Discussion

Datasets used:

Task 1: https://huggingface.co/datasets/legacy-datasets/wikipedia

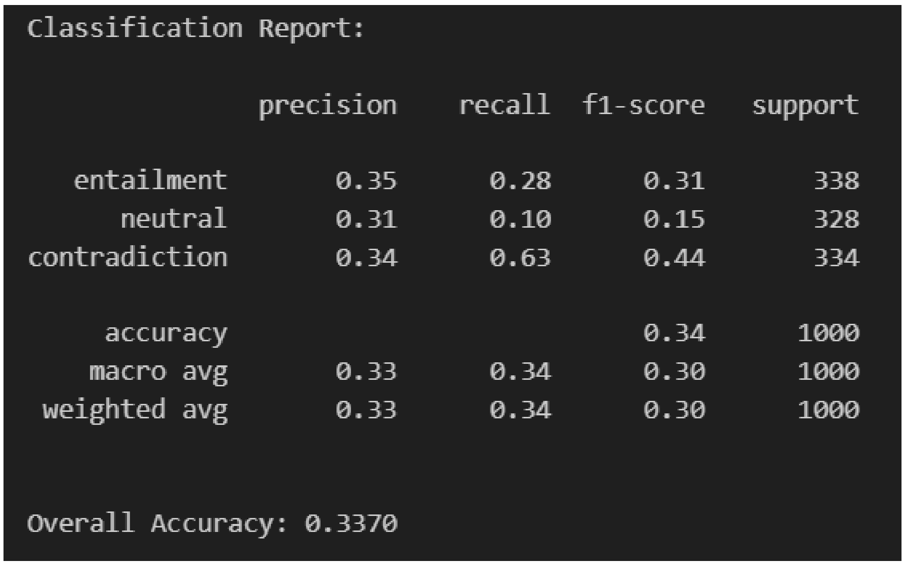
```
@ONLINE{wikidump,
    author = "Wikimedia Foundation",
    title  = "Wikimedia Downloads",
    url    = "https://dumps.wikimedia.org"
}
```

Task 2:

https://huggingface.co/datasets/glue/viewer/mnli

https://huggingface.co/datasets/nyu-mll/glue/viewer/mnli

Performance metrics (classification Report) based on the SNLI or MNLI datasets for the Natural Language Inference (NLI) task.



```
Classification Report:

               precision    recall  f1-score   support

   entailment       0.35      0.28      0.31       338
      neutral       0.31      0.10      0.15       328
contradiction       0.34      0.63      0.44       334

     accuracy                           0.34      1000
    macro avg       0.33      0.34      0.30      1000
 weighted avg       0.33      0.34      0.30      1000


Overall Accuracy: 0.3370
```

Table 1. Classification Report

The classification report shows the precision, recall, and F1-score for the three NLI classes: entailment, neutral, and contradiction. The overall accuracy achieved was approximately 0.33, which indicates that the model performs slightly above random guessing (since there are three classes).

The model performs better on the neutral class compared to entailment and contradiction. This suggests that the learned representations may not be sufficiently discriminative for fine-grained semantic distinctions required in NLI tasks.

Limitations or challenges encountered during the implementation

(a) Limited Training Data or Epochs. The model may not have been trained long enough to fully converge with limited computer resources.

(b) Custom BERT from Scratch. Unlike pretrained HuggingFace BERT models, the implementation was trained from scratch, which significantly limits performance due to lack of large-scale pretraining.

(c) Class Imbalance. As the dataset contains imbalance across entailment, neutral, and contradiction classes, it can bias predictions.

(d) Pooling Strategy. Using mean pooling may not capture sentence-level semantics as effectively as CLS pooling or attention pooling.

(e) Computational Constraints. Training a full BERT architecture requires substantial GPU memory and training time.

Potential improvements or modifications

To improve performance, the following modifications can be done:

- Use pretrained BERT weights (e.g., bert-base-uncased)
- Increase training epochs
- Apply learning rate scheduling
- Use dropout regularization
- Apply class-weighted loss to handle imbalance
- Replace mean pooling with CLS token representation

Hyperparameters

Training BERT model to take as pretained model for the assignment

- n_layers = 12   # number of Encoder of Encoder Layer
- n_heads = 12   # number of heads in Multi-Head Attention
- d_model = 768 # Embedding Size
- d_ff = d_model * 4 # 4*d_model, FeedForward dimension
- d_k = d_v = 64 # dimension of K(=Q), V
- n_segments = 2
- num_epoch = 1000