Adversarial Natural Language Inference

Federico Matarante,
2133034

Abstract

The task of Natural Language Inference (NLI) consists in telling whether an hypothesis entails or contradicts a premise. This report explains the process and results of training two models for this task. Furthermore the existing dataset has been augmented, creating an adversarial version which improves the performances of the models.

1 Datasets

The datasets used are all alterations of the Fever Dataset[1] adapted for the NLI task. All the labels proportions are in Table 1.

1.1 Original dataset

The main dataset is a down-sampled version of the Fever Dataset[2]. The original data has been down-sampled keeping 25% of the original size. Furthermore, information about Semantic Role Labelling and Word Sense Disambiguation has been added.
The dataset has been divided in 3 splits: test, validation and training.
Each record contains: the premise, the hypothesis, the label of the Inference task and information about SRL and WSD.
The three possible classes:

- Entailment: The hypothesis logically follows from the premise.
- Contradiction: The hypothesis contradicts the premise.
- Neutral: The hypothesis neither follows from nor contradicts the premise.

[1] https://huggingface.co/datasets/fever/fever

[2] https://huggingface.co/datasets/tommasobonomo/sem_augmented_fever_nli

1.2 Adversarial test dataset

The 'test' split of the previous dataset has been augmented to create an adversarial[3] version for testing purpose only. The augmentation techniques used are the following:

- manual_simple (62): examples where the hypothesis are modified manually following as modification constraint SRL and WSD tags.
- manual_adversarial (134): examples where the hypothesis are modified manually to fool 3 expert NLI models.
- gpt_generated (141): examples where the hypothesis are created by GPT-4o to be tricky to solve.contradicts the premise.

1.3 Adversarial dataset

The original dataset has been augmented on the three splits to create an adversarial dataset to train the model. The techniques used are the followings:

- Random Hypothesis Substitution: an hypothesis of a record is substituted with another random hypothesis from the dataset, making the inference label always "Neutral".
- Hypernymy Substitution: a random noun (chosen between the 'ARG0', 'ARG1' and 'ARG2' semantic labels of the English Propbank [4]) of the hypothesis has been replaced by its hypernymy.
- Verb Negation: the verb of the hypothesis has been negated by hand-crafted grammatical rules. In this case the labels are inverted.
- Verb Synonym Substitution: the verb of the hypothesis have been replaced by a random synonym.

[3] https://huggingface.co/datasets/iperbole/adversarial_fever_nli

[4] https://propbank.github.io/

- Premise Summarization: the premise is summarized using a Distilbart model[5].

Each record of the main dataset have been altered individually by choosing a random augmentation technique, giving different weight to the random choice considering the estimated probability of it being correctly applicable.
After some cross validation, it's been noticed that other methods (named "Date Substitution" and "Adverb Inversion") would impact negatively on the model's accuracy, so they've not been included.

2 Models and training

After some manual testing and considerations about the datasets size, it's been chosen to use the pre-trained distilbert model[6] - which is a smaller version of Bert.
The base distilbert encoder has been used to create two different models:

1. The normal one FC layer classifier. The whole model has been trained by freezing up to the second layer's parameter and then fine-tuned.

2. A two FC layers classifier followed by a softmax function. The model has been through 2 training processes: the first one consist in only training the Distilbert base with a masked training in which one random relevant word of the hypothesis was masked and predicted. The second one is a training on the multi-layer classifier. The idea was to make it learn explicitly some features about the NLI as a pretext task.

The parameters used for all the trainings are in the tab. 2, while the metrics after the evaluations are in the tab 3.

2.1 Training on original dataset

The model 1 (fig. 1) outperformed the model 2 (fig. 3 and fig. 4) in every aspect; the losses evolution makes clear that the multi layer classifier would need more training and possibly more data to improve its performance. Also the predictions on this model are always uniform, meaning that the model didn't learn the task.
The unbalanced confusion matrix of the first model's evaluation probably reflects the fact that the original dataset is not well balanced.

2.2 Training on adversarial dataset

The model 1 trained on the original dataset has been fine-tuned the adversarial version. For the model 2 It's been chosen to keep only the weights of the Distilbert base and training the classifier from scratch on the adversarial dataset.
The performances of both the models have improved, but the consideration about the difference between the two has remained comparable. One further positive aspects relies on the fact that the confusion matrix of the one-layer classifier model are now more balanced, probably due to the fact that the adversarial dataset labels distribution is more balanced.

3 Code instructions

1. Download the code from the .rar files.

2. Make sure to have Python 3 installed.

3. Go to the directory in which there are the files “requirements.txt”,“2133034-augment.py” and “2133034-main.py”.

4. Install the dependencies:

```
pip install -r requirements.txt
```

5. Execute the scripts:

   - To generate the adversarial dataset:

   ```
   python 2133034-augment.py
   ```

   - To train or test the model:

   ```
   python 2133034-main
   [train|test]
   [original|adversarial]
   ```

Further information is on the readme.txt file.

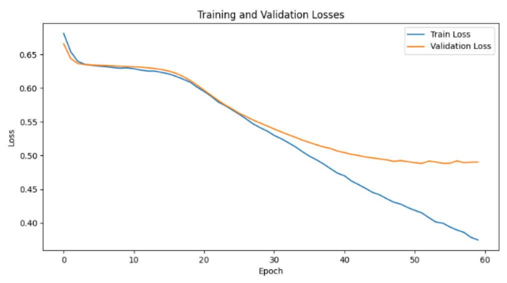

Figure 1: Training of model 1 on original dataset

[5] https://huggingface.co/sshleifer/distilbart-cnn-12-6
[6] https://huggingface.co/distilbert/distilbert-base-uncased

Dataset	**Entailement**	**Neutral**	**Contradiction**	**Total elements**
Original (Train split)	31128 (60.93%)	7627 (14.93%)	12331 (24.14%)	51086
Original (Test split)	792 (34.63%)	683 (29.86%)	812 (35.51%)	2287
Original (Validation split)	821 (35.88%)	692 (30.24%)	775 (33.87%)	2288
Adversarial Test	111 (32.94%)	110 (32.64%)	116 (34.42%)	337
Adversarial (Train split)	17837 (34.92%)	8219 (16.09%)	25030 (49.00%)	51086
Adversarial (Test split)	786 (34.37%)	709 (31.00%)	792 (34.63%)	2287
Adversarial (Validation split)	751 (32.82%)	724 (31.64%)	813 (35.53%)	2288

Table 1: Dataset proportions

Model	**Dataset**	**Epochs**	**Batch size**	**Learning rate**
Model 1	Original	60	16	1e-06
Model 1	Adversarial	10	16	1e-05
Model 2 (body)	Original	30	16	1e-04
Model 2 (classifier)	Original	20	16	1e-04
Model 2 (classifier)	Adversarial	20	16	1e-04

Table 2: Training configurations

Model	**Trained on**	**Test set**	**Accuracy**	Precision	Recall	F1-Score	Conf. Matrix
Model 1	Original	Original	79.54%	79.60%	79.54%	79.11%	Fig 6
Model 1	Original	Adversarial	48.37%	47.62%	48.37%	47.61%	Fig 7
Model 1	Adversarial	Original	83.03%	83.83%	83.03%	82.96%	Fig 8
Model 1	Adversarial	Adversarial	49.55%	50.24%	49.55%	49.78%	Fig 9
Model 2	Original	Original	35.81%	40.95%	35.81%	25.25%	Fig 10
Model 2	Original	Adversarial	32.34%	21.19%	32.34%	20.27%	Fig 11
Model 2	Adversarial	Original	34.46%	35.73%	34.46%	31.52%	Fig 12
Model 2	Original	Adversarial	33.53%	30.30%	33.53%	24.44%	Fig 13

Table 3: Models evaluation metrics

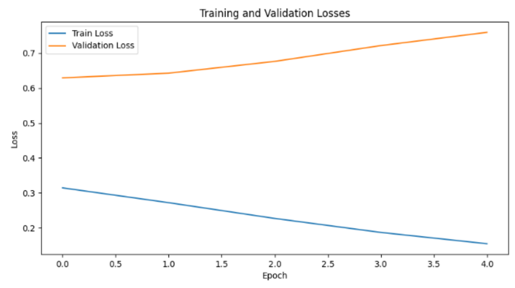

Figure 2: Training of model 1 on adversarial dataset

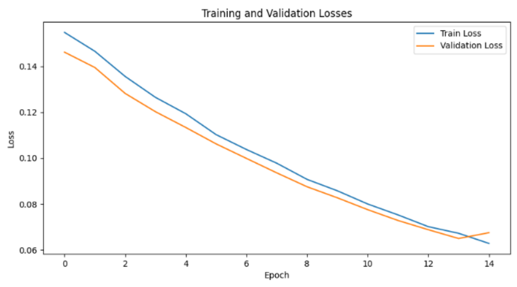

Figure 3: Training of model 2 body on original dataset

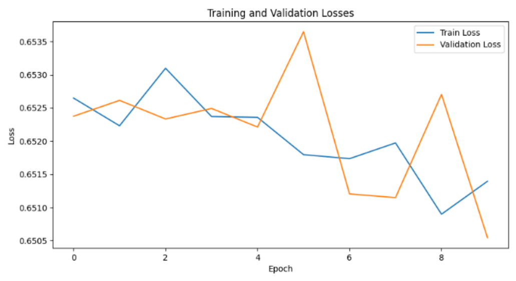

Figure 4: Training of model 2 classifier on original dataset

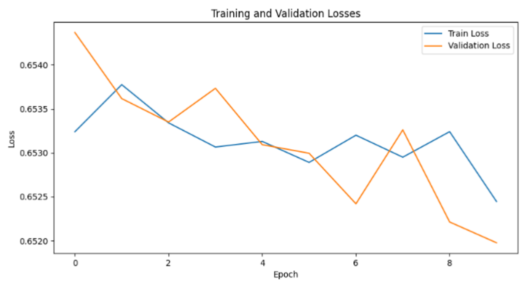

Figure 5: Training of model 2 classifier on adversarial dataset

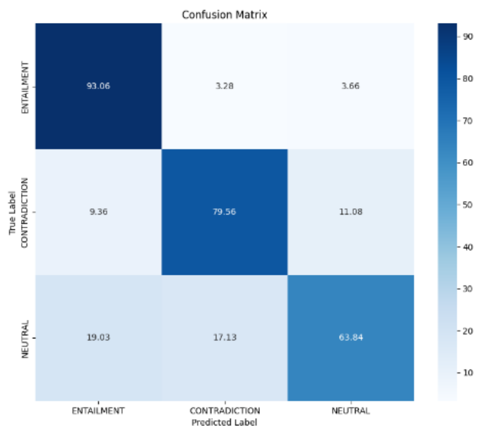

Figure 6: CF 1

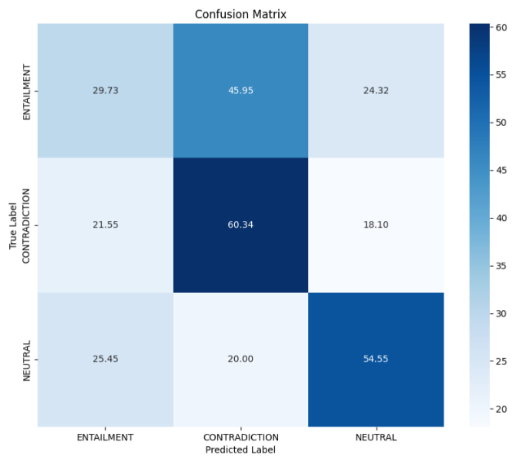

Figure 7: CF 2

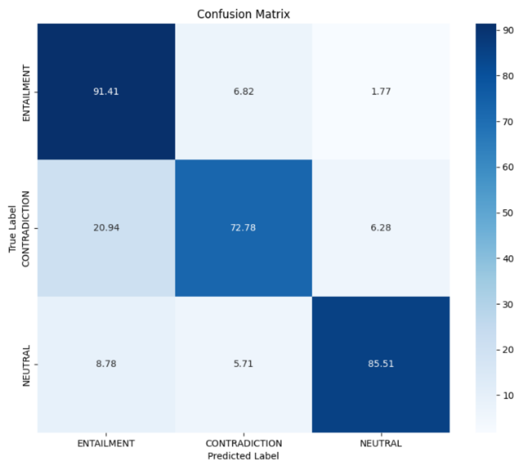

Figure 8: CF3

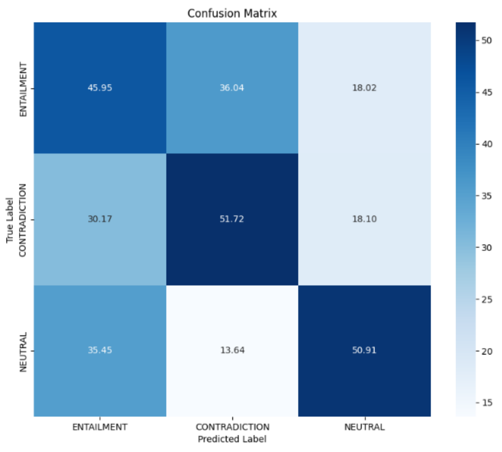

Figure 9: CF4

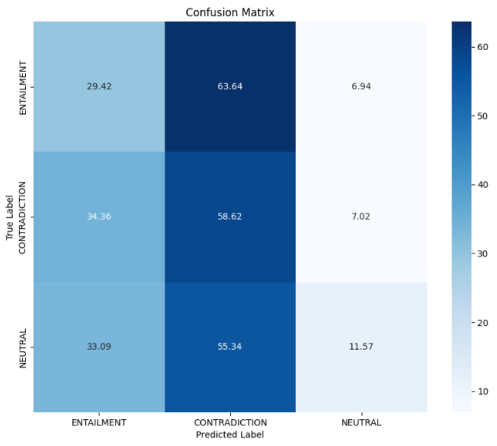

Figure 12: CF7

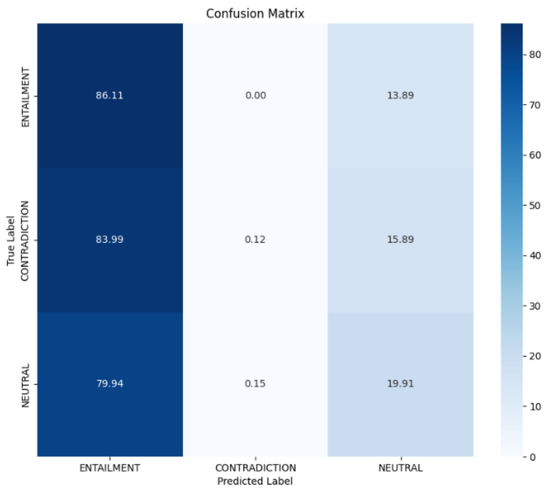

Figure 10: CF5

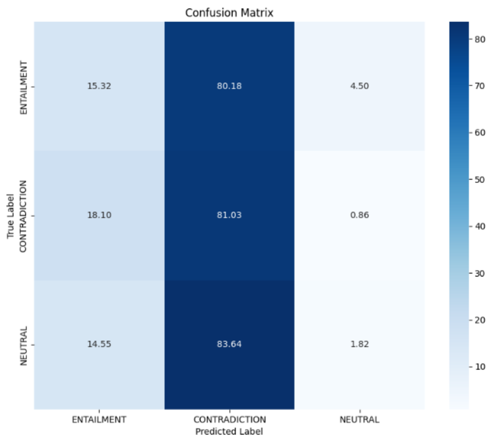

Figure 13: CF8

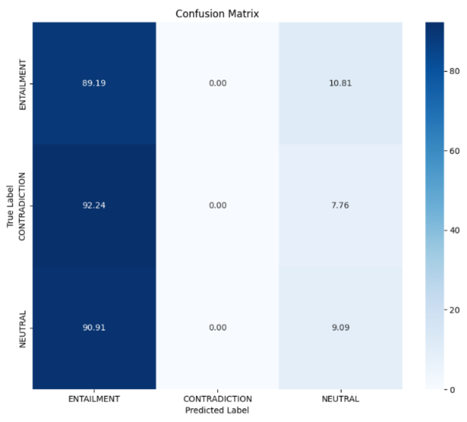

Figure 11: CF6