MCGILL U.

Replication / ML Reproducibility Challenge 2022

On the Importance of Data Size in Probing Fine-tuned Models

Sana Agarwal, Khoi Nguyen
McGill University

Reproducibility Summary

Scope of Reproducibility — The main claim of the original paper is that the size of the dataset on which a model is fine-tuned has a significant impact on its performance and the extent of encoded linguistic knowledge. Our objective is to reproduce the original experiments, and extend the results on more tasks from a different benchmark.

Methodology — To conduct our experiments, we used the probing code and some fine-tuned models provided by the authors of the original paper. We fine-tuned additional models using the SuperGLUE dataset, and then evaluated the performance of all of the models on their respective datasets. We then performed probing tasks on all of our fine-tuned models. We also performed additional fine-tuning to test the recoverability of knowledge in our models. We primarily used Colab, Kaggle and Compute Canada with approximately 261 GPU hours to complete our experiments (assuming a P100 GPU).

Results — Our experiment reproduced the results of the original paper, showing that the size of the training dataset has a significant impact on the performance of fine-tuned models on downstream tasks. We also found that the recoverability of linguistic knowledge through re-fine-tuning is dependent on the size of the dataset for the target task.

What was easy — The author's code for performing the probing tasks was easy to use, allowing us to easily reproduce the experiments and obtain similar results to those reported in the paper.

What was difficult — The large number of tasks that required a GPU made the study more difficult and time-consuming than expected. We had to use Colab and Kaggle, which presented challenges in terms of slow speeds and storage limitations. Additionally, we encountered problems when running experiments on Compute Canada servers. This required significant effort and resources to verify the results of our experiments.

Communication with original authors — We reached out to the authors to obtain the checkpoints of the fine-tuned models. They were able to provide us with some additional checkpoints that were not included in the original paper as the original checkpoints file was corrupted and could not be used.

Correspondence should be addressed to Sana Agarwal, Khoi Nguyen (sana.agarwal@mail.mcgill.ca, khoi.nguyen3@mail.mcgill.ca)
Code is available at https://github.com/KhoiTienNguyen/datasize-probing

1 Introduction

The problem addressed in the original paper is the neglect of the role of the size of the dataset in the performance of fine-tuned models. In many studies, the effectiveness of fine-tuning is studied through the lens of probing, but the size of the dataset is often overlooked. The most often-cited explanation is that fine-tuning allows models to learn domain-specific information that is not captured by pre-trained models. However, this explanation is unsatisfactory because it does not explain why fine-tuning works better than other methods of learning domain-specific information, such as training from scratch.

In the paper, the authors provide a more satisfactory explanation of why fine-tuning works. They argue that the success of fine-tuning is due to the fact that it allows models to learn from a larger amount of data than other methods. In other words, the size of the dataset is an important factor that can significantly affect the performance of fine-tuned models.

The authors used BERT on a set of tasks from the GLUE Benchmark. However, we will attempt to generalize their work by performing the same experiments on three additional tasks from the SuperGLUE benchmark, in an attempt to investigate whether the findings of the original paper can be extended to other tasks and models. This will provide a more comprehensive understanding of the role of the size of the dataset in the performance of fine-tuned models, and will help to confirm or refute the hypothesis that the success of fine-tuning is due to its ability to learn from a larger amount of data.

The main contribution of the original paper is that it highlights the importance of the size of the training dataset in the performance of fine-tuned models. The authors carried out a number of experiments to study the effect of training data size on the probing performance of fine-tuned models. Through their experiments, the authors found that the size of the dataset has a significant impact on the encoded linguistic knowledge of the model, and that the changes made to the model during fine-tuning are dependent on the number of iterations used during training. They also showed that the size of the dataset affects the recoverability of changes made to the model's linguistic knowledge. These findings suggest that when studying the linguistic knowledge of fine-tuned models, it is important to control the size of the training dataset. Additionally, the authors argue that probing accuracy may not be sufficient to fully represent the linguistic knowledge captured by fine-tuned models.

In order to extend the findings of the original paper, we performed additional experiments on three tasks from the SuperGLUE benchmark: WiC, MultiRC, and BoolQ. We fine-tuned models on these tasks using different data sizes and then performed all four probing tasks on the resulting models. Additionally, we carried out Linguistic Knowledge Recoverability experiments with models fine-tuned on GLUE using Object Number and Coordination Inversion, since the original paper only performed these experiments with Semantic Odd Man Out and Bigram Shift. Finally, we performed Linguistic Knowledge Recoverability experiments with two sequences of models fine-tuned with SuperGLUE, using BoolQ → WiC → BoolQ and WiC → BoolQ → WiC. At each step in these experiments, we also performed all four probing tasks on the models. These experiments allowed us to investigate whether the findings of the original paper can be extended to other tasks and models, and to provide a more comprehensive understanding of the role of the size of the dataset in the performance of fine-tuned models.

2 Scope of reproducibility

The main claims of the paper are that the size of the dataset on which a model is fine-tuned plays a significant role in its performance, and that the extent of encoded linguistic knowledge depends on the number of fine-tuning samples. The paper also suggests that larger training data mainly affects higher layers of the model, and that the changes made to the model's linguistic knowledge are dependent on the number of iterations used during fine-tuning. Finally, the paper presents evidence that the size of the fine-tuning data affects the recoverability of changes made to the model's linguistic knowledge.

We will be reproducing the paper using the author's original codebase. We aim to reproduce the experiments carried out by the authors in order to determine the effect of training data size on the probing performance of fine-tuned models. Then we will be replicating the results of the investigation into why data size affects the probing performance of fine-tuned models. Lastly, we will demonstrate that fine-tuning data size affects the extent to which the modifications made to a model's linguistic knowledge are recoverable.

3 Methodology

To conduct our experiments, we used the probing code and some fine-tuned models provided by the authors of the original paper. We fine-tuned additional models using the SuperGLUE dataset, and then evaluated the performance of all of the models on their respective datasets. To run the probing tasks, we transferred the models to Compute Canada servers and batch queued as many tasks as possible at a given time. We wrote the results to a text file for later analysis. For the Linguistic Knowledge Recoverability experiments, we fine-tuned additional models using both the SuperGLUE and GLUE datasets, and then sent them to the server to perform the probing experiments. We set up a shared Excel file where we could upload and share the results of the experiments. This allowed us to efficiently conduct and analyze the results of our experiments.

To begin, we first fine-tuned 10 models on different SuperGLUE tasks using different data size intervals. The author of the original paper also released 19 models trained on 5 different GLUE tasks, so we had a total of 29 models to evaluate. We evaluated each of the models on their respective tasks and recorded the results.

We divided the tasks into two categories based on the size of the data: those with more than 7k samples and those with less than 7k samples. For the tasks with more than 7k samples, we trained and evaluated 4 models with different data size intervals, for a total of 20 models. For the tasks with less than 7k samples, we trained and evaluated 3 models with different data size intervals, for a total of 9 models. This gave us a total of 29 models that we evaluated in our experiments.

Next, we performed four probing tasks from the SentEval benchmark on each of these 29 models to study the linguistic knowledge encoded in the models. This resulted in a total of 126 probing tasks. This allowed us to gain a better understanding of the performance and limitations of the fine-tuned models.

The four binary classification, syntactic and semantic probing tasks are as follows:

- **Bigram Shift**: This task tests a model's ability to identify when a sequence of two words has been shifted within a sentence.

- **Object Number**: This task tests a model's ability to identify the number of objects (singular or plural) in a sentence.
- **Coordination Inversion**: This task tests a model's ability to identify when the order of words in a coordinating phrase has been inverted.
- **Semantic Odd Man Out**: This task tests a model's ability to identify the word in a sentence that does not semantically fit with the other words.

To conduct the probing experiments, we used the code provided by the authors of the original paper. This was the only part of the experiments for which we were given code, but it was very helpful. The authors also provided us with 19 fine-tuned models, which were not code but were still very useful for our experiments.

We then also performed additional fine-tuning on a subset of these models to test their knowledge recoverability. This involved fine-tuning a model with one dataset, and then fine-tuning the resulting model with another dataset. We repeated this process 6 times with different datasets and performed 4 probing tasks on each resulting model. This resulted in an additional 12 fine-tuned models and 48 probing tasks.

Overall our methodology involved fine-tuning and evaluating a large number of models on multiple tasks, followed by probing to gain insights into their abilities. These experiments allowed us to evaluate the performance and linguistic knowledge of our models. We also performed additional fine-tuning and probing to test the recoverability of knowledge in our models. This approach allowed us to thoroughly investigate the capabilities of our models and make informed conclusions about their performance.

3.1 Model descriptions

In our experiments, we used the pre-trained bert-base-uncased transformer model from Hugging Face. BERT (Bidirectional Encoder Representations from Transformers) is a type of transformer-based model that has been widely used in natural language processing tasks [1]. It uses a combination of an encoder and a decoder to represent input text in a way that captures the underlying semantic and syntactic relationships. The model we used was pre-trained, which means that it had already been trained on a large corpus of text data, allowing us to fine-tune it for specific tasks without the need for a large amount of labeled data.

Studies have shown that pre-trained language models, like BERT, encode certain linguistic knowledge in their internal representations. Lower layers tend to encode surface-level knowledge, middle layers encode syntactic information, and higher layers capture semantic knowledge. Fine-tuning affects BERT in various ways, including its attention mode and feature extraction mode. Fine-tuning primarily affects higher layers, but can also impact lower layers depending on the downstream task. Durrani, Sajjad, and Dalvi [2] showed that fine-tuning transfers most of the model's linguistic knowledge to lower layers to reserve capacity in higher layers for task-specific knowledge.

Our probing model was built on top of the BERT model, and consisted of a dropout (0.1) layer and a dense layer for classification. The dropout layer was used to regularize the model and prevent overfitting, while the dense layer was used to predict the output class based on the encoded input text. This simple yet effective architecture allowed us to probe the linguistic knowledge encoded in the fine-tuned models.

Our BERT model has 109,482,240 parameters, making it a relatively large model. We also froze all 12 layers of BERT, resulting in a model with 109,483,778 parameters but only 1,538 trainable parameters. This allows us to analyze the impact of fine-tuning on

Benchmark	Dataset	Train	Validation	Test	Average example length	Label 0	Label 1	Label 2
GLUE	SST2	67349	872	1821	10.41	44.21	55.78	-
GLUE	COLA	8551	1043	1063	7.69	29.56	70.43	-
GLUE	MNLI	392702	9815	9796	15.01	33.33	33.33	33.33
GLUE	QQP	363846	40430	390965	11.06	63.07	36.93	-
GLUE	MRPC	3668	408	1725	21.94	32.55	67.45	-
SuperGLUE	BoolQ	9427	3270	3245	53.13	37.69	62.31	-
SuperGLUE	MultiRC	27243	4848	9693	95.12	55.86	44.14	-
SuperGLUE	WiC	5428	638	1400	7.18	50	50	-

Table 1. Statistics of datasets used in tasks from GLUE and SuperGLUE benchmarks.

BERT's linguistic knowledge. Moreover, this allows the Probing model to retain the linguistic knowledge encoded in BERT while only allowing a small number of trainable parameters to be adjusted for the specific task at hand. This design allowed us to isolate the effect of the size of the dataset on the model's performance.

The BERT model was fine-tuned on tasks from the GLUE and SuperGLUE benchmarks. The tasks used in this project include the Corpus of Linguistics Acceptability (CoLA), Microsoft Research Paraphrase Corpus (MRPC), Stanford Sentiment Treebank (SST-2), Quora Question Pairs (QQP), and Multi-Genre Natural Language Inference (MNLI) from GLUE, as well as Multi-Sentence Reading Comprehension (MultiRC), Boolean Questions (BoolQ), and Word in Context (WiC) from SuperGLUE. This allows us to evaluate the model's performance on a variety of natural language processing tasks and analyze the effect of fine-tuning on its linguistic knowledge.

3.2 Datasets

We used the GLUE and SuperGLUE benchmarks for this project. While we were reproducing the results for GLUE, we used the SuperGLUE benchmark to extend this project. The datasets used in these benchmarks are listed in Table 1, along with their statistics.

From the GLUE benchmark [3], we used:

- **SST-2** (binary classification of movie reviews as positive or negative) task dataset with 67,349 training examples, 872 validation examples, and 1,821 test examples.
- **COLA** (single sentence classification of whether a sentence is grammatically correct or not) task dataset with 8,551 training examples, 1,043 validation examples, and 1,063 test examples.
- **MNLI** (multi-class classification of the entailment relationship between two sentences) task dataset with 392,702 training examples, 9,815 validation examples, and 9,796 test examples.
- **QQP** (binary classification of whether two questions are semantically equivalent) task dataset with 363,846 training examples, 40,430 validation examples, and 390,965 test examples.
- **MRPC** (binary classification of whether two sentences are semantically equivalent) task dataset with 3,668 training examples, 408 validation examples, and 1,725 test examples.

From the SuperGLUE benchmark [4], we used:

- **BoolQ** (binary classification of natural language questions) task dataset with 9,427 training examples, 3,270 validation examples, and 3,245 test examples.

- **MultiRC** (multiple-choice reading comprehension) task dataset with 27,243 training examples, 4,848 validation examples, and 9,693 test examples. It is important to note here that when we fine-tuned our model on the Jiant platform, we only had access to 5,500 training samples. This is because the Jiant platform divides the training set differently than the original SuperGlue benchmark, resulting in a smaller number of samples for the MultiRC task.
- **WiC** (word-in-context classification) task dataset with 5,428 training examples, 638 validation examples, and 1,400 test examples.

The datasets used in the GLUE and SuperGlue benchmarks were collected in different ways, depending on the specific task. Some of the datasets, such as MNLI, were originally created by crowdsourcing [4], where a large number of people were asked to annotate text data according to specific guidelines. We used the datasets from GLUE and SuperGLUE as provided by Hugging Face datasets library.

3.3 Hyperparameters

In the original paper, the author used a learning rate of 5e-5 and a batch size of 16 for fine-tuning and training the probe model. Multiple runs were performed for each experiment with different seeds, and the results were averaged. We however, did not conduct a hyperparameter search, and only ran each experiment once. For fine-tuning and probing training, the author used a different seed for each run. We used a seed of 42 for fine-tuning and 60 for probing training, which are seeds also used by the author.

Due to a mistake, we used a batch size of 128 to fine-tune BoolQ models. These models were used for the probing accuracy data and graphs. Due to time constraints, we were only able to fine-tune again the BoolQ model trained with the full dataset using batch size of 16. This model was used for knowledge recoverability, since the other knowledge models were also fine-tuned with a batch size of 16.

3.4 Experimental setup and code

We set up several experiments to evaluate the performance of natural language processing (NLP) models. The first set of experiments involved fine-tuning tasks from GLUE using Hugging Face's datasets and evaluate library. The second set of experiments involved fine-tuning tasks from SuperGLUE, using Jiant which is a convenient tool that allows us to easily fine-tune NLP models with just a few lines of code, by handling tasks such as tokenizing, dataset splitting, and evaluation. Finally, we used a file provided by the author to perform a set of probing experiments, in which we probed the models on the SentEval dataset. Overall, our experimental setup involved fine-tuning and evaluating the performance of the models, followed by probing the models to investigate their capabilities.

We evaluated the performance of our natural language processing (NLP) models using several measures. For the CoLA task, we used the Matthews correlation coefficient (MCC) to evaluate the model's performance. MCC is a measure of the quality of binary and multiclass classifications, and is commonly used in NLP tasks. For the MultiRC task, we used the F1 measure, which is a metric that balances precision and recall, to evaluate the model's performance. Finally, for the remaining tasks, we used accuracy as the evaluation measure. Accuracy is a simple and intuitive measure that calculates the proportion of correct predictions made by the model, and is often used as a primary evaluation metric in NLP tasks. Overall, our evaluation measures were chosen to appropriately reflect the performance of the models on each specific task.

	Full	7k	2.5k	1k	Baseline
CoLA	58.25	58.18	45.63	42.00	11.86
SST-2	92.78	91.06	89.79	86.58	53.89
MNLI	82.41	73.45	67.63	58.42	35.96
QQP	90.63	81.36	79.52	76.87	39.65
MRPC	86.03	-	82.60	77.45	68.38
BoolQ	71.28	64.68	65.41	62.17	62.07
Wic	65.56	-	64.60	62.43	45.29
MultiRC	69.28	-	67.08	66.46	61.42

Table 2. The performance of fine-tuned BERT on five tasks from GLUE and three tasks from SuperGLUE after fine-tuning on training data of varying size. The numbers are reported based on accuracy for SST, MNLI, QQP, MRPC, BoolQ, WiC; Matthew's correlation for CoLA, and F1 for MultiRC.

In Table 2, we present the performance of our fine-tuned models on various tasks from GLUE and SuperGLUE. We show how reducing the amount of training data affects a model's performance on each task. It's worth noting that even though the performance of these tasks decreases with less training data, it's still significantly better than the performance of the pre-trained model. This indicates that the fine-tuned models have successfully learned the target tasks to some extent. Also note that the performance models fine-tuned with tasks from GLUE is very similar to the performance reported in the original paper. Here we were able to successfully reproduce the experimental setup to build on our further experiments.

3.5 Computational requirements

To fine-tune our models, we primarily used Colab, which provides access to either a K80 or T4 GPU, depending on availability, as well as 2 CPU cores and 12 GB of RAM. For some of the Linguistic Knowledge Recoverability fine-tuning experiments (QQP → MRPC → QQP and MRPC → QQP → MRPC), we used a Kaggle P100 GPU, 2 CPU cores, and 13 GB of RAM. For the probing tasks, we used a mixture of V100, P100, and T4 GPUs, depending on availability. We estimate that we used approximately 45% V100, 45% P100, and 10% T4 GPUs. Each of these GPUs came with 3 CPU cores and 16 GB of RAM. This allowed us to efficiently fine-tune and evaluate our models.

Moreover, for the V100 GPU, the average runtime for probing tasks was 35-40 minutes. The P100 GPU had a slightly longer average runtime of 1 hour and 10 minutes for probing tasks. The T4 model had an average runtime of 1 hour and 35 minutes for probing tasks. For fine-tuning on small datasets, both the P100 and T4 models had an average runtime of 5-10 minutes per run. On the other hand, when fine-tuning on a large dataset like QQP, the P100 model had an average runtime of 4 hours. We did not fine-tune the V100 model on any datasets.

To reproduce this paper, assuming the user is using a P100 GPU and the 19 fine-tuned models provided by the authors, it would take an average of 1.17 hours (1 hour and 10 minutes) to perform probing on each of the models. Fine-tuning the entire dataset would take an average of 0.17 hours for small datasets like CoLA, or 4 hours for large datasets like QQP and MNLI. Therefore, it would take a total of 143 hours (122 models x 1.17 hours/model) to perform probing on all of the models. To reproduce the results of the original paper, the user would also need to fine-tune large datasets twice and small datasets twice, for a total of 9 hours (2 x 4 + 2 x 0.17). This means that reproducing the author's work would take a total of 154 GPU hours (143 + 9).

Our expanded version of the paper performed 174 probing tasks and 22 fine-tuning tasks. This would take a total of 204 hours (174 models x 1.17 hours/model) to perform the probing experiments, and 12 hours to perform the fine-tuning (2 x 4 + 20 x 0.17). However, we made a mistake when converting the models fine-tuned with Jiant, which resulted in incorrect experiment results. We had to re-run some of the experiments, which added an additional 45 GPU hours. This means that we probably used a total of 261 GPU hours (216 + 45) to complete our experiments (assuming a P100 GPU).

Moreover, we mostly used the Hugging Face libraries to facilitate our experiments. These libraries, such as the tokenizer, datasets, evaluate, and trainer, are easy to use and provide a convenient interface for working with natural language processing tasks. We found these libraries to be particularly useful and recommend them for any future work for reproduction. Additionally, we advise avoiding the use of TensorFlow (if not proficient), as it can be more complicated and time-consuming to work with. Overall, the Hugging Face libraries offer a user-friendly and efficient way to perform natural language processing experiments.

4 Results

We reproduced the results of the original study on the impact of data size on the performance of fine-tuned language models. The results showed that the size of the training dataset has a significant impact on the performance of fine-tuned models on downstream tasks, supporting the hypothesis that data size plays a significant role in probe accuracy. We also found that the recoverability of linguistic knowledge through re-fine-tuning is dependent on the size of the dataset for the target task. Overall, these results highlight the importance of considering both the type of downstream task and the size of the training dataset when fine-tuning large language models.

4.1 Probing Linguistic Knowledge and Impact of Data Size

The authors in the original paper hypothesised that the type of downstream task and the size of its corresponding dataset can have a significant impact on the linguistic knowledge encoded in a model. To reproduce the results for this, we fine-tuned pre-trained BERT on a selection of downstream tasks with varying amounts of training data. We used pre-trained BERT as our baseline and limited the number of samples to 7k, 2.5k, and 1k to analyze the effect of training set size on encoded linguistic knowledge. Figure 1 shows the results of this experiment done with models fine-tuned on tasks from both GLUE and SuperGLUE benchmark. The results for the probing performance on models fine-tuned on GLUE tasks seem to be reproduced well, and are similar to the results in the original paper, further validating the original hypothesis that data size plays a significant role in probe accuracy.

Our experiment not only supports the original hypothesis that the size of the training dataset can have a significant impact on the performance of fine-tuned models on downstream tasks, but also suggests that this impact can be seen in models fine-tuned on different types of tasks, as shown with the results for models fine-tuned on SuperGLUE tasks. In other words, the results of our experiment indicate that data size should be considered not only when selecting a downstream task for fine-tuning, but also when deciding on the size of the training dataset. This is important because using a large dataset can potentially improve the performance of a fine-tuned model, but can also increase the time and resources required for training. Therefore, it is important to strike a balance between the size of the dataset and the expected performance of the model on the downstream task. Overall, our experiment provides evidence for the importance of

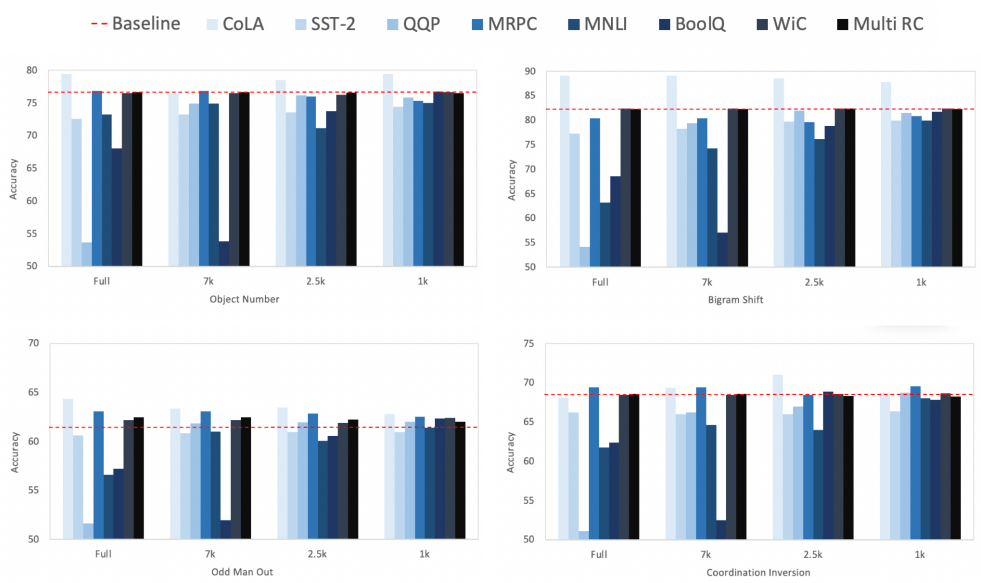

Figure 1. An illustration of the probing performance of models fine-tuned on fixed-size training sets of eight different tasks. The pre-trained BERT's performance on each of the four probing tasks has been shown by the dashed red line. The figures suggest that different fine-tuned models, irrespective of the fine-tuning task, almost encode similar linguistic knowledge when trained on equal-sized data.

considering both the type of downstream task and the size of the training dataset when fine-tuning large language models.

4.2 Linguistic Knowledge Recoverability

For this section, we wanted to replicate the results for assessing the recoverability of modifications made during the fine-tuning process. As in the original paper, we used a fine-tuned model on a specific task as our baseline, and further fine-tuned it on another task. We then compared the probing performance of the resulting models with their corresponding baselines. We wanted to confirm the hypothesis that when the performance is similar, it suggests that the modifications made during fine-tuning can be recovered. Moreover, we wanted to extend the results to more tasks and see if it can be generalized.

We reproduced the experiments for CoLA and SST-2 and MRPC and QQP as two pairs of tasks where we reported the probing results for the Bigram Shift and Semantic Odd Man Out tasks. We also conducted a similar set of experiments for WiC and BoolQ models, fine-tuned on SuperGLUE task to extend the paper. The results are presented in Figure 2.

Our reproduced results suggest that the recoverability of linguistic knowledge through re-fine-tuning is dependent on the size of the dataset for the target task. Earlier we observed that CoLA and SST-2 have significantly different performances on the Bigram Shift and Semantic Odd Man Out tasks. While in the original paper, after re-fine-tuning, both tasks were able to recover the knowledge that was modified by the previous fine-tuning step, we were unable to reproduce it for SST-2 → CoLA → SST-2 and CoLA → SST-2 for any of the probing tasks. However, we did see the same result for all the four probing task when the base model was CoLA.

On the other hand, for the QQP and MRPC pair, we observed the same result as the original paper where the small size of the QQP dataset greatly limited the recoverability of

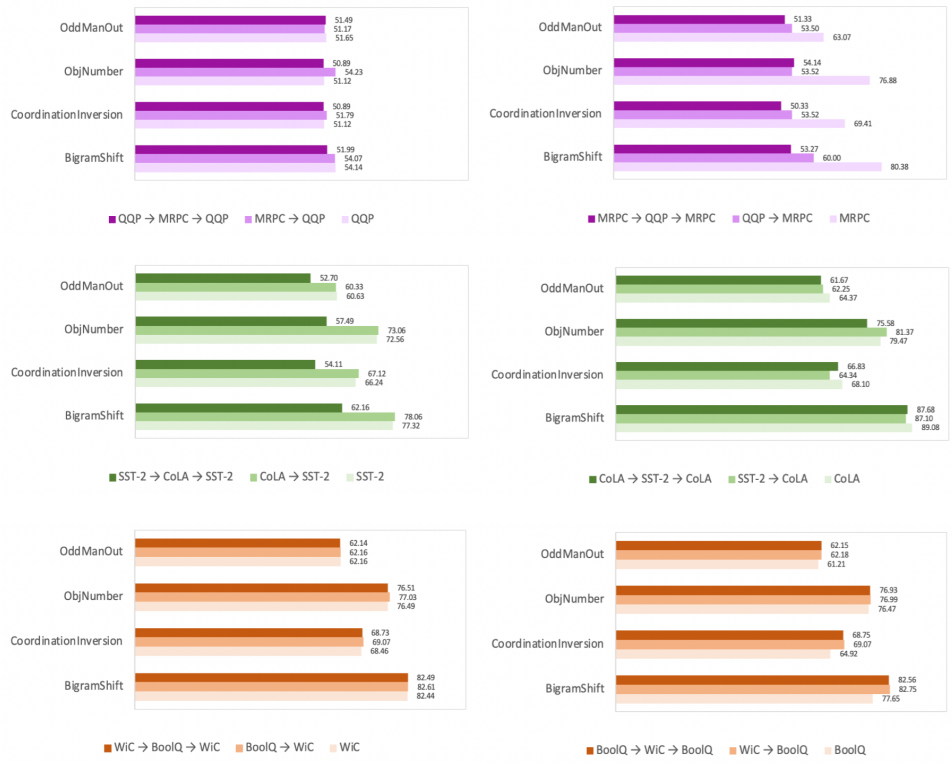

Figure 2. The performance of the models after being sequentially fine-tuned on different tasks. The figures demonstrate that the modified knowledge recoverability depends on the fine-tuning data size.

linguistic knowledge. The final MRPC fine-tuning in the QQP → MRPC and MRPC → QQP → MRPC settings could not recover the modification introduced by QQP (the probing results remained similar to QQP's) for all the probing tasks. In the reverse setting (MRPC → QQP and QQP → MRPC → QQP), the probing performance was only slightly affected by the MRPC data size, resulting in performance that was similar to QQP's.

Moreover, we were also able to extend these results for WiC and BoolQ from SuperGLUE as seen in the figure. Here the modifications introduced by WiC and BoolQ were recovered well after re-fine-tuning the model on the other task for both the settings (BoolQ → WiC → BoolQ and WiC → BoolQ → WiC and BoolQ → WiC). This suggests that the recoverability of linguistic knowledge through re-fine-tuning is also possible for these tasks, since these tasks have relatively small datasets. This supports the idea that the recoverability of linguistic knowledge is dependent on the size of the dataset for the target task, and that this applies to a variety of different tasks

5 Discussion

Based on the reproduced results, it appears that the results support the claims made in the original paper. The significant gaps between the results of models fine-tuned on larger data sizes and the pre-trained BERT baseline support the idea that data size plays a significant role in the performance of fine-tuned models on downstream tasks. The results also support the claim that as the number of samples increases, the gap between fine-tuned models and the pre-trained BERT becomes more apparent. This in-

dicates that fine-tuning data size does indeed affect the linguistic knowledge encoded by the model. In other words, our results support the conclusion that data size should be considered when analyzing fine-tuned models and their ability to encode linguistic knowledge.

Our experimental results not only support the original claims made in the paper, but they also show similar trends when applied to SuperGLUE tasks. This suggests that the conclusions drawn from the original experiment are generalizable to a broader range of downstream tasks. Specifically, our results indicate that data size plays a significant role in the performance of fine-tuned models on downstream tasks, and that this should be considered when analyzing the ability of fine-tuned models to encode linguistic knowledge. These conclusions are supported by both the results for GLUE tasks and those for SuperGLUE tasks, indicating that they are likely to hold true for a wide range of downstream tasks.

In regards to linguistic knowledge recoverability, our reproduced and extended results indicate that the recoverability of knowledge during fine-tuning is tied to the size of the fine-tuning data. In particular, further fine-tuning a fine-tuned model with a comparable data size (e.g., SST-2 → CoLA and CoLA → SST-2 → CoLA; BoolQ → WiC and WiC → BoolQ → WiC) has the same effect as fine-tuning a pre-trained model (e.g., CoLA, WiC). However, increasing the data size in one of these tasks reduces the recoverability of the other task. Therefore, we can say that the extent to which linguistic knowledge can be recovered after it has been modified by a different task depends on the size of the dataset for the target task.

One strength of our approach is the extensive experimentation we performed. We conducted a large number of experiments in order to thoroughly evaluate the performance of our models and methodologies. Additionally, we performed a number of additional experiments in order to extend our results and investigate specific questions and areas of interest.

However, one weakness of our approach is that we only ran each experiment once, which can make the results uncertain. Ideally, we would have run each experiment multiple times and taken the average in order to better understand the variance and standard deviation. This would have given us more confidence in our results.

Another weakness is that we were unable to tokenize and evaluate some tasks from SuperGLUE, such as MultiRC. This meant that we had to use a separate library called Jiant to do so. The results from Jiant could very well vary from the results we would have obtained using our own tokenization and evaluation methods, which can affect the accuracy of our results. Additionally, we encountered difficulties when converting models from Jiant format (model.p) to pytorch format (pytorch_model.bin), which resulted in invalid results for some experiments. Although we eventually found a solution, we are not fully confident in the correctness of our conversion method.

In our research, we were unable to run all of the experiments we had planned due to time constraints. Specifically, we were unable to run the ReCORD task from the SuperGLUE dataset as it would have taken too long to fine-tune the models. We would have benefited greatly by experimenting with a large dataset in the SuperGLUE benchmark. Additionally, we were unable to explore other experiments mentioned in the original paper, such as layer-wise analysis. We prioritized the experiments that we deemed the most important and were unable to fully explore the full scope of the original paper.

5.1 What was easy

The author's code for performing the probing tasks was well-written and easy to use, which made it easy to reproduce the experiments described in the paper. The code had clear documentation and examples, which made it straightforward to understand and run. As a result, we were able to easily reproduce the experiments and obtain similar results to those reported in the paper. This demonstrates the quality of the author's code and their efforts to make it accessible to others.

5.2 What was difficult

One aspect of the study that took more time than anticipated was the large number of tasks that required the use of a GPU. The original paper required us to perform 19 x 6 = 114 probing tasks, fine-tune 4 additional models, and perform 2 probing tasks on each of those models, for a total of 126 tasks that required a GPU. In addition, we performed additional experiments on tasks from SuperGLUE, which increased the number of models we had to fine-tune to 22 (plus the 19 provided by the author, for a total of 41) and the number of probing tasks to 174. This required a significant amount of storage and GPU compute hours, and managing the 196 total tasks was time-consuming. Additionally, we made some mistakes and had to rerun some experiments, which added to the time and effort required for the study. Overall, the large scale of the study and the need for GPU resources made it more difficult and time-consuming than expected.

For this project we had to fine-tune all of our models on Colab and Kaggle. Colab provided a useful platform for creating multiple accounts and running multiple experiments simultaneously. However, it had slow upload and download speeds and storage limitations, which posed challenges for our experiments. In order to overcome these limitations, we had to connect our Google Drive to Colab in order to upload and download models. This resulted in additional storage issues, as Google Drive only provides 15 GB of storage per account. Additionally, Colab would sometimes randomly disconnect us in the middle of experiments if we ran out of GPU hours, with no warning beforehand. It also required us to constantly monitor the experiments by answering captcha pop-ups, asking if we were still using Colab interactively.

Kaggle provided a better platform for our experiments, with faster upload and download speeds and 20 GB of storage per account. It also gave us a clear indication of how many GPU hours were available in our quota. However, it had its own limitations, such as a weekly limit of 30 GPU hours per account and the requirement that each person could only create one account. We used the P100 GPU on Kaggle for our experiments. Overall, both platforms presented challenges and required significant work and resources to verify the results of our fine-tuned models.

Another difficulty that we faced when trying to re-use the original paper was performing probing experiments on Compute Canada servers. These servers had faster GPUs than Colab, and we were able to queue multiple tasks. However, the server had problems installing certain libraries from Hugging Face, such as 'datasets' and 'evaluate'. It was also unable to download models and tokenizers from the Hugging Face hub. This required us to fine-tune our models on Colab and Kaggle, and then transfer them to the servers using scp. This process was slow and required significant work and resources to verify the results of our experiments.

5.3 Communication with original authors

In our attempt to communicate with the original authors, we reached out to them in order to obtain the experiment checkpoints for their fine tuned models as their github link

for the checkpoints was broken. It took some time for the authors to respond, but when they did, they informed us that the original checkpoints file had been corrupted. However, they were able to provide us with some other checkpoints that were not originally included in the paper, but may still be useful for our reproducibility project.

6 Contributions

In this project, KN contributed to the experimental design in conceiving and planning the experiments, and carried out a majority of the experiments. KN also played a key role in revising the article, providing valuable feedback and insights throughout the process.

SA was responsible for the final data analysis and visualization, and played a crucial role in presenting the results in a clear and concise manner. SA also carried out a few key experiments, and wrote the paper.

Overall, KN played a leading role in the experimental design and execution, while SA played a key role in the writing and data analysis. Both authors were instrumental in discussing the results, making the final revisions and ensuring the accuracy and clarity of the paper. Their combined efforts and collaboration were essential in bringing this project to fruition.

References

1. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: Proceedings of the 2019 Conference of the North (2019).
2. N. Durrani, H. Sajjad, and F. Dalvi. How transfer learning impacts linguistic knowledge in deep NLP models? ACLWeb, Aug. 2021.
3. A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. 2019.
4. A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems.
5. H. Mehrafarin, S. Rajaee, and M. T. Pilehvar. "On the Importance of Data Size in Probing Fine-tuned Models." In: Findings of the Association for Computational Linguistics: ACL 2022. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 228–238.
6. T. Singh and D. Giovanardi. How much does pre-trained information help? Partially re-initializing BERT during fine-tuning to analyze the contribution of layers Stanford CS224N Custom Project. 2020.