

TAKE-HOME MIDTERM EXAM REPORT

MULTI-LABEL TEXT CLASSIFICATION MODEL ON SCOPUS DATASET

PREPARED FOR

Assoc. Prof. Peerapon Vateekul

PREPARED BY

Mr. Chaiyaphop Jamjumrat

(6670056421)

Data Science and Data Engineering Tools course (2110531) 2023/1

Faculty of Engineering at Chulalongkorn University

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION.....	2
CHAPTER 2: DATA PREPARATION.....	3
CHAPTER 3: MODEL.....	4
CHAPTER 4: RESULTS.....	5
CHAPTER 5: DISCUSSION.....	6
CHAPTER 6: CONCLUSION.....	6
BONUS POINT MODEL REPORT.....	7

Chapter 1: Introduction

This is the individual take-home midterm examination report for the Data Science and Data Engineering Tools course (2110531) in Computer Science, Faculty of Engineering at Chulalongkorn University.

The examination's objective is to research and develop a multi-label text classification model to categorize multi-content text (18 subject areas of engineering) on the limited Scopus dataset.

Data structure

Abstract, Title, and Classes

Subject areas classes

1. CE - Civil Engineering
2. ENV- Environmental Engineering
3. BME - Biomedical Engineering
4. PE - Petroleum Engineering
5. METAL- Metallurgical Engineering
6. ME - Mechanical Engineering
7. EE - Electrical Engineering
8. CPE - Computer Engineering
9. OPTIC - Optical Engineering
10. NANO - Nano Engineering
11. CHE - Chemical Engineering
12. MATENG - Materials Engineering
13. AGRI - Agricultural Engineering
14. EDU – Education
15. IE - Industrial Engineering
16. SAFETY - Safety Engineering
17. MATH - Mathematics and Statistics
18. MATSCI - Material Science

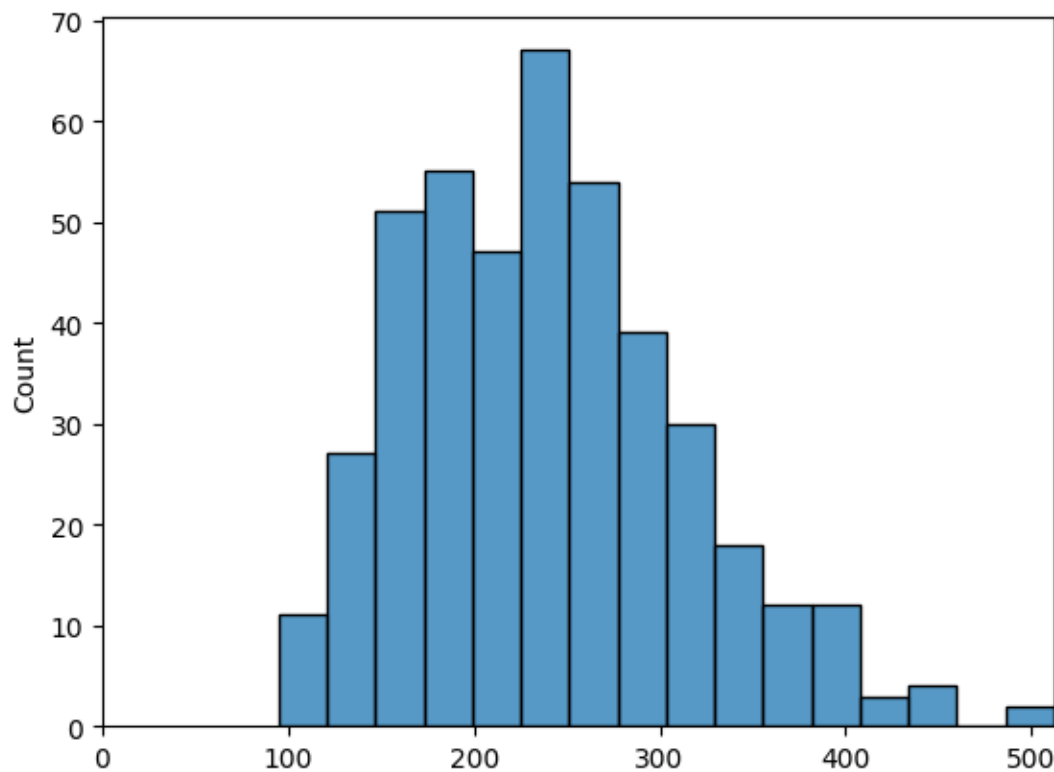
Chapter 2: Data preparation

Data Cleansing

1. Resetting the index because the index of the training dataset began with 1, which is inconvenient for future use and must be reset.
2. Checking for missing values in each feature.
3. Matching the abbreviation of subject areas classes with their full names.
4. Removing Unicode characters, links, punctuation, single letters, and numbers surrounded by space.

Data Preprocessing

After splitting the dataset into a training set and a test set, I used the AutoTokenizer from the pre-trained BART model to tokenize the data. And setting the number of maximum lengths by plotting the token distribution on the training set.



Chapter 3: Model

Methodology

I used the Pre-trained Facebook BART model (facebook/bart-large-mnli) on Hugging Face and fine-tuned it on Scopus dataset with 10 Epochs, Batch size = 4, Learning rate = 0.00002, Metric = F1 score

```
(classification_head): BartClassificationHead(
  (dense): Linear(in_features=1024, out_features=1024, bias=True)
  (dropout): Dropout(p=0.0, inplace=False)
  (out_proj): Linear(in_features=1024, out_features=18, bias=True)
```

The class imbalance problem

To solve this problem, I got the best threshold by finding the threshold that maximizes the F1 score.

```
Best Threshold at [0.179, 0.18] F1: 0.6103896103896105
```

Number of articles in each subject area:

```
# of articles in civil engineering: 52 articles, 0.11%
# of articles in environmental engineering: 59 articles, 0.13%
# of articles in biomedical engineering: 36 articles, 0.08%
# of articles in petroleum engineering: 86 articles, 0.19%
# of articles in metallurgical engineering: 72 articles, 0.16%
# of articles in mechanical engineering: 90 articles, 0.2%
# of articles in electrical engineering: 118 articles, 0.26%
# of articles in computer engineering: 142 articles, 0.31%
# of articles in optical engineering: 31 articles, 0.07%
# of articles in nano engineering: 32 articles, 0.07%
# of articles in chemical engineering: 177 articles, 0.39%
# of articles in materials engineering: 64 articles, 0.14%
# of articles in agricultural engineering: 20 articles, 0.04%
# of articles in education: 32 articles, 0.07%
# of articles in industrial engineering: 74 articles, 0.16%
# of articles in safety engineering: 22 articles, 0.05%
# of articles in mathematics and statistics: 112 articles, 0.25%
# of articles in material science: 119 articles, 0.26%
```

Chapter 4: Results

Model Evaluation

Training and validation loss with F1 score, ROC AUC, and accuracy:

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
1	No log	0.383813	0.370044	0.650150	0.000000
2	No log	0.330461	0.511905	0.748177	0.000000
3	No log	0.319538	0.526882	0.777778	0.000000
4	No log	0.301788	0.548780	0.773059	0.000000
5	0.330800	0.274604	0.566265	0.788932	0.045455
6	0.330800	0.289237	0.564706	0.792364	0.045455
7	0.330800	0.274564	0.610390	0.806950	0.045455
8	0.330800	0.281606	0.581560	0.769841	0.045455
9	0.330800	0.283488	0.583333	0.774775	0.045455
10	0.138000	0.279889	0.575342	0.771772	0.045455

TrainOutput(global_step=1080, training_loss=0.22412561884632817, metrics={'train_runtime': 1987.5917, 'train_samples_per_second': 2.173, 'train_steps_per_second': 0.543, 'total_flos': 4695120074833920.0, 'train_loss': 0.22412561884632817, 'epoch': 10.0})

Summary of model evaluation on validation set:

```
[ ] trainer.evaluate()

[6/6 00:02]
{'eval_loss': 0.27456387877464294,
 'eval_f1': 0.6103896103896105,
 'eval_roc_auc': 0.806949806949807,
 'eval_accuracy': 0.045454545454545456,
 'eval_runtime': 2.8757,
 'eval_samples_per_second': 7.65,
 'eval_steps_per_second': 2.086,
 'epoch': 10.0}
```

Model evaluation on test set:

- Before optimizing the threshold.




Macro F1 score: 0.54086

Rank: 18th

18	6670056421_Chaiy aphop_Jam		0.54086	22	6d
----	-------------------------------	---	---------	----	----

- After optimizing the threshold.

Macro F1 score: 0.58461

Submission and Description		Private Score 	Public Score 
	cj_final_result (1).csv Complete (after deadline) · 1h ago	0.58461	0.58461

Chapter 5: Discussion

I believe there are many things that I have not done due to knowledge, time, and computer unit limitations. The model can be improved by performing more data cleansing processes, experimenting on other pre-trained models such as Roberta or other tokenizers, and training on larger batch sizes with larger GPUs and unlimited compute units to reduce training time and increase experimental time.

Chapter 6: Conclusion

This report explains how to build the multi-label text classification model on Scopus dataset by using the Pre-trained Facebook BART model on Hugging Face with fine-tuned the model on Scopus dataset, optimized the threshold for finding the optimal threshold, prepared data with AutoTokenizer, and presented the model's results before and after optimized the threshold.

Bonus point model report

Data structure

Abstract, Title, and Classes

Subject areas classes

1. CE - Civil Engineering
2. ME - Mechanical Engineering
3. EE - Electrical Engineering
4. CPE - Computer Engineering
5. CHE - Chemical Engineering
6. IE - Industrial Engineering
7. AUTO - Automotive Engineering

Methodology

I applied the processes of the 18 classes model to the 7 classes model and obtained a better result because the 7 classes model had fewer classes and a lower class imbalance problem, which may lead to a better result than the 18 classes model.

Number of articles in each subject area:

```
# of articles in civil engineering: 154 articles, 0.34%
# of articles in mechanical engineering: 121 articles, 0.27%
# of articles in electrical engineering: 178 articles, 0.39%
# of articles in computer engineering: 124 articles, 0.27%
# of articles in chemical engineering: 204 articles, 0.45%
# of articles in industrial engineering: 287 articles, 0.63%
# of articles in automotive engineering: 119 articles, 0.26%
```

Result

Training and validation loss with F1 score, ROC AUC, and accuracy:

Epoch	Training Loss	Validation Loss	F1	Roc AUC	Accuracy
1	No log	0.628061	0.567376	0.524835	0.000000
2	No log	0.589670	0.618401	0.644033	0.000000
3	No log	0.531369	0.667707	0.702603	0.011111
4	No log	0.504321	0.661491	0.695378	0.011111
5	No log	0.541268	0.697761	0.747823	0.122222
6	0.449900	0.547333	0.686726	0.733802	0.100000
7	0.449900	0.583882	0.692699	0.736291	0.077778
8	0.449900	0.619641	0.699065	0.749115	0.144444
9	0.449900	0.631230	0.682657	0.733372	0.133333
10	0.449900	0.639700	0.692593	0.742655	0.144444

TrainOutput(global_step=910, training_loss=0.2868095481788719, metrics={'train_runtime': 1770.4027, 'train_samples_per_second': 2.056, 'train_steps_per_second': 0.514, 'total_flos': 3955947318435840.0, 'train_loss': 0.2868095481788719, 'epoch': 10.0})

Summary of model evaluation on validation set:

```
trainer.evaluate()  
[23/23 00:10]  
{'eval_loss': 0.6196413636207581,  
  'eval_f1': 0.6990654205607476,  
  'eval_roc_auc': 0.7491147478227581,  
  'eval_accuracy': 0.1444444444444443,  
  'eval_runtime': 11.4829,  
  'eval_samples_per_second': 7.838,  
  'eval_steps_per_second': 2.003,  
  'epoch': 10.0}
```

Model evaluation on test set - Macro F1 score: 0.71655

```
f1 0.7165532879818594  
auc 0.7751982579554142  
acc 0.152317880794702
```