Is BERT a Cross-Disciplinary Knowledge Learner? A Surprising Finding of Pre-trained Models' Transferability

Wei-Tsung Kao
National Taiwan University
r09942067@ntu.edu.tw

Hung-Yi Lee
National Taiwan University
hungyilee@ntu.edu.tw

Abstract

In this paper, we investigate whether the power of the models pre-trained on text data, such as BERT, can be transferred to general token sequence classification applications. To verify pre-trained models' transferability, we test the pre-trained models on (1) text classification tasks with meanings of tokens mismatches, and (2) real-world non-text token sequence classification data, including amino acid sequence, DNA sequence, and music. We find that even on non-text data, the models pre-trained on text converge faster than the randomly initialized models, and the testing performance of the pre-trained models is merely slightly worse than the models designed for the specific tasks.

1 Introduction

In recent NLP research, pre-trained masked language models (MLMs) such as BERT (Devlin et al., 2019) are widely used by practitioners. After pre-trained on large corpora such as Wikipedia, the models can be fine-tuned quickly on NLP tasks like text classification and question answering and generalize well on low-resource datasets such as RTE in GLUE (Wang et al., 2018). To apply and improve BERT in a more specialized domain such as scientific articles or clinical data, several MLMs are proposed by pre-training on the domain-specific text data (Lee et al., 2020; Beltagy et al., 2019). The concept of MLM can also be extended to other disciplines (maybe non-linguistic) as long as the input is discrete. For example, Min et al. (2019) pre-train MLMs called PLUS on amino acid sequence data and achieve state-of-the-art performance on several protein classification tasks.

This paper examines whether the model pre-trained on large text corpora, such as BERT, can be efficiently adapted to data with numbers of tokens, distribution, and structure very different from natural language (the target data could even be non-text). We refer to this ability as *discipline adaptability*[1]. Our contributions are the following.

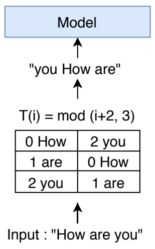
- We propose settings to examine the *discipline adaptability* of the pre-trained models.
- We find that BERT can reduce training loss much more quickly and generalize better than the randomly initialized models across disciplines, even on the non-text data. The testing performance of BERT is just slightly worse than the performance of the models designed for the non-text data. Sometimes, BERT even has better performance.
- While more pre-training steps improve the performance, we find that the first 20k steps (2% of the total steps) contribute to most of the performance gain.
- We find that embedding layer also contributes to discipline adaptability. Randomly initializing the embedding layer or using the unused tokens of BERT to fine-tune degrades the performance drastically.

The findings intrigue us to ponder what is learned in the pre-training procedure. The findings are also essential for practitioners. The discipline adaptability of BERT can be helpful to the disciplines that large scale datasets are not available.

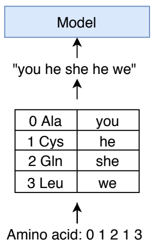
2 Method

To examine the *discipline adaptability* of the models pre-trained on text corpora, we fine-tune the pre-trained models on two types of downstream tasks. In the first type (section 2.1), the text in the downstream tasks are perturbed, so the meaning

[1]We use the term *discipline adaptation* instead of *domain adaptation*. In NLP, domain adaptation usually refers to the setting like the transfer from general text to specialized text data such as scientific articles. We use the term *discipline* to emphasize data with very different distribution and structure.



(a) *shift c, c=2*



(b) Amino acid sequence

Figure 1: Examples of (a) *shift c* setting with *c=2, D=3* (b) amino acid sequence as the input data for protein classification.

of each token is changed. The second one (section 2.2) considers a more challenging situation in which the downstream tasks are not relevant to human language.

2.1 *Token shifting* on text data

We first test the models on text sequence classification data with *token shifting*. We define a mapping table $T : \mathbb{N} \rightarrow \mathbb{N}$ that changes the i-th (subword) token in the embedding tables of the models to another token $T(i) = (i + c) \bmod D$ for some constant c and the embedding tables size D as in figure 1a. We called this setting *shift c*. For example, if we set $c = 1000$, the sentence "his healthy sense of satire is light and fun..." in GLUE dataset will be changed to "canadian franzme 1988pia leader watch sports czech at at at". In *token shifting* setting, we first pre-train the model on text corpora without token shifting, and then we fine-tune the model on the shifted training data and test the model on the shifted testing data.

Here we try to elaborate on the difference between the original text data and the shifted text data. Consider the tokens in the text corpora as nodes in a graph, while the relationship among tokens are edges. Although the original text data and shifted text data form different graphs, the graph of the shifted tokens is an isomorphism of the original graph. Hence, the structure of the shifted data is identical to the original one. Thus, performing token shifting does not change downstream tasks' difficulty (if pre-trained models are not considered). Suppose the pre-trained model can still outperform the model trained from scratch on this artificial data, then this indicates that the pre-trained models can transfer knowledge to the downstream tasks with meanings of tokens different from pre-trained corpora. And therefore, it is probable that we can further take advantage of the pre-trained model when processing non-text data, as shown in the next subsection.

2.2 Non-text data

To further validate the *discipline adaptability* of the pre-trained model, we fine-tune the pre-trained model on non-text data. In these downstream tasks, both token distributions and token numbers could be very different from the text data for pre-training. So this is a more difficult setting to evaluate the transferability of pre-trained models.

To process non-text data by BERT, we assign each token of the non-text data to one subword token of the text data as in figure 1b. Different assignments lead to similar results as long as we do not use the unused tokens of BERT. Therefore, in the following experiments, we only show the results of assigning the tokens of the non-text data to the most frequent subword tokens in the text corpora. In the fine-tuning phase, we just add a randomly initialized linear classifier on top of the pre-trained model without randomly initializing any pre-trained parameters, including the embedding layer. Then we fine-tune the whole model.

3 Experiment

3.1 Setup

We use GLUE as the text data. The validation sets are used to test the models. WNLI is excluded as in (Devlin et al., 2019); For the non-text data, we include the following tasks with different numbers of tokens, distributions, and structures:

- Protein classification (3 tasks): *Localization* (Almagro Armenteros et al., 2017), *Stability* (Rocklin et al., 2017), and *Fluorescence* (Sarkisyan et al., 2016) used in Min et al. (2019). The input is amino acid sequences consisting of 20 different tokens.

- DNA classification (4 tasks): *H3*, *H4*, *H3K9ac* from (Pokholok et al., 2005), and *Splice* from (Asuncion and Newman, 2007) used in Yin et al. (2018). The input is DNA subsequences consisting of 4 different tokens.

- Music composer classification (1 task): We use *MAESTRO-v1* dataset (Hawthorne et al., 2019). The input is pitch sequences consisting of 128 different tokens.

Table 1: Results of different models on GLUE validation set. The metric for each task is listed below the task name. "spr": spearman correlation. "rand": model trained from scratch. "re-emb": BERT with randomly initialized word embedding.

	QQP F1	QNLI acc	STS-B spr
shift 1000			
BERT	81.2	80.0	77.8
BERT (rand)	69.3	61.1	8.3
ALBERT	77.8	79.2	73.9
ALBERT (rand)	66.9	60.3	7.2
shift 1000	1% data		
BERT	66.5	72.9	57.0
BERT (rand)	54.0	54.7	4.3
normal data			
BERT (re-emb)	78.7	64.5	19.0

All the models in the experiments are the 12-layer, 768-hidden dimension transformer models (the same size as BERT-base). The models trained from scratch are initialized by the default distribution widely adopted for pre-training the models (e.g., $\mathcal{N}(0, 4 \times 10^{-4})$ for BERT). For detailed hyperparameters, please refer to the appendix.

3.2 *Token shifting* on text data

The rows under "*shift* 1000" in table 1 shows the results of pre-trained BERT, ALBERT, and the models trained from scratch under *shift* 1000 setting. Due to space limitations, we only show the results of three tasks. Results of the other tasks and different settings of *shift* are similar and left to the appendix. Even under *shift* 1000 setting, pre-trained models outperform the trained from scratch ones consistently. The phenomenon is general over different model structures and tasks.

To further examine the generalization ability, we train all the models on only 1% of the *shift* 1000 downstream data until their training loss converges to zero. The two rows under "*shift* 1000 1% data" in table 1 show that the pre-trained models still surpass the trained from scratch ones on the validation set. Therefore, the generalization ability of pre-trained models can be transferred as well.

At first sight, loading the whole pre-trained parameters of BERT and then fine-tuning BERT on the shifted text data seems equivalent to randomly initializing the word embedding of BERT and then fine-tuning BERT on the normal data, which we called re-emb. If the equivalence is true, an explanation for the performance gain under *shift* 1000 setting is just that the intermediate layers are already trained. Nevertheless, the last row in table 1 shows the equivalence does not hold. BERT (re-emb) degrades the performance of BERT in most cases. Accordingly, the pre-trained word embedding layer benefits the downstream tasks even though the meanings of the tokens are different from pre-training.

3.3 Non-text data

For the non-text data, we compare BERT not only with the models trained from scratch but also with the models designed for the corresponding non-text token sequence, which we called *discipline-specific*. For protein classification, we compare BERT with PLUS-TFM pre-trained model proposed in Min et al. (2019). For DNA classification, we compare BERT with Hilbert-CNN proposed in Yin et al. (2018). For music composer classification, we use all the classes in the dataset to classify. But previous works (Kim et al., 2020; Spijker, 2020) use only part of the classes, so no *discipline-specific* models are available.

Table 2 shows the testing performance of BERT, the models trained from scratch, and the discipline-specific models on the real-world non-text data. BERT outperforms the trained from scratch models in all disciplines. Furthermore, the performance of BERT is just slightly worse than PLUS-TFM on protein sequences and Hilbert-CNN designed for DNA sequences. Surprisingly, BERT even surpasses the discipline-specific models in two of the tasks without using any discipline-specific knowledge.

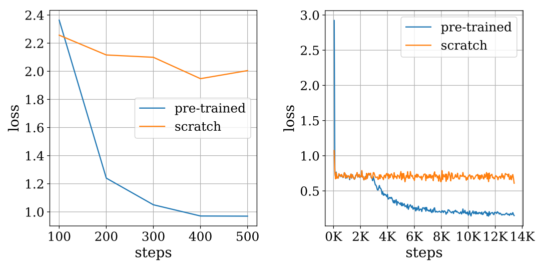
The performance of re-emb in table 2 is much worse than BERT with all pre-trained parameters (BERT in table 2). The experimental results show that the pre-trained word embedding layer also benefits the non-text downstream tasks. Also, using unused tokens of BERT even makes the performance of BERT degenerate to the trained from scratch baseline.

3.4 Training speed

The results in section 3.2 and 3.3 validate the potential of BERT as a strong cross-disciplinary knowledge learner. Now we turn to study whether BERT can learn cross-disciplinary knowledge more effi-

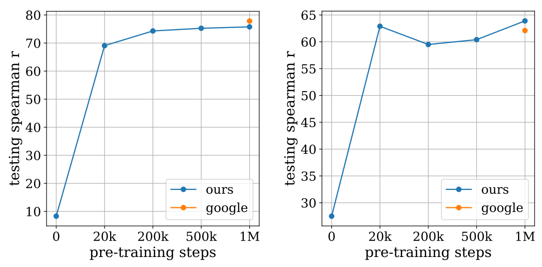
Table 2: Results of BERT on protein classification, DNA classification, and music composer classification. The metric is spearman correlation for *fluorescence* and *stability*. And the metric is accuracy for all the other tasks. "rand" and "re-emb" means BERT (rand) and BERT (re-emb). "specific": the *discipline-specific* models.

	Protein			DNA				Music
	localization	stability	fluorescence	H3	H4	H3K9ac	Splice	composer
specific	69.0	76.0	63.0	87.3	87.3	79.1	94.1	-
BERT	64.8	74.5	63.7	83.0	86.2	78.3	97.5	55.2
re-emb	63.3	75.4	37.3	78.5	83.7	76.3	95.6	55.2
rand	58.6	65.8	27.5	75.6	66.5	72.8	95	36



(a) STS-B (*shift* 1000)
(b) fluorescence

Figure 2: Training loss of BERT (blue lines) and the models trained from scratch (orange lines).



(a) STS-B (*shift* 1000)
(b) fluorescence

Figure 3: Testing performance of BERT with different pre-training steps. "google" means the official BERT.

ciently or not and dig into the training loss dynamics. Figure 2 shows that BERT always reduces the training loss faster than the models trained from scratch. Especially, for *fluorescence* task in figure 2b, the model trained from scratch seems to stuck at local minimum rapidly, while BERT gets out of local minimum as fine-tuning proceeds. For a low-resource task such as *STS-B* in figure 2a, BERT can reduce the training loss in only hundreds of steps, but the training loss of the model trained from scratch is still high. The results of the other tasks are similar and left to the appendix.

3.5 Effect of pre-training steps

While we can exploit the optimization and generalization power of BERT on the shifted text data and even the non-text data, the pre-training procedure requires many computational resources. So the question emerges: Could we get most of the performance gain but consume computational resources as limited as possible? To answer the question, we pre-train a 12-layer BERT model with the same size of pre-training data as in (Devlin et al., 2019). Then we use checkpoints of different pre-training steps as the initialization to repeat the experiment in section 3.2 and 3.3.

Figure 3 shows the testing performance of the checkpoints with 0K, 20K, 200K, 500K, 1M pre-training steps, and the performance of the official BERT. Although more pre-training steps result in better performance, the first 20K pre-training steps contribute to most of the performance gain on the shifted text data (figure 3a) and the non-text data (figure 3b). The results indicate that BERT captures *discipline adaptability* early in the pre-training procedure. Hence, it is feasible to utilize the power of BERT even with limited resources.

4 Conclusion

In this paper, we investigate the potential of BERT as a cross-disciplinary knowledge learner. By fine-tuning BERT on the text data with meanings of tokens changed and the non-text data, we verify that BERT can be adapted to data of different disciplines efficiently and generalizes well. Besides, we find that 20K pre-training steps are enough to get most of the performance gain, so it is feasible to utilize BERT as a cross-disciplinary learner with limited computational resources. We hope that the proposed settings can act as new analysis tools for researchers and provide new insight into the power of pre-trained models.

Broader impact

The results of this paper are helpful for practitioners of other disciplines when large-scale pre-training datasets are unavailable. The *discipline adaptability* of the pre-trained models also helps to reduce computational costs since we may not need to pre-train one model for each discipline. We think that the results in this paper will not cause any ethical issues.

References

José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. 2017. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395.

Arthur Asuncion and David Newman. 2007. Uci machine learning repository.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*.

Sunghyeon Kim, Hyeyoon Lee, Sunjong Park, Jinho Lee, and Keunwoo Choi. 2020. Deep composer classification using symbolic representation. *arXiv preprint arXiv:2010.00823*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Seonwoo Min, Seunghyun Park, Siwon Kim, Hyun-Soo Choi, and Sungroh Yoon. 2019. Pre-training of deep bidirectional protein sequence representations with structural information. *arXiv preprint arXiv:1912.05625*.

Dmitry K Pokholok, Christopher T Harbison, Stuart Levine, Megan Cole, Nancy M Hannett, Tong Ihn Lee, George W Bell, Kimberly Walker, P Alex Rolfe, Elizabeth Herbolsheimer, et al. 2005. Genome-wide map of nucleosome acetylation and methylation in yeast. *Cell*, 122(4):517–527.

Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. 2017. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175.

Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. 2016. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401.

BC Spijker. 2020. Classifying classical piano music into time period using machine learning. Master's thesis, University of Twente.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Bojian Yin, Marleen Balvert, Davide Zambrano, Alexander Schoenhuth, and Sander Bohte. 2018. An image representation based convolutional network for dna classification. In *International Conference on Learning Representations*.

Appendix for Is BERT a Cross-Disciplinary Knowledge Learner? A Surprising Finding of Pre-trained Models' Transferability

Wei-Tsung Kao
National Taiwan University
r09942067@ntu.edu.tw

Hung-Yi Lee
National Taiwan University
hungyilee@ntu.edu.tw

A Hyperparameters for experiments

The transformer models used in the experiments are 12-layer, 768-hidden, 12-attention heads models. The total number of parameters is 110M, which is the same size as BERT-base. PLUS-TFM has the same structure as BERT-base and the total number of parameters is 110M. For the Hilbert-CNN model, the total number of parameters is 961K according to the original paper.

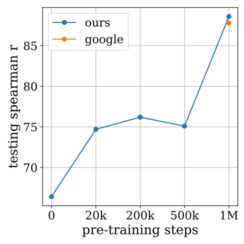
We use Adam optimizer for all experiments in the paper, and the learning rate is set to 10^{-5}. The optimizer and learning rate are chosen by applying grid search on MRPC from GLUE dataset. We search learning rate from 10^{-4} to 10^{-7}. We uniformly sample 5 points in this range, and further sample 5 points between 10^{-5} and 10^{-6}. We search optimizer including Adagrad, Adam, Adamax, RAdam, and NovoGrad for three independent runs. The parameters that make the randomly initialized transformer models achieve the highest F1 score on the training set are chosen. We do not use gradient clipping and warm-up, so the learning rate schedule is the same as linear learning rate decay. All models are trained with batch size 32 on two RTX 2080-Ti GPUs. We only use early stopping on all the non-text data since we use the validation set of GLUE as testing set.

B Full results on shifted GLUE

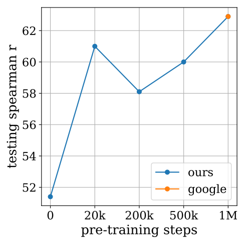
The full results on shifted GLUE dataset are shown in table 1. Among the eight tasks, the exceptions are SST-2 and CoLA. For SST-2, pre-trained models generalize worse than the models trained from scratch. For CoLA, all models fail to be trained. But for the other six tasks, pre-trained models outperform the models trained from scratch under *shift* 1000 settings. When the models are trained using only 1% data, pre-trained models generalize better than the models trained from scratch, too.

We can also consider *random shift* setting: Construct (and then fix) a list $T' = \text{shuffle}([1, 2, ..., D])$ and let $T(i) = T'(i)$. The results of *random shift* are shown in the two rows under the text "random shift" in table 1. The results are similar to the ones of *shift* 1000.

For BERT with word embedding layer randomly initialized and then fine-tuned (re-emb), the performance is worse than the one using the whole pre-trained weights, which indicate that even pre-trained word embedding is necessary.



(a) H4



(b) MAESTRO

Figure 1: Validation performance of BERT with different pre-training steps for (a) DNA classification and (b) music composer classification. "google" means the official BERT.

C Validation performance on non-text data

Table 2 shows the validation results of fine-tuning BERT on non-text data. BERT outperforms the models trained from scratch and the re-emb models on the validation set as well.

D Validation performance of different checkpoints

Figure 1 and figure 2 shows the validation results of checkpoints with different pre-training steps on the DNA classification, music composer classification, and protein classification. The results are consistent with the ones on the testing set. The first 20k pre-training steps (2% of the total steps) contribute to

Table 1: Full results on GLUE validation set. The evaluation metrics are listed below the task names. "spr" means spearman correlation. "mcc" means matthews correlation coefficients. "rand": model trained from scratch. "re-emb": BERT with randomly initialized word embedding layer and then fine-tuned. The results on MNLI are averaged over the matched and the mismatched sets.

	MNLI acc	QQP F1	QNLI acc	SST-2 acc	CoLA mcc	STS-B spr	MRPC F1	RTE acc
shift 1000								
BERT	69.3	81.2	80.0	79.6	0.0	77.8	82.9	60.2
BERT (rand)	61.8	69.3	61.1	81.0	0.0	8.3	81.3	54.2
ALBERT	67.0	77.8	79.2	76.1	0.0	73.9	83.0	56.0
ALBERT (rand)	58.9	66.9	60.3	78.6	0.0	7.2	81.2	54.0
random shift								
BERT	68.5	80.6	79.7	78.7	0.0	75.6	83.4	58.5
BERT (rand)	61.8	69.0	61.5	79.6	0.0	8.3	81.3	51.0
shift 1000	1% data							
BERT	52.7	66.5	72.9	59.3	0.0	57.0	80.5	51.5
BERT (rand)	39.2	54.0	54.7	65.3	0.0	4.3	76.2	50.8
normal data								
BERT (re-emb)	66.5	78.7	64.5	80.0	0.0	19.0	78.6	49.0

Table 2: Validation results of BERT on protein classification, DNA classification, and music composer classification. The metric is spearman correlation for *fluorescence* and *stability*. And the metric is accuracy for all the other tasks. "rand" and "re-emb" means BERT (rand) and BERT (re-emb).

	Protein			DNA				Music
	localization	stability	fluorescence	H3	H4	H3K9ac	Splice	composer
BERT	71.0	73.1	63.9	84.8	87.8	77.7	96.9	62.9
re-emb	70.0	69.2	33.4	80.0	86.2	77.9	96.2	57.1
rand	63.5	62.2	28.3	75.8	66.4	71.6	96.2	51.4

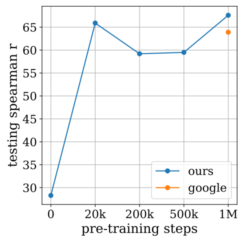


Figure 2: Validation performance of BERT with different pre-training steps for protein classification. "google" means the official BERT.

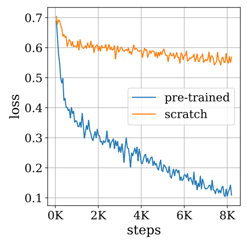
much performance gain.

E Training loss for the other disciplines

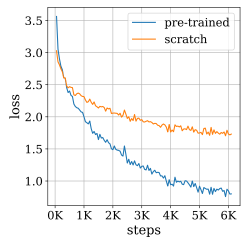
Figure 3 shows the training loss of BERT and the models trained from scratch for the DNA classification and music composer classification. BERT can reduce the training loss more quickly than the models trained from scratch. The results are consistent over disciplines.

F Effect of pre-training steps for the other disciplines

Figure 4 shows the effect of pre-training steps for DNA classification and music composer classification. The first 20k pre-training steps contribute to a lot of performance gain. Although for *H4* task, the performance gain of 20K pre-training steps is relatively small, we still get 34% gain of 1M pre-

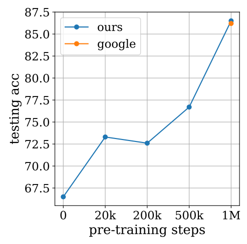


(a) H4

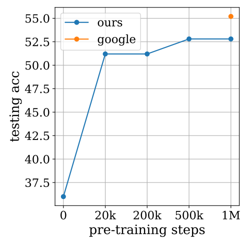


(b) MAESTRO

Figure 3: Training loss of BERT (blue line) and the models trained from scratch (orange line) on (a) DNA classification and (b) music composer classification.



(a) H4



(b) MAESTRO

Figure 4: Testing performance of BERT with different pre-training steps for (a) DNA classification and (b) music composer classification. "google" means the official BERT.

training steps using only 2% pre-training steps. For the music composer classification, the results are also consistent.

G Statistics of datasets

G.1 GLUE

GLUE is an English dataset consists of several tasks. Table 3 shows the statistics of GLUE. We use the validation set as the test set in our experiments. The train/validation split can be found in the downloaded data.

G.2 Protein classification

Table 4 shows the statistics of protein classification datasets. For pre-processing, we truncate the length of input sequences to 512.

G.3 DNA classification

Table 5 shows the statistics of DNA classification datasets. For the train/validation/test splits, we use randomly chosen 90% samples as training data, 5% samples as validation data, and 5% samples as testing data as Hilbert-CNN does. We do not apply any additional pre-processing for these datasets.

dataset	train	validation
CoLA	8551	1043
SST-2	67349	872
MRPC	3668	408
QQP	363849	40430
STS-B	5749	1500
MNLI	392702	9815/9832
QNLI	104743	5463
RTE	2490	277

Table 3: train/validation examples of GLUE dataset. The numbers of MNLI validation set are the matched subset and the mismatched subset respectively. Data can be downloaded at `https://gluebenchmark.com/`

dataset	train	validation	test
fluorescence	21446	5362	27217
stability	53614	2512	12851
localization	9977	1108	2773

Table 4: train/validation/test examples of protein classification datasets. Data can be downloaded at `http://ailab.snu.ac.kr/PLUS/`. The train/validation/test can be found in the downloaded files.

G.4 Music composer classification

Table 6 shows the statistics of MAESTRO-v1 dataset. The train/validation/test splits can be found in the downloaded files. We read the midi data and convert it to pitch sequence. For sequences longer than 128, we divide them into several segments of length 128. For training data, each segment is one training example. For validation and testing, we inference on all the segments and decide the final output by voting.

dataset	#samples
H3	14965
H4	14601
H3K9ac	27782
Splice	3190

Table 5: Numbers of samples of DNA classification datasets. Data can be downloaded at https://github.com/Doulrs/Hilbert-CNN

dataset	train	validation	test
MAESTRO-v1	954	105	125

Table 6: train/validation examples of MAESTRO-v1 dataset. Data can be downloaded at https://magenta.tensorflow.org/datasets/maestro