Case Presentation 1

Team CT06

資科工碩一 310551056 何政儒
資科工碩一 310551118 簡維成
資科工碩一 310551165 簡言安

Github Page : https://reurl.cc/L7lg3e

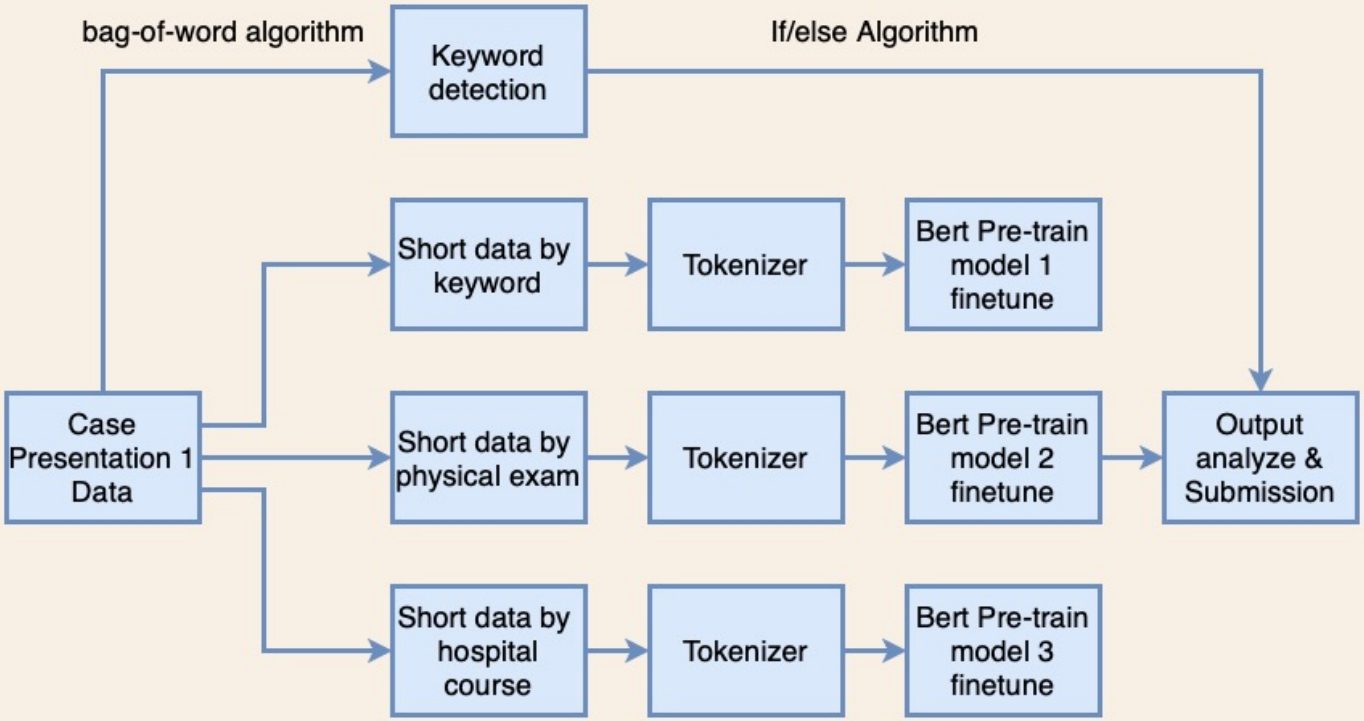
Time Cost

Index

01
Project Pipeline

What’s our plan?

Project Pipeline



02
Data Analysis

What’s important in the data?

Data Analysis

Use Keyword to capture data.

- Keyword: obese, obesity, overweight, heavy, fat…
- However,
  - heavy lifting, heavy smoking, heavy exercise
  - Saturated fat

obese	texture	intuitive
Y	191	149
N/U	0	11

obesity	texture	intuitive
Y	189	129
N/U	0	21

- "Obese" and "Obesity" means Label Y?
  - Usually Y: morbidly obese, an obese male/female
  - Not Sure: abdomen : obese (may be different in texture-based and intuitive-based)
  - Detect if there is an negative word. (non-obese, not obese)

Data Analysis

Use Keyword to capture data.

- How to find more keyword? Bag-of-words

Keyword	Y ratio
Sleep apnea 睡眠窒息症	0.776 0.946
Nasal 鼻子	0.638 0.725
Hypoxia 缺氧	0.8 0.778
Obstructive 阻礙	0.698 0.765
OSA 阻塞型睡眠呼吸中止症	0.818 0.885
Morbid obese 病態肥胖	0.922 0.983

Data Analysis

Use title name to capture data.

- How about those article without keyword?
  - Bert model has the limitation: up to 512 token.
  - We need to capture the more informative content.

- Unfortunately, not all the title names appear in all data.
  - History of present illness
  - Medical history
  - Physical examination
  - Other diagnoses
  - Hospital course
  - Discharge diagnosis

Data Analysis

Use title name to capture data.

- Why Physical examination and Hospital course ?
  - Those two title name usually appear in train/test data.
  - We find that the word "obesity", "obese" usually appears in those p aragraph. So we believe there should be some useful information in those paragraph.
  - Moreover, those two word appear in all the validation data.

	Physical exam	Hospital course	One of them
Train	198	383	390
Test	199	384	391
Validation	42	46	50

03
Data Pre-Processing

How to prepare the useful data for Bert model?

Data Pre-Processing

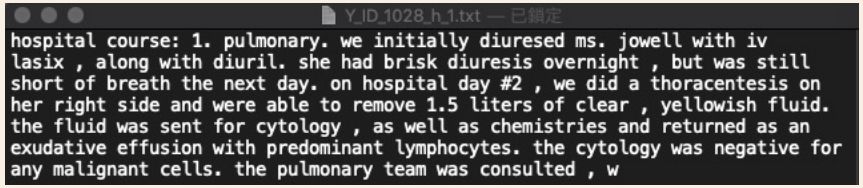
Get the clean data

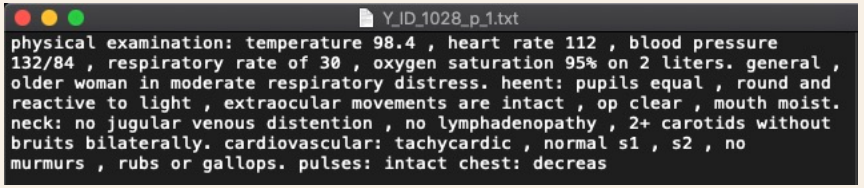
- Remove the redundancy space and "\n" then change it to lower case.
- Capture the useful content from txt file and save it to a new txt file.
  - Y_ID_1028.txt contain ["obese" x1, "obesity" x1, "hospital course" x 1, "physical examination" x1]
  - We will produce
    - Y_ID_1028_keyword_1.txt (for obese)
    - Y_ID_1028_keyword_2.txt (for obesity)
    - Y_ID_1028_h_1.txt (for hospital course)
    - Y_ID_1028_p_1.txt (for physical examination)
- We have 230 training data from "physical examination" and 231 training data from "hospital course".

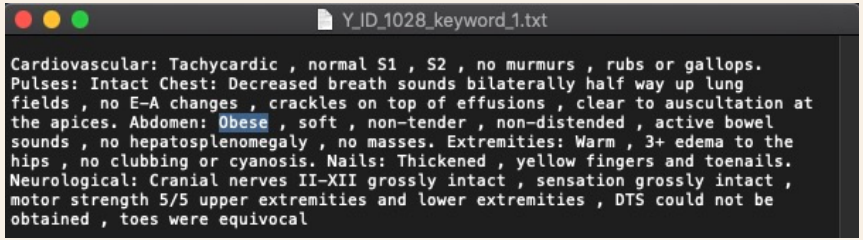
Data Pre-Processing

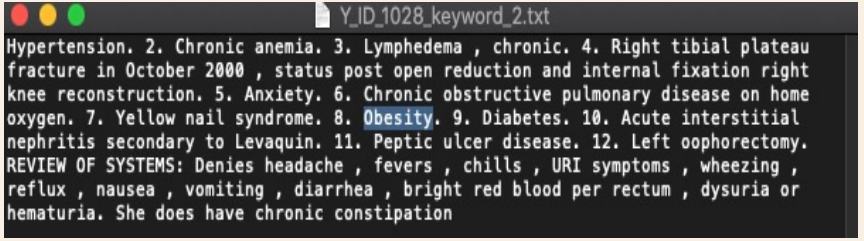
Prepare the small but useful data

- Our data would be…









- How to control the length of the content?
  - Search_Len (100, 200, 500, 1000)

04
Model Training

Use Pre-train model to improve the performance!

Model Training

What is Bert?

- [NIPS17] Attention is all you need.
- Use the Transformer (encoder-decoder) architecture.
- Pre-train model are powerful and can better solve NLP problem.

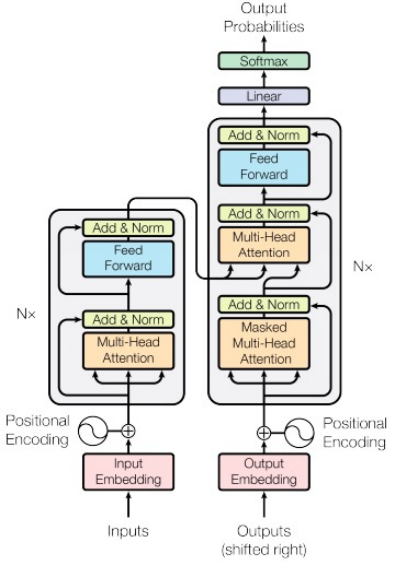
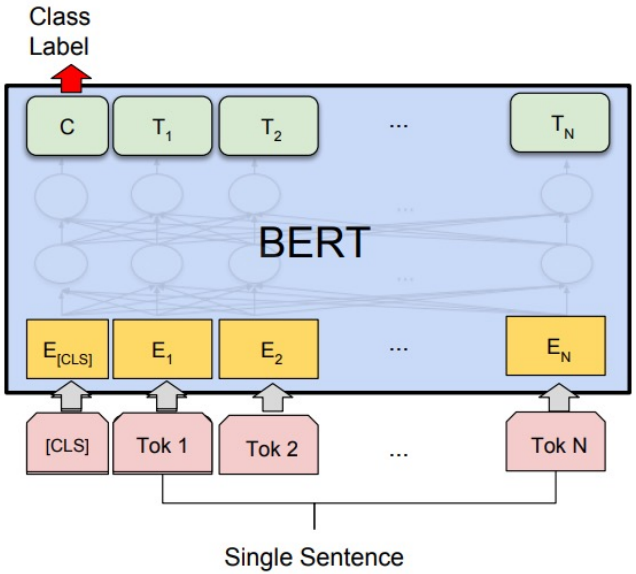


Figure 1: The Transformer - model architecture.



Model Training

Training process

- Model Version = 'bert-base-uncased' (12-layer, 768-hidden, 12-heads, 1 10M parameters)
- tokenizer = BertTokenizer.from_pretrained(model_version)

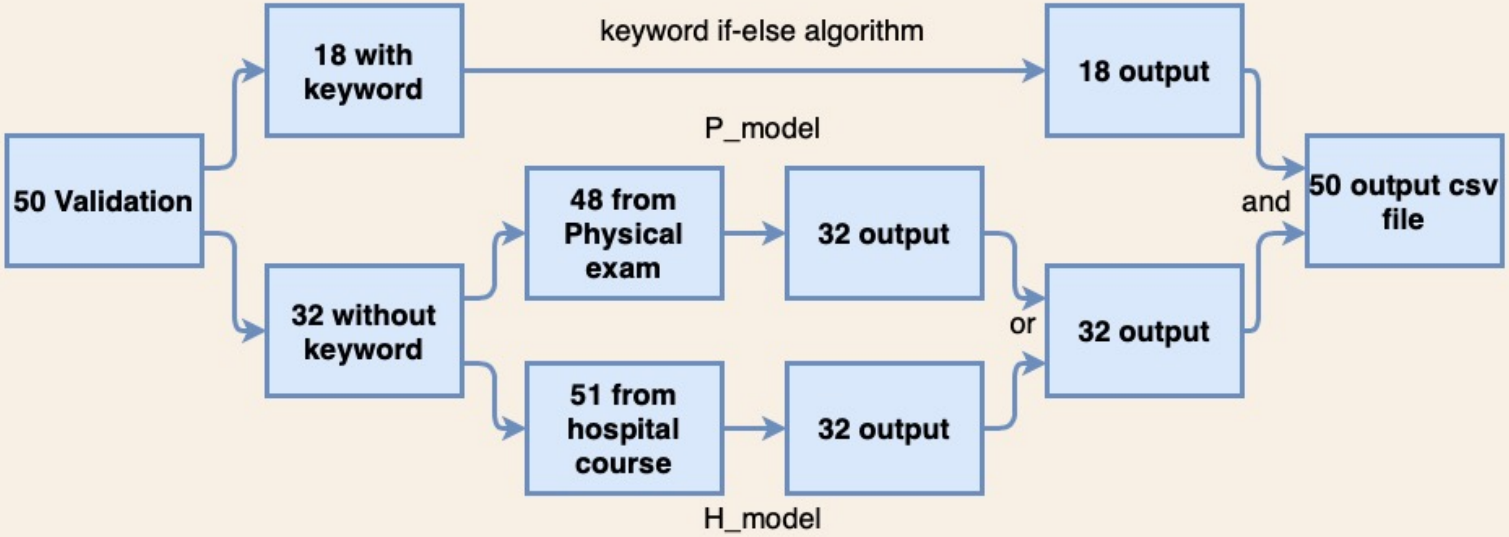
- Dataloader
  - Tokenize our input data by tokenizer.
  - Add [cls] and [sep] and create the segment.
  - Batch padding and create the mask.

- Model = BertForSequenceClassification.from_pretrained()
- Optimizer : Adam

Model Training

Output

- We will combine the multiple Bert model output (from physical exam / hospital course).
- Also, we will also combine the output from keyword if-else algorithm.



05
Result Analysis

How is the model performance and how to choose the final answer?

Result Analysis

Kaggle Leaderboard

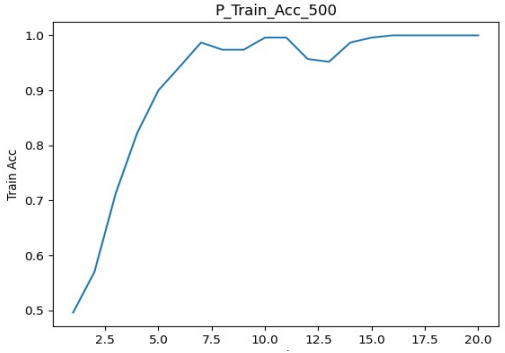
- We achieve 0.65714 on Public leaderboard test.

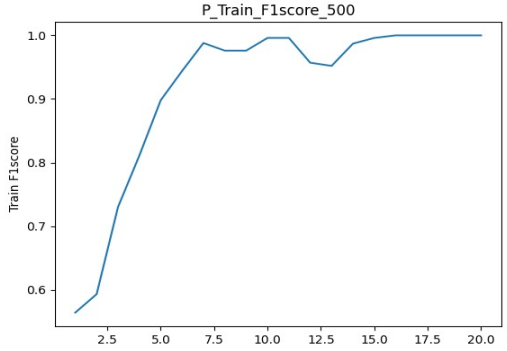
#	Team Name	Notebook	Team Members	Score	Entries	Last
1	CT_05			0.68571	19	4d
2	CT_03			0.65714	45	7h
3	YM_3			0.65714	20	1d
4	CT_01			0.65714	41	11h
5	CT_06			0.65714	23	2h

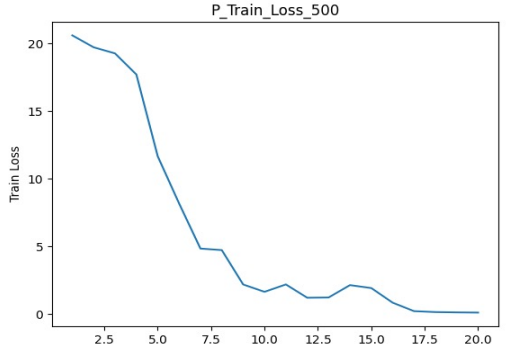
- We find Search_Len = 500 and 1000 can achieve better performance.

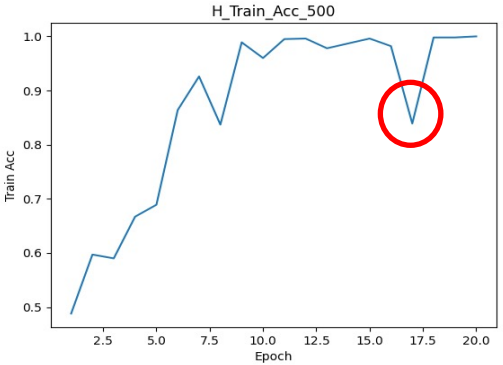
Result Analysis

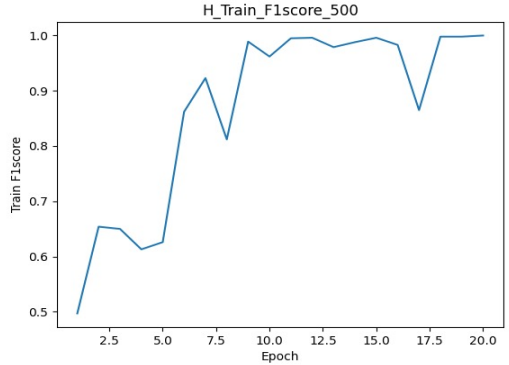
Train Result when Search_Len = 500

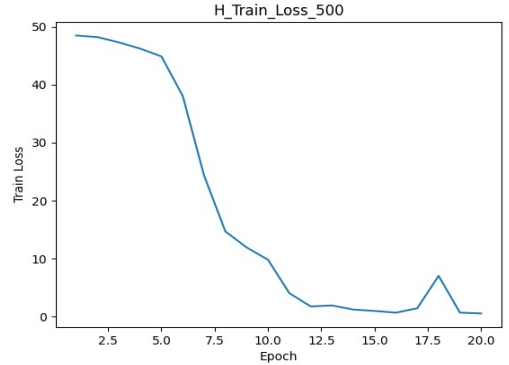






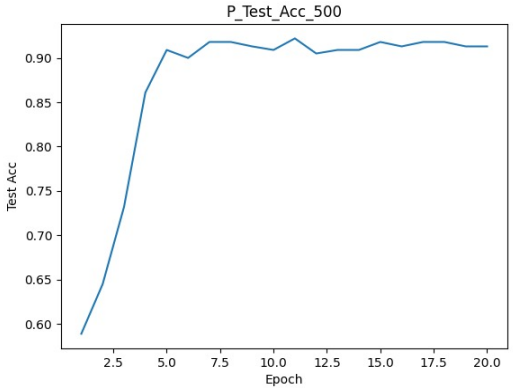


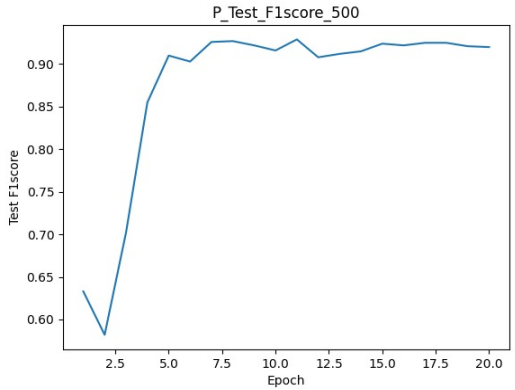


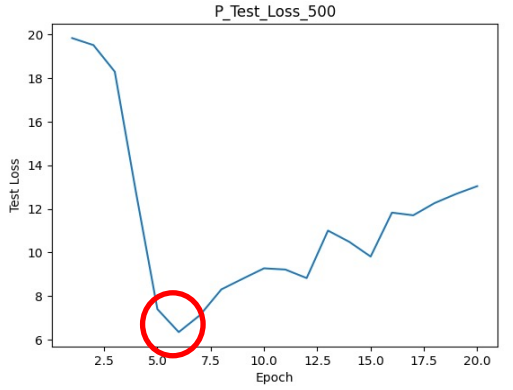


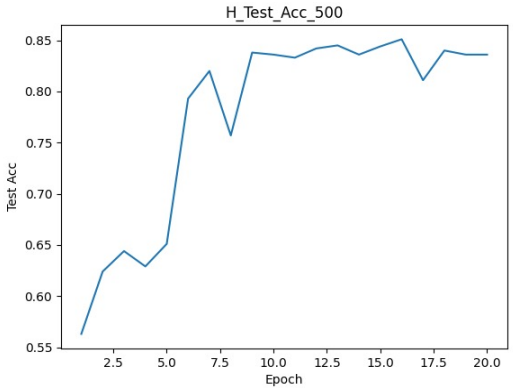
Result Analysis

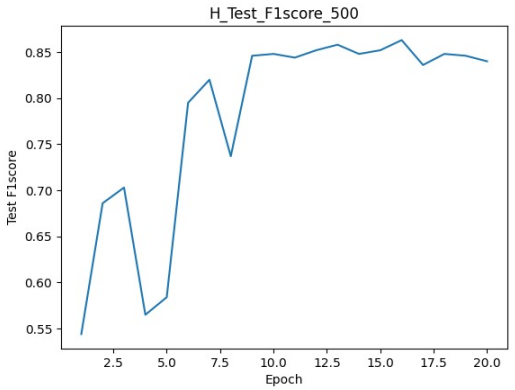
Test Result when Search_Len = 500

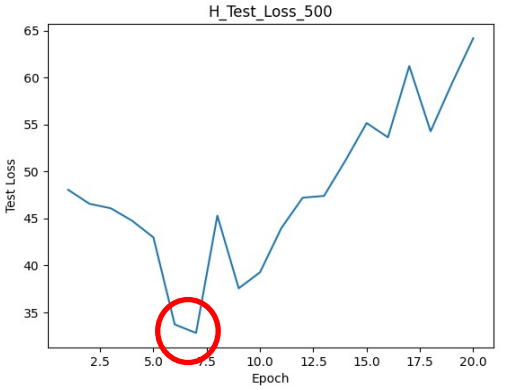












Result Analysis

Feedback

- It is obvious that we still have overfitting problem.
  - We can choose smaller Search_Len.
  - We can try larger model (Bert-large-uncased)
  - We may adjust the learning rate.
  - We need to put more effort on data pre-processing.

- Bert model vs LSTM vs SVM
  - Bert model is SOTA method.
  - The transformer architecture has been proved to be powerful.
  - Pre-train model can save time and increase performance.
  - It has the length limitation, so we need to carefully deal with the da ta.

Thank you

Contribution

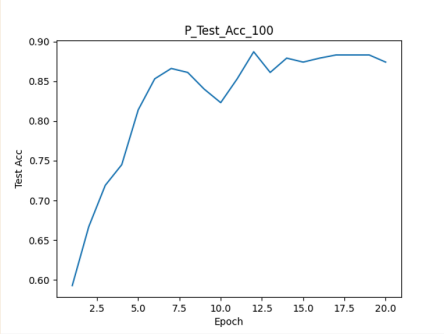
- 310551056　何政儒　40 %
- 310551118　簡維成　30 %
- 310551165　簡言安　30 %

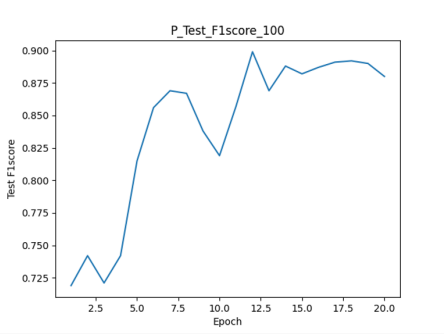
Reference

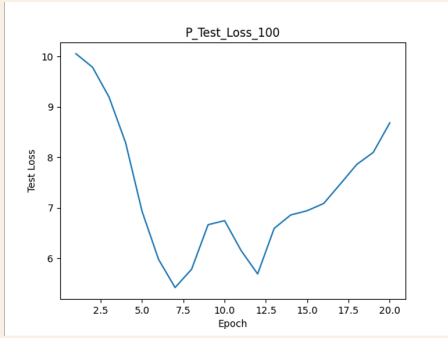
- [NIPS17] Attention is all you need
- Implementation: 進擊的BERT：NLP界的巨人之力與遷移學習
- ModelZoo：pytorch-pretrained-BERT
- Med-BERT
- 臨床NLP – 從病例紀錄預測病患是否再次入院
- Build Your First Text Classification model using PyTorch

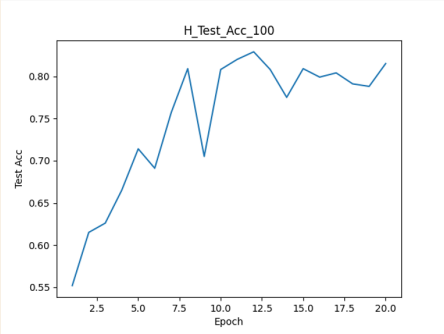
Appendix

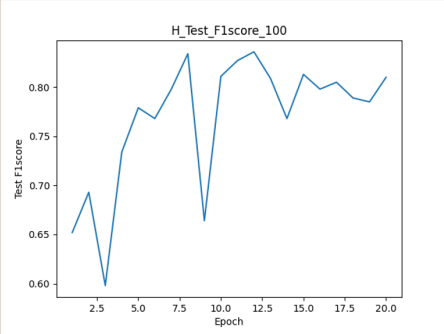
Experiment Result

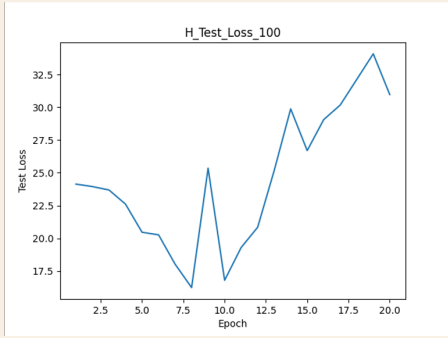






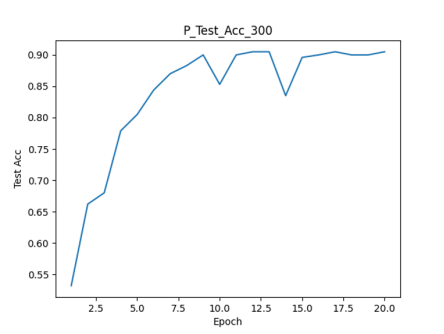


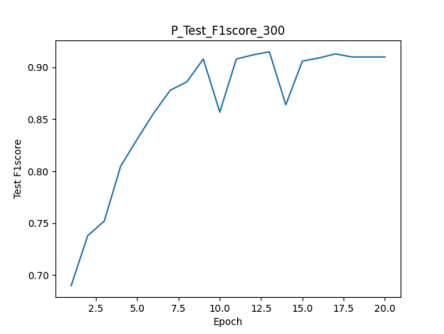


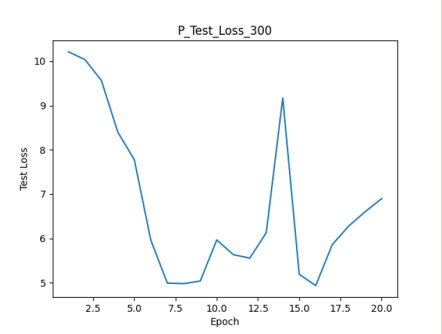


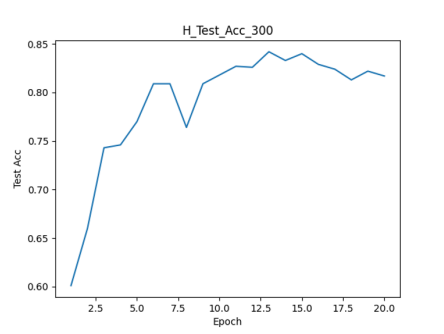
Appendix

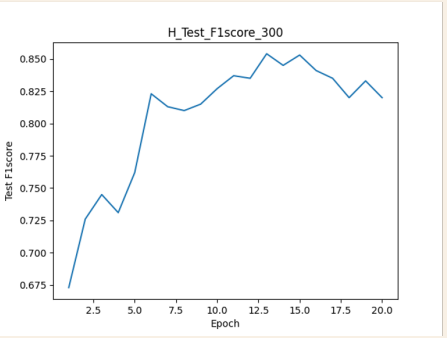
Experiment Result

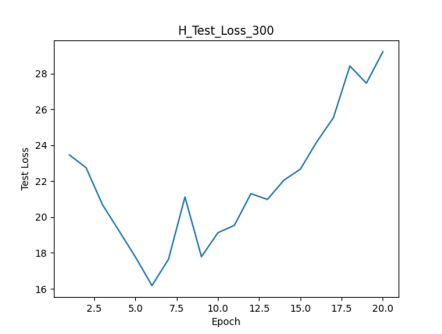






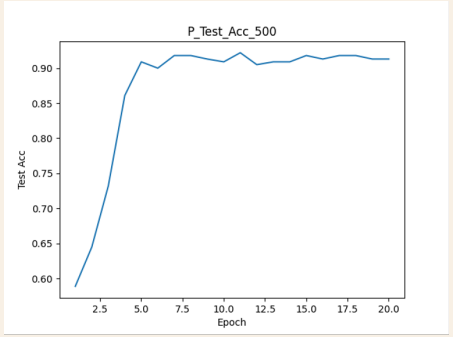


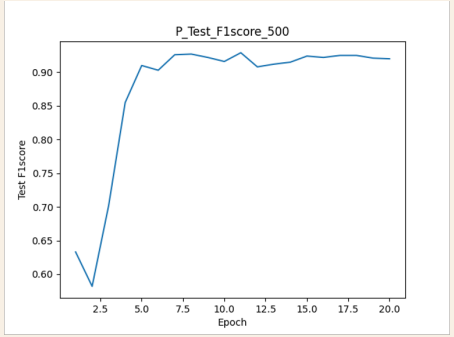


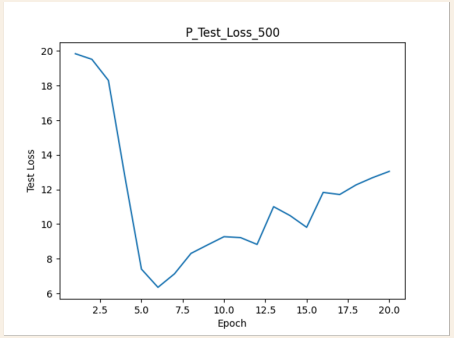


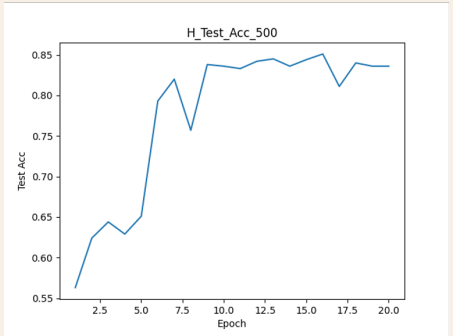
Appendix

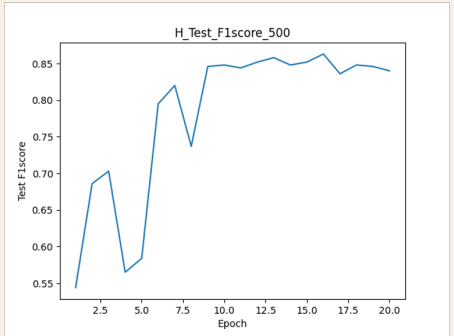
Experiment Result

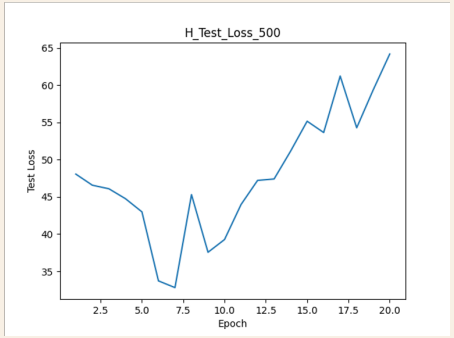






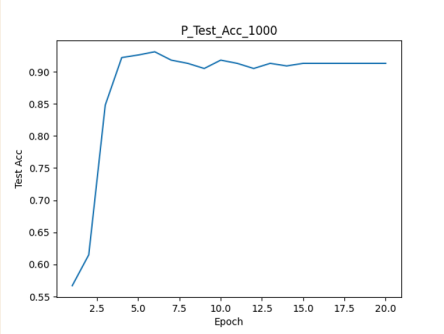


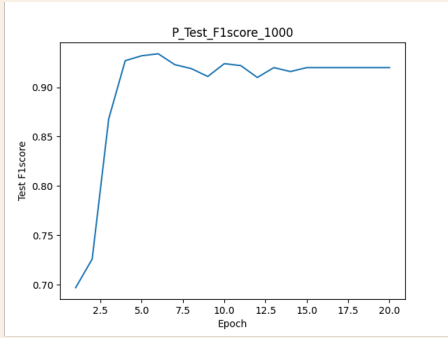


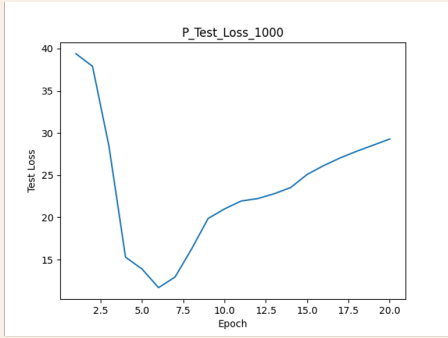


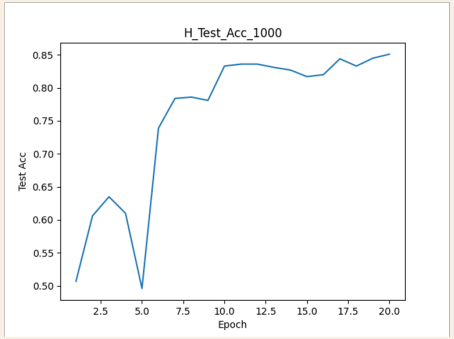
Appendix

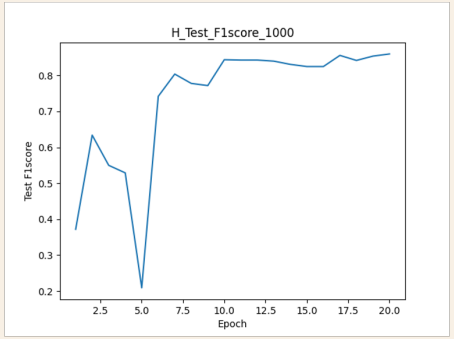
Experiment Result

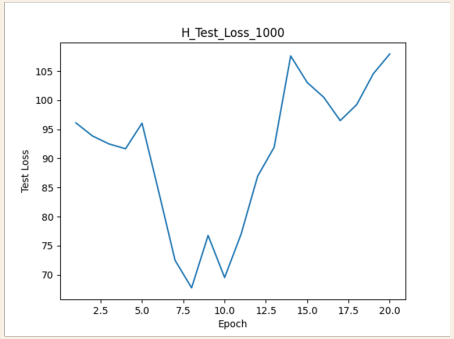






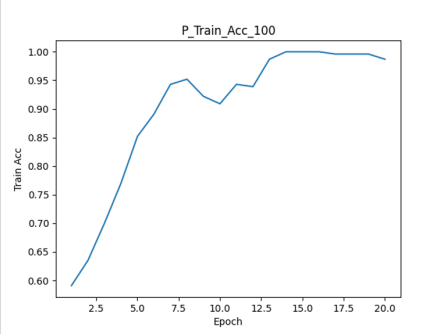


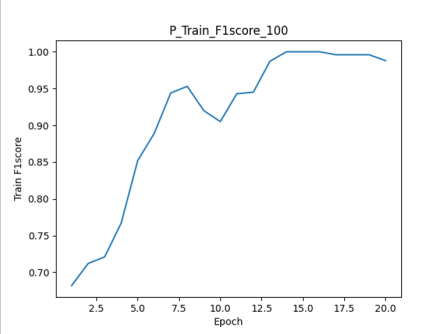


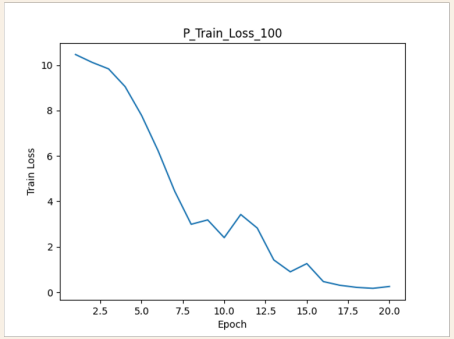


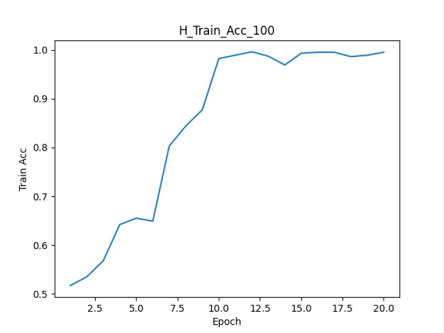
Appendix

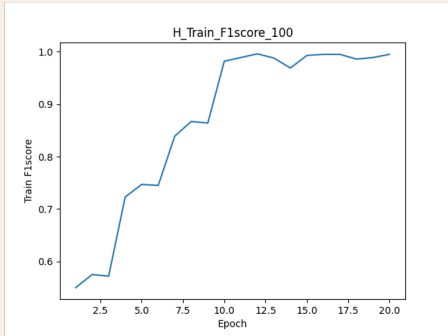
Experiment Result

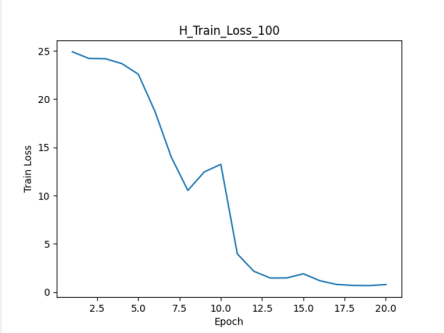






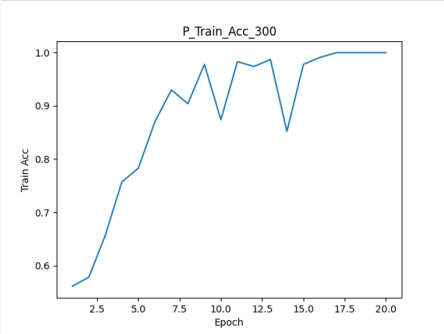


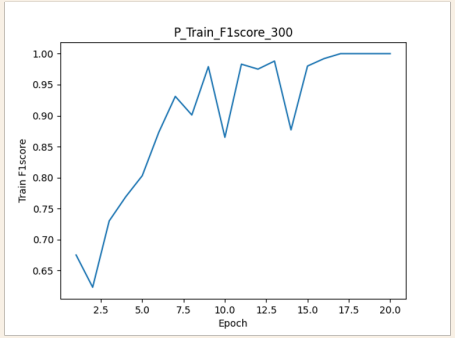


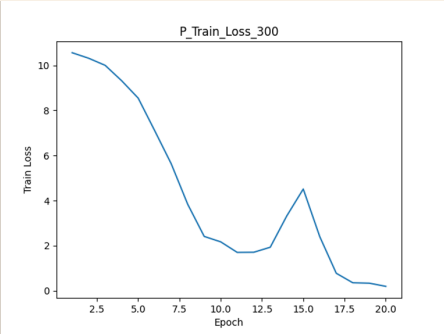


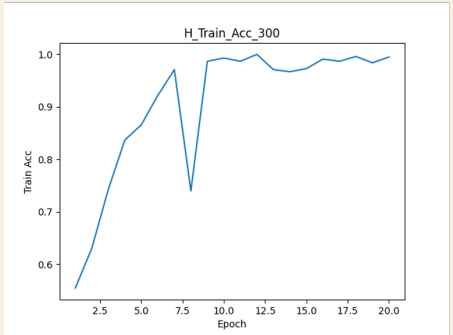
Appendix

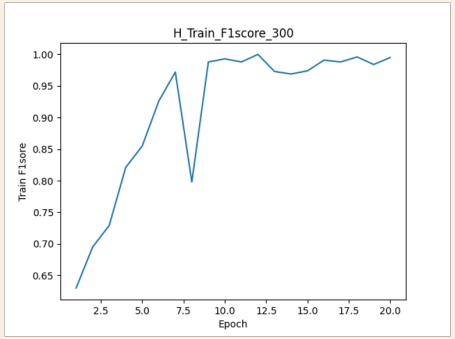
Experiment Result

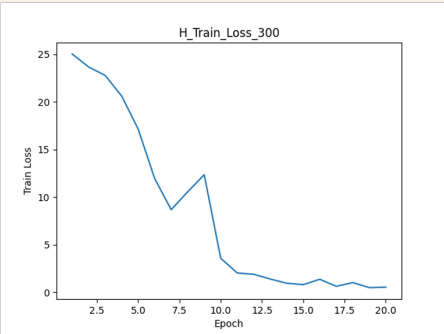






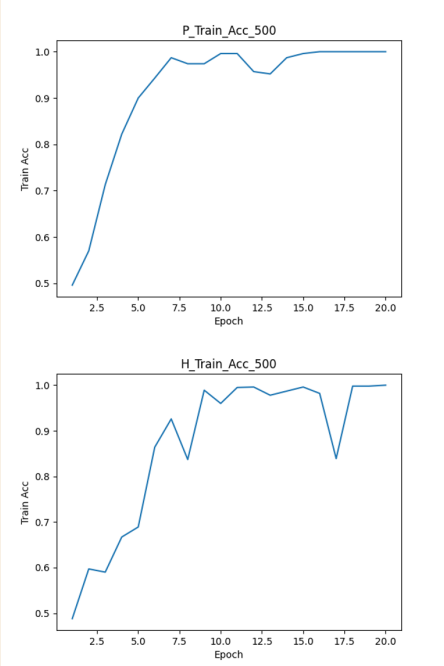


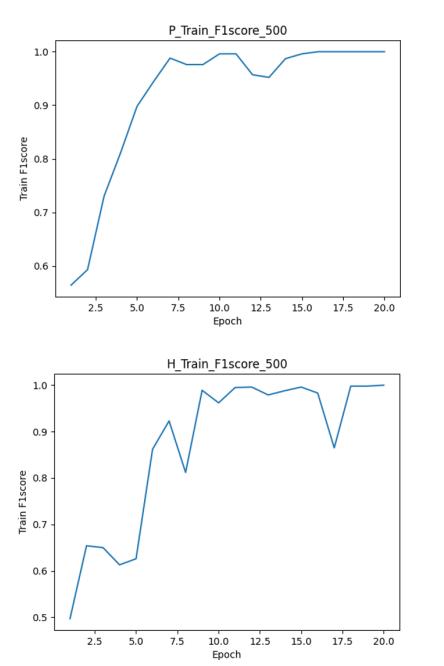


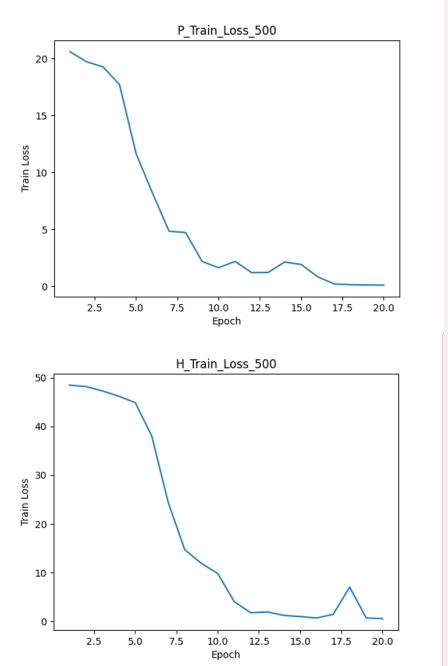


Appendix

Experiment Result







Appendix

Experiment Result