Adversarial Natural Language Inference on FEVER Dataset Using Data Augmentation

Emanuele Frasca
Sapienza University of Rome
frasca.1836098@studenti.uniroma1.it

Abstract

This paper describes the development and evaluation of three BERT models for Natural Language Inference (NLI) tasks using a subset of the FEVER dataset. Models are evaluated on two test sets: a simpler one and an adversarial one. Each model is trained on both the original dataset and an augmented one using various techniques. Results show that all models outperform baselines, with the best one achieving an accuracy of 76.87% on the simpler test-set and 68.25% accuracy on the adversarial test set. Findings suggest data augmentation enhances model performance on complex examples, and that choosing the right model is crucial for handling complexity.

1 Task and Dataset Description

The task involved is NLI, which involves determining the relationship between a given pair of sentences: a premise and a hypothesis. The goal is to classify this relationship into one of three categories: *entailment*, *contradiction*, or *neutral*.

The dataset used for this task is a subset of the FEVER dataset, a hand-curated dataset for fact extraction and verification built from Wikipedia. The samples consists of premises, hypotheses, and corresponding labels. Additionally, the dataset is enhanced with *Word Sense Disambiguation* (WSD) and *Semantic Role Labeling* (SRL) annotations to provide more detailed linguistic information, which can help in better understanding the content of texts.

2 Model Architecture

The architecture used is based on RoBERTa-Base, which is a transformer-based model that has been pre-trained on a large corpus of text.
The model consists of three main components:

- **Input Layer**: Takes tokenized sequences and attention masks, which are essential for the Roberta model to process the sequences correctly.
- **Transformer Layer**: The core RoBERTa model processes the input sequences through multiple transformer layers, generating hidden states for each token at each layer.
- **Classification Head**: A fully connected layer that takes the final hidden state of the [CLS] token as input, producing logits representing the scores for each one of the three classes.

A second model has been implemented modifying the base architecture adding a dropout layer after the transformer to prevent overfitting and freezing the weights of 6 layers of the Roberta model to check if the model can learn more complex patterns maintaining more of its base training weights.

3 Methodology

To ensure meaningful learning from the data, a random baseline and a majority baseline has been implemented.

The dataset preprocessing involved several steps. First, all text was converted to lowercase and tokenized using the RoBERTa tokenizer with a maximum length of 512 tokens. Labels were converted to integers to serve as targets for the model. *Cross-Entropy* was used as the loss function.

Given the original dataset's imbalance, with a majority of examples labeled as entailment, as it can be seen in Figure 1a, the loss function was weighted to emphasize the minority classes. These weights were computed as the inverse of the class frequencies in the train-set.

The *AdamW* optimizer was used for training, with a learning rate of *1e-5*, a batch size of *8* and a patience of *2*. After each training epoch, model evaluation was performed on the validation set, and the best model was selected based on the highest validation accuracy.

Model performance was tested using *accuracy*, *precision*, *recall*, and *F1-score* metrics on the test-sets.

4 Data Augmentation

Data augmentation techniques have been used to generate new samples from the original train-set. These include methods used to generate new samples that are still sound but have a different label and some methods used to generate noisy samples.

- **Synonyms substitution:** Words in the premises are substituted using synonyms and related words from WordNet.
- **Sentence negation:** The hypothesis is negated to create a contradictory sample.
- **Gender guesser:** Guesses and generates entailment or contradiction samples based on the inffered gender.
- **Replace names:** Proper names in the hypotheses are replaced with other names.
- **Change dates:** Dates in the hypotheses are altered to create new samples.
- **Shuffle sentences:** The order of sentences in the hypotheses is shuffled.
- **Random phrase insertion:** Random phrases are inserted into the hypotheses.
- **Random characters remover:** Random characters are removed from the hypotheses.
- **Random words remover:** Random words are removed from the hypotheses.

An example usage of the data augmentation can be seen in Table 2. The number of new samples generated can be seen in Figure 1

5 Advanced Model Architecture

From Table 1 it can be seen that both the RoBERTa-Base model and RoBERTa-Base-DF trained on the original dataset performs well on the simple test-set but struggles on the adversarial one even using the augmented dataset. To understand if the problem is caused by the poor quality of the augmented data or by the model's capabilities, a more complex model has been implemented. The DeBERTa-v3-Large model has been trained on the original and augmented datasets to see if this model can learn more patterns from the datasets and so improve the performance on the adversarial test-set.

6 Results

As shown in Table 1 and Figure 2, all models outperform the baselines on both test-sets. The DeBERTa-v3-Large model trained on the original dataset achieves 76.78% accuracy on the original test-set and 66.77% on the adversarial test-set. When trained on the augmented dataset, it achieves 76.87% accuracy on the original test-set and 68.25% on the adversarial test-set. In contrast, the RoBERTa-Base model trained on the original dataset achieves 74.46% accuracy on the original test-set and 53.71% on the adversarial test-set. With the augmented dataset, the RoBERTa-Base model achieves 72.98% accuracy on the original test-set and 54.90% on the adversarial test-set.

The results show that the DeBERTa-v3-Large model can learn more complex patterns and generalize better to unseen examples compared to the RoBERTa-Base model. The RoBERTa-Base model, while capable of learning meaningful representations, struggles to generalize to more complex examples. Furthermore, adding a dropout layer and freezing the initial layers of the RoBERTa-Base model does not significantly improve its performance.

Overall, the augmentation techniques used to create the augmented dataset effectively enhance the DeBERTa-v3-Large model's performance on the more complex test set, highlighting the limitations of the RoBERTa-Base model in handling complex examples, even with data augmentation.

7 Scripts Usage

The implementation of the described methodologies are provided as Python scripts. To ensure reproducibility of the results, a fixed seed has been set. A *readme.md* file is provided to facilitate the use.

In the terminal, navigate to the directory containing the scripts and run the following commands to install the required libraries, save plots images, create the augmented set and to train and evaluate the models:

```
> pip install -r requirements.txt
> python 1836098-plots.py
> python 1836098-augment.py
> python 1836098-main.py [train, test, baselines] --data [original, adversarial] --model [roberta, roberta_df, deberta]
```

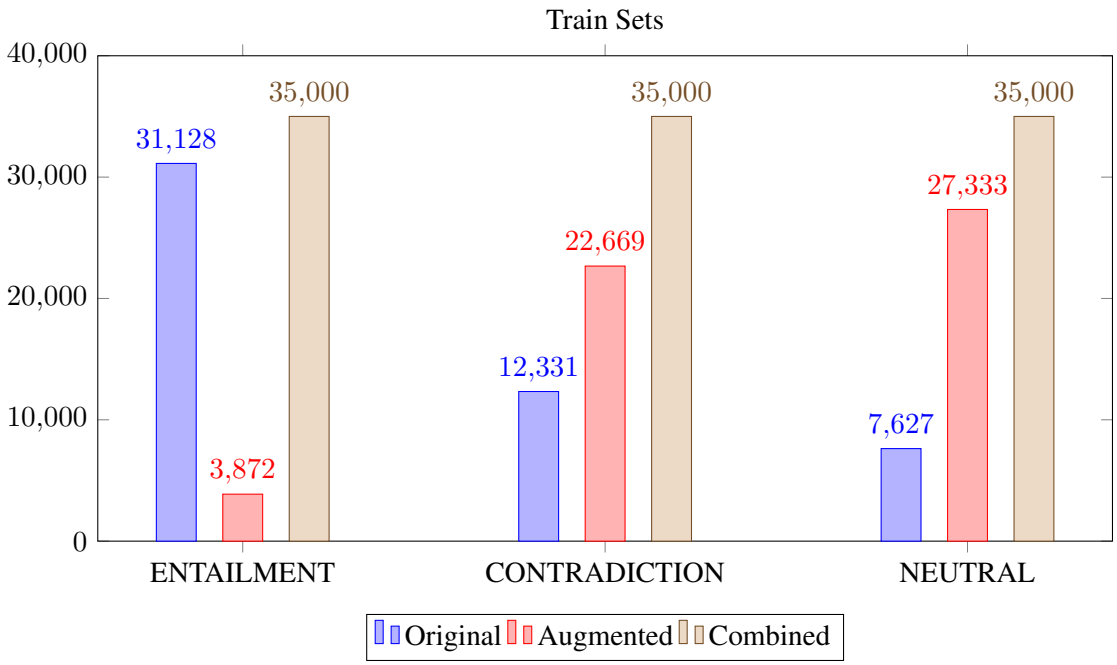
The models and the augmented sets will be saved on the scripts folder.

Model	Train-set	Test-set	Accuracy	Precision	Recall	F1-score
Random Baseline	N/A	Original	33.10%	33.39%	33.10%	33.18%
Random Baseline	N/A	Adversarial	34.12%	34.35%	34.12%	34.07%
Majority Baseline	N/A	Original	35.51%	12.61%	35.51%	18.61%
Majority Baseline	N/A	Adversarial	34.42%	11.85%	34.42%	17.63%
(1) RoBERTa-Base	Original	Original	74.46%	74.05%	74.46%	73.75%
(2) RoBERTa-Base	Original	Adversarial	53.71%	53.88%	53.71%	53.75%
(3) RoBERTa-Base	Augmented	Original	72.98%	72.65%	72.98%	72.06%
(4) RoBERTa-Base	Augmented	Adversarial	54.90%	57.64%	54.90%	55.28%
(5) RoBERTa-Base-DF	Original	Original	72.67%	74.27%	72.67%	72.93%
(6) RoBERTa-Base-DF	Original	Adversarial	54.60%	56.70%	54.60%	53.00%
(7) RoBERTa-Base-DF	Augmented	Original	72.45%	72.45%	72.45%	71.49%
(8) RoBERTa-Base-DF	Augmented	Adversarial	55.19%	56.69%	55.19%	55.53%
(9) DeBERTa-v3-Large	Original	Original	76.78%	77.28%	76.78%	76.96%
(10) DeBERTa-v3-Large	Original	Adversarial	66.77%	70.35%	66.77%	65.79%
(11) DeBERTa-v3-Large	Augmented	Original	76.87%	76.42%	76.87%	76.53%
(12) DeBERTa-v3-Large	Augmented	Adversarial	68.25%	68.86%	68.25%	67.62%

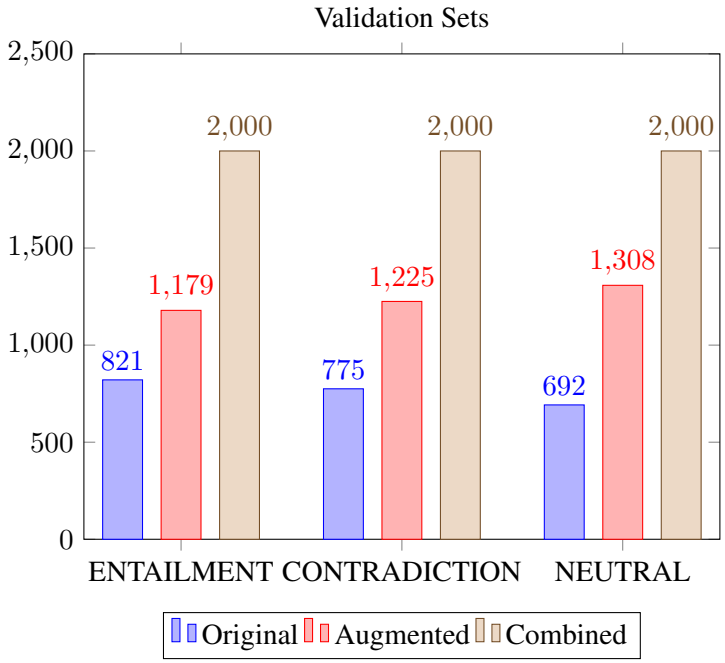
Table 1: Performance metrics of the described models and baselines. Values in red indicate the best performance metrics.

Technique	Result
Original sample	**P:** Mario Rossi is an hydraulic. He is born the 12 October 1922 **H:** Mario was born the 12 October 1922, **L:** Entailment
(P) Sentence negation	**H:** Mario wasn't born the 12 October 1922 **L:** Contradiction
(H) Gender guesser	**H:** Mario Rossi is a male **L:** Entailment
(P) Replace names	**H:** Jane Smith was born the 12 October 1922 **L:** Contradiction
(H) Change dates	**H:** Mario was born the 24 October 1922 **L:** Contradiction
(P/H) Synonyms substitution	**H:** Mario was born the 12 Oct 1922 **L:** Entailment
(H) Shuffle sentences	**H:** No change **L:** No change
(P) Random phrase insertion	**P:** Mario Rossi is an idraulic. Nevertheless, he was born the 12 October 1922. **L:** Entailment
(P) Random characters remover	**P:** Mario Rossi is an idrulic. He is bor the 12 ocbr 92 **L:** Entailment
(H) Random words remover	**H:** He is born 12 October **L:** Entailment

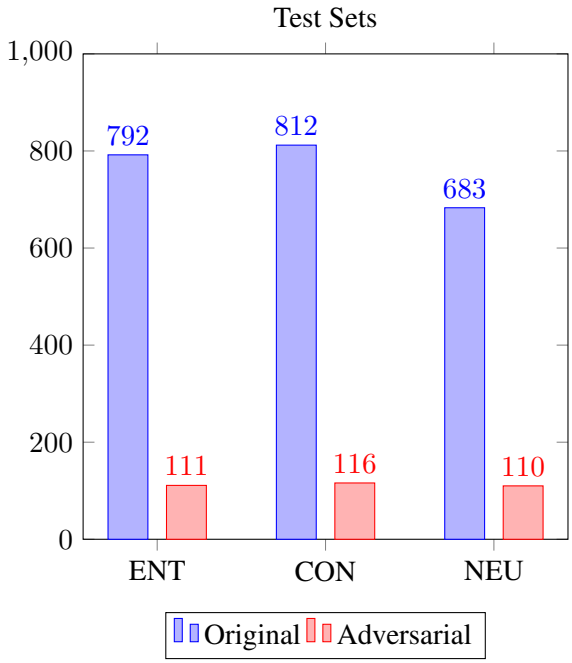
Table 2: Usage examples of data augmentation techniques implemented. (P) indicates a modification of the premise, (H) a modification of the hypothesis and (P/H) a modification of both of them.



(a) Comparison of sample numbers in train-sets.



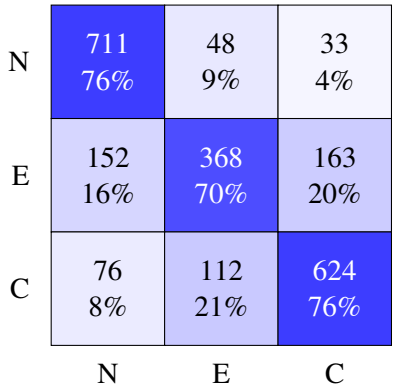
(b) Comparison of sample numbers in validation-sets.



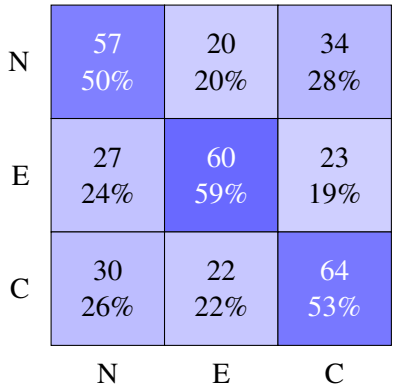
(c) Comparison of sample numbers in test-sets

Figure 1: Comparison of sample numbers in validation-sets and test-sets.

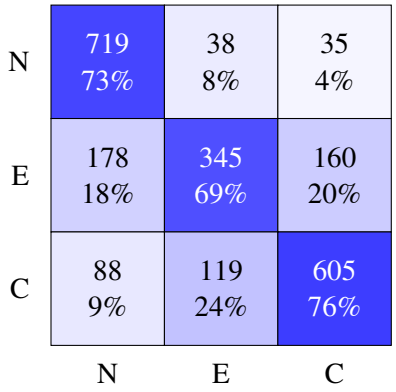
Figure 2: Confusion matrices for each model present in Table 1.



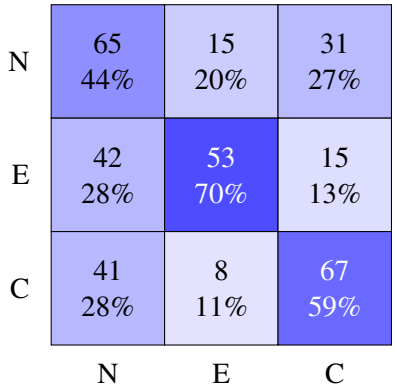
(a) Model 1



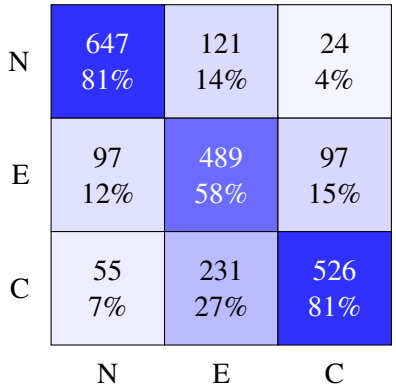
(b) Model 2



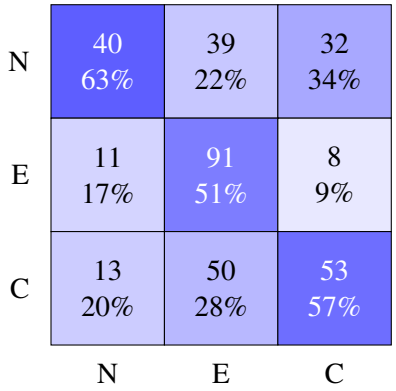
(c) Model 3



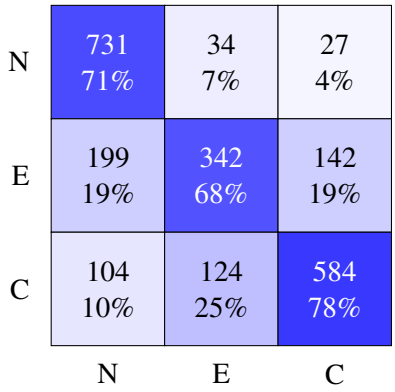
(d) Model 4



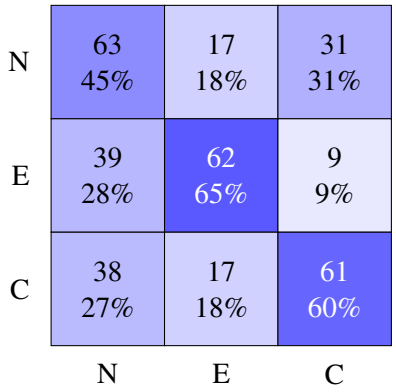
(e) Model 5



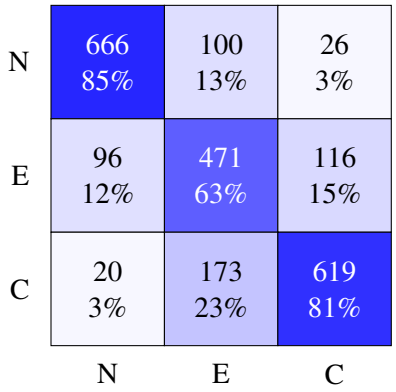
(f) Model 6



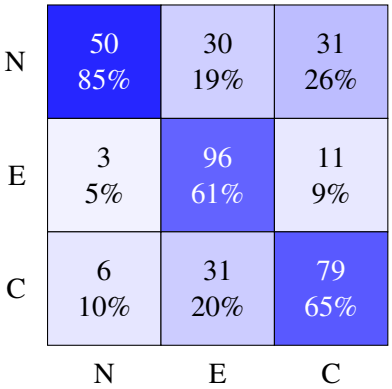
(g) Model 7



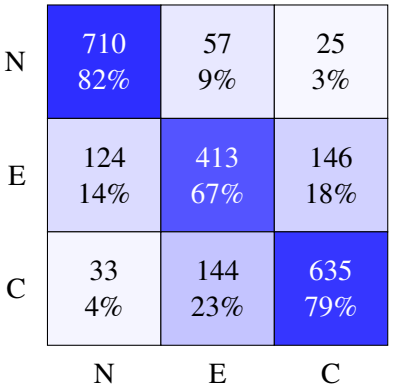
(h) Model 8



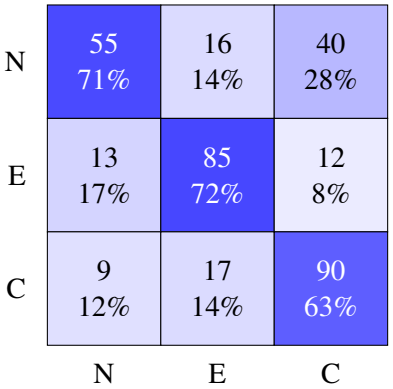
(i) Model 9



(j) Model 10



(k) Model 11



(l) Model 12