Report

1 Task 2

In this task, three pretrained models `roberta-base` , `bert-base-uncased` and `allenai/scibert_scivocab_uncased` are fine-tuned on `restaurant_sup` , `acl_sup` and `agnews_sup` . Each model-dataset pair is trained for 5 times to ensure reliability.

The batch size, epoch number and other configurations are adjusted for each model-dataset pair to make the model converge stably.

The script for this task is `train.py` in the repository.

1.1 roberta-base

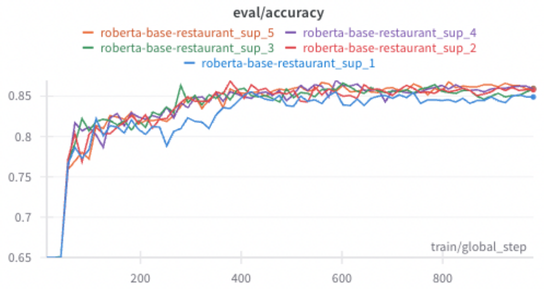
1.1.1 restaurant_sup

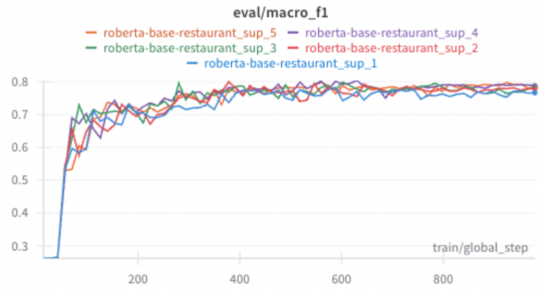
The configuration is as follows:

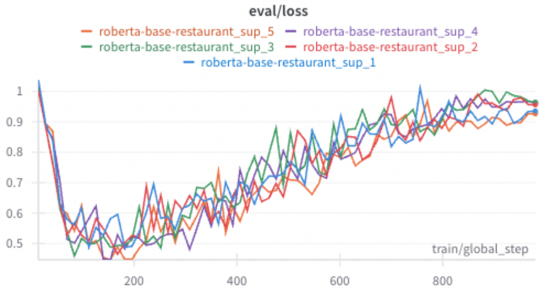
Dropout Rate	0.3

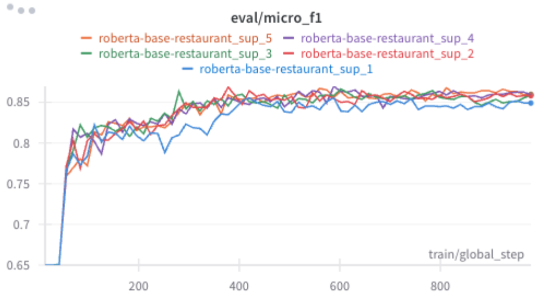
Epochs	70
Batch Size	256
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

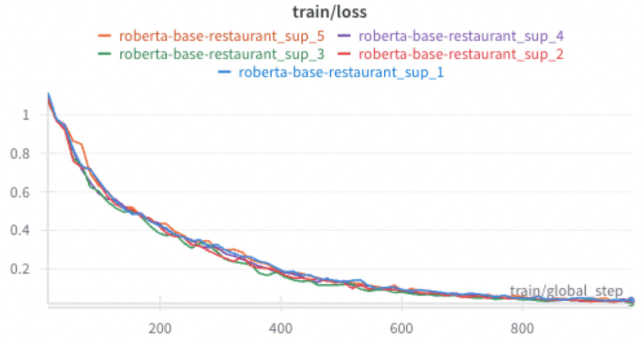
The following results are obtained:











	Mean	Standard Deviation
eval/loss	0.94924	0.01518

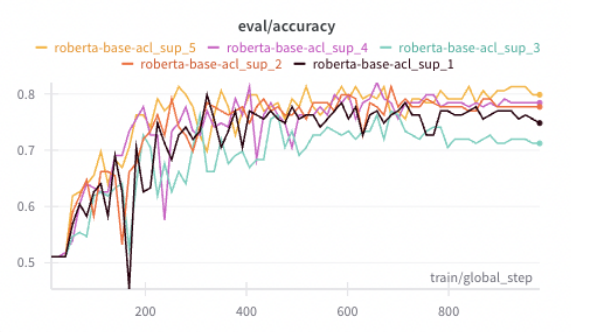
	Mean	Standard Deviation
eval/accuracy	0.85696	0.00397
eval/macro_f1	0.77995	0.00725
eval/micro_f1	0.85696	0.00397
train/loss	0.03342	0.00666

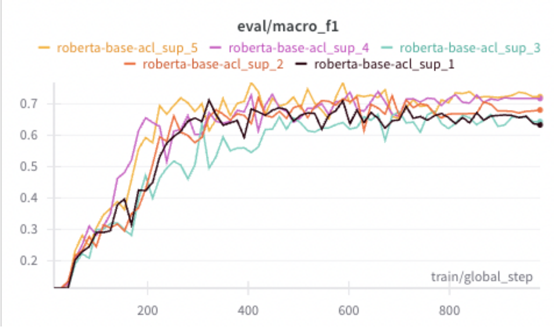
1.1.2 acl_sup

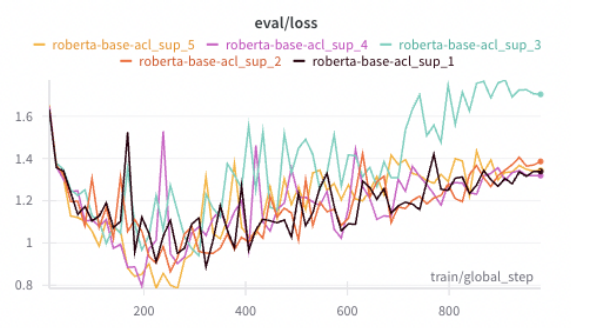
The configuration is as follows:

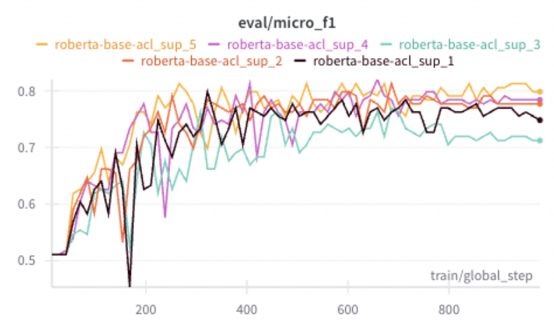
Dropout Rate	0.3
Epochs	70
Batch Size	128
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

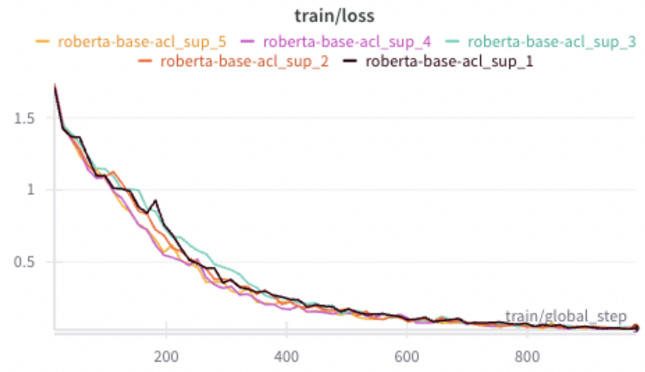
The following results are obtained:











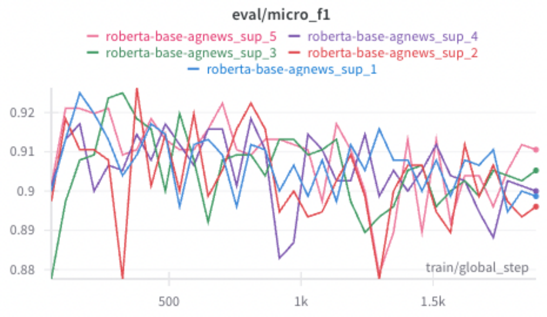
	Mean	Standard Deviation
eval/loss	1.41823	0.14449
eval/accuracy	0.76403	0.02861
eval/macro_f1	0.67944	0.03674
eval/micro_f1	0.76403	0.02861
train/loss	0.04146	0.00528

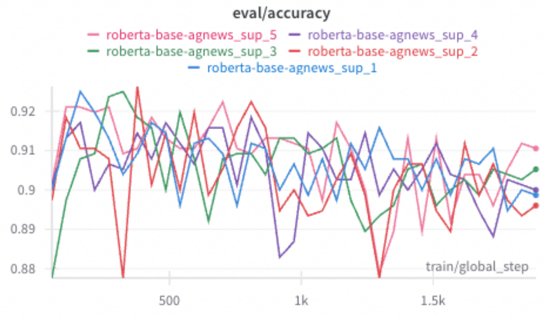
1.1.3 agnews_sup

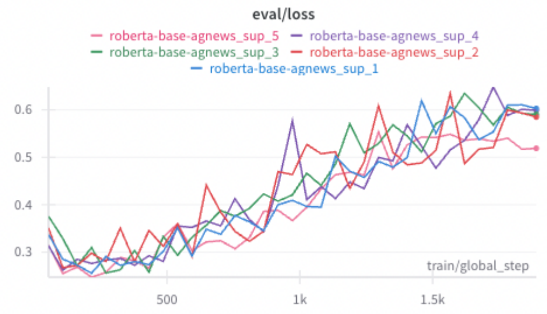
The configuration is as follows:

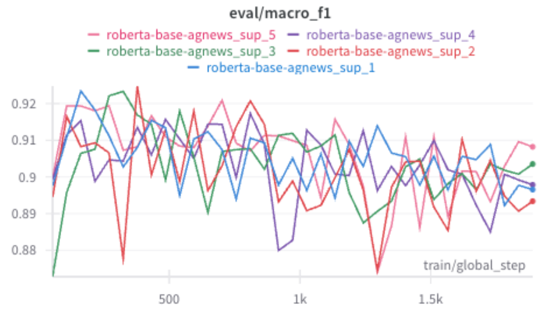
Dropout Rate	0.3
Epochs	35
Batch Size	128
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

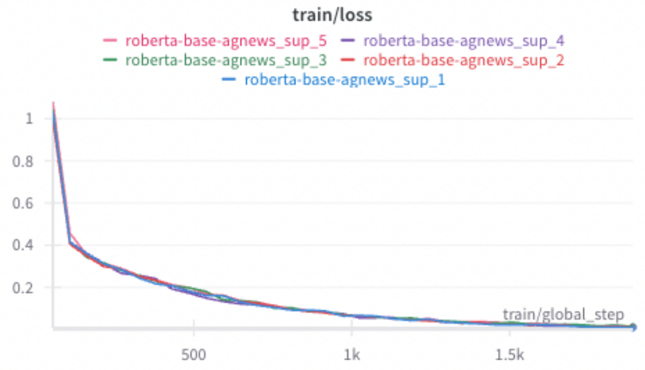
The following results are obtained:











	Mean	Standard Deviation
eval/loss	0.57913	0.03075
eval/accuracy	0.90211	0.00517
eval/macro_f1	0.89990	0.00529
eval/micro_f1	0.90211	0.00517
train/loss	0.01064	0.00218

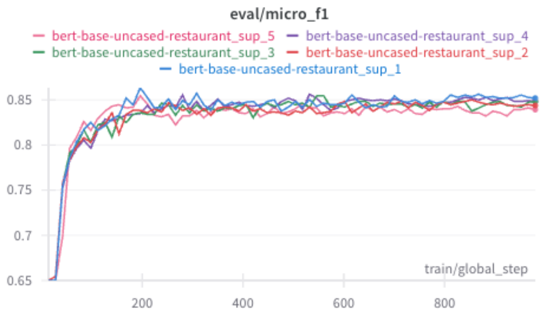
1.2 bert-base-uncased

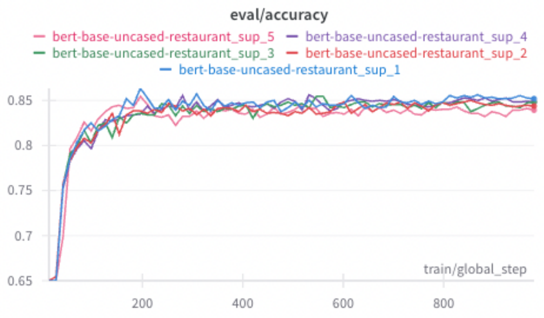
1.2.1 restaurant_sup

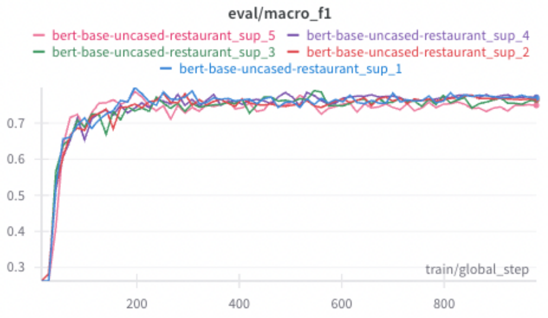
The configuration is as follows:

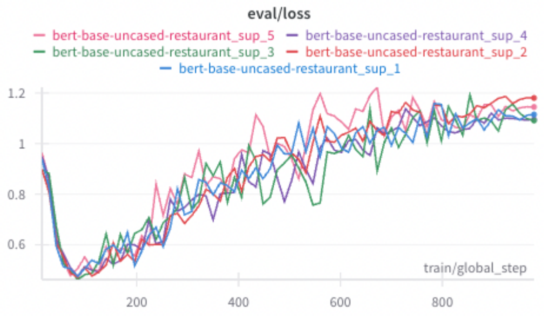
Dropout Rate	0.3
Epochs	70
Batch Size	256
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

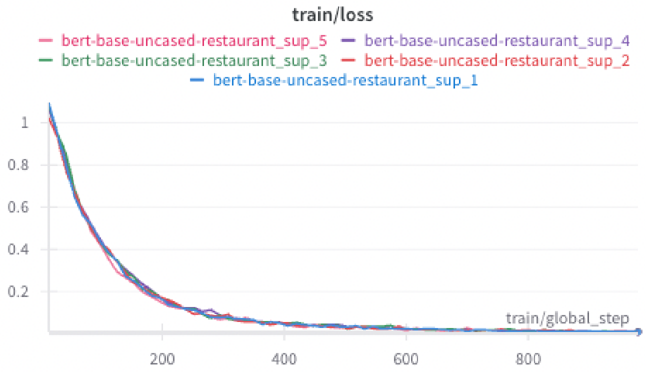
The following results are obtained:











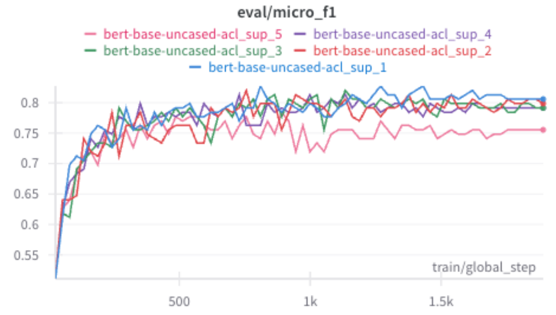
	Mean	Standard Deviation
eval/loss	1.12554	0.03338
eval/accuracy	0.84625	0.00432
eval/macro_f1	0.76383	0.00776
eval/micro_f1	0.84625	0.00432
train/loss	0.00844	0.00063

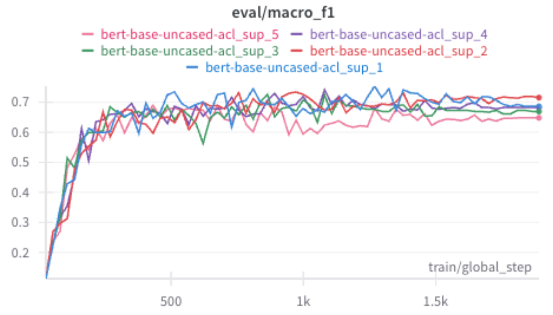
1.2.2 acl_sup

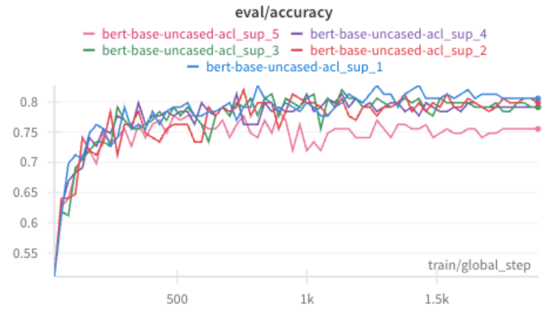
The configuration is as follows:

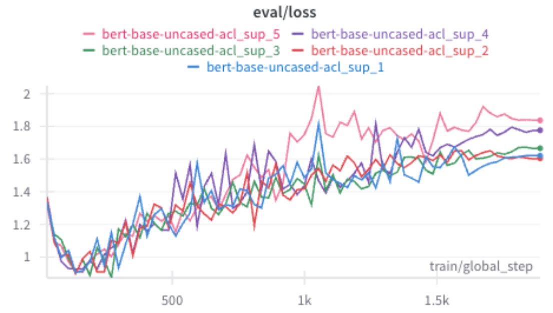
Dropout Rate	0.3
Epochs	70
Batch Size	64
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

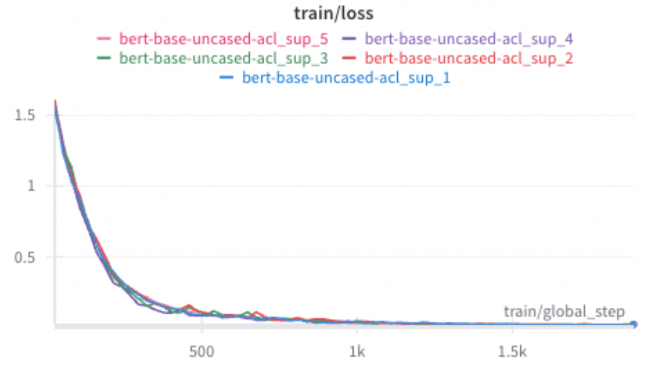
The following results are obtained:











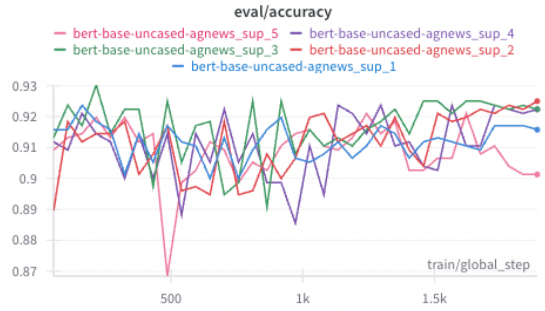
	Mean	Standard Deviation
eval/loss	1.70068	0.09082
eval/accuracy	0.78849	0.01739
eval/macro_f1	0.68031	0.02205
eval/micro_f1	0.78849	0.01739
train/loss	0.02206	0.00116

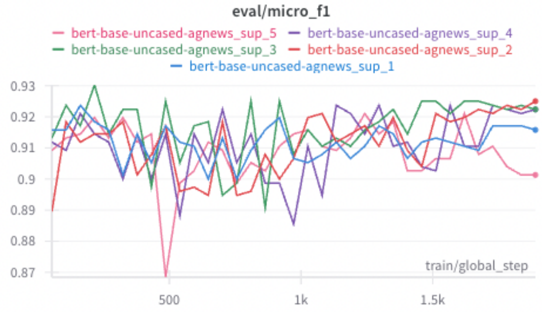
1.2.3 agnews_sup

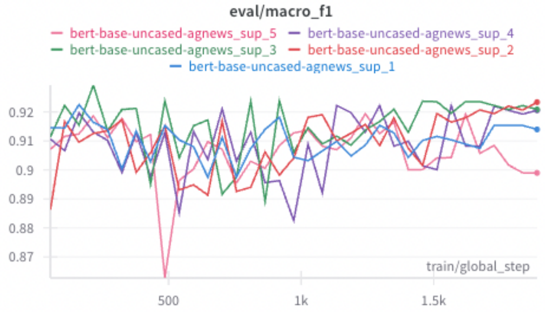
The configuration is as follows:

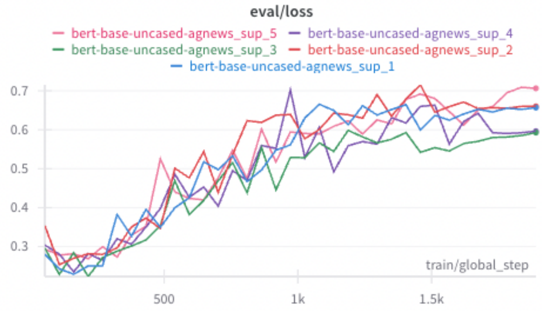
Dropout Rate	0.3
Epochs	35
Batch Size	128
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

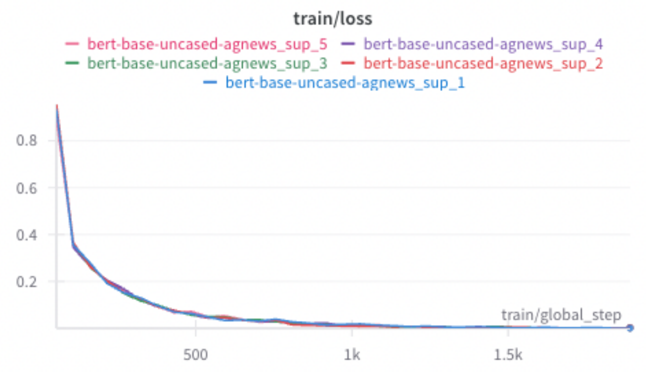
The following results are obtained:











	Mean	Standard Deviation
eval/loss	0.64246	0.04291
eval/accuracy	0.91737	0.00858
eval/macro_f1	0.91556	0.00885
eval/micro_f1	0.91737	0.00858
train/loss	0.00148	0.00082

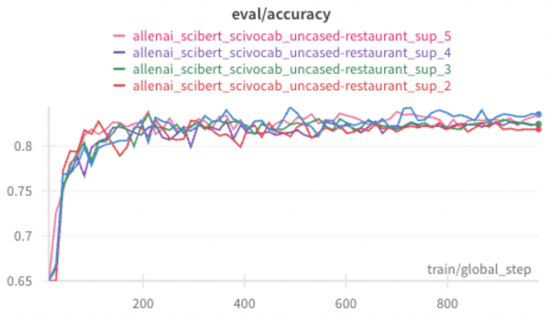
1.3 allenai/scibert_scivocab_uncased

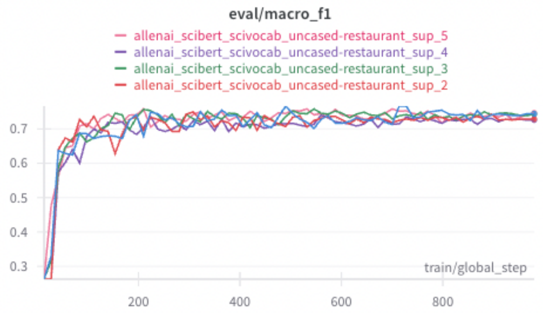
1.3.1 restaurant_sup

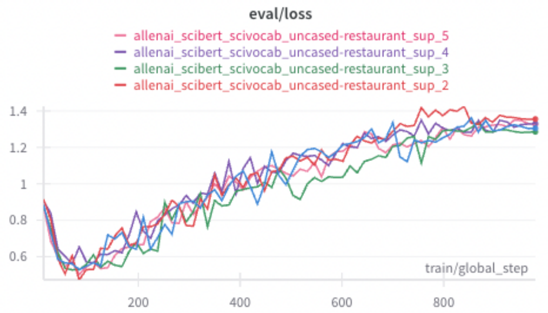
The configuration is as follows:

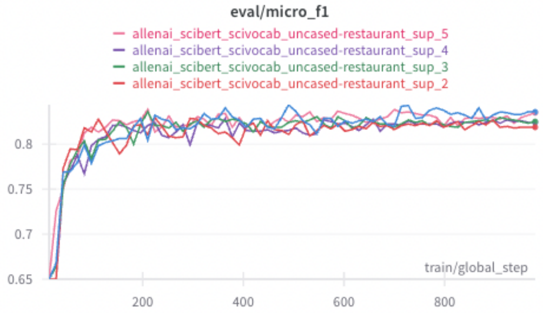
Dropout Rate	0.3
Epochs	70
Batch Size	256
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

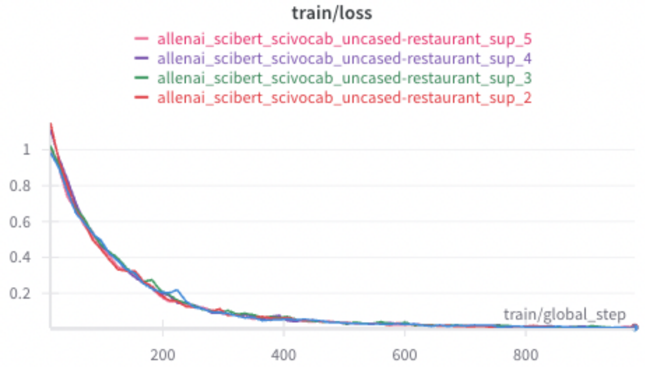
The following results are obtained:











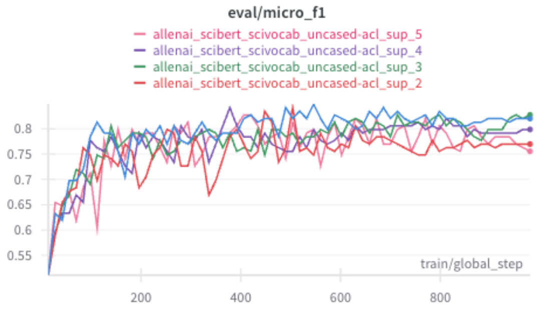
	Mean	Standard Deviation
eval/loss	1.32040	0.02376
eval/accuracy	0.82768	0.00656
eval/macro_f1	0.73767	0.00851
eval/micro_f1	0.82768	0.00656
train/loss	0.00956	0.00164

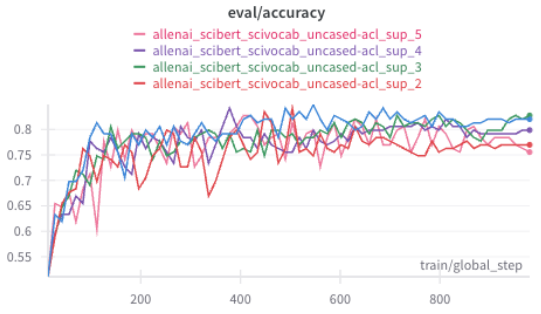
1.3.2 acl_sup

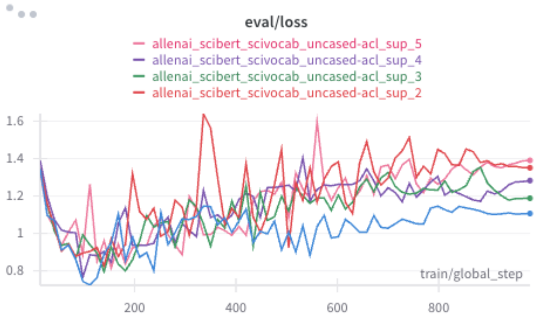
The configuration is as follows:

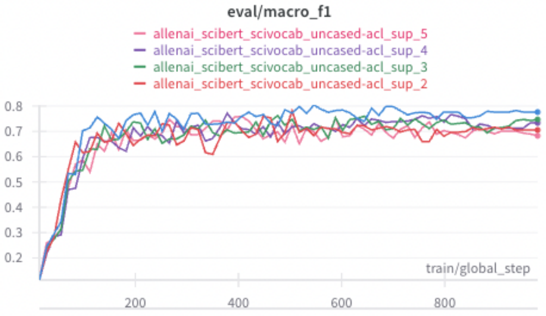
Dropout Rate	0.3
Epochs	70
Batch Size	128
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

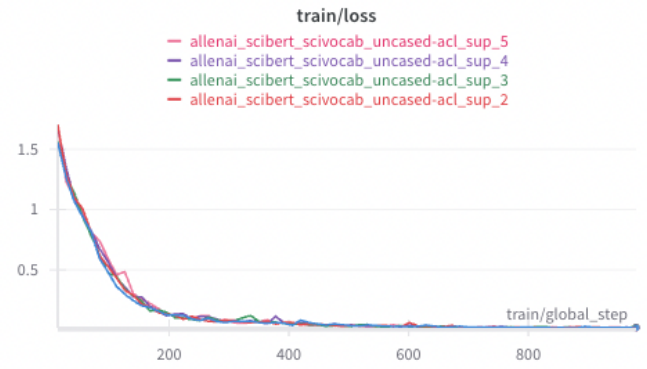
The following results are obtained:











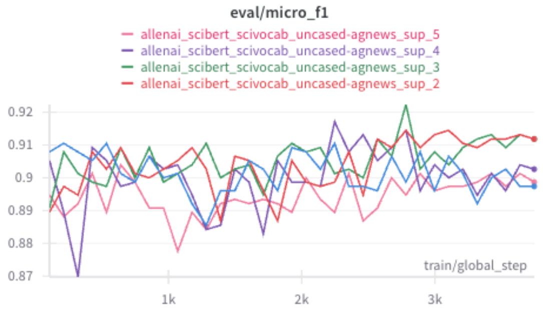
	Mean	Standard Deviation
eval/loss	1.26377	0.10430
eval/accuracy	0.79424	0.02790
eval/macro_f1	0.72893	0.03251
eval/micro_f1	0.79424	0.02790
train/loss	0.02068	0.00189

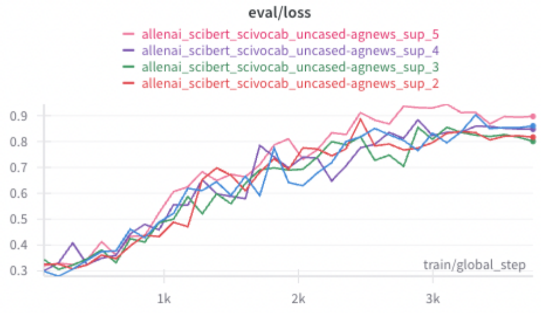
1.3.3 agnews_sup

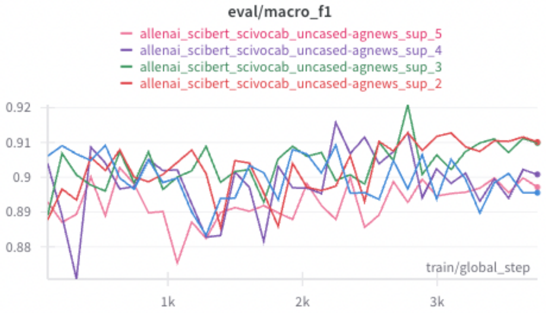
The configuration is as follows:

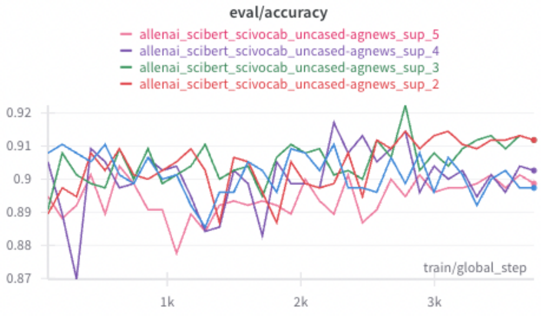
Dropout Rate	0.3
Epochs	35
Batch Size	64
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

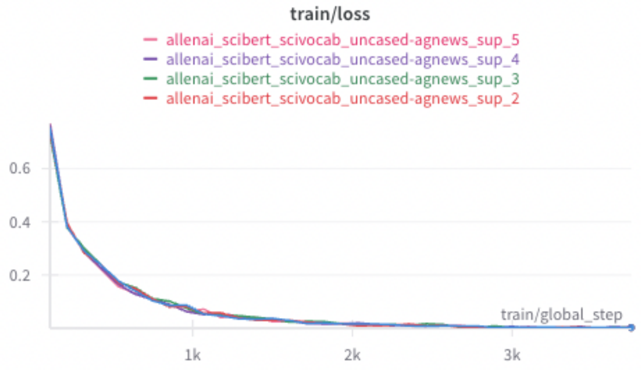
The following results are obtained:











	Mean	Standard Deviation
eval/loss	0.84449	0.03378
eval/accuracy	0.90447	0.00626
eval/macro_f1	0.90271	0.00620
eval/micro_f1	0.90447	0.00626
train/loss	0.00180	0.00045

1.4 Analysis

The `roberta-base` model achieves the highest accuracy (0.85696) and macro F1-score (0.77995) among all models on `restaurant_sup`, which makes sense for the reason that it is an improved version of BERT, and SciBERT is pretrained specifically for scientific text, making it possibly unfamiliar with comments on restaurants.

The `allenai/scibert_scivocab_uncased` model achieves the highest accuracy (0.79424) and macro F1-score (0.72893) on the `acl_sup` dataset, which is expected given SciBERT's specialization in scientific text. Since `acl_sup` consists of scientific texts, this result is particularly fitting.

Lastly, the `bert-base-uncased` model achieves the highest accuracy (0.91737) and macro F1-score (0.91556) on the `agnews_sup` dataset. This result is somewhat unexpected, considering BERT's pretraining data primarily consists of BooksCorpus and English Wikipedia, without specific news datasets. In contrast, RoBERTa's pretraining included exposure to news-related content, which might have suggested better performance on this task. It is possible that the configuration used wasn’t optimal for RoBERTa’s finetuning.

From the dataset perspective, `agnews_sup` yielded the highest finetuned model accuracy and macro F1-score, followed by `restaurant_sup`, and then `acl_sup`. This ranking suggests the relative difficulty of the task corresponding to the dataset.

2 Task 3

In this task, the pretrained `roberta-base` model is finetuned using PEFT. Adapters are inserted into the pretrained model, and the adapters are tuned instead of the entire model.

The corresponding training script is `train_adapter.py`.

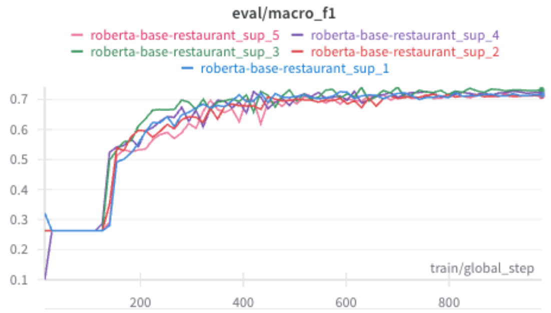
2.1 restaurant_sup

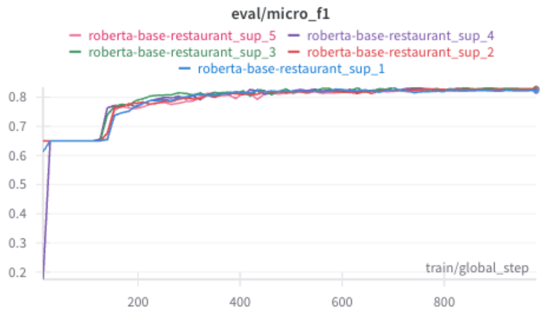
The configuration is as follows:

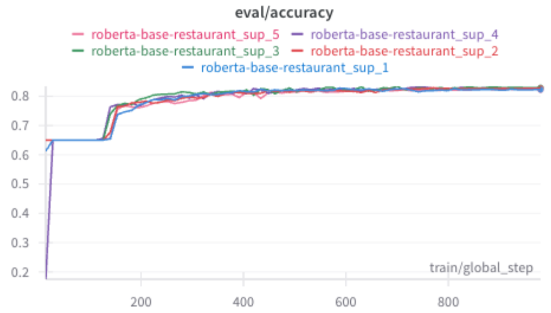
Dropout Rate	0.3
Epochs	70
Batch Size	256
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01

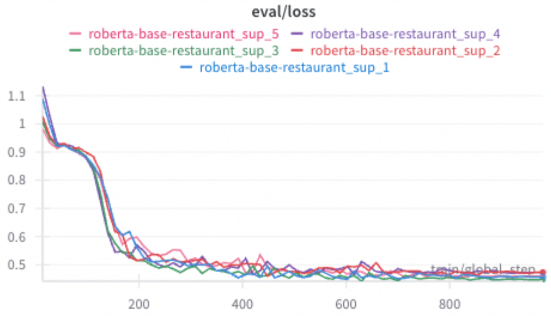
Warmup Steps	70

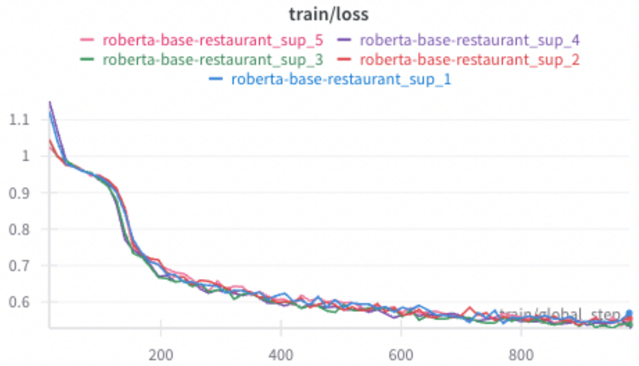
The following results are obtained:











	Mean	Standard Deviation
eval/loss	0.46201	0.00973
eval/accuracy	0.82571	0.00326
eval/macro_f1	0.71861	0.00751
eval/micro_f1	0.82571	0.00326

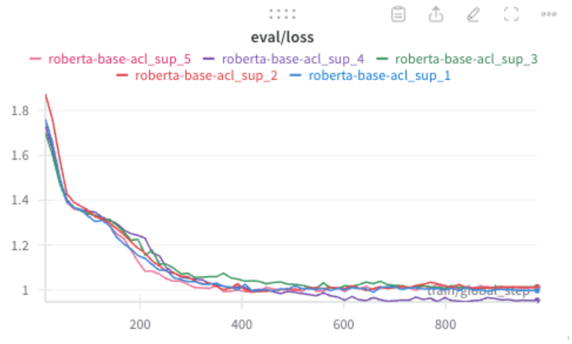
	Mean	Standard Deviation
train/loss	0.55012	0.01218

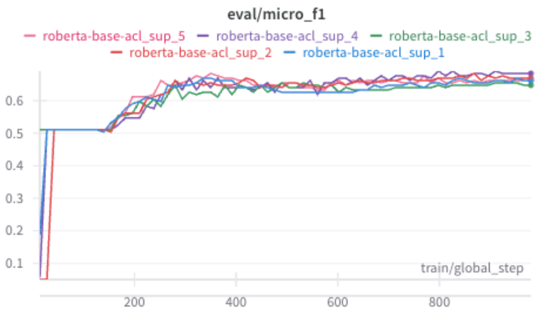
2.2 acl_sup

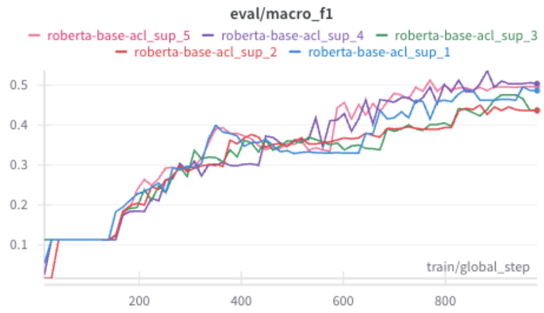
The configuration is as follows:

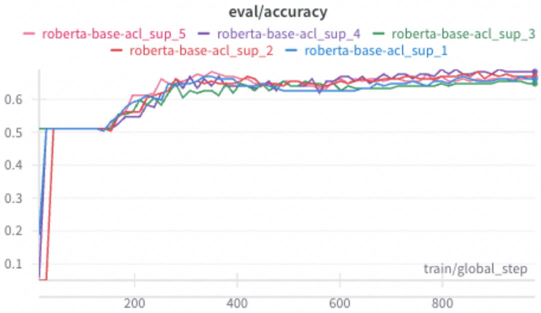
Dropout Rate	0.3
Epochs	70
Batch Size	128
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

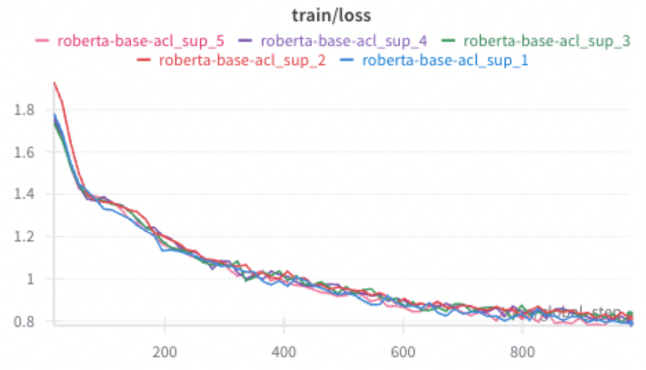
The following results are obtained:











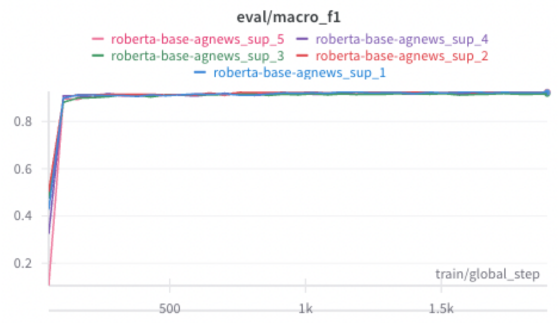
	Mean	Standard Deviation
eval/loss	0.99737	0.02260
eval/accuracy	0.66475	0.01169
eval/macro_f1	0.47183	0.02924
eval/micro_f1	0.66475	0.01169
train/loss	0.80362	0.01701

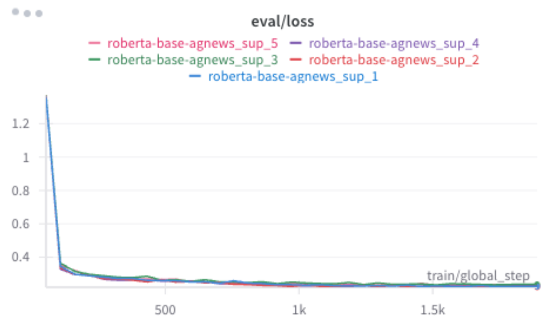
2.3 agnews_sup

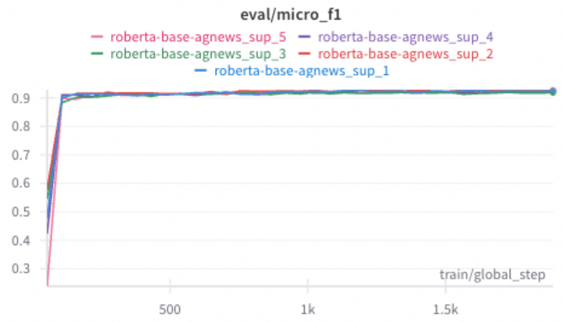
The configuration is as follows:

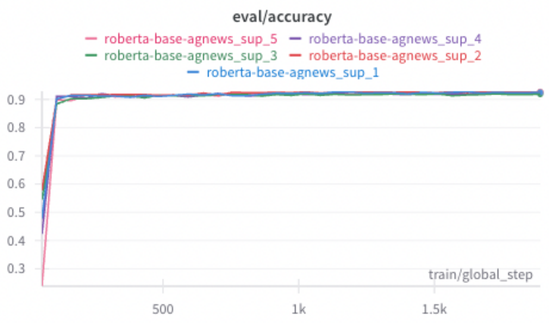
Dropout Rate	0.3
Epochs	35
Batch Size	128
Learning Rate	9e-5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	70

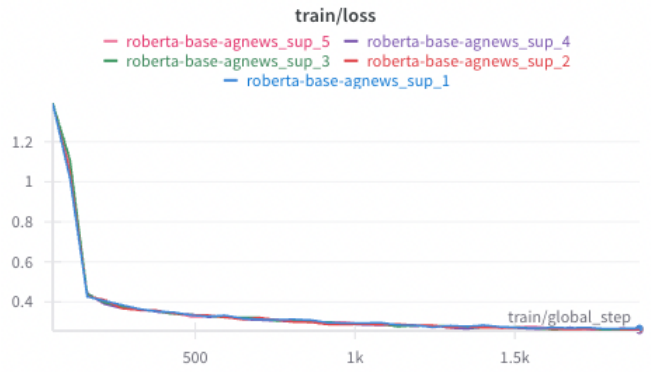
The following results are obtained:











	Mean	Standard Deviation
eval/loss	0.22841	0.00589
eval/accuracy	0.92368	0.00343
eval/macro_f1	0.92218	0.00346
eval/micro_f1	0.92368	0.00343
train/loss	0.26374	0.00443

2.4 Analysis

1. If you directly fine-tune a 3B model without PEFT, how much GPU memory do you need?

   Assuming that during finetuning, the model uses FP32 precision and uses Adam as its optimizer,

   a. The model itself takes up 3e9 ∗ 4B = 11.176 GB.

   b. The stored gradient invoked by loss.backward() takes up about the same memory.

   c. Adam keeps track of the mean and variance of gradients, taking up 2x model size.

   In total, it takes up about 3e9 ∗ 4B * 4 = 44.703 GB to fine-tune a 3B model without PEFT.

2. With PEFT, how much GPU memory is saved?

   With PEFT,

   a. The model itself is still needed, and additionally we have to take the adapter parameters into account. In the adapter paper, is it said that

      > Training adapters with sizes 0.5 − 5% of the original model, performance is within 1% of the competitive published results on BERT-large.

      so let's just assume the adapter size is 5% of the original model.

   b. Only the gradients of the adapters are stored, for that the pretrained parameters are frozen.

   c. Adam keeps track of the mean and variance of gradients of the adapters, taking up 2x adapter size.

   In total, it takes up about 3e9 ∗ 4B * 1.05 + 3 ∗ 3e9 ∗ 4B * 0.05 = 13.411 GB, meaning that PEFT saves 31.292 GB of GPU memory.

3. Other observations

   Compared with conventional finetuning, it is suprising to see that even though PEFT on these datasets makes the model to converge to a relatively higher train loss, the eval loss converges well and shows no sign of overfitting, while the eval loss bounces back quickly and the model overfits after merely a few epochs in conventional finetuning.

Besides, even though the results on `restaurant_sup` and `acl_sup` are relatively worse than conventional finetuning, the results on `agnews_sup` is better, which proves that adapter is capable of finetuning models decently.