# 2 solutions for NLI: Hypothesis – Premise Relationship Classification

## Introduction

Natural Language Inference (NLI) is a core task in Natural Language Processing (NLP) that involves determining the logical relationship between a given premise and hypothesis. It plays a crucial role in applications such as automated reasoning, question-answering, and fact verification.

In this study, we were given **24,432** labeled hypothesis-premise pairs for training and **6,736** pairs for validation. Our goal was to classify these pairs into one of two categories: **entailment** or **contradiction** (1/0). We developed and compared two distinct approaches:

- A **transformer-based model**, following the third option in the coursework.
- A **bidirectional LSTM model** (deep learning without transformers), following the second coursework option.

To guide our model development, we conducted an extensive literature review to identify trends in state-of-the-art NLI models. This review inspired innovative solutions, leading us to implement and experiment with complex architectures from recent research papers to enhance classification performance.

## FIRST SOLUTION: Transformer-Based Meta-Learning

**Transformers** have revolutionized deep learning, consistently achieving state-of-the-art results in NLP tasks. However, model performance can vary significantly depending on the specific transformer architecture used—especially in low-data scenarios, where overfitting becomes a concern.

Following an extensive literature review and inspired by the 2024 paper *"Natural Language Inference with Transformer Ensembles and Explainability Techniques"*, we explored an **innovative, ensemble-based approach** to NLI. Our goal was to move beyond conventional methods and experiment with a **novel meta-learning strategy** that aggregates insights from multiple transformer models.

### Data preprocessing and tokenization

- "bert-base-uncased"
- "roberta-base"
- "albert-base-v2"

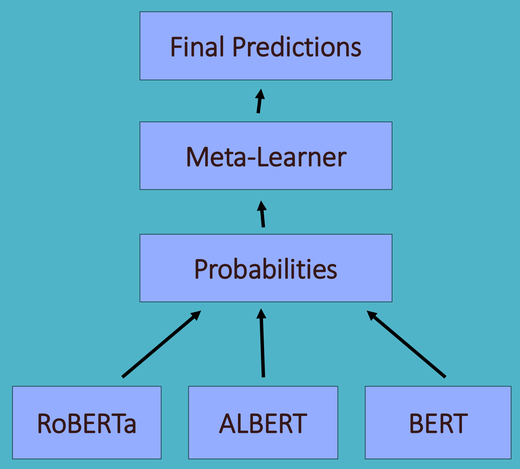
### Fine-tuning the transformers

Hyper-parameters:
- Epochs: 3
- Batch size: 64
- Focal loss for Albert, cross-entropy for Bert and Roberta
- Maximum length of sentence: 124
- Learning Rate scheduler
- Optimizer: Adam

### Preparing the data for the training of the meta-learner

- Using the saved fine-tuned transformers, we generate probabilities for the training dataset for the meta-learner
- The loggit outputs are converted to probabilities using softmax
- The probabilities of the 3 transformers are stacked using: torch.cat

### Main architecture design




### Model Performance and Results

| Model | Accuracy | F1 Score |
|---|---|---|
| BERT | 0.793 | 0.791 |
| RoBERTa | 0.824 | 0.830 |
| Albert | 0.842 | 0.841 |
| Meta-Learner | 0.891 | 0.893 |

Our proposed **Meta Learner model** demonstrated **superior performance**, outperforming all three individual transformer models. This highlights that the **three transformers capture different relationships** within the data — and the Meta Learner effectively **leverages these diverse patterns** to make more accurate predictions.

Below, we present the **validation set results** comparing the individual models and the Meta Learner.

## SECOND SOLUTION: BiLSTM Model with Combined Feature Vector

**Bidirectional LSTMs** process sequences bidirectionally, capturing richer contextual information. This architecture is effective for NLP tasks like **Natural Language Inference (NLI)**, where understanding sentence relationships requires considering both preceding and following contexts. As a non-transformer-based approach, ***BiLSTMs*** offer robust performance but generally lag behind ***Transformers*** due to architectural limitations. Enhancing their efficacy often requires task-specific heuristics. Accordingly, this work draws upon techniques from *"GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding"* (Wang *et al.*, 2019) to improve BiLSTM performance.

### Combined Feature Vectors

The proposed heuristic for enhancing ***NLI*** performance involves encoding each sentence independently using ***BiLSTMs***, producing vectors $u$ and $v$. A joint feature vector is then constructed comprising four components: the original vectors $u$ and $v$ , their absolute difference $|u-v|$, and their element-wise product $u \odot v$.

This design aims to better capture semantic differences between sentences, thereby enabling the model to make more informed and accurate predictions.

### Model Performance and Results

After training our ***BiLSTM*** model on the provided 24,432 pairs of premises and hypothesi and using another set of 6k of pair of sentences of validation the model's performance metrics on trial dataset were as follows:

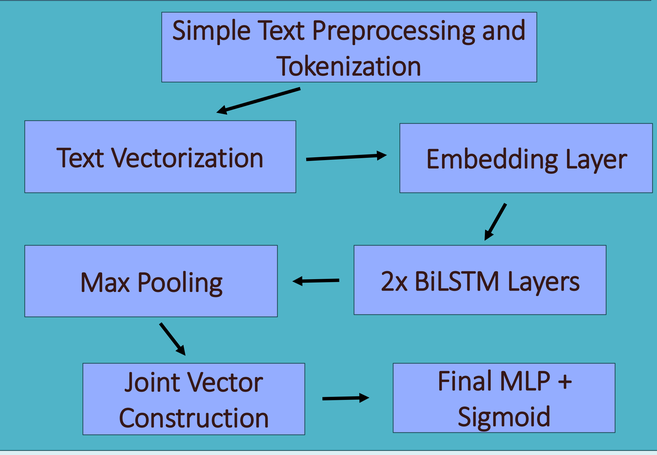
- *F1-score* = 0.8
- *Accuracy* = 0.8

This indicates that combined feature vectors leverage good relation extraction and therefore higher performance.

### Combined Feature Vectors

To enhance performance, ***two stacked BiLSTMs*** were employed, allowing the model to capture more complex hierarchical patterns and dependencies. Stacking multiple ***BiLSTM*** layers deepens the network, improving its ability to extract richer features from input sentences, which is particularly beneficial for the ***NLI*** task.

### Main architecture design




Overall, ***BiSTM*** performance with described heuristics can be considered good. However, there is still room for improvement, *i.e* it is possible to separately learn ***GLoVe*** embeddings for the provided text instead of learning embeddings at the same time with the model training or, simply, increase the number of parameters in order to increase model's performance.