A Report on

Experiments

Prepared By:

Siddharth P. Shah(19PH01)

Ph.D. Scholar, Faculty of Technology (Computer Engineering)

Dharmsinh Desai University, Nadiad

Under the supervision of:

Dr. Brijesh S. Bhatt

Professor, Faculty of Technology (Computer Engineering)

Dharmsinh Desai University, Nadiad

March 24, 2022

Contents

List of Tables

Experiment 1

Fine-tune and Test t5 Models

1.1 Objective

In this experiment t5-small and t5-base models are fine-tuned for text summarization task and then tested. The CNN / Daily Mail dataset is used for fine-tuning as well as testing purpose.

1.2 Expectations

We expect the similar/comparable results claimed in the original research paper describing t5 models.

1.3 Steps

1.3.1 Fine-tuning t5-small model

We fine tune t5-small model on cnn/daily mail train and validation dataset using standard python script (run_summarization.py) available on hugging face platform. Following is the command, that we used to start the fine-tuning. The fine-tuning process took around **5 days** of time to train and validate the model for summarization task.

```
python run_summarization.py \
  --model_name_or_path t5-small \
  --do_train \
  --do_eval \
  --dataset_name cnn_dailymail \
```

```
    --dataset_config "3.0.0" \
    --source_prefix "summarize: " \
    --output_dir ./fine-tuned-models/t5-small-cnn  \
    --per_device_train_batch_size=4 \
    --per_device_eval_batch_size=4 \
    --overwrite_output_dir \
    --predict_with_generate
```

1.3.2 Fine-tuning t5-base model

We fine tune t5-base model on cnn/daily mail train and validation dataset using standard python script (run_summarization.py) available on hugging face platform. Following is the command, that we used to start the fine-tuning. The fine-tuning process took around **22 days** of time to train and validate the model for summarization task.

```
python run_summarization.py \
    --model_name_or_path t5-base \
    --do_train \
    --do_eval \
    --dataset_name cnn_dailymail \
    --dataset_config "3.0.0" \
    --source_prefix "summarize: " \
    --output_dir ./fine-tuned-models/t5-base-cnn  \
    --per_device_train_batch_size=2 \
    --per_device_eval_batch_size=2 \
    --overwrite_output_dir \
    --predict_with_generate
```

1.3.3 Test t5-small model using test dataset

We tested the fine-tuned t5-small model using standard test dataset from CNN/DM. Following is the command, that we used to test the fine-tuned model. The testing took around **6 h 33 m** of time to test the model for summarization task.

```
python3 run_summarization.py \
    --model_name_or_path t5-small-cnn \
    --do_predict \
    --dataset_name cnn_dailymail \
    --dataset_config "3.0.0" \
    --source_prefix "summarize: " \
    --output_dir ./fine-tuned-models/t5-small-cnn-test \
    --per_device_eval_batch_size=16 \
    --overwrite_output_dir \
    --predict_with_generate
```

1.3.4 Test t5-base model using test dataset

We tested the fine-tuned t5-base model using standard test dataset from CNN/DM. Following is the command, that we used to test the fine-tuned model. The testing took around **1 day** of time to test the model for summarization task.

```
python3 run_summarization.py \
  --model_name_or_path t5-base-cnn \
  --do_predict \
  --dataset_name cnn_dailymail \
  --dataset_config "3.0.0" \
  --source_prefix "summarize: " \
  --output_dir ./fine-tuned-models/t5-base-cnn-test \
  --per_device_eval_batch_size=16 \
  --overwrite_output_dir \
  --predict_with_generate
```

1.4 Results

We have used ROUGE - 1, 2, L and L_{sum} as evaluation metrics. The following table shows results of fine-tuning and testing the t5-small and t5-base models. In case of fine-tuning the results are taken on standard validation dataset, whereas in case of testing they are taken on standard test dataset.

Table 1.1: Results - Fine-tuned t5 models

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L_{sum}
Validation Results (ours)				
t5-small	41.62	19.18	29.48	38.82
t5-base	43.81	20.96	31.17	40.97
Test Results (ours)				
t5-small	41.05	18.73	29.13	38.25
t5-base	43.14	20.35	30.74	40.26
Test Results (from original paper)				
t5-small	41.12	19.56	–	38.35
t5-base	42.05	20.34	–	39.40

1.5 Conclusion

From the results, it can be seen that we also achieved the similar results as claimed by the authors.

Experiment 2

Test t5 Models with trained adapters

2.1 Objective

In this experiment t5-small and t5-base models along with summarization task adapter trained on CNN / Daily Mail are tested for text summarization task.

2.2 Expectation

We expect poor result as the adapter is trained with BART-large model and we are using it with t5 model.

2.3 Steps

2.3.1 Test t5-small model with adapter

We tested the t5-small model along with summarization task adapter using standard test dataset from CNN/DM. Following is the command, that we used to test the fine-tuned model. The testing took around **1 d 7 h 27 m** of time to test the model for summarization task.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-small \
--do_predict \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
```

```
--output_dir ./adapter-transformer/t5-small-cnn-adapter \
--per_device_eval_batch_size=16 \
--overwrite_output_dir \
--load_adapter @ukp/facebook-bart-large_sum_cnn_dailymail_pfeiffer \
--predict_with_generate
```

2.3.2 Test t5-base model with adapter

We tested the t5-base model along with summarization task adapter using standard test dataset from CNN/DM. Following is the command, that we used to test the fine-tuned model. The testing took around **7 h 29 m** of time to test the model for summarization task.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-base \
--do_predict \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
--output_dir ./adapter-transformer/t5-base-cnn-adapter \
--per_device_eval_batch_size=16 \
--overwrite_output_dir \
--load_adapter @ukp/facebook-bart-large_sum_cnn_dailymail_pfeiffer \
--predict_with_generate
```

2.4 Results

We have used ROUGE - 1, 2, L and L_{sum} as evaluation metrics. The following table shows results of testing the t5-small and t5-base models with adapter on standard test dataset.

Table 2.1: Results - T5 models with trained adapter

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L_{sum}
t5-small	30.63	11.50	21.66	28.41
t5-base	11.50	1.93	8.81	10.54

2.5 Conclusion

- The results are very poor as specially with t5-base model.

- t5-base model took less time to run than t5-small model.

Experiment 3

Train adapter on t5 Models

3.1 Objective

In this experiment a new task adapter for summarization will be trained and tested on t5-small and t5-base models.

3.2 Expectation

We expect comparative results with fine-tuned models (as created in Experiment 1).

3.3 Steps

3.3.1 Train task adapter for summarization on t5-small

We trained an adapter on t5-small model using CNN/DM dataset. Following is the command, that we used to train the adapter. The adapter training took around **5 d 0 h 21 m** of time to complete the training and around **1 h 28 m**of time to complete the validation.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-small \
--do_train \
--do_eval \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
```

```
--output_dir ./t5-small-train-adapter-cnn \
--per_device_train_batch_size=32 \
--per_device_eval_batch_size=32 \
--overwrite_output_dir \
--train_adapter \
--predict_with_generate
```

3.3.2 Train task adapter for summarization on t5-base

We trained an adapter on t5-base model using CNN/DM dataset. Following is the command, that we used to train the adapter. The adapter training took around **5 d 19 h 33 m** of time to complete the training and around **4 h 34 m**of time to complete the validation.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-base \
--do_train \
--do_eval \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
--output_dir ./t5-base-train-adapter-cnn \
--per_device_train_batch_size=16 \
--per_device_eval_batch_size=16 \
--overwrite_output_dir \
--train_adapter \
--predict_with_generate
```

3.3.3 Test trained task adapter for t5-small using test dataset

We tested the trained adapter for t5-small model using standard test dataset from CNN/DM. Following is the command, that we used to test the adapter model. The testing took around **1 h 17 m** of time to test the model for summarization task.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-small \
--do_predict \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
--output_dir ./t5-small-train-adapter-cnn/test \
--per_device_eval_batch_size=32 \
--overwrite_output_dir \
--load_adapter ./t5-small-train-adapter-cnn/cnn_dailymail \
--predict_with_generate &
```

3.3.4 Test trained task adapter for t5-base using test dataset

We tested the trained adapter for t5-base model using standard test dataset from CNN/DM. Following is the command, that we used to test the adapter model. The testing took around **3 h 50 m** of time to test the model for summarization task.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-base \
--do_predict \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
--output_dir ./t5-base-train-adapter-cnn/test \
--per_device_eval_batch_size=16 \
--overwrite_output_dir \
--load_adapter ./t5-base-train-adapter-cnn/cnn_dailymail \
--predict_with_generate &
```

3.4 Results

We have used ROUGE - 1, 2, L and L_{sum} as evaluation metrics. The following table shows validation and test results when adapter is trained and used with the t5-small and t5-base models.

Table 3.1: Results - Adapter training with T5 models

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L_{sum}
Validation Results				
t5-small	41.20	18.81	29.09	38.42
t5-base	43.43	20.64	30.86	40.59
Test Results				
t5-small	40.56	18.40	28.74	37.80
t5-base	42.86	20.07	30.44	39.97

3.5 Conclusion

From the results, it can be seen that using trained adapter we achieved the comparable results with fine-tuned models. Whereas the time taken to train the adapter in case of t5-base models is very less. Table 3.2 shows the results comparison for standard test data.

Table 3.2: Results - Adapter vs Fine-tuned on Test data

Scores	**t5-small**		**t5-base**	
	Adapter	**Fine-tuned**	**Adapter**	**Fine-tuned**
ROUGE-1	40.56	41.05	42.86	43.14
ROUGE-2	18.40	18.73	20.07	20.35
ROUGE-L	28.74	29.13	30.44	30.74
ROUGE-L_{sum}	37.80	38.25	39.97	40.26

Experiment 4

Train and Test adapter on t5 Models for Headline Generation task

4.1 Objective

In this experiment a new task adapter for headline generation will be trained and tested on t5-small model.

4.2 Expectation

We expect good results.

4.3 Steps

4.3.1 Train task adapter for headline generation on t5-small

We trained an adapter on t5-small model using XLSUM (English) dataset. Following is the command, that we used to train the adapter. The adapter training took around **1 d 16 h 53 m** of time to complete the training and around **35 m** of time to complete the validation.

We also modified the code of adapter library file **loading.py** to support creation of sub directories. We replace **mkdir()** function by **makedirs()** function.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-small \
```

```
--do_train \
--do_eval \
--dataset_name GEM/xlsum \
--dataset_config "english" \
--source_prefix "summarize: " \
--output_dir ./t5-small-train-adapter-HL-XLSum \
--per_device_train_batch_size=16 \
--per_device_eval_batch_size=16 \
--overwrite_output_dir \
--train_adapter \
--predict_with_generate
```

4.3.2 Test trained task adapter for t5-small using test dataset

We tested the trained adapter for t5-small model using standard test dataset from XL-SUM (English). Following is the command, that we used to test the adapter model. The testing took around **0 day** of time to test the model for summarization task.

```
python3 run_summarization_adapter.py \
--model_name_or_path t5-small \
--do_predict \
--dataset_name cnn_dailymail \
--dataset_config "3.0.0" \
--source_prefix "summarize: " \
--output_dir ./t5-small-train-adapter-cnn/test \
--per_device_eval_batch_size=32 \
--overwrite_output_dir \
--load_adapter ./t5-small-train-adapter-cnn/cnn_dailymail \
--predict_with_generate &
```

4.4 Results

We have used ROUGE - 1, 2, L and L_{sum} as evaluation metrics. The following table shows validation and test results when adapter is trained and used with the t5-small model.

4.5 Conclusion

From the results, it can be seen that using adapter training we achieved the comparable results with fine-tuned models. Whereas the time taken to train the adapter in case of t5-base models is very less.

Table 4.1: Results - Train Adapter for Headline Generation

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L_{sum}
Validation Results				
t5-small	28.54	8.82	25.28	25.29
Test Results				
t5-small	28.45	8.83	25.22	25.23

Appendix A

Configuration

A.1 Hardware

All the experiments are performed on the machine with following hardware profile.

A.2 Software

A.3 Dataset

A.3.1 CNN / Daily Mail

The CNN / DailyMail Dataset is an English-language dataset containing just over 300k unique news articles as written by journalists at CNN and the Daily Mail. The version 3.0, used here, is not anonymized, so individuals' names can be found in the dataset. Information about the original author is not included in the dataset.

Data Fields

- **id:** a string containing the heximal formated SHA1 hash of the url where the story was retrieved from.
- **article:** a string containing the body of the news article
- **highlights:** a string containing the highlight of the article as written by the article author

Data Splits

The CNN/DailyMail dataset has 3 splits: train, validation, and test. Below are the statistics for Version 3.0.0 of the dataset.

Table A.1: CNN / Daily Mail - Data Splits

Dataset Split	**Number of Instances in Split**
Train	287,113
Validation	13,368
Test	11,490

A.4 Pre-trained Models

A.5 Adapter