Report HW 2

Daniele Santino Cardullo
2127806
AIRO
cardullo.2127806@studenti.uniroma1.it

1 Introduction

In this homework, the goal is to deal with Natural Language Inference (in particular with an adversarial dataset), training a transformer-based model on the FEVER dataset. The chosen model for my implementation is `base DistilBERT` due to its performance compared to its relatively small size, which allows for training on a local machine.

2 Task Description

The natural language inference task (NLI) refers to the problem of determining if a given hypothesis entails, contradicts, or is neutral with respect to a previously given premise.

The adversarial version of this task is more challenging because it has to deal with an adversarial dataset, i.e., a dataset specifically designed to fool models.

3 Base Dataset and Adversarial Dataset Description

The provided base dataset is extracted from the original base FEVER dataset and adapted to the natural language inference task. In particular, the most notable modifications applied to the dataset are:

- Only one premise and one hypothesis for each entry;
- Labels have been modified into `["ENTAILMENT", "CONTRADICTION", "NEUTRAL"]`.

The given dataset has 51,086 train samples, 2,288 validation samples, and 2,287 test samples. The train set, in particular, is quite unbalanced in favor of class `"ENTAILMENT"` samples, as can be seen in Fig. 1. For this reason, an always-output-most-common baseline might be useful to evaluate the models.

The provided adversarial dataset consists of only the test set, which contains 337 samples evenly spread over the three classes (Fig. 2).

4 Designed Architecture and Performance

To face the NLI classification task, I relied on an architecture based on a pre-trained DistilBERT model that can be found in the `distilbert/distilbert-base-uncased` repository on HuggingFace (the model has never been trained on the adversarial set).

The DistilBERT model is the output of a distillation process having BERT as the teacher. BERT is an encoder-only transformer model designed by Google for masked language modeling tasks. In particular, it has been trained to infer masked words given their context. In my implementation, only the text encoding part has been used, and then it has been integrated with an average pooling operation over the last hidden states along with a single-layer classifier. The architecture of the model used with the base dataset is shown in Fig. 3. The designed architecture receives as input the tokenized sequence where the first token is the classification token, and premise and hypothesis are separated by a separation token. It also receives an attention mask to let the model know where the padding tokens are and ignore them.

The finetuning process has been performed with batch size 16, `AdamW` optimizer, learning rate $1e-5$, and it started overfitting after the first epoch. That is why the selected model has been the one trained in the first epoch, which achieved 69.8% accuracy on the validation set, with a 0.698 F1-score (Fig. 4). On the given base test set, the model achieved 66% accuracy, which outperforms the baseline (Tab. 1).

On the adversarial test set, the model can only achieve 48% accuracy, which still outperforms the baseline but is clearly insufficient.

To better visualize the results on the base and adversarial sets, it is possible to refer to confusion matrices in Fig. 5. The majority of confusion happens in distinguishing entailment and neutral classes, which are respectively the most and the least common classes in the train set.

5 Adversarial NLI Problem

In order to perform better on the adversarial test set, I performed data augmentation on the original training set.

5.1 Data Augmentation

The data augmentation pipeline I designed is formed by two parts: synonym substitution and neutral hypothesis generation.

Synonym substitution can be easily done using WSD information given in the dataset and the WordNet corpus provided by the NLTK library. In particular, the changes focus on modifying the premises (that are longer, hence allowing for more modifications), changing the words tagged as nouns, verbs, and adverbs with their synonyms randomly extracted from the WordNet proposals. This results in sentences that have the same meaning but with different (and often unusual) words.

A second step in the data augmentation pipeline is to generate new neutral samples because it has been observed that this is the least common class of samples and also the most misclassified one (especially in the adversarial case). For this reason, some randomly selected entailment-class samples have been rephrased to become neutral. The procedure is straightforward and requires the use of the GPT-2 model, which is fed with a truncated portion of the premise and outputs a continuation of the sentence, generating a more general hypothesis that can be classified as being neutral.

5.2 Performances

After generating the augmented (and adversarial) training dataset, it has been attached to the original training dataset in order to train a new, more robust model.

After training the model on the new augmented dataset for 5 epochs, the results are better than the base case, not quantitatively but qualitatively. The accuracy on the base set has remained around 66%, and the accuracy on the adversarial set has slightly increased to 49%. Observing the confusion matrices (Fig. 6), it is possible to see how the neutral-labeled samples are now recognized much better than in the base case, and the confusion distribution is generally more balanced.

A Figures

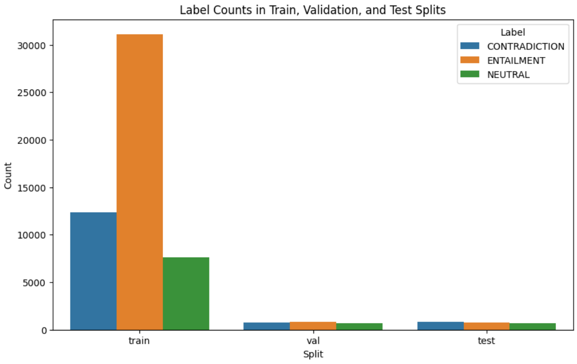
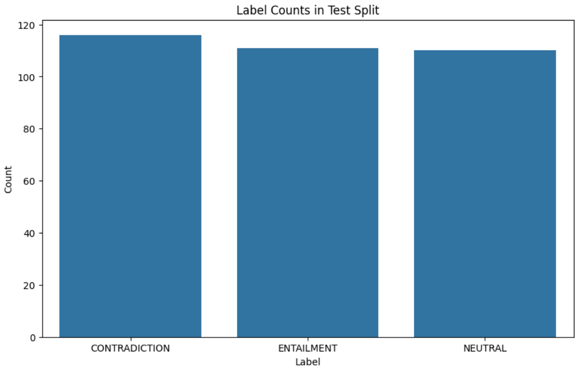
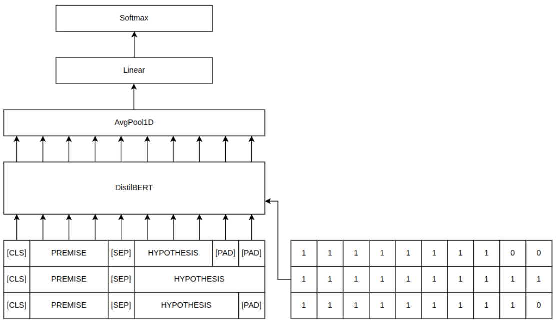


Figure 1: Class count for each split in the base dataset.





[CLS]	PREMISE	[SEP]	HYPOTHESIS	[PAD]	[PAD]
[CLS]	PREMISE	[SEP]	HYPOTHESIS		
[CLS]	PREMISE	[SEP]	HYPOTHESIS	[PAD]	

1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0

Figure 2: Class count in the adversarial dataset.

Figure 3: Model architecture to deal with the base set.

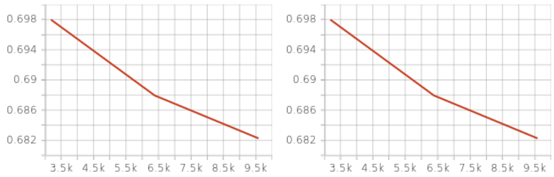


Figure 4: Accuracy and F1 score in finetuning on the validation set.

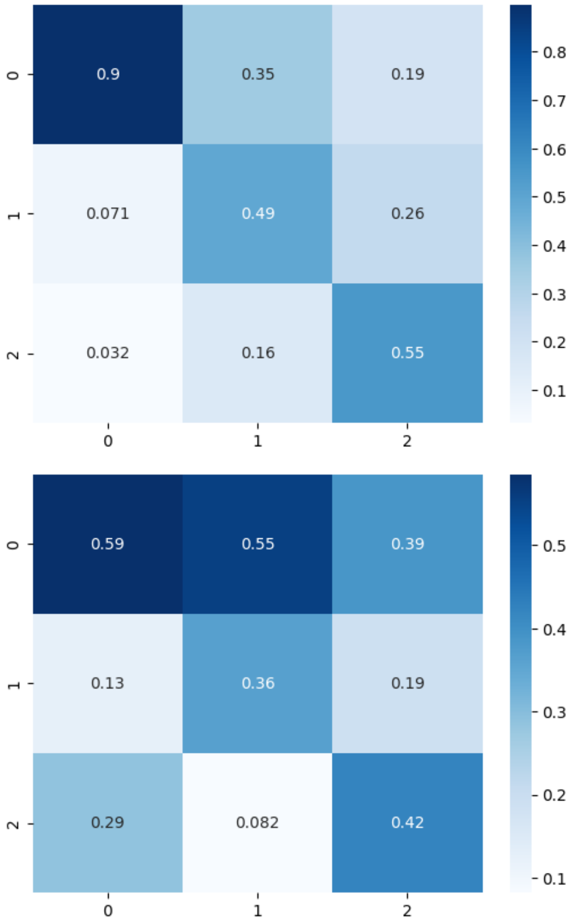


Figure 5: Confusion matrices on base and adversarial test sets.

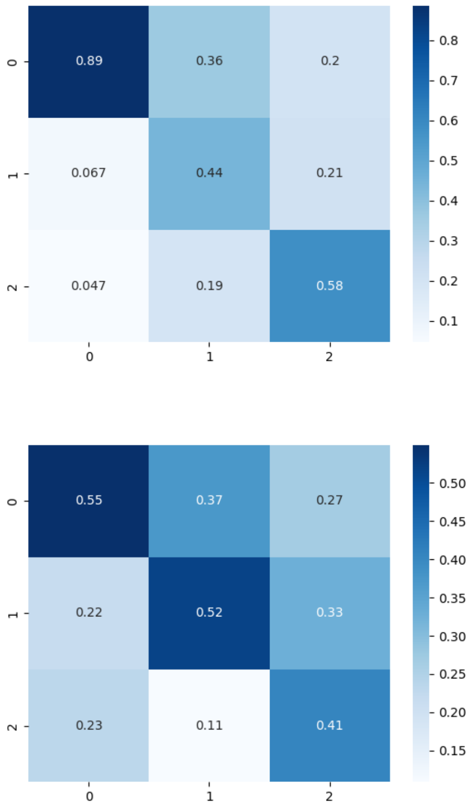


Figure 6: Confusion matrices on base and adversarial test sets after finetuning the model on the augmented adversarial train set.

B Tables

Set	**Accuracy**	**Loss**
Base	42.3%	0.55
Adversarial	39.8%	0.91

Table 1: Baseline results on base and adversarial sets.