Homework 2 - Multilingual Natural Language Processing

Leonardo Sandri
2137374

Abstract

This document presents the methodology employed to tackle an **adversarial Natural Language Inference (NLI)** task using a **FEVER** dataset provided. The chosen model is **DistilBERT**, a pre-trained transformer from the **Hugging Face** library. The model was fine-tuned on the **NLI** task by training its classifier on the **FEVER** dataset. To enhance performance metrics such as accuracy, data augmentation techniques were applied. These techniques included **Word Sense Disambiguation (WSD)** and **Semantic Role Labelling (SRL)**. The model was trained also on the augmented datasets and performances compared between each other.

1 Dataset description

The **FEVER (Fact Extraction and VERification)** dataset is a large-scale benchmark designed for verifying factual claims using textual evidence. It consists of a certain number of claims which in the specific case of our project were down-sampled. In the context of our **NLI** task, pairs of context and claim were used as **(premise, hypothesis)**, and our model was tasked with labeling these pairs as **Entailment, Contradiction, or Neutral**. Samples of the dataset had a structure consisting of a premise, a hypothesis, a label, and two dictionaries called **SRL** and **WSD**, which contained important information about context, semantic roles, and word senses, respectively. An adversarial dataset was also given, constructed using a variety of data augmentation techniques. The role of adversarial datasets is crucial in training robust models as they introduce challenging examples designed to test the model's ability to generalize beyond the standard training set.

2 Model Architecture

The **distilbert-base-uncased model** is a smaller, faster version of **BERT**, using 6 layers instead of 12. In our project, it was fine-tuned for sequence classification with 3 labels (entailment, contradiction, neutral). The training process included **3 epochs**, a **batch size = 8** and a **linear learning rate scheduler** with a **starting learning rate = 1e-5**, including **500 warm-up steps**. Scheduling the learning rate is beneficial for fine-tuning problems as it helps in gradually adapting the pre-trained model to the new task. This method helps the model to learn more steadily and reliably. Evaluations were conducted every 500 steps, and metrics such as **accuracy** and **F1 score** were used to assess performance.

3 Design choices

I chose to run the training for only 2 epochs because I observed that between the second and third epochs, the model began to show signs of overfitting. This was indicated by fluctuations in both the loss and accuracy metrics, suggesting that the model was becoming too specialized to the training data. Observing significantly higher performance on the validation set of FEVER compared to the adversarial set was surely due to the test set's adversarial nature. This served as a way to measure generalization, giving hints when performance was truly being optimized, and when potential overfitting was being mitigated, which became a critical focus. Initially, I set a high **learning rate = 0.0001**, but the model did not make meaningful progress during training. To address this, I reduced the starting learning rate to **1e-5** and introduced warmup steps. Warmup steps gradually increase the learning rate from a very low value to its initial value, helping the model to stabilize and learn effectively from the training data.

4 Data augmentation and its results

In augmenting the dataset, two primary semantic approaches were adopted: **Word Sense Disam-**

biguation (WSD) which substituted words with synonyms and **Semantic Role Labelling (SRL)**, which exchanged words between different semantic roles in sentences.

4.1 Word Sense Disambiguation

Exploiting the **'wsd'** dictionary from the FEVER training set, offsets for each word in premise samples were extracted, including context and **part-of-speech (POS) tags** aligned with **WordNet** database. Synonym sets from WordNet were then used to substitute each word in samples with its first associated synonym, augmenting the dataset while preventing creation of overly similar samples. Changing only one word with its synonym, or generating a new sample for each possible synonym of a word, would result in training the model on numerous nearly identical samples.

This was deduced through another approach: generating additional samples by substituting each word with each of its identified synonyms, one per sample, which significantly increased the dataset size. While achieving validation accuracy of 0.98 on the FEVER set, performance notably declined on the adversarial set, prompting abandonment due to pronounced overfitting. This underscored the risk of training on numerous nearly identical samples, guiding a shift toward diverse approaches aligning with project guidelines.

4.2 Semantic Role Labelling

The next implemented approach is **Semantic Role Labelling (SRL)**. I wrote a function that can extract roles of words in the samples from the **'srl'** dictionary. A particular focus was placed on samples where roles included "agent" and "patient". Using a loop, spans of word indexes associated with these roles were identified and used to interchange words assigned to the agent role with those assigned to the patient role.

4.3 Augmentation Results

While **WSD** improved adversarial test accuracy from **0.51** to approximately **0.55**, **SRL** maintained accuracy at **0.51**. When combining the augmented datasets from both WSD and SRL, the overall improvement was modest, increasing accuracy to around **0.52**, which reflects how SRL negatively influenced the results. This slight decrease in accuracy may have been influenced by issues such as inadvertent generation of similar samples. Additionally, semantic transformations introduced by exchanging agent and patient roles occasionally resulted in nonsensical or "not-sound" samples, leading to ambiguous sentence meanings. These results highlight the importance of strategic dataset augmentation and the refinement of methods to mitigate overfitting, while enhancing model robustness.

5 Conclusion

While the application of WSD significantly improved the model's performance on adversarial tests, increasing accuracy from 0.51 to approximately 0.55, the substantial increase in dataset size due to WSD could also lead to pronounced overfitting. To address this, SRL was introduced to add diversity through a different and varied approach. However, misconsiderations in SRL functioning and in the construction of "sound samples" caused the overall performance to slightly decrease, despite the initial improvements brought by WSD.

These results underscore the importance of balancing dataset diversity and augmentation volume. Exaggerating dataset size without ensuring diversity can lead to overfitting, while diverse and well-thought-out augmentation strategies are crucial to enhancing model robustness and generalization.

6 Instruction to run the code

The code is organized into two Jupyter notebooks: one dedicated to dataset augmentation and another to model training and evaluation.

The dataset augmentation notebook requires execution with the data directory path accessible, where the original FEVER dataset must be stored. It will generate six datasets as output divided in three pairs of training and validation sets, created respectively using the WSD approach, the SRL approach, and a combination of WSD and SRL.

The model training and evaluation notebook is configured to perform model evaluation by default. To initiate model training, uncomment the trainer.train() cell corresponding to the desired model and ensure the associated model cell is uncommented as well. This precaution is implemented to avoid inadvertent training runs. Additionally, pretrained model weights are provided in the notebook's directory to streamline the process.

7 Performances metrics

In the *Table 1* are reported the respective results of accuracy of each different trained model.

Model	Train set	Validation set	Accuracy	f1
model1	FEVER	FEVER	0.74	0.73
model1	FEVER	Adversarial set	0.52	0.52
model2	F+WSD	Adversarial set	0.55	0.55
model3	F+SRL	Adversarial set	0.51	0.51
model4	F+W+S	Adversarial set	0.52	0.52

Table 1: Performance results

8 References

- https://pytorch.org/text/stable/data_utils.html
- https://www.kaggle.com/code/mlwhiz/bilstm-pytorch-and-keras
- https://raw.githubusercontent.com/propbank/propbank-documentation/master/annotation-guidelines/Propbank-Annotation-Guidelines.pdf
- https://verbatlas.org/
- https://huggingface.co/docs/datasets/process
- https://huggingface.co/distilbert/distilbert-base-uncased

9 figures

For model names clarity see performance metrics section

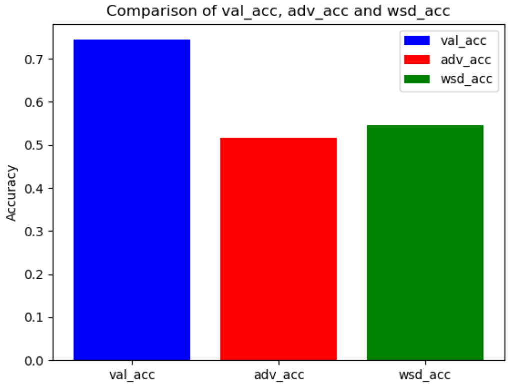

Figure 1: accuracy comparison between model1, model1 evaluated on adversarial and model2

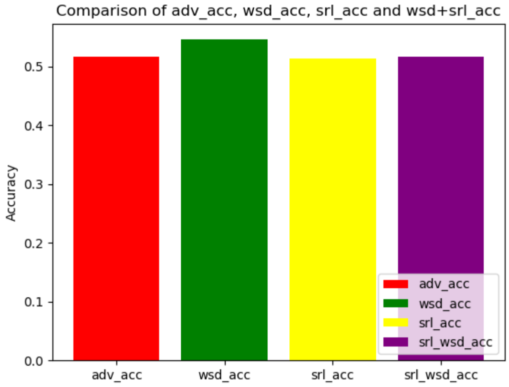

Figure 2: accuracy comparison between model1 evaluated on adversarial, model2, model3 and model4

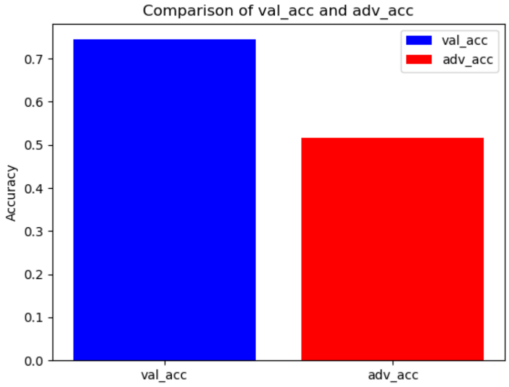

Figure 3: accuracy comparison between model1 and model1 evaluated on adversarial