methods for neural network fine-tuning include FreeLB (Zhu et al., 2020) and SMART (Jiang et al., 2020). FreeLB is an adversarial method, which perturbs the input during training via gradient ascent steps to improve robustness. SMART constrains the model updates during fine-tuning with smoothness-inducing regularisation in order to reduce overfitting. These approaches could theoretically be combined with both the standard GBDT and the FreeGBDT.

3 Methodology

We introduce the standard GBDT (Algorithm 1) and the FreeGBDT (Algorithm 2), our new method of using features generated during fine-tuning. *Features* refers to the hidden state corresponding to the `[CLS]` token of BERT-like pretrained models. We use these features as training data for the GBDT and FreeGBDT.

3.1 The standard GBDT classification head

In order to train the standard GBDT, we apply the feature extraction procedure shown in Algorithm 1. Using the fine-tuned neural network, we perform one additional forward pass over each sample in the training data. We store the features as training data for the GBDT, denoted ’feature storage’ in Figures 1 and 2. The GBDT can be then used as a substitute for the MLP classification head.

Algorithm 1 Standard GBDT training procedure. Features are extracted after fine-tuning.

```
Require: training data $X$, pretrained network $f_p$ parametrised by $\theta_p$, classification head $f_h$ parametrised by $\theta_h$.
  for $epoch = 1..N_{epochs}$ do
    for minibatch $(X_b, y_b) \subset X$ do
      $cls \leftarrow f_p(X_b, \theta_p)$
      $y_{pred} \leftarrow f_h(cls, \theta_h)$
      $loss \leftarrow LossFn(y_{pred}, y_b)$
      update $\theta_h$ and $\theta_p$ via backpropagation of $loss$
    end for
  end for

  $features \leftarrow$ empty list
  $labels \leftarrow$ empty list

  for minibatch $(X_b, y_b) \subset X$ do
    $cls \leftarrow f_p(X_b, \theta_p)$
    extend $features$ with $cls$
    extend $labels$ with $y_b$
  end for

  $gbdt \leftarrow train_{gbdt}(features, labels)$
```

3.2 The FreeGBDT classification head

Instead of extracting features once after fine-tuning, the training data for the proposed FreeGBDT is obtained *during* fine-tuning. The features computed in every forward pass of the neural network are stored as training data, shown in Algorithm 2. As no additional computation by the neural network is required, this new classification head is called *FreeGBDT*. Accumulating features in this manner allows the FreeGBDT to be trained on $N \times E$ samples while the standard GBDT is trained on N samples where N is the size of the dataset and E is the number of fine-tuning epochs.

Corpus	Train	Dev	Test	Classes
ANLI	162k	3.2k	3.2k	3
CNLI	6.6k	800	1.6k	3
RTE	2.5k	278	3k	2
CB	250	57	250	3
QNLI	104k	5.4k	5.4k	2

Table 1: NLI evaluation datasets. The tasks with 3 classes contain labels: *entailment*, *neutral* and *contradiction*, tasks with 2: *entailment* and *not_entailment*.

Algorithm 2 FreeGBDT training procedure. Features are accumulated throughout fine-tuning.

```
Require: training data $X$, pretrained network $f_p$ parametrised by $\theta_p$, classification head $f_h$ parametrised by $\theta_h$.
  $features \leftarrow$ empty list
  $labels \leftarrow$ empty list

  for $epoch = 1..N_{epochs}$ do
    for minibatch $(X_b, y_b) \subset X$ do
      $cls \leftarrow f_p(X_b, \theta_p)$
      extend $features$ with $cls$
      extend $labels$ with $y_b$
      $y_{pred} \leftarrow f_h(cls, \theta_h)$
      $loss \leftarrow LossFn(y_{pred}, y_b)$
      update $\theta_h$ and $\theta_p$ via backpropagation of $loss$
    end for
  end for

  $gbdt \leftarrow train_{gbdt}(features, labels)$
```

4 Experimental Setup

We now describe the featured models, details of training procedures and evaluation datasets.

4.1 Datasets

We evaluate our methods on the following NLI datasets, summarised in Table 1.

- Adversarial NLI (ANLI) (Nie et al., 2020). This corpus consists of three rounds of data collection. In each round, annotators try to break a model trained on data from previous rounds. We use the concatenation of R1, R2 and R3.

- Counterfactual NLI (CNLI) (Kaushik et al., 2020). The CNLI corpus consists of counterfactually-revised samples of SNLI (Bowman et al., 2015). We use the full dataset i.e. samples with the revised premise and with the revised hypothesis.

- Recognising Textual Entailment (RTE) (Wang et al., 2019b). We use the data and format as used in the GLUE benchmark: a concatenation of RTE1 (Dagan et al., 2006), RTE2 (Bar Haim et al., 2006), RTE3 (Giampiccolo et al., 2007) and RTE5 (Bentivogli et al., 2009), recast as a binary classification task between *entailment* and *not_entailment*.

Method	**CB**	**RTE**	**CNLI**	**ANLI**	**QNLI**
MLP head	93.57 (2.2)	89.51 (0.8)	82.49 (0.8)	57.56 (0.5)	**94.32 (0.1)**
standard GBDT	94.11 (1.7)	89.33 (0.8)	80.84 (0.9)	57.22 (0.5)	94.29 (0.1)
FreeGBDT	**94.20 (1.7)**	**89.69 (0.8)**	**82.53 (0.7)**	**57.63 (0.5)**	94.30 (0.1)

Table 2: Mean Accuracy (Standard Deviation) on the development sets from 20 runs with different random seeds. A Wilcoxon signed-rank test conducted across all five datasets confirms significance with $p \approx 0.01$ c.f. Section 5.

- CommitmentBank (CB) (de Marneffe et al., 2019). We use the subset of the data as used in SuperGLUE (Wang et al., 2019a).

- Question-answering NLI (QNLI) (Demszky et al., 2018). This is a converted version of the Stanford Q&A Dataset (Rajpurkar et al., 2016), aiming to determine whether a given context contains the answer to a question.

4.2 Model and Training

We start all experiments from the RoBERTa-large model (Liu et al., 2019) with intermediate pretraining on the Multi-Genre Natural Language Inference (MNLI) corpus (Williams et al., 2018). The MNLI checkpoint is provided by the fairseq[2] library (Ott et al., 2019). Note that no task-specific tuning of hyperparameters was performed. Instead, we use one learning rate cycle (Smith, 2017) with a maximum learning rate of 1×10^{-5} for each task to fine-tune RoBERTa for 10 epochs with a batch size of 32. We use the Adam optimiser (Kingma and Ba, 2015) to optimise the network. In order to compare the FreeGBDTs with standard GBDTs, we apply Algorithm 1 and Algorithm 2 during the same fine-tuning session to eliminate randomness from different model initialisations.

We use LightGBM[3] to train the GBDT. We do not manually shuffle the data before training. The individual trees of a GBDT are learned in a sequence where each tree is fit on the residuals of the previous trees. One important parameter of the GBDT is thus the number of trees to fit. This is commonly referred to as *boosting rounds*.

We observe that the optimal number of boosting rounds varies significantly across tasks, with a tendency towards more boosting rounds for larger datasets. Thus, we select the number of boosting rounds from the set $\{1, 10, 20, 30, 40\}$ for each task. This is the only task-specific hyperparameter in our experiments. The hyperparameters are identical for the standard GBDT and the FreeGBDT, shown in Table 3, the model shown in Figure 2. The time it takes to train the standard GBDT and the FreeGBDT is negligible compared to the time it takes to fine-tune the RoBERTa model.

Parameter	**CB**	**RTE**	**CNLI**	**ANLI**	**QNLI**
learning rate	0.1	0.1	0.1	0.1	0.1
max. leaves	256	256	256	256	256
boosting rounds	10	10	10	40	30

Table 3: Hyperparameters of the standard GBDT and the FreeGBDT. All other hyperparameters are set to the default value as per LightGBM version 2.3.1.

4.3 Evaluation

Development set We evaluate our methods using accuracy. Each experiment is repeated 20 times with different random seeds. We report the mean and standard deviation.

Test set For each task, we train the GBDT with the following boosting rounds: $\{1, 10, 20, 30, 40\}$. We select the GBDT with the best score on the development set. Test scores are obtained with a submission to the SuperGLUE benchmark[4] for CB and the GLUE benchmark[5] for RTE and QNLI. We calculate the test scores on ANLI and CNLI ourselves as the test labels are publicly available. We report accuracy on the test set for each task except for CB, where we report the mean of F1 Score and Accuracy, same as the SuperGLUE leaderboard.

5 Results and Analysis

We summarise the results on the development sets in Table 2. The FreeGBDT is compared with a standard GBDT and the MLP classification head. The standard GBDT achieves a higher score than the MLP head on 1 out of 5 tasks. The FreeGBDT outperforms the standard GBDT on 5 out of 5 tasks and the MLP head on 4 out of 5

[2] https://github.com/pytorch/fairseq
[3] https://lightgbm.readthedocs.io
[4] https://super.gluebenchmark.com/
[5] https://gluebenchmark.com/

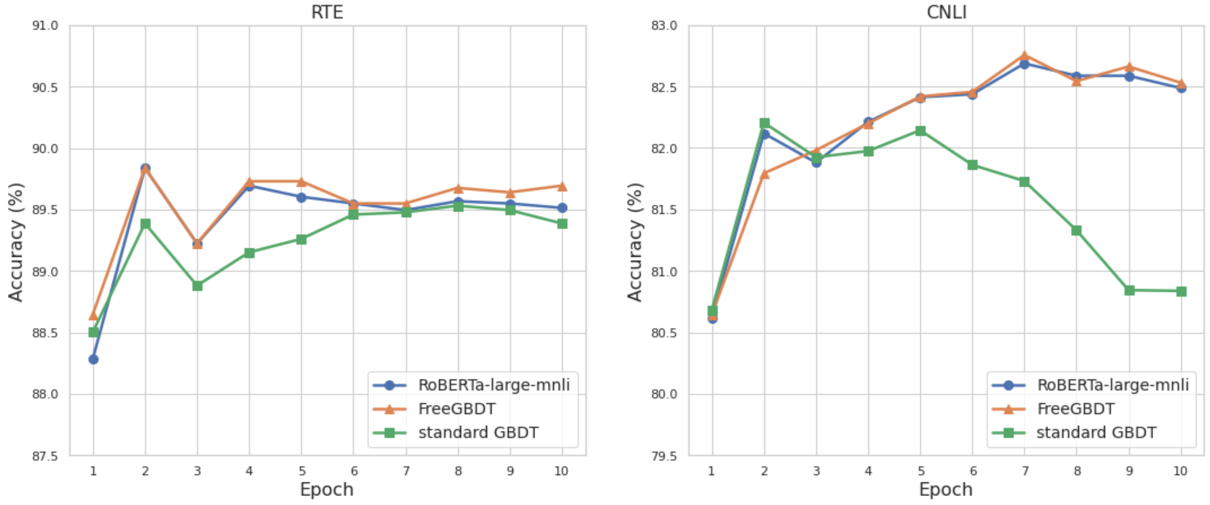

Figure 3: Accuracy on RTE and CNLI development sets. Training is paused after each epoch to compare the GBDT, FreeGBDT and MLP heads. We plot the mean from 20 runs (same hyperparameters but different seeds).

tasks. As recommended for statistical comparison of classifiers across multiple datasets (Demšar, 2006) we conduct a Wilcoxon signed-rank test (Wilcoxon, 1992) with the accuracy differences between the FreeGBDT and the MLP across the 20 seeds and 5 datasets. The test confirms that the improvement from our FreeGBDT method is significant with $p \approx 0.01$.

Method	**CB**	**RTE**	**CNLI**	**ANLI**	**QNLI**
MLP head	92.9	87.5	83.6	57.4	**94.3**
GBDT	91.3	87.5	82.1	57.2	**94.3**
FreeGBDT	**93.3**	**87.8**	**83.7**	**57.6**	**94.3**

Table 4: Results on the test sets.

Results on the test sets are shown in Table 4. The FreeGBDT achieves a small but consistent improvement over the MLP head on each task except QNLI. This task is not a conventional NLI task but a question-answering task converted to an NLI format (Demszky et al., 2018). It has been shown that QNLI does not benefit from MNLI pretraining (Liu et al., 2019) hence this result is not unexpected. Out of the four datasets which do benefit from MNLI pretraining, the FreeGBDT improves over the MLP head on each one with an average score difference of +0.23%. As our experiments start from a competitive baseline, RoBERTa-large with MNLI pretraining, we consider the results important because (a) to the best of our knowledge, this is the first tree-based method that achieves near state-of-the-art performance on benchmark NLI tasks and (b) our method is 'free' as it requires no additional computations by the model. We were able to demonstrate that a FreeGBDT head can be successfully integrated with modern transformers and is a good alternative to the commonly used MLP classification head.

For the Adversarial NLI (ANLI) dataset, we report results which are competitive with the state-of-the-art shown in Table 5, surpassing both SMART (Jiang et al., 2020) and ALUM (Liu et al., 2020). The RoBERTa-large model pretrained with SNLI, MNLI, FEVER (Thorne et al., 2018) and ANLI reported 53.7% accuracy on the ANLI test set (Nie et al., 2020). The state-of-the-art result of 58.3% accuracy on the ANLI dataset was achieved by InfoBERT (Wang et al., 2021). The FreeGBDT achieves a new state-of-the-art on the A2 subset of ANLI with 52.7%. Interestingly, it does not yield an improvement on the easier A1 subset but compares favourably to other recent approaches on the more difficult A2 and A3 subsets of ANLI.

To better understand how performance of the GBDTs evolves during fine-tuning, we carry out an additional experiment. We pause training after each epoch to extract features and train a standard GBDT. We compare it with the MLP classification head and a FreeGBDT trained on the features accumulated up to the current epoch. The result on the RTE and CNLI datasets is shown in Figure 3. Notably, the FreeGBDT does not improve on the standard GBDT after the first epoch where the number of instances the GBDT is trained on is

Method	**A1**	**A2**	**A3**	**All**
RoBERTa-large-mnli *(ours)*	72.0	52.5	49.4	57.4
RoBERTa-large-mnli + GBDT *(ours)*	72.1	52.2	49.0	57.2
SMART (Jiang et al., 2020)	72.4	49.8	**50.3**	57.1
ALUM (Liu et al., 2020)	72.3	52.1	48.4	57.0
InfoBERT (Wang et al., 2021)	**75.0**	50.5	49.8	**58.3**
RoBERTa-large-mnli + FreeGBDT *(ours)*	71.9	**52.7**	49.7	57.6

Table 5: Accuracy across different rounds of the ANLI test set. *All* denotes a sample-weighted average. Our FreeGBDT achieves SOTA on the A2 subset. InfoBERT (Wang et al., 2021) is the SOTA on the full test set.

equal to the size of the training dataset for both. As the FreeGBDT starts accumulating more training data, however, it consistently outperforms the standard GBDT and eventually, the MLP head.

The state-of-the-art in NLI provides some context for our method of combining tree-based models with modern neural networks. RoBERTa (large with MNLI pretraining) reports 89.5% accuracy on the development set of RTE (Liu et al., 2019) and 94.7% accuracy on the development set of QNLI. The same model obtains an F1 Score / Accuracy of $90.5/95.2$ on the CB test set and an accuracy of 88.2% on the RTE test set. Note that ensembles of 5 to 7 models (Liu et al., 2019) were used while our test figures achieve similar scores of $91.3/95.2$ for CB and 87.8% for RTE with a *single* model. These are not direct comparisons, however, the figures demonstrate that FreeGBDT can operate at SOTA levels while matching and exceeding the 'default' MLP head classifier accuracy. Across all datasets, the FreeGBDT improves by an average of 0.2% and 0.5% over the MLP head and the standard GBDT head, respectively.

6 Discussion

The FreeGBDT improves over the MLP head on each task where intermediate supervised pretraining on MNLI is effective. The improvement is significant but not large. This is expected since the input features of the classification head are already a highly abstract representation of the input. Thus, there is limited potential for improvement. However, our results show that a tree-based method is a viable alternative to the commonly used MLP head and can improve over a baseline chosen to be as competitive as possible. Notably, the FreeGBDT improves the MLP baseline on the CB dataset by $> 0.6\%$ solely by switching to our tree-based classification head.

Furthermore, the FreeGBDT outperforms a standard GBDT by a large margin in some cases. For instance, we observe a $+1.5\%$ improvement on the CNLI dataset. Figure 3 shows the gap forming towards the end of training. We think this may be due to overfitting to the training data. Recall that the standard GBDT is trained only on features extracted *after* fine-tuning. At this point, the features may exhibit a higher degree of memorisation of the training data. The FreeGBDT is able to mitigate this problem as it was trained with features collected throughout training. Let $f(x, \theta_t)$ denote a mapping from the input text x to the output space parameterised by θ_t where $t \in \{0..T\}$ and T is the total amount of steps the model is fine-tuned for. Then, the standard GBDT is trained on features from $f(x, \theta_T)$, while the FreeGBDT is trained on features from every t in $\{0..T\}$. As such, it may help to think of the FreeGBDT as a type of regularisation through data augmentation (from the FreeGBDT's point of view), having trained on several perturbed views of each training instance.

Figure 4 helps illustrate the regularisation effect by showing the differences between the FreeGBDT and standard GBDT training data beyond just size. The figure shows the temporal changes in a typical feature collected during fine-tuning and the same feature extracted after fine-tuning. We can see that the distribution gradually changes from earlier epochs but remains similar to the distribution of the feature at the end of fine-tuning. FreeGBDT is able to exploit the information at the start of fine-tuning as the features at $t = 0$ already encode information highly relevant to the target task hence all training data is useful. The FreeGBDT head compares favourably to an MLP head, which is randomly initialized at the start of the fine-tuning

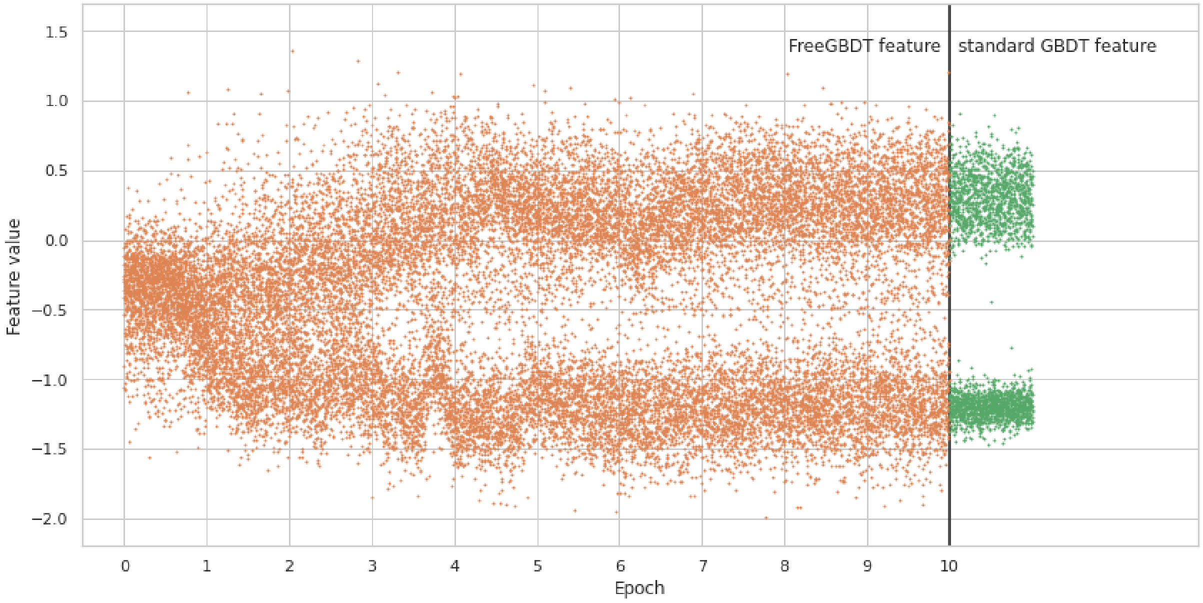

Figure 4: Values of a typical dimension from the 1,024 dimensional vector stored *during* fine-tuning to train a FreeGBDT (left). The same dimension extracted *after* fine-tuning to train a standard GBDT (right).

stage and must thus learn to exploit the latent information from a potentially small amount of training examples. Therefore, we believe intermediate supervised pretraining is essential for the effectiveness of the FreeGBDT, supported by the results from preliminary experiments on the BoolQ dataset (Clark et al., 2019) and QNLI, which does not benefit from pretraining on MNLI (Liu et al., 2019) where FreeGBDT matches the accuracy of the MLP head but does not exceed it. Our experiments also suggest that the potential for improvement from FreeGBDT depends on the size of the training dataset. The gap between FreeGBDT and the MLP head in Table 4 is larger for the smaller datasets CB and RTE and smaller for the larger datasets (ANLI, CNLI, QNLI). This is consistent with prior work showing that GBDTs are especially effective compared to other methods if $n \ggg p$ (Couronné et al., 2018) and hints that the FreeGBDT method might be especially useful for smaller datasets.

7 Future Work

One possible avenue for future work is exploring different features to train the GBDT, e.g. the hidden states from different layers of the pretrained model. This includes new combinations of top layer representations of the Transformer to generate richer input features for the classification head. This could lead to potential improvement by leveraging a less abstract representation of the input. Given that our method operates on distributed representations from a pretrained encoder, applications in other domains such as Computer Vision may be possible, e.g. using features extracted from a ResNet (He et al., 2016) encoder. Furthermore, a GBDT might not be the best choice for each task hence the use of Random Forests (Breiman, 2001) or Support Vector Machines (Cortes and Vapnik, 1995) may also be evaluated to investigate the effectiveness of combining Transformer neural networks with traditional supervised learning methods.

8 Conclusion

State-of-the-art transfer learning methods in NLP are typically based on pretrained transformers and commonly use an MLP classification head to fine-tune the model on the target task. We have explored GBDTs as an alternative classification head due to their strong performance on dense, numerical data and their effectiveness when the ratio of the number of samples w.r.t the number of features is low. We have shown that tree-based models can be successfully integrated with transformer-based neural networks and that the free training data generated during fine-tuning can be leveraged to improve model performance with our proposed FreeGBDT classification head. Obtaining consistent improvements over the MLP head on several NLI tasks confirms that tree-based learners are relevant to state-of-the-art NLP.

References

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.

Sebastian Bruch, Jan Pfeifer, and Mathieu Guillame-Bert. 2020. Learning representations for axis-aligned decision forests through input perturbation. *CoRR*, abs/2007.14761.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Raphael Couronné, Philipp Probst, and Anne-Laure Boulesteix. 2018. Random forest versus logistic regression: a large-scale benchmark experiment. *BMC bioinformatics*, 19(1):270.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58(9):92–103.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ji Feng, Yang Yu, and Zhi-Hua Zhou. 2018. Multi-layered gradient boosting decision trees. In *Advances in neural information processing systems*, pages 3551–3561.

Yoav Freund, Robert Schapire, and Naoki Abe. 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Hussein Hazimeh, Natalia Ponomareva, Petros Mol, Zhenyu Tan, and Rahul Mazumder. 2020. The tree ensemble layer: Differentiability meets conditional computation.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154.

Guolin Ke, Zhenhui Xu, Jia Zhang, Jiang Bian, and Tie-Yan Liu. 2019. Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 384–394.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yunchuan Kong and Tianwei Yu. 2018. A deep neural network model using random forest to extract feature representation for gene expression data classification. *Scientific reports*, 8(1):1–9.

Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. 2015. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision*, pages 1467–1475.

Mourad Lazri and Soltane Ameur. 2018. Combination of support vector machine, artificial neural network and random forest for improving the classification of convective and stratiform rain using spectral features of seviri data. *Atmospheric research*, 203:118–129.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. In *Advances in Neural Information Processing Systems*, volume 33, pages 18470–18481. Curran Associates, Inc.

Pan Li, Zhen Qin, Xuanhui Wang, and Donald Metzler. 2019. Combining decision trees and neural networks for learning-to-rank in personal search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2032–2040.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

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. *arXiv preprint arXiv:1907.11692*.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE.

Wilson L Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface’s transformers: State-of-the-art natural language processing.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Yufei Xia, Chuanzhe Liu, Bowen Da, and Fangming Xie. 2018. A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, 93:182–199.

Yongxin Yang, Irene Garcia Morillo, and Timothy M Hospedales. 2018. Deep neural decision trees. *arXiv preprint arXiv:1806.06988*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhi-Hua Zhou and Ji Feng. 2017. Deep forest: Towards an alternative to deep neural networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3553–3559.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.