

# CentraleSupélec - Natural Language Processing

Abel Capitant

28/03/2025

I initially started by working directly within the provided notebook. However, as the project progressed, I decided to change my approach by structuring the code into Python modules. This made the overall pipeline more organized and allowed me to run training and testing more efficiently.

You can check the first ideas I tried in the notebook tp8. But to check my best results, please look at the torchtml folder.

## Testing My Model

To test my best-performing model, I used the following command:

```
python -m torchtml.main config.yaml test
```

The file Results.Roberta.png shows the results of this model, and best\_model.pt contains the saved weights.

## Description of My Work

### 1. Understanding the Dataset

I began by exploring the SNLI dataset. It is structured as a dictionary containing three subsets: train, validation, and test. Each subset includes three fields: premise, hypothesis, and label. (See Notebook)

### 2. Preprocessing

I tokenized the input sentences and computed the maximum sentence length. Then, I created a function prep that concatenates each sentence pair (premise and hypothesis) and applies zero-padding up to the maximum length. The resulting data was converted into PyTorch TensorDataset objects to be used by the model.

### 3. Training

I first trained an ALBERT model with all layers unfrozen, which gave me an accuracy of 0.89.

Next, I experimented with different configurations using the Roberta model:

- Training all layers resulted in an accuracy of 0.90.
- Training only the last layer from a randomly initialized model led to a disappointing 0.60 accuracy.
- Freezing all but the last layer and starting from pretrained weights also led to an accuracy of 0.90.

### Best Model

I selected the Roberta model trained on all layers as my final and best-performing model. The training results can be seen in Results\_Roberta.png, and the corresponding weights are saved in best\_model.pt.