arXiv paper classification

BERT & SciBERT

Anshul Verma

30^{th} May 2021

Motivation

- A starting point to implement NLP pipelines which can be resused for other projects later.
- To implement BERT, Sci-BERT and RoBERTa for different NLP experiments on the Newspaper article data.
- To start using Compute Canada and other resources, so that training on new data becomes easier.

[1] Iz Beltagy and Kyle Lo and Arman Cohan, SCIBERT: A Pretrained Language Model for Scientific Text.

Build-up

- Built a scraper to scrape records from arXiv for different categories of papers.
- With aim to be able to classify these papers based on titles and abstracts.
- Overall I could scrape 40,000 records from arXiv with a variety of topics and attempted a few modelling approaches

What we want :

- A model which rapidly generalizes to new tasks with only a few samples. To solve the problem of transferring model across different eras.
- Enable models to perform under practical scenarios where data annotations is infeasible or new classes are dynamically included with time. (Possibility to add new categories)

Basic Analysis

Its evidently visible that we have a lot of classes with very few samples. So I decided to merge these classes.

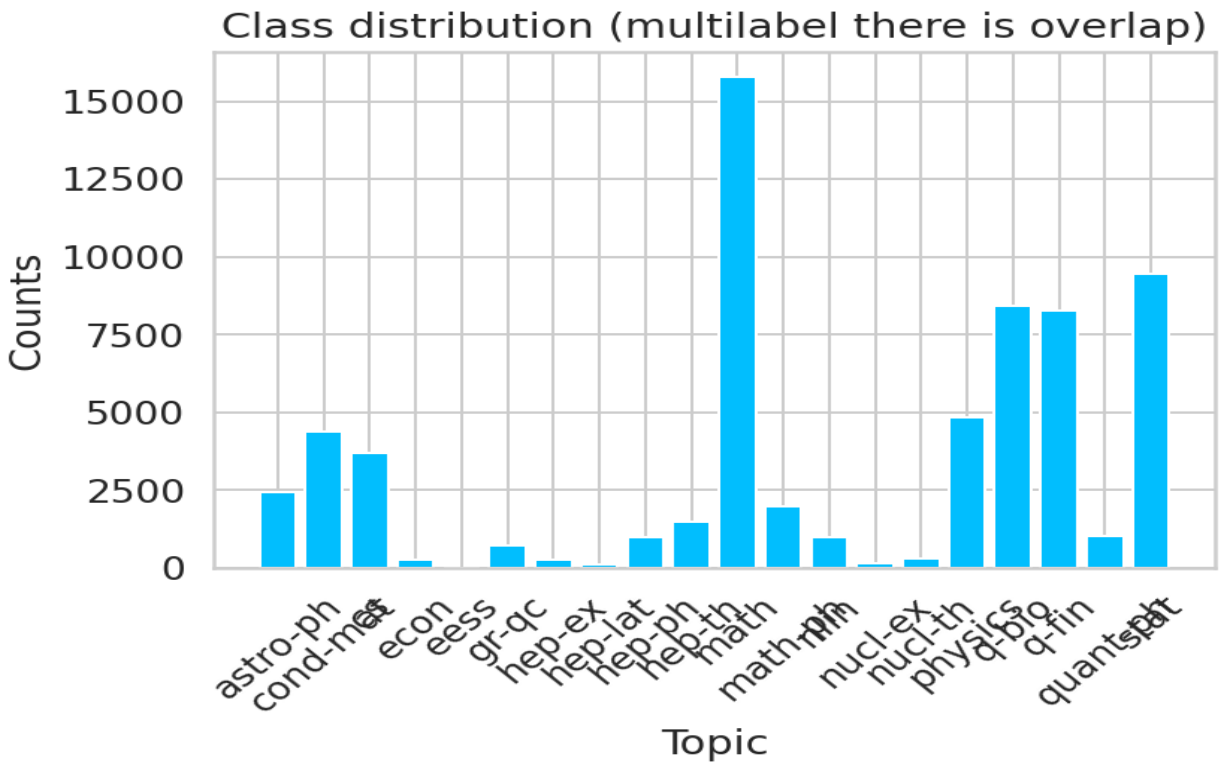


Figure: Class distribution.

Merged Class Distribution

merged *'astro-ph'*, *'cond-mat'*, *'gr-qc'*, *'hep-ex'*, *'hep-lat'*, *'hep-ph'*, *'hep-th'*, *'nucl-ex'*, *'nucl-th'*, *'quant-ph'* to **physics** and merged *'cs'*, *'nlin'* to **maths**

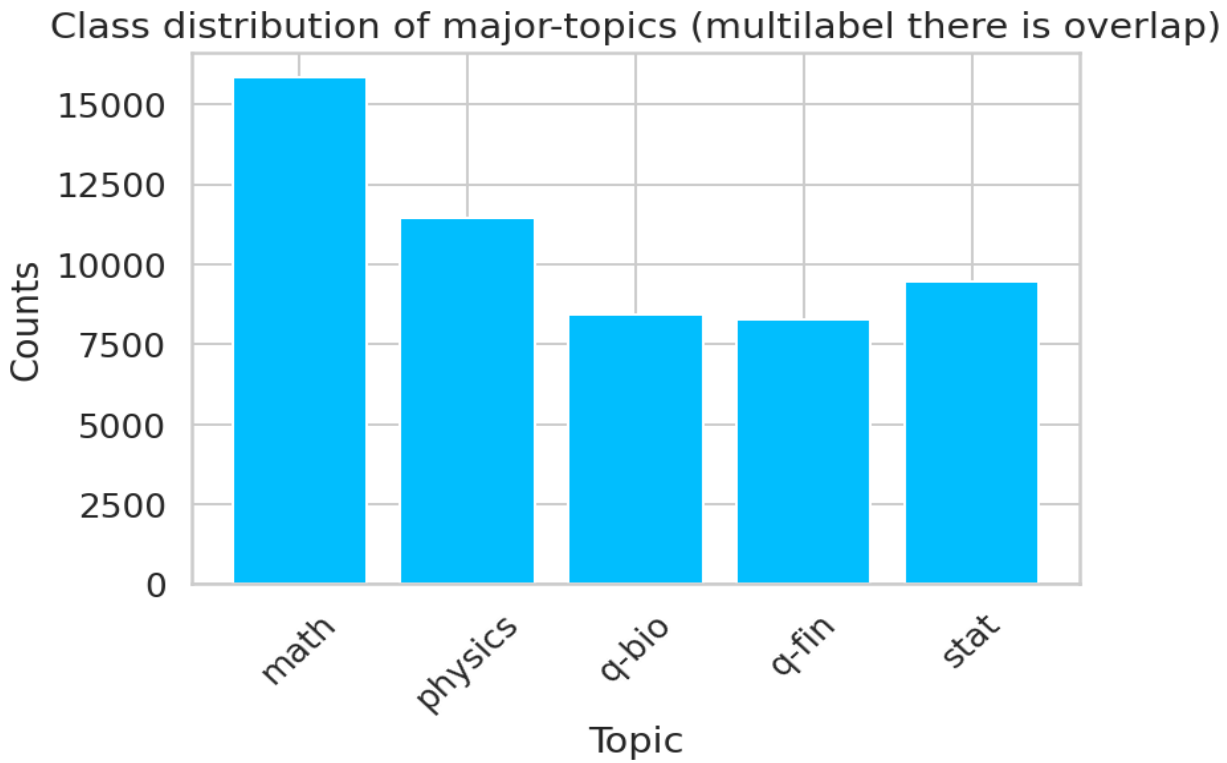


Figure: Merged Class distribution.

Multi-Class Problem

The problem in hand is a multi-class problem where one paper can have more than one classes. A 2-class overlap is shown below just to give the context, but a paper can have more than two classes.

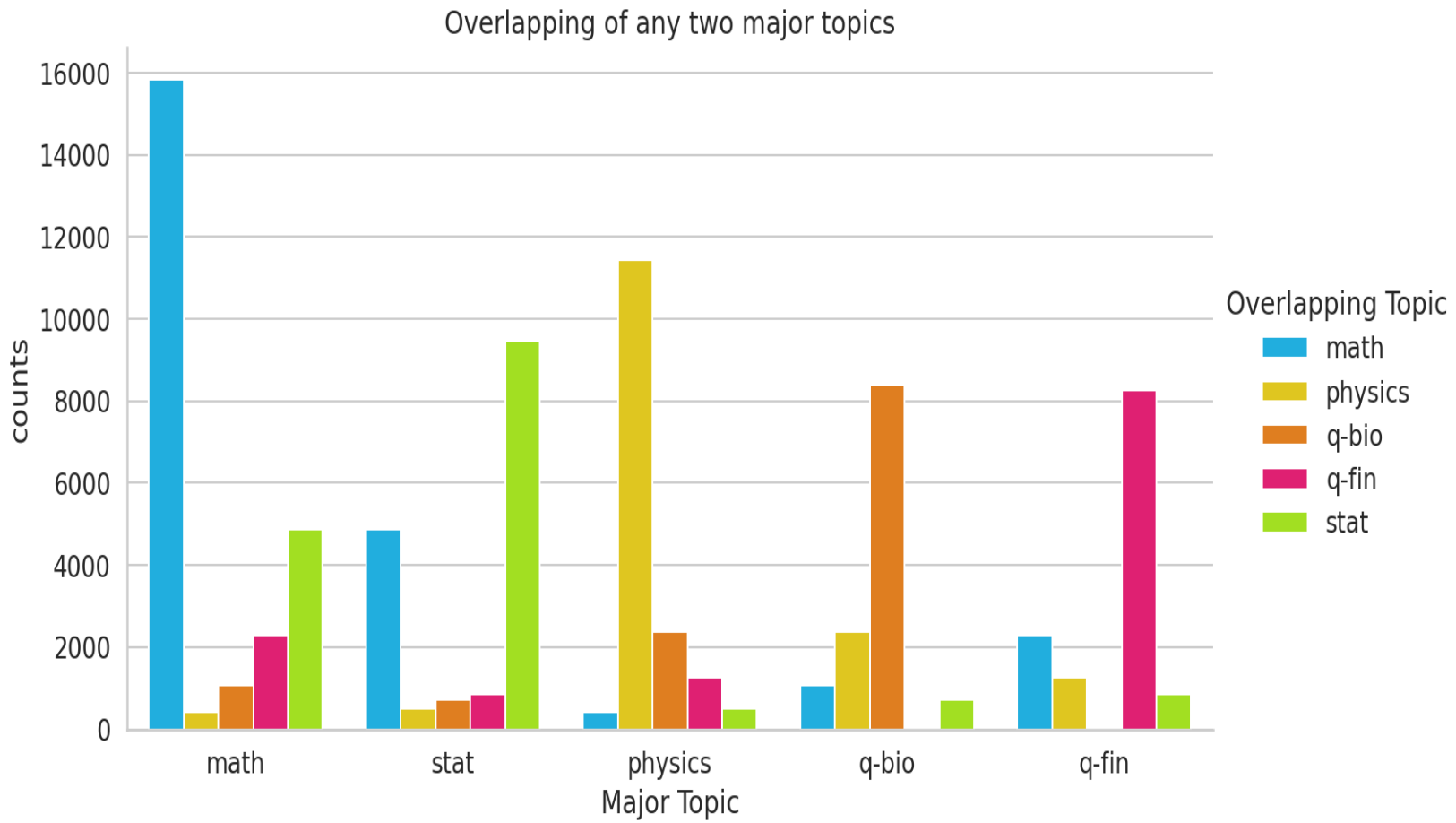


Figure: Class Overlap.

Experiment - 01

Classifying papers using **Title** only.

Experiment 01-01

For the first experiment I fine-tuned a BERT model on the title of all the papers in the dataset without training the tokenizer and using the trained BERT-tokenizer.

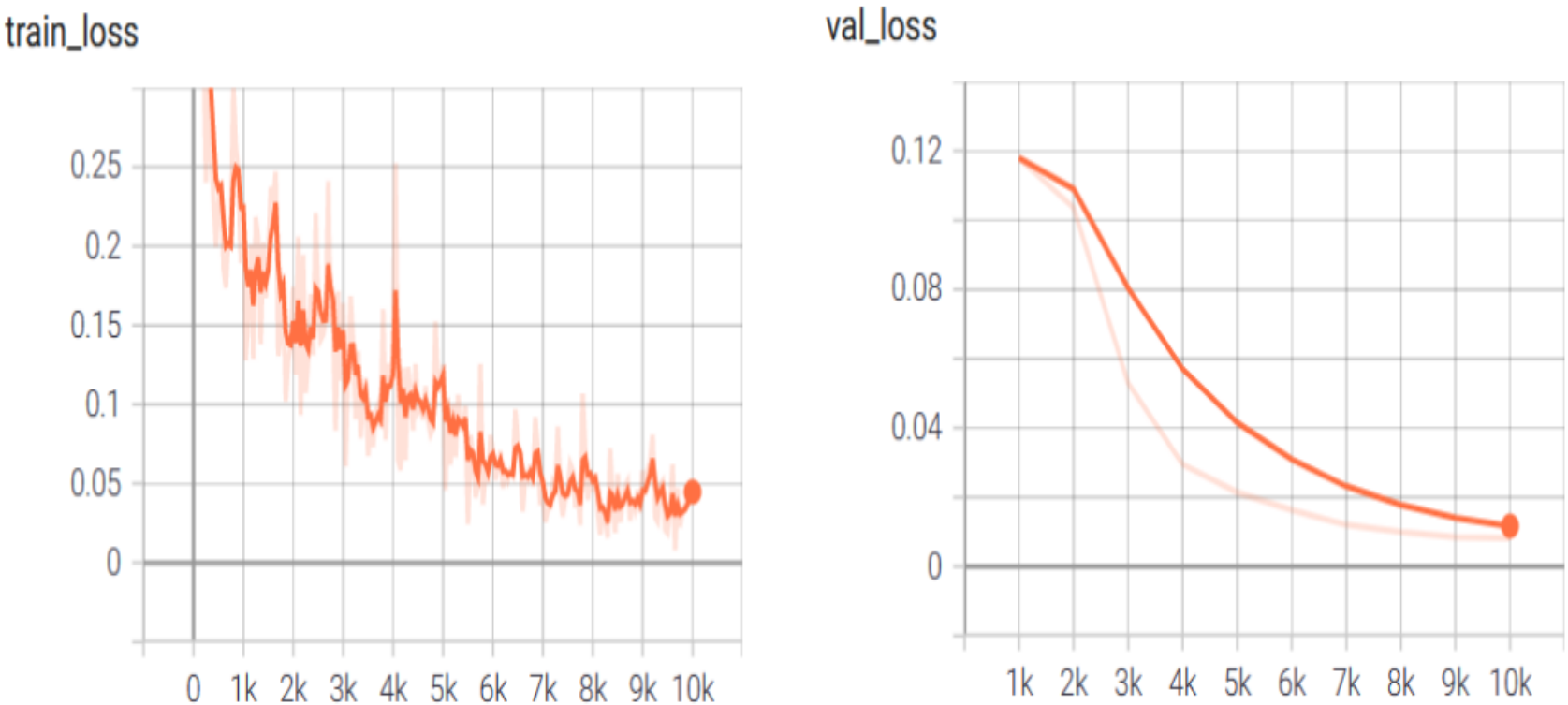


Figure: Loss over different epochs.

Experiment 01-01; Model Performance

Below is the un-normalized CM of all the classes in the problem, the final **F1-score** of the model is **0.996** and the **exact match accuracy** of the model is **99.839%**.

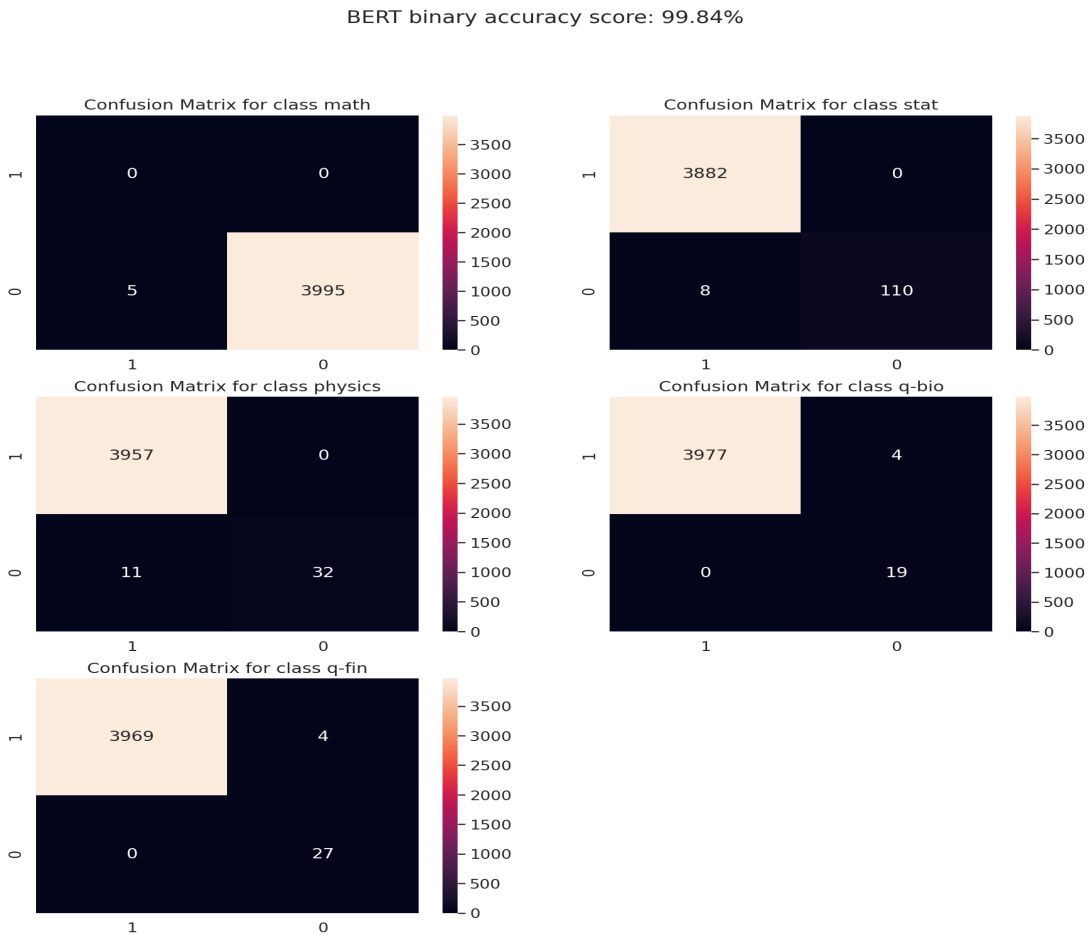
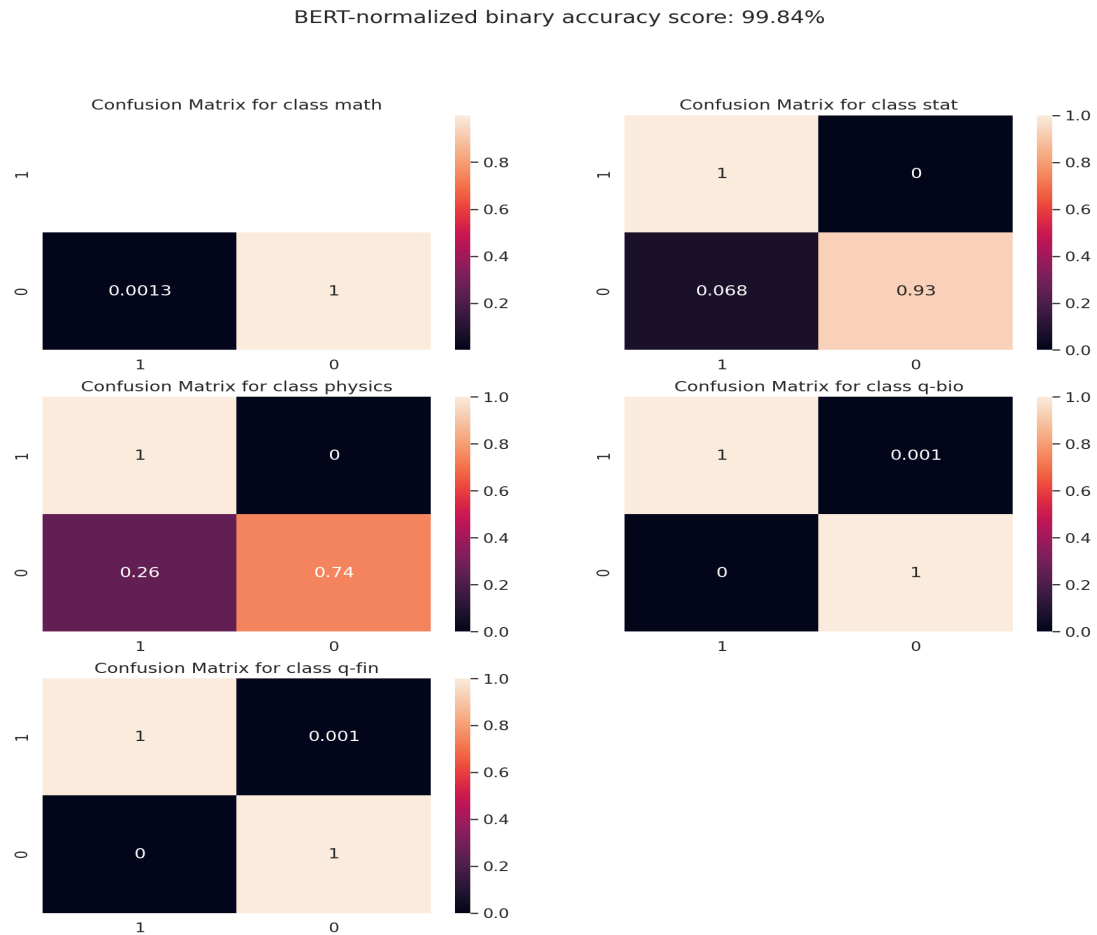


Figure: Un-normalized confusion matrix.

Experiment 01-01; Model Performace

Below is the normalized CM of all the classes in the problem, this is useful to see the accuracy of the model per class. Per class F1-score of the model is { **math**: **0.999**, **stat**: **0.965**, **physics**: **0.853**, **q-bio**: **0.905**, **q-fin**: **0.931** }.



Experiment 02-01

For the first experiment I fine-tuned a SciBERT model on the title of all the papers in the dataset without training the tokenizer and using the trained SciBERT-tokenizer. This model is expected to be slightly better than the BERT model because the dataset on which the tokenizer was trained for this model is more relevant for the problem in hand.

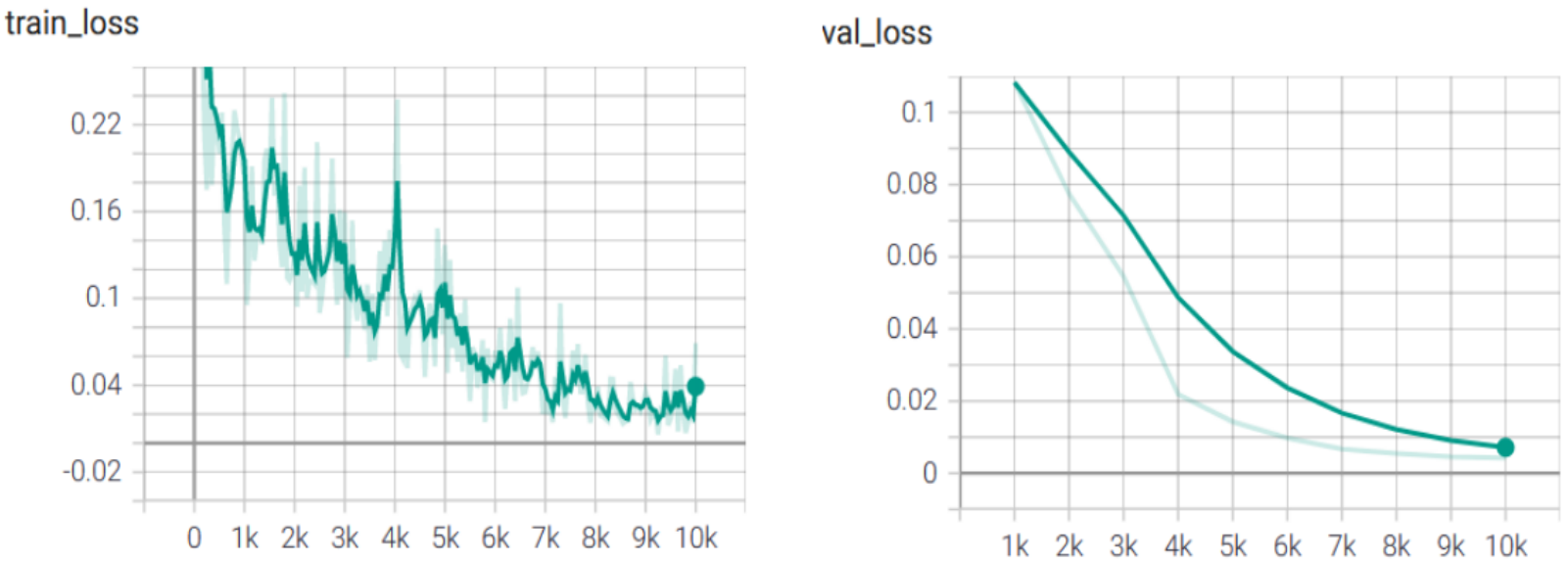


Figure: Loss over different epochs.

Experiment 02-01; Model Performance

Below is the un-normalized CM of all the classes in the problem, the final **F1-score** of the model is **0.991** and the **exact match accuracy** of the model is **99.94%**.

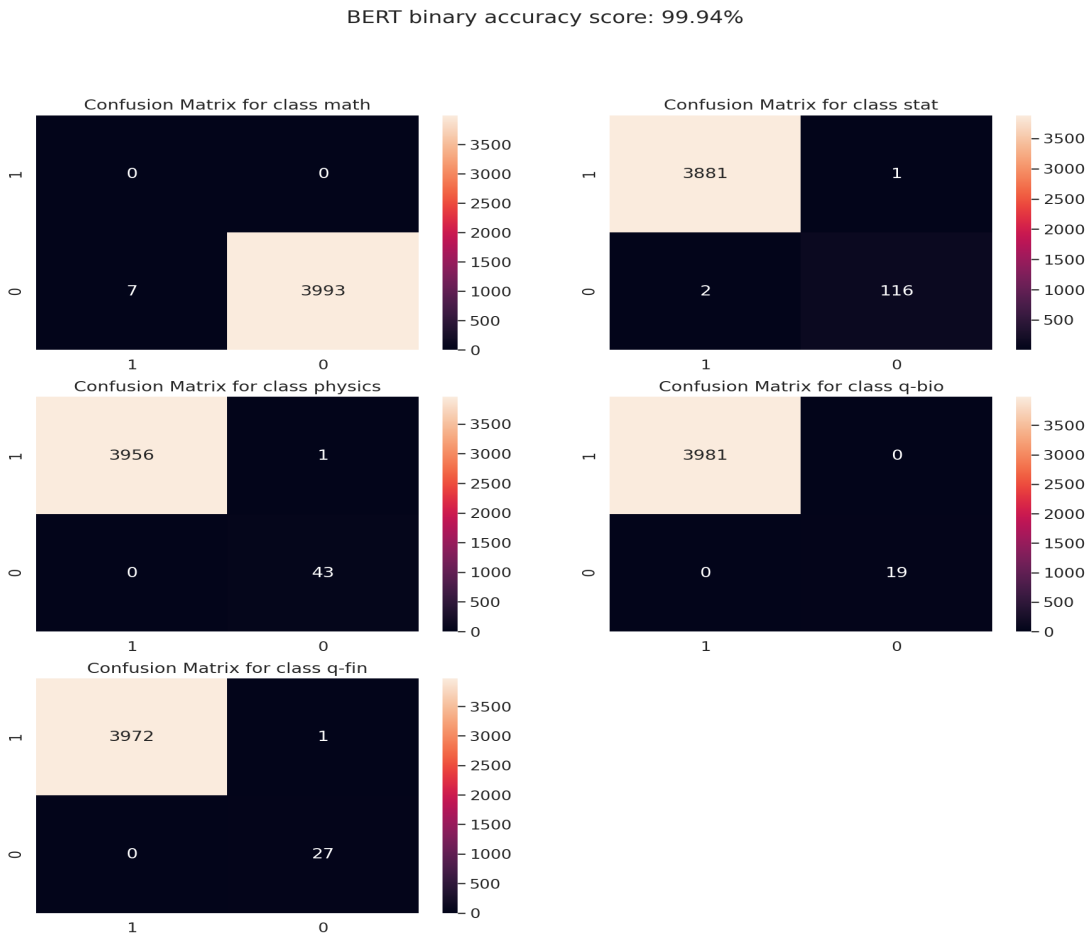


Figure: Un-normalized confusion matrix.

Experiment 02-01; Model Performace

Below is the normalized CM of all the classes in the problem, this is useful to see the accuracy of the model per class. Per class F1-score of the model is { **math**: **0.999**, **stat**: **0.987**, **physics**: **0.988**, **q-bio**: **1.0**, **q-fin**: **0.982** }