Online Appendix: A Tutorial on the Pretrain-Finetune Paradigm for Natural Language Processing*

Practical Exercises

In this online appendix, we provide two additional practical exercises to illustrate how researchers can finetune a large language model with relative ease to achieve state-of-the-art results in classification and regression, respectively.[1] All our exercises are written in Python in the format of Jupyter notebooks so that readers can follow along in an interactive manner.[2] For easy reproducibility and access to computation resources, all our computation is done on Google Colab.[3]

0.1 Multi-class classification

In the following case study, we classify a given text into one of eight topics. The texts are the speech transcripts from the New Zealand Parliament from 1987 to 2002 (Osnabrügge et al., 2021). In Figure 1, we illustrate the data input and list all the eight classes. We use using 2,915 annotated speech transcripts for training, 625 for validation, and another 625 for testing.

We finetune a RoBERTa-base model (Liu et al., 2019) for topic classification. RoBERTa-base has 12 layers of transformers and 125 million parameters in total. On top of its 12 layers of transformers, we add a classification layer for 8-topic classification. We use cross

*Our replication package can be accessed at https://doi.org/10.7910/DVN/QTT84C.
[1]Some of the original results on classification and regression have been published at *Political Analysis*.
[2]https://jupyter.org.
[3]https://colab.research.google.com.

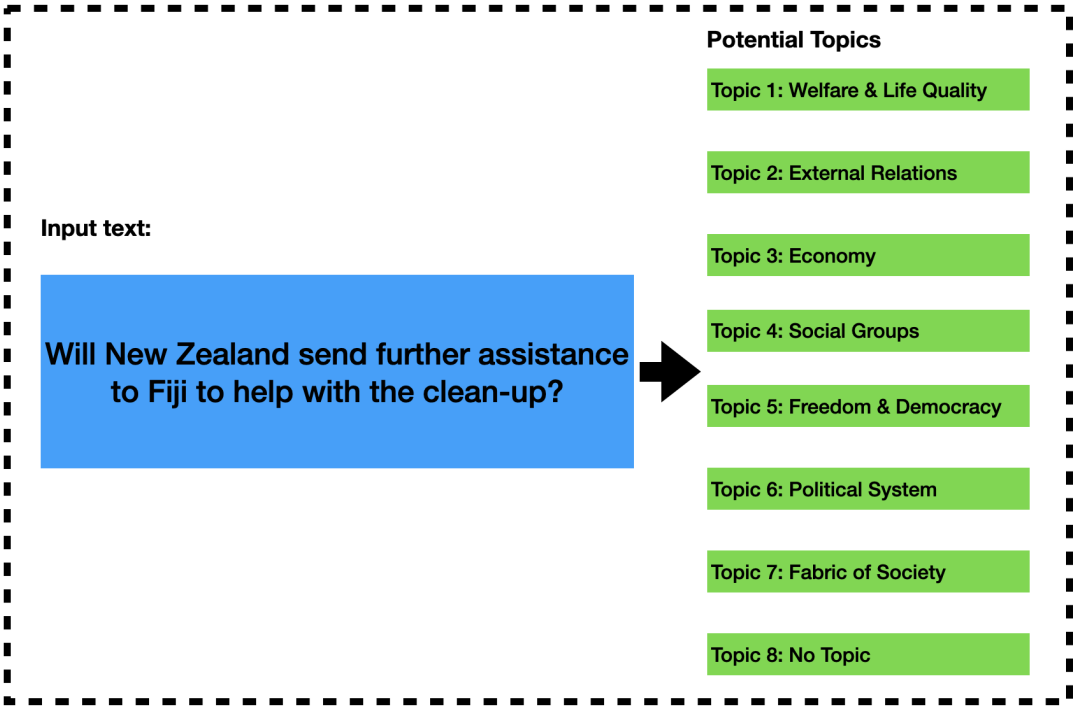

Figure 1: Given an input text, we categorize it into one of eight mutually exclusive classes. In the provided example, the text is classified under the second topic, External Relations.

entropy as the loss function. We finetune the RoBERTa-base model for 20 epochs with a learning rate of 2e-5, a batch size of 16, and an input sequence length of 512 on an A100 GPU. We use the validation set's cross entropy loss to select the best epoch and the optimal checkpoint. We then use the optimal checkpoint to make inferences on the test set with a batch size of 64 (see Listing 1). As our baseline, we use results from Osnabrügge et al. (2021), where the authors use 115,410 related but out-of-domain policy statements from Australia, Canada, Ireland, New Zealand, the United Kingdom, and the United States to train a regularized multinomial logistic regression model. For easy comparison, we use the same evaluation metrics as in Osnabrügge et al. (2021).

```
1
2     training_args = TrainingArguments(
3         output_dir="./results",          # output directory
4         num_train_epochs=epochs,         # total number of training epochs
5         per_device_train_batch_size=16,  # batch size for training
```

```
        per_device_eval_batch_size=64,   # batch size for evaluation
        learning_rate = 2e-5,            # learning rate
        save_strategy= "epoch",          # save checkpoints after each
    epoch
        evaluation_strategy="epoch",
        load_best_model_at_end= True,
    )

    def model_init():
        return RobertaForSequenceClassification.from_pretrained("roberta-
    base", num_labels=8)                  # number of labels
```

Listing 1: Specify the number of labels to be 8 for the multi-class classification and set the hyperparameters such as batch size and learning rate.

Across all metrics, finetuning the RoBERTa model with 2,915 New Zealand parliamentary speeches substantially outperforms the cross-domain topic classifier by Osnabrügge et al. (2021), which is trained using 115,420 annotated policy statements (Table 1). For example, for top-1 accuracy the finetuned RoBERTa model outperforms the cross-domain baseline by 26%. For top-3 accuracy the finetuned RoBERTa model outperforms the cross-domain baseline by 10%.

Table 1: Finetuning a RoBERTa-base model with 2,915 labeled in-domain samples can outperform the cross-domain regularized multinomial logistic regression model, trained with 115,420 out-of-domain samples, in the 8-topic classification task by a large margin. Cross-domain classifiers are from Osnabrügge et al. (2021). Test set is the same for both models. Mean of three random runs is reported, with standard deviation in the brackets. Better results are in bold.

Metrics	8 topics	
	Cross-domain	Finetuning LM
Top-1 accuracy	0.512 (0.007)	**0.643 (0.007)**
Top-3 accuracy	0.821 (0.001)	**0.899 (0.001)**
Top-5 accuracy	0.917 (0.007)	**0.968 (0.007)**
Balanced accuracy	0.465 (0.004)	**0.592 (0.004)**
F1 macro	0.456 (0.011)	**0.584 (0.011)**

0.2 Regression

In regression tasks, researchers use an input text to predict a numerical value. In the following exercise, we predict the natural logarithm of fatalities on a country month level using the CrisisWatch text (Häffner et al., 2023). For a visual representation of the input and output, please refer to Figure 2.

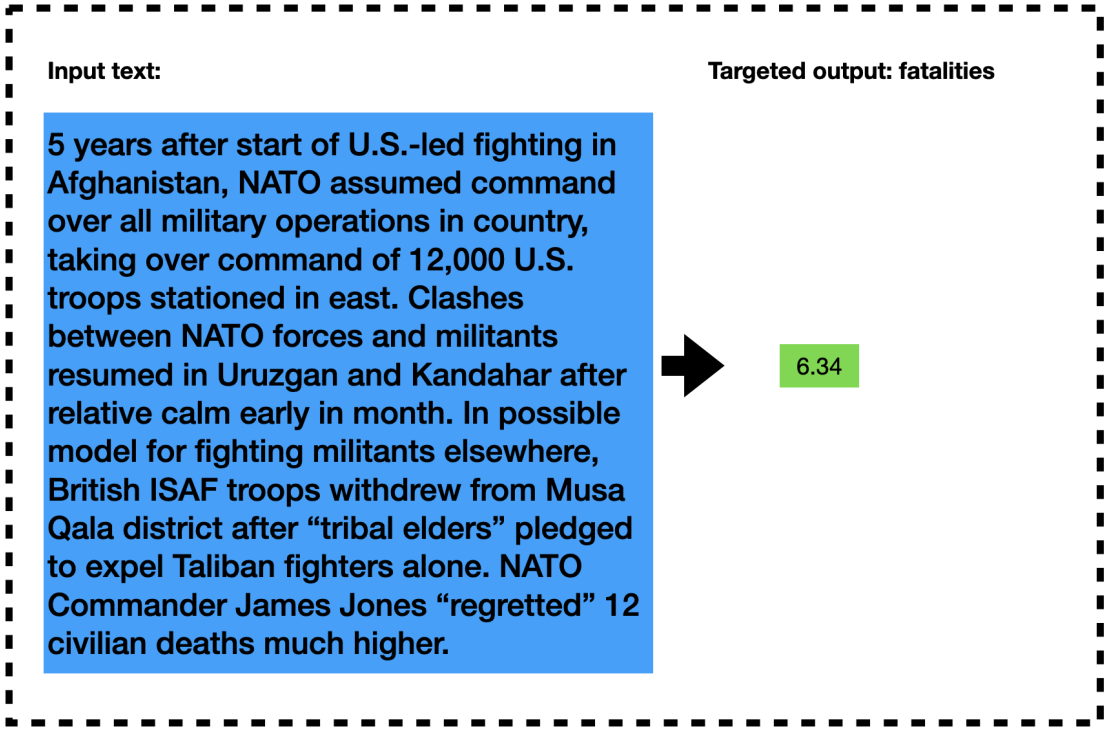

Figure 2: Given an input text, we predict a numerical value. In the example above, given an expert-written International Crisis Group (ICG) CrisisWatch report on Afghanistan for October, 2006, the model is expected to predict 6.34 as the natural logarithm of fatalities.

In Häffner et al. (2023), for the task of predicting the natural logarithm of fatalities on a country month level using the CrisisWatch text, the authors use random forest and XGBoost.[4] The training set covers the period from 2003 to first half of 2020 and has 21,924 samples, the validation set covers the second half of 2020 and has 648 samples, and the test set covers the year 2021 and has 1296 samples. These two models achieve an R^2 of 0.64 and 0.63 respectively and an MSE of 1.59 and 1.60 respectively. We use the same

[4]For an introduction to these tree-based models, readers can refer to Pargent et al. (2023) and Wang (2019).

training set, validation set and testing set as in Häffner et al. (2023). The large language model we use is ConfliBERT (Hu et al., 2022). ConfliBERT has the same architecture as BERT-base and is pretrained from scratch using a large corpus in the politics and conflicts domain.[5] When we finetune ConfliBERT using CrisisWatch texts, we set the learning rate to 2e-5 and set the number of training epochs from 10 (see Listing 2).

```
BASE_MODEL = "snowood1/ConfliBERT-scr-uncased"  # pretrained model
LEARNING_RATE = 2e-5
MAX_LENGTH = 256
BATCH_SIZE = 16
EPOCHS = 10

training_args = TrainingArguments(
    learning_rate=LEARNING_RATE,                # learning rate
    per_device_train_batch_size=BATCH_SIZE,     # training batch size
    per_device_eval_batch_size=BATCH_SIZE,      # test batch size
    num_train_epochs=EPOCHS,                    # number of training epochs
    seed=123
)

tokenizer = AutoTokenizer.from_pretrained(BASE_MODEL)

def model_init():
    model = AutoModelForSequenceClassification.from_pretrained(BASE_MODEL,
    num_labels=1, ignore_mismatched_sizes=True)  # number of labels
    return model

def preprocess_function(examples):
    label = examples["fatalities_log"]
    examples = tokenizer(examples["final_text"], truncation=True, padding="
    max_length", max_length=256)                # max sequence length
    examples["label"] = float(label)
    return examples
```

[5] https://github.com/eventdata/ConfliBERT/.

```
26
27 class RegressionTrainer(Trainer):
28     def compute_loss(self, model, inputs, return_outputs=False):
29         labels = inputs.pop("labels")
30         outputs = model(**inputs)
31         logits = outputs[0][:, 0]
32         loss = torch.nn.functional.mse_loss(logits, labels)
33         return (loss, outputs) if return_outputs else loss
```

Listing 2: Specify the number of labels to be 1 for the regression task and set the hyperparameters such as batch size and learning rate.

We report two groups of finetuning results: *ConfliBERT Length-256*, where we set the max number of tokens to 256 and *ConfliBERT Length-512*, where we set the maximum sequence length to 512 (Wang, 2023).[6] In Table 2, we compare the performance of finedtuned ConfliBERT models with that of two baseline models. OCoDi-Random Forest is the dictionary-based model that leverages random forest, where OCoDi stands for Objective Conflict Dictionary. OCoDi-XGBoost is the dictionary-based model that leverages XGBoost (Häffner et al., 2023).

Table 2: Finetuned ConfliBERT models outperform dictionary-based models by a large margin. Results in Columns 1 and 2 are from Table 2 in Häffner et al. (2023). By increasing the maximum sequence length to 512, we are able to further improve the performance of those finetuned models. Best results in bold.

Model	OCoDi Random Forest (1)	OCoDi XGBoost (2)	ConfliBERT Length-256 (3)	ConfliBERT Length-512 (4)
MSE	1.59	1.60	0.99	**0.82**
R^2	0.64	0.63	0.77	**0.81**

We observe that by finetuning ConfliBERT (Column 3), we are able to achieve a much lower mean squared error and a substantially higher R^2 than the dictionary-based

[6] We note that 3,386 out of 23,868 samples (14%) contain more than 256 tokens. By setting the max sequence length to 256, we are effectively truncating these long samples to 256 tokens, thus reducing the amount of information that we give to the model. Setting the max sequence length to 512, which is the longest input length possible for BERT models, helps alleviate this problem.

approaches. Further, by increasing the maximum sequence length from 256 to 512 (Column 4), we observe that the mean squared error on the test set decreases from 0.99 to 0.82 and that the R^2 on the test set increases from 0.77 to 0.81. Comparing *OCoDi-XGBoost* and *ConfliBERT Max Length*, we observe that a finetuned ConfliBERT is able to achieve an MSE that is 49% lower than OCoDi-XGBoost and an R^2 that is 29% higher.

References

Hu, Y., Hosseini, M., Parolin, E. S., Osorio, J., Khan, L., Brandt, P. T., & D'Orazio, V. J. (2022). Conflibert: A pre-trained language model for political conflict and violence. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*.

Häffner, S., Hofer, M., Nagl, M., & Walterskirchen, J. (2023). Introducing an interpretable deep learning approach to domain-specific dictionary creation: A use case for conflict prediction. *Political Analysis*, 1–19. doi: 10.1017/pan.2023.7

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*.

Osnabrügge, M., Ash, E., & Morelli, M. (2021). Cross-Domain Topic Classification for Political Texts. *Political Analysis*.

Pargent, F., Schoedel, R., , & Stachl, C. (2023). Best practices in supervised machine learning: A tutorial for psychologists. *Advances in Methods and Practices in Psychological Science*.

Wang, Y. (2019). Comparing Random Forest with Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data: A Comment. *Political Analysis*, *21*(1), 107-110.

Wang, Y. (2023). Topic Classification for Political Texts with Pretrained Language Models. *Political Analysis*.