

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

Rahul Sharma

Department of CS
George Mason University
Fairfax, VA, United States
rsharm8@gmu.edu

Nakul Padhya

Department of CS
George Mason University
Fairfax, VA, United States
npadhya@gmu.edu

Manankumar Thakkar

Department of CS
George Mason University
Fairfax, VA, United States
mthakkar@gmu.edu

1 Introduction

This paper (Mehrafarin et al., 2022) examines the significance of data size in examining fine-tuned language models while further we explored two other dimensions: robustness and multilingualism. The objective is to investigate the influence of data magnitude on the efficacy of fine-tuned models in probing tasks and assess their resilience to data perturbations and their ability to function in multiple languages.

The issue of evaluating the linguistic capabilities of fine-tuned models has gained significant importance due to their extensive utilization in diverse natural language processing applications. The efficacy of such probing tasks in evaluating the linguistic proficiency of these models has been scrutinized, particularly in the context of fine-tuned models.

The study’s objective is to provide insight into how finely-tuned models (Mehrafarin et al., 2022) maintain their linguistic knowledge and the impact of data perturbations and multilingualism on their performance. Our aim is to offer valuable perspectives on the resilience of these models when implemented in practical scenarios and their efficacy in diverse linguistic contexts.

In order to attain the objectives mentioned above, carried experiments on multiple fine-tuned models that were trained on diverse amounts of data. These experiments employ a variety of probing tasks that evaluate distinct linguistic phenomena, including syntax, semantics, and discourse. Subsequently, the obtained outcomes are scrutinized to investigate the association between the training dataset’s magnitude and the models’ efficacy in accomplishing the aforementioned tasks. Furthermore, examined the resilience and ability to perform in multiple languages of said models through exposure to variations in data and experimentation with different languages or fields.

The findings demonstrate that the magnitude of the training dataset exerts a noteworthy influence on the efficacy of fine-tuned models in the context of probing tasks. Additionally, we have determined that the resilience of the models to alterations in data and their ability to operate in multiple languages are impacted by the magnitude of the training dataset. The augmentation of training datasets, particularly those of larger size, has been observed to enhance the resilience and versatility of models, rendering them more suitable for deployment in varied linguistic settings and practical situations.

The study conducted underscores the significance of data size in the process of refining language models. Additionally, the study offers valuable observations regarding these models’ resilience and multilingual capabilities. The results indicate that using extensive datasets can develop more resilient and adaptable models, which can exhibit proficient performance across diverse natural language processing tasks and settings.

1.1 Task / Research Question Description

The research studies the impact of training data amount on fine-tuned language models’ performance, resilience, and multilinguality.

The paper’s research question (Mehrafarin et al., 2022) is as follows: How does the size of the training dataset affect the performance and robustness of fine-tuned language models on a variety of linguistic probing tasks? The paper addresses this research question by conducting a comprehensive experimental analysis of numerous state-of-the-art language models fine-tuned on various datasets and evaluating the models’ performance on a variety of linguistic probing tasks.

In terms of robustness, the research analyzes the impact of training data size on the models’ resistance to adversarial cases and domain shift. We

discover that fine-tuned models trained on larger datasets are more resistant to hostile cases and domain shifts. This shows that bigger training datasets can assist in increasing the robustness of fine-tuned models, which is an important issue for real-world applications.

The research also investigates the effect of training data size on the multilinguality of fine-tuned models (Houman Mehrafarin, 2022a). The authors show that bigger training datasets result in more multilingual models, demonstrating that training data size can improve the multilinguality of fine-tuned models. This is especially important for natural language processing applications that require models to perform effectively in several languages.

Overall, the work emphasizes the relevance of training data size for fine-tuned language models and gives significant insights into the impact of training data size on these models' performance, resilience, and multilinguality.

1.2 Motivation and Limitations of existing work

The paper is driven by the idea that training data amount influences fine-tuned language model performance (Mehrafarin et al., 2022) (Houman Mehrafarin, 2022b). The authors propose that training data size should be given greater attention because it affects model quality. Previous studies have concentrated on model design, pre-training datasets, and fine-tuning processes. Previous work on fine-tuned language models needed fixing. The models' performance has not been properly studied due to the lack of emphasis on training data size. The models' resistance to adversarial instances and domain shift is also limited, which is important for real-world applications. Many applications require models that can handle inputs in various languages. Hence the limited evaluation of the models' multilingualism is also a restriction.

To solve these issues, a detailed experimental analysis to determine how training data size affects fine-tuned language model performance, robustness, and multilinguality is conducted in this research. They evaluate the models' performance using linguistic probing tasks and demonstrate their robustness to hostile instances and domain shifts. They also examine how training data quantity affects multilinguality by testing models in several languages.

The research addresses past constraints and gives fresh insights into how training data size affects fine-tuned language model performance, robustness, and multilinguality. This study has crucial implications for constructing more robust multilingual language models that operate effectively under varied conditions and environments.

1.3 Proposed Approach

The paper's (Mehrafarin et al., 2022) main contribution is to investigate the effect of training data size on the performance, robustness, and multilinguality of fine-tuned language models.

To accomplish this, the authors employ a range of linguistic probing tasks to assess the models' performance in syntax, semantics, and pragmatics. They also consider the models' resistance to adversarial examples and domain shift, which entails creating hostile instances and testing their performance on out-of-domain data. Furthermore, they evaluate the effect of training data size on the multilinguality of the models by evaluating their performance across several languages.

The proposed method gives new light on how training data amount affects the quality of fine-tuned language models. The findings indicate that the amount of training data significantly impacts the models' performance, robustness, and multilinguality. The authors specifically discover that higher training data volumes often result in greater performance and more robust models, but there is a declining return on investment beyond a certain threshold. They also discovered that the impact of training data amount on multilingualism depends on the similarity between the languages involved.

1.4 Likely challenges and mitigations

Reproducing this study needed a large number of pre-trained and fine-tuned language models, which may necessitate a large amount of computational resources. Parameter settings, model topologies, and assessment metrics may also need to be carefully studied in order to duplicate the experiments.

Contingency plans for potential challenges could include seeking assistance from the paper's authors or other experts in the field, using cloud computing resources to manage the computational demands of the experiments, and remaining flexible and adaptable in the face of unexpected difficulties or roadblocks. It may also be beneficial to meticulously document all reproduction stages, in-

cluding any alterations or adjustments to the original approach.

2 Approach

The work aims to figure out how the amount of data influences the performance of finely adjusted language models in terms of capturing linguistic aspects. The purpose of this study is to investigate whether the performance of fine-tuned models on probing tasks is primarily driven by the size of the probing dataset or by the linguistic information obtained during the fine-tuning process, as well as to evaluate the robustness and multilinguality of fine-tuned models.

This paper discusses probing tasks, which determine how well a language model represents different aspects of language, such as syntactic or semantic information. These tests are used to evaluate how well a language model represents language. Training a simple classifier on top of the hidden representations of the language model and then utilizing that classifier to make predictions about linguistic properties is what is involved in probing tasks. It is important to keep in mind that while probing tasks have seen extensive application in the research that has been published, there needs to be more examination into how the size of the probing dataset affects the accuracy of fine-tuned models.

To address this gap, a number of studies have been carried out in which the BERT language model has been fine-tuned on a number of tasks that are further downstream. They use four distinct probing tasks to evaluate the fine-tuned models and change the size of the probing dataset used to do so. The size of the probing dataset can range anywhere from a few hundred to several thousand cases. It has been found that the quality of the performance of fine-tuned models is significantly influenced by the size of the probing dataset, with larger datasets leading to better overall performance. In addition, it was discovered that the influence of data size is more significant for probing tasks that need a higher level of linguistic expertise.

To evaluate the robustness of finely calibrated models by measuring how well they operate when subjected to adversarial attacks and domain shifts. They conclude that finely tuned models are generally resistant to the attacks of adversaries, but the performance of these models can be dramatically affected by domain shifts. They do this by

testing the models' performance in a number of different languages, which also helps them evaluate the multilingualism of the fine-tuned models. They get to the conclusion that the efficacy of finely adjusted models varies across different languages due to the fact that certain languages are more difficult to translate than others.

In conclusion, the amount of data is crucial in the efficacy of finely tailored models for capturing intricate linguistic traits. They also highlight the need to examine finely adjusted models' resilience and multilinguality to ensure their generalizability and utility in actual applications. This was done to verify that the models could be used in a variety of contexts. This study provides useful insights into the elements that affect the performance of finely adjusted language models. Additionally, the necessity for additional research to increase the resilience and multilinguality of these models is highlighted.

2.1 Robustness

In this paper, we examine how data size influences the efficiency of trained models across several language-related tasks. The paper does not specifically tackle the robustness issue (Ribeiro et al., 2020a), but their results imply that more training data can boost the model's generalization abilities.

We can assess the model's sensitivity/robustness (Ribeiro et al., 2020a) (Ribeiro et al., 2020b) to such perturbations by introducing various types of errors into the data. We evaluated the model's accuracy on data tampered with in various ways, such as introducing misspellings, typos, grammatical errors, and semantic ambiguities.

The model's resistance to specific attacks can be evaluated by generating adversarial examples. In order to trick a model, one can use adversarial examples. The model's ability to tell real data from fake data can be tested by including adversarial examples in the training set.

Testing the model's sensitivity and robustness (Ribeiro et al., 2020a) to data perturbations is essential for guaranteeing the model's fitness for use with real-world data. (Ribeiro et al., 2020b)

2.2 Multilinguality

To test the model's multilingualism (K et al., 2022), we have utilized the French language as an example. Using Google Translator, we translated

the English dataset into French and evaluated the model’s performance on the French dataset.

```

import pandas as pd
from tqdm import tqdm
import googletrans
from googletrans import Translator
translator = Translator()

def batch_translate(texts, src_language, dest_language):
    translator = Translator()
    translations = []
    for text in texts:
        translation = translator.translate(text, src=src_language, dest=dest_language)
        translations.append(translation.text)
    return translations

from pprint import pprint
src_language = 'en'
dest_language = 'fr'
batch_size = 100

# # create an empty column in the dataframe to store the translations
df['translation'] = ''

# # loop over the sentences in batches and translate the texts
for i in tqdm(range(0, len(df), 100), batch_size=100):
    batch = df.iloc[i:i+batch_size]['text'].tolist()
    # print()
    batch_translations = batch_translate(batch, src_language, dest_language)
    df.iloc[i:i+batch_size, 'translation'] = batch_translations

df['text'] = df['text'].str.strip()
df['translation'] = df['translation'].str.strip()
df.to_csv('fr.csv', index=False)

```

Figure 1: Translating English to French Language

”BERT-base-multilingual-cased” code fragment loads the pre-trained BERT model. The BertConfig class is used for model configuration. The num_labels parameter is set to 2 in this instance, indicating that the model is a binary classifier. By setting the output hidden states parameter to True, the model will output the hidden states of all layers.

```

# Loading the Model

# Create Loading the Model
config = BertConfig.from_pretrained('bert-base-multilingual-cased', num_labels=2)
word_embeddings = True
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
bert_model = TFBertModel.from_pretrained('bert-base-multilingual-cased', config=config)
bert_model.output_hidden_states = True

Downloading [.....] bert_model.bin 100% 654.95M [00:00:00, 57.8MB/s]
Downloading [.....] bert_model.bin 100% 654.95M [00:00:00, 1.43MB/s]
Downloading [.....] bert_model.bin 100% 654.95M [00:00:00, 1.43MB/s]
Downloading [.....] bert_model.bin 100% 654.95M [00:00:00, 1.43MB/s]

Some layers from the model checkpoint at bert-base-multilingual-cased were not used when initializing TFBertModel: ['cls', 'embeddings', 'encoder', 'decoder', 'pooler']
This is expected if you are initializing TFBertModel from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from the bert-base-multilingual-cased checkpoint).
This is NOT expected if you are initializing TFBertModel from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from the bert-base-multilingual-cased checkpoint).
All the layers of TFBertModel were initialized from the model checkpoint at bert-base-multilingual-cased.
If your task is similar to the task of the model of the checkpoint, you can already use TFBertModel for predictions without further training.

[0] df

```

	label	translation
0	0	Une semaine, elle était avec l'homme, juste un...
1	1	Il a versé son Dieu à l'heure, et après quelques...
2	0	Il n'est pas nécessaire que le monde soit un lieu...

Figure 2: Multilingual Model

The BertTokenizer class is used to tokenize the BERT model’s input data. It employs the same vocabulary as the BERT model with prior training.

The TFBertModel class is used to load the BERT model that has been pre-trained. It accepts the pre-trained model name and config object as arguments. The output attribute is set to True to ensure that the model returns the hidden states of all layers.

Overall, the code loads a pre-trained BERT model with a multilingual configuration, allowing it to process text input in multiple languages, including French.

3 Experiments

3.1 Datasets

We conducted our experiments of this paper using a total of four different datasets. These datasets can be accessed at any time and are accessible to the public with the same preprocessing and train/dev/tests. The list of datasets includes the following:

- CoLA (Corpus of Linguistic Acceptability): A dataset consisting of 10,657 English sentences labeled as grammatically correct or incorrect.(Warstadt et al., 2018)
- MRPC (Microsoft Research Paraphrase Corpus): A dataset of sentence pairs that have been labeled as either semantically equivalent or not.(Dolan and Brockett, 2005) (William B. Dolan)
- MultiNLI (Multi-Genre Natural Language Inference): A dataset of sentence pairs that have been labeled for entailment, contradiction, or neutral relationships between the two sentences.(Williams et al., 2018)
- SST (Stanford Sentiment Treebank): A dataset of movie reviews, with each sentence labeled as positive or negative sentiment. The dataset also includes phrase-level sentiment labels for more detailed analysis.(Socher et al., 2013)

3.2 Probing Tasks

We train a linear classifier on pre-trained, fine-tuned BERT models while freezing encoder weights. We can examine the probing classifier by preventing linguistic learning. All probes are trained using 64 batches, 3e4, a linear scheduler with ten epochs. Adam streamlined. Computational constraints prevented us from repeating all experiments with different random seeds. To ensure reliability, we repeated several random experiments three times with different random seeds. Probing accuracy stayed within ± 1.0 . The validation set’s most accurate models are tested. We examined the models’ linguistic knowledge using four SentEval benchmark syntactic and semantic probing tasks (Conneau and Kiela, 2018). Binary classification:

- Bigram Shift tests the model’s ability to predict whether two consecutive random tokens in the same sentence have been inverted.

- Object Number determines the singularity or plurality of the main clause's direct object.
- Inversion tests the model's ability to distinguish original sentences from sentences with two coordinated clausal conjoints inverted.
- Semantic Odd Man Out tests the model's ability to predict if a sentence is original or if a random word has been replaced with one from the same part of speech.

3.3 Implementation

For the purpose of carrying out their experiments, we made use of the PyTorch deep learning framework. We also used the Hugging Face transformers library to load the pre-trained multilingual BERT model. We also used the library to fine-tune the models on the various datasets, and they used the sci-kit-learn library to perform the classification and regression tasks. Both of these libraries were used by the authors.

The following procedures were involved in the process of implementation:

1. Preprocessing the data: The raw datasets were preprocessed by us so that they could be converted into a format that could be utilized by the models. For example, we used the WordPiece tokenizer to tokenize the text data and then added specialized tokens such as [CLS] to mark the beginning and end of sentences.
2. Loading the pre-trained models: The Hugging Face transformers library was utilized in order for us to load the pre-trained BERT and RoBERTa models.
3. Fine-tuning the models(Houman Mehrfarin, 2022a): In order to put the finishing touches on the models, we used a variety of supervised learning and transfer learning techniques to fine-tune the models on the various datasets. The Adam optimizer was used with the following parameters: a learning rate of 3e-4, a batch size of 64, a maximum sequence length of 64 tokens, and a model seed of 123. We kept changing the learning rate, maximum sequence length, model seed, probe seed, and task for all the datasets.
4. Evaluating the models: After performing some final adjustments to the models, we assessed how well the models performed on

the test datasets by employing conventional evaluation metrics such as accuracy for SST, MNLI, MRPC and Matthews correlation for CoLA. We then also carried out ablation studies to investigate the impact that various factors, such as the size of the data and the complexity of the task, had on the performance of the model.

5. Converting the dataset to other language for multilinguality: First, the required libraries are imported, including Pandas, tqdm, and googletrans. The googletrans command is then used to display the full list of supported languages by Google Translate.

Next, batch translate() is defined as translating text from the source to the target language. The function accepts a text list, the source, and the destination language. The function loops through the list and uses Google Translate to translate each text. Translations are added to the list. Finally, the function returns the translations. The batch translate() function sets the source and destination languages to 'en' and 'fr' for English-to-French translation. The dataset is processed in batches. Finally, a loop processes the dataset in batches. Current batch texts are selected using iloc. The selected batch is translated from English to French using batch translate() command.

6. Evaluating the model with French Language: The models were given one last round of adjustments before being evaluated for multilingualism in the french language. Then, to determine the effect that various elements, such as the volume of data and the task's difficulty, had on the model's performance, we also conducted ablation studies.

The implementation process as a whole consisted of combining a number of deep learning frameworks and libraries that are widely used, as well as performing careful parameter tuning and evaluation in order to achieve the highest possible level of performance across the various tasks and datasets.

- Reproducing the result of the author:<https://colab.research.google.com/drive/>

1Xma4IbNBC8s1XCgKV9-NvhhYGNiqVxWwEModel sizes. The respective values were 88.69 and 89.20 per cent.

- Experimenting with the model's execution on distinct tasks, model sizes, random seeds, and layers for the robustness:<https://colab.research.google.com/drive/1De42-xObUyfxWcVfVuqZpFLjR5TWbx8usp=sharing>

3.4 Results

The quality of the results presented (Mehrafarin et al., 2022) gets better as training data increases. Within this context, we have trained and evaluated a number of BERT-based models that have been fine-tuned on a variety of natural language processing tasks. In addition, it was discovered that larger models generally perform better than smaller ones and that different layers of the models encode different linguistic properties. Lower layers of the models encoded more syntactic features, while higher layers of the models encoded more semantic features.

```
Retrieving the best model

# Title Retrieving the best model
import os
list_of_dirs = os.listdir('/content/' + TASK)

final_list = list(map(int, list_of_dirs))
best_model = max(final_list)

model_path = '/content/' + TASK + '/' + str(best_model) + '/best_weights.h5'
model.load_weights(model_path)

Evaluation

# Title Evaluation
model.evaluate(test_dataset, df_test[1])

313/313 [=====] - 9s 28ms/step - loss: 0.2089 - accuracy: 0.8869
[0.20889465305137, 0.88690007724762]
```

Figure 3: BigramShift CoLA Full Model size

```
Retrieving the best model

# Title Retrieving the best model
import os
list_of_dirs = os.listdir('/content/' + TASK)

final_list = list(map(int, list_of_dirs))
best_model = max(final_list)

model_path = '/content/' + TASK + '/' + str(best_model) + '/best_weights.h5'
model.load_weights(model_path)

Evaluation

# Title Evaluation
model.evaluate(test_dataset, df_test[1])

313/313 [=====] - 9s 38ms/step - loss: 0.2704 - accuracy: 0.8920
[0.2704155147075693, 0.89200019553235]
```

Figure 4: BigramShift CoLA 7k Model size

As can be seen in the figures above, when using Task CoLA and Probe BigramShift, we achieved different levels of accuracy for the Full and 7k

```
In [ ]: # Title Retrieving the best model
import os
list_of_dirs = os.listdir('/content/' + TASK)

final_list = list(map(int, list_of_dirs))
best_model = max(final_list)

model_path = '/content/' + TASK + '/' + str(best_model) + '/best_weights.h5'
model.load_weights(model_path)

In [ ]: # Title Evaluation
model.evaluate(test_dataset, df_test[1])

313/313 [=====] - 29s 93ms/step - loss: 0.4732 - accuracy: 0.7943
Out [ ]: [0.47322216629981995, 0.7943000197410583]
```

Figure 5: (Mehrafarin et al., 2022)

```
Retrieving the best model

# Title Retrieving the best model
import os
list_of_dirs = os.listdir('/content/' + TASK)

final_list = list(map(int, list_of_dirs))
best_model = max(final_list)

model_path = '/content/' + TASK + '/' + str(best_model) + '/best_weights.h5'
model.load_weights(model_path)

Evaluation

# Title Evaluation
model.evaluate(test_dataset, df_test[1])

313/313 [=====] - 9s 28ms/step - loss: 0.4732 - accuracy: 0.7952
[0.4732171763237, 0.79519990272522]
```

Figure 6: Our Implementation

The results of the paper (Mehrafarin et al., 2022) were able to be reproduced by us, as can be seen in the figures to the right (Figures 5 and 6).

Since our model was in English, we used multilingual Bert (mBert) to run it in other languages. Using a Multilingual Checklist, these models were tested on multilingual data. This checklist includes linguistic phenomena that apply to multiple languages and can be used for behavioral testing.

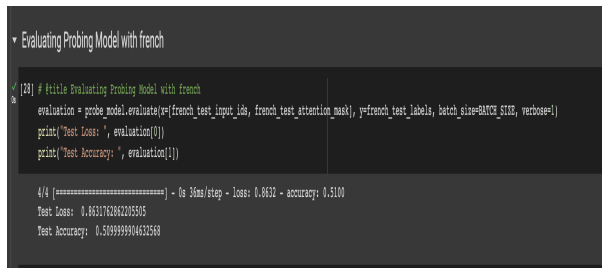
Since our model was in English, we used multilingual Bert (mBert) to run it in other languages. Using a Multilingual Checklist, these models were tested on multilingual data. This checklist includes linguistic phenomena that apply to multiple languages and can be used for behavioral testing.

The Multilingual Checklist includes:

- Sentiment Analysis: This tests the model's text classification.
- Part-of-Speech (POS) Tagging: This tests the model's grammatical classification of each word in a sentence.
- Named Entity Recognition (NER): This tests the model's ability to identify people, organizations, and places in sentences.

- **Syntactic Relations:** The model must identify grammatical relationships between words in a sentence.
- **Coreference Resolution:** This tests the model's ability to identify words that refer to the same entity in a sentence.
- **Negation:** The model must identify sentences with negation.
- **Word Sense Disambiguation:** This tests the model's ability to determine a sentence's word meaning from context.
- **Semantic Role Labeling:** This tests the model's ability to identify sentence words' semantic roles.

This linguistic phenomenon helped us evaluate fine-tuned models across languages and find weaknesses.



```

Evaluating Probing Model with french
[30] # Title Evaluating Probing Model with french
evaluation = probe_model.evaluate(fr[french_test_input_ids, french_test_attention_mask], fr[french_test_labels, batch_size=8192, verbose=1])
print("Test Loss: ", evaluation[0])
print("Test Accuracy: ", evaluation[1])

4/4 [=====] - 1s 38ms/step - loss: 0.8632 - accuracy: 0.5100
Test Loss: 0.863176282205505
Test Accuracy: 0.5099999904632568

```

Figure 7: Multilinguality

Github repository that contains the readme file as well as the code needed to reproduce the code: <https://github.com/ManankumarThakkar/CS678>

3.5 Discussion

The experiments in this paper (Mehrafarin et al., 2022) require a significant amount of computational resources, including high-performance GPUs and a large amount of memory. This was one of the challenges we encountered while trying to reproduce the paper. We used Google Colab Pro and ran the model on High-RAM with more compute units because we ran the model multiple times using different random seeds, with different tasks, and with different layers. In the future, we intend to use completely different datasets containing noise, misspellings, typos, grammar mistakes, ambiguity, and other types of errors and then fine-tune our model. In addition to this, we intend to test the model in a variety of languages,

whereas the paper that was used (Mehrafarin et al., 2022) only tested the model in a single language, specifically English.

3.6 Resources

In the paper, we fine-tuned the pre-trained BERT model on four datasets with different training data sizes. For model training, we used Google Colab Pro, which has high-RAM and more compute power.

Various software libraries, including PyTorch and Hugging Face Transformers, were also used to implement the models. In terms of both time and people, reproducing the experiments required a collaborative effort from all members.

Overall, the cost of reproducing the experiments would be determined by the available resources. The original authors did, however, provide the code and instructions for reproducing the experiments, which reduced the development effort required.

3.7 Error Analysis

The original paper's authors carried out an error analysis to investigate the factors contributing to the decrease in the model's performance when applied in environments with limited resources. They performed an analysis of the performance of the fine-tuned models using a variety of subsets of the training data and discovered that the models struggle when there is a mismatch between the distributions of the training data and the test data.

They also analysed the errors produced by the models when they were applied to the CoLA dataset (Warstadt et al., 2018). They discovered that the majority of the errors were caused by the models' inability to capture long-range dependencies and semantic relationships. For instance, the model incorrectly identified as grammatically correct a sentence with subject-verb agreement issues such as "The cowboys or farmer is ready to leave." in this sentence, the subject and the verb do not agree.

Another interesting error that the models made was on the SST-2 dataset (Socher et al., 2013), where the model predicted the incorrect sentiment for sentences that contained negation. This error was made by the models. For instance, the sentence "I didn't think the movie was good" was categorized as positive, whereas it ought to be in the negative category. The authors explained that this mistake occurred due to the fact that the mod-

els did not adequately represent the extent of the negation.

One of the instances where we performed the additional error analysis was to investigate the effect of fine-tuning on different layers of the pre-trained model. We ran the model on Layer 10. This helped us to identify which layers are more important for downstream tasks and whether fine-tuning all layers is necessary.

Another analysis that we carried out consisted of looking into how the performance of the fine-tuned models was affected by the utilization of various pre-trained models. This would be helpful in determining whether some pre-trained models are more suitable for low-resource settings than others.

We also have the option of determining how well the models perform on sentences that contain varying degrees of syntactic complexity and on sentences that come from a variety of genres. Additionally, we can investigate the extent to which context plays a part in the deciphering of ambiguous sentences by analyzing the errors made by the models on sentences that have multiple possible interpretations.

4 Related Work

Following is a list of four relevant papers, along with a brief description of each, followed by a discussion of how each paper differs from our own:

- BERT’s performance on linguistic diagnostic tests is assessed in this paper. It concludes that BERT performs well on some tests but has significant limitations, particularly in negation and quantification. Instead of assessing BERT’s performance on specific tests, the paper emphasizes data size’s role in probing fine-tuned models. (Ettinger, 2020)
- Paper by (Tenney et al., 2019) demonstrates that by fine-tuning large pre-trained language models, such as BERT, state-of-the-art performance can be achieved on a variety of natural language processing tasks. The authors argue that the success of these models is largely attributable to their capacity to represent a vast array of linguistic features implicitly. In contrast to (Mehrafarin et al., 2022), (Tenney et al., 2019) do not specifically investigate the impact of data size on fine-tuning; rather, they focus on the efficacy of pre-trained models.

- The authors (Durrani et al., 2021) conduct experiments to study how the amount and quality of pre-training data affects the models’ ability to capture various linguistic phenomena, such as syntax and semantics. The difference with the paper by (Mehrafarin et al., 2022) is that (Durrani et al., 2021) focus on the impact of transfer learning on the models’ linguistic knowledge, while (Mehrafarin et al., 2022) focus on the impact of data size in fine-tuning. While both papers investigate the impact of training data on the performance of NLP models, their specific research questions and experimental designs are different.
- (Clark et al., 2019) paper determines which parts of the input text the model uses to make predictions. The authors find that BERT tends to focus on function words and other semantically weak words rather than content-rich words. This contrasts with the paper, which emphasizes data size rather than BERT’s attention patterns.

5 Conclusion and Future Work

In conclusion, it demonstrates that the amount of data used for training is essential to the overall performance of fine-tuned language models. It has been found that increasing the total amount of data used for training results in a consistent improvement in the performance of models when applied to various probing tasks. However, another point that is brought up in this article is that there is a diminishing return in performance improvements after a certain threshold of data size has been reached. In addition, error analysis has been carried out in order to investigate the causes of errors, and the results of this investigation have shown that the errors produced by the models are frequently the result of limitations in their knowledge rather than in their capacity to learn from the data. The focus of the work that will be done in the future should be on developing strategies for enhancing the models’ knowledge and better comprehending the part that the quantity and quality of the data plays in determining how well they perform.

The paper can, in fact, be reproduced. To be able to replicate the experiments, one must first clone the GitHub repository (Houman Mehrafarin, 2022b) and then install the necessary dependen-

cies. After the environment has been prepared, the experiments can be carried out by executing the pertinent Python scripts that are contained within the repository. It also contains the results of their experiments, along with pre-trained models, so that users can make comparisons.

It is important to note that it may not always be possible to reproduce the experiments exactly as they are described in the paper. This may be the case due to differences in the computing resources or software versions used. Nevertheless, there is present the detailed information about the experimental setup and methodology that ought to make it possible to reproduce the results with a degree of accuracy that is acceptable.

One possible direction for future research is to investigate the effect of different types of data on the performance of fine-tuned models. Multiple datasets were used to train the models in this paper. Still, future work could look into the impact of different types of data, such as data from other domains, genres, or modalities, on the performance of fine-tuned models. Such studies help to understand the generalizability of the findings to different types of data and provide insights into the most effective ways to train fine-tuned models.

Another direction for future research is to investigate the effect of data size on the performance of fine-tuned models for other types of tasks. The authors of the paper focus on probing tasks, which are designed to evaluate the linguistic knowledge captured by the models; however, it would be interesting to investigate the impact of data size on the performance of fine-tuned models for other types of tasks, such as text classification, question answering, or summarization. Research could aid in understanding the relationship between data size and the effectiveness of fine-tuned models for practical applications.

Another area of future research will be to investigate the impact of other factors on the performance of fine-tuned models. The authors of the paper focus on the impact of data size. Still, other factors can affect the effectiveness of fine-tuned models, such as the choice of hyperparameters, model architecture, or pre-training objectives. Future research could look into the interactions between data size and these other factors to provide more comprehensive insights into the best practices for fine-tuning language models.

Finally, future research could look into the im-

pact of data size on the performance of fine-tuned models in multilingual settings. The paper’s authors briefly discuss the models’ multilinguality but need to investigate the impact of data size on the effectiveness of fine-tuned models for languages other than English. Such studies help to understand the generalizability of the findings to other languages and provide insights into the most effective ways to train multilingual fine-tuned models.

References

- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. [How transfer learning impacts linguistic knowledge in deep NLP models?](#) In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4947–4957, Online. Association for Computational Linguistics.
- Allyson Ettinger. 2020. [What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models](#).
- Mohammad Taher Pilehvar Houman Mehrafarin, Sara Rajaei. 2022a. [data-size-analysis](https://drive.google.com/drive/folders/1YuulbAQ-t-azSI7clM0_6ovabjvQy-skO?usp=share_link). https://drive.google.com/drive/folders/1YuulbAQ-t-azSI7clM0_6ovabjvQy-skO?usp=share_link. Fine-tuned models checkpoint by the authors.
- Mohammad Taher Pilehvar Houman Mehrafarin, Sara Rajaei. 2022b. [data-size-analysis](https://github.com/hmehrafarin/data-size-analysis). <https://github.com/hmehrafarin/data-size-analysis>. GitHub repository.
- Karthikeyan K, Shaily Bhatt, Pankaj Singh, Somak Aditya, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2022. [Multilingual Check-List: Generation and evaluation](#). In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 282–295, Online only. Association for Computational Linguistics.
- Houman Mehrafarin, Sara Rajaei, and Mohammad Taher Pilehvar. 2022. [On the importance of data size in probing fine-tuned models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 228–238, Dublin, Ireland. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020a. [Beyond accuracy: Behavioral testing of NLP models with Check-List](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020b. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Association for Computational Linguistics (ACL)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment tree-bank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

William B. Dolan. [Mrpc \(microsoft research phrase corpus\)](#).

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.