ECE-GY 9123 Deep Learning Project Report

May 18, 2021

1 Project Group Members

Ashwin Shukla as13351@nyu.edu
Ayesha Nazneen Ahmed ana487@nyu.edu

2 Title

Finding the relationship between a given premise and a hypothesis.

3 Problem Statement

Training a Natural Language Inference (NLI)[1] model based on BERT that takes a multilingual set of a premise and a hypothesis and classifies the hypothesis as being an entailment, neutral or a contradiction to the premise. This is an ongoing Kaggle competition: Contradictory, My Dear Watson.

4 Description of Dataset

This dataset contains premise-hypothesis pairs in fifteen different languages, including: Arabic, Bulgarian, Chinese, German, Greek, English, Spanish, French, Hindi, Russian, Swahili, Thai, Turkish, Urdu, and Vietnamese.
The training dataset contains 12120 rows with a fairly even split of the 3 output classes, as shown in figure 1.

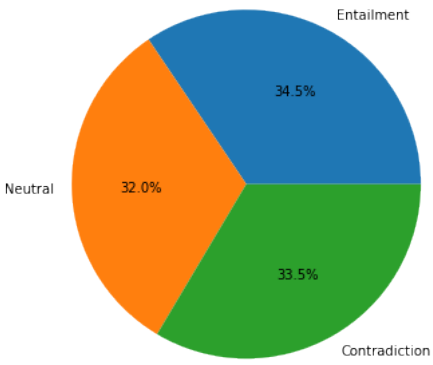

Figure 1: Distribution of classes

4.1 Data Files

train.csv This file contains 6 columns – the ID, premise, hypothesis, and label, as well as the language of the text and its two-letter abbreviation.

test.csv This file contains 5 columns – the ID, premise, hypothesis, language, and language abbreviation, without labels.

4.2 Example Case

Here's an example of each of the cases, taken from the Kaggle competition website.

Premise: *He came, he opened the door and I remember looking back and seeing the expression on his face, and I could tell that he was disappointed.*

Hypothesis 1: *Just by the look on his face when he came through the door I just knew that he was let down.*
Based on the information in the premise, we know that this an **entailment**.

Hypothesis 2: *He was trying not to make us feel guilty but we knew we had caused him trouble.*
We can't conclude anything about this hypothesis based on the information in the premise. Therefore, this relationship is **neutral**.

Hypothesis 3: *He was so excited and bursting with joy that he practically knocked the door off it's frame.*
Since this is the opposite of the premise, this relationship is a **contradiction**.

5 Description of Models

We considered the following models for this project, mainly because they are based on BERT and they are the latest in the field of NLP.
In all cases, the input will be pairs of sentences in the form of a premise and a hypothesis; and the output will be the classification of the relationship between the premise and hypothesis.
Class labels: 0 for entailment, 1 for neutral, 2 for contradiction.

5.1 M-BERT

M-BERT[2] is Google Research's multilingual variant of BERT. It has a 110K WordPiece[3] vocabulary in 104 languages. It also provides a sort of a shared representation across multiple languages.
M-BERT does not perform any normalisation on the input like, lower casing, accent stripping or Unicode normalisation. Therefore, we will follow the same rules while tokenising the data for this model. For the BERT model, the input is represented in the following format: `CLS` Premise `SEP` Hypothesis `SEP`.
The `CLS` and `SEP` are special tokens, where `CLS` is used in the beginning of a sequence for sentence-level classification while `SEP` separates the sentence pairs. We need to add a classification layer and retrain the model on the dataset in section 4.

5.2 XLM-RoBERTa

Proposed by Facebook in 2019 as a multilingual successor of RoBERTa[4], trained on 100 different languages. It uses a SentencePiece tokenizer (which is a language-independent tokenizer that treats input text as a sequence of Unicode characters) on a 250K vocabulary.
The input needs to be tokenized using the SentencePiece tokenizer. Tokenization process includes normalization and encoding of the input data such that it replaces the whitespace in front of every word with a special meta symbol "_" (U+2581) and tokenizes the input into an arbitrary subword sequence. We need to add a classification layer and retrain the model on the dataset in section 4.

5.3 DeBERTa

DeBERTa is a model proposed by Microsoft in 2021 in the paper "DeBERTa: Decoding-enhanced BERT with Disentangled Attention"[6]. It's built on top of BERT/RoBERTa with two improvements – disentangled attention and enhanced mask decoder; and provides a state-of-the-art solution for Natural Language Inference.
It uses a 128K vocabulary and a SentencePiece tokenizer. The input needs to be tokenized using

the SentencePiece tokenizer. Tokenization process includes normalization and encoding of the input data as explained in section 5.2. We need to add a classification layer and retrain the model on the dataset in section 4.

6 Description of Loss Function

We will be using the **sparse categorical cross entropy loss** function for this task because there are more than two label classes. The `SparseCategoricalCrossentropy` function from Keras expects the class labels to be provided as integers.

7 Data Exploration and Hyperparameter Selection

First, we have split the training dataset 80-20 as training and validation dataset.
During the data exploration stage, we plotted the histogram of the lengths of all input sequences, including premises and hypotheses, in order to determine the hyperparameter **max length**. The distribution is shown in figure 2. Since majority of the input sequences have less than 50 tokens,

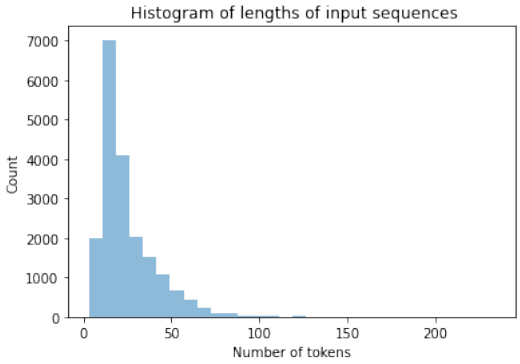

Figure 2: Histogram of tokens in input sequences

we set **max length** as 100, to account for a premise and hypothesis pair with 50 tokens each.
We retrained the model for **3 epochs** and restricted the **batch size** to 64. The **learning rate** is kept as 0.00001.

8 Training Algorithm

This is a high-level workflow of the training the models for the NLI task. For the final results, we will implement this workflow for M-BERT, XLM-RoBERTa and DeBERTa models. For the preliminary results, we have only implemented the M-BERT model with this workflow.

Algorithm 1: Workflow for the NLI Task

1. Load libraries and dependencies
2. Load data
3. Data exploration and analysis
4. Split training data
5. Implementing models
   - 5.1. Setup tokenizer
   - 5.2. Configure hyperparameters
   - 5.3. Encode input sequences
   - 5.4. Build Model and retrain
   - 5.5. Evaluate on validation data
   - 5.6. Plot results

9 Results

9.1 M-BERT

We implemented the M-BERT model with the workflow in section 8. We found that M-BERT does not perform as well as we expected (accuracy around 65%) against the validation data after training for 3 epochs with the hyperparameters specified in the section 7. Figures 3 and 4 show the plots for average loss and average accuracy on training vs. validation data.

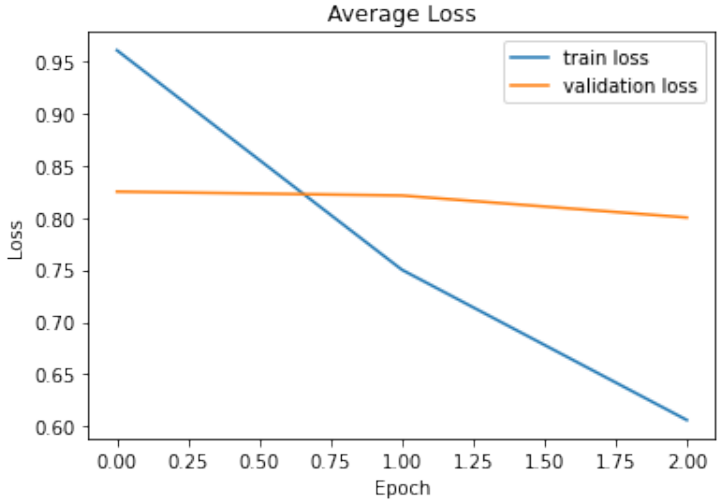

Figure 3: Average loss of training vs. validation data on M-BERT.

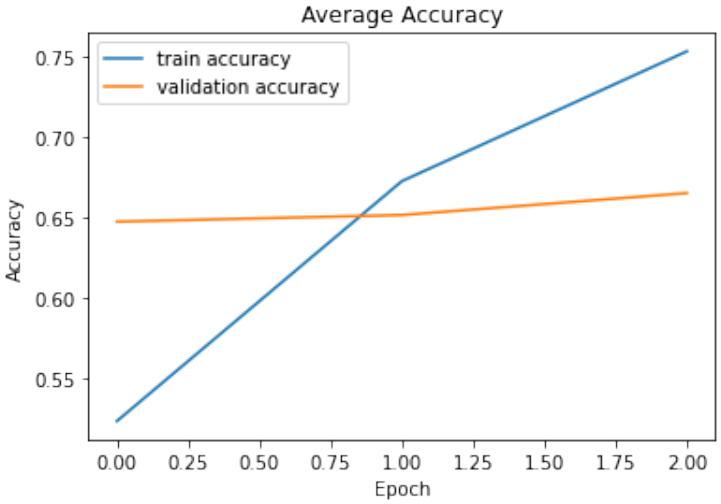

Figure 4: Average accuracy of training vs. validation data on M-BERT.

The confusion matrix in figure 5 of M-BERT model shows that it confuses the neutral classes quite a bit.

Also, figure 6 shows the accuracy of the model in different languages.

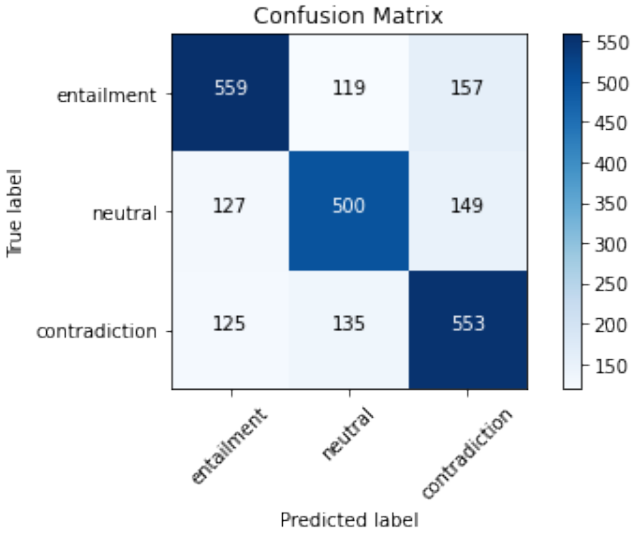

Figure 5: Confusion Matrix

```
Accuracy of Arabic is 68.0
Accuracy of Bulgarian is 61.0
Accuracy of Chinese is 68.0
Accuracy of English is 69.0
Accuracy of French is 68.0
Accuracy of German is 66.0
Accuracy of Greek is 60.0
Accuracy of Hindi is 66.0
Accuracy of Russian is 71.0
Accuracy of Spanish is 71.0
Accuracy of Swahili is 56.99999999999999
Accuracy of Thai is 45.0
Accuracy of Turkish is 56.99999999999999
Accuracy of Urdu is 56.00000000000001
Accuracy of Vietnamese is 72.0
```

Figure 6: Accuracy of M-BERT in All Languages

9.2 XLM-RoBERTa

While implementing XLM-RoBERTa, we faced a couple of challenges. First was that the RoBERTa model on HuggingFace that was fine-tuned for the XNLI dataset was too big for us to run on a free-tier Google Colab environment. To mitigate this problem, we developed and ran the notebook in the Kaggle environment that was facilitated by the competition.
Secondly, while implementing the model, we noticed that the class labels on which the model was trained were different from the class labels in the data. As shown in fig. 7, we can see that the class labels for `contradiction` and `entailment` are 0 and 2, whereas the class labels as shown in section 5 are flipped. So we had to swap the class labels in the training routine to achieve better accuracy.

The loss and accuracy plots for XLM-RoBERTa are shown in figures 8 and 9.

The confusion matrix of the model is shown in figure 10.

Figure 11 shows the accuracy of the model per language.

```
Model config XLMRobertaConfig {
  "architectures": [
    "XLMRobertaForSequenceClassification"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "eos_token_id": 2,
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 1024,
  "id2label": {
    "0": "contradiction",
    "1": "neutral",
    "2": "entailment"
  },
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "label2id": {
    "contradiction": 0,
    "entailment": 2,
    "neutral": 1
  },
```


Figure 7: Configuration of XLM-RoBERTa fine-tuned on cross-lingual data.

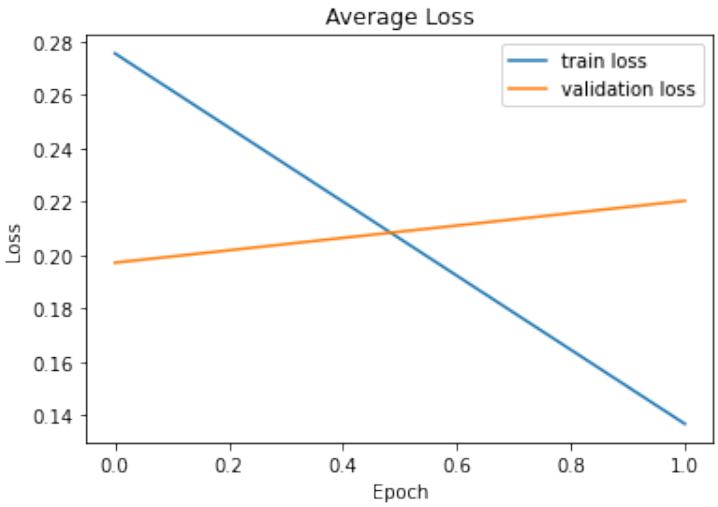

Figure 8: Average loss of training vs. validation data on XLM-RoBERTa.

9.3 DeBERTa

We wanted to evaluate DeBERTa for the NLI task as well, but unfortunately, no instance of the DeBERTa model exists yet that is fine-tuned for cross-lingual NLI. We simply did not have the resources to retrain the model ourselves. The dataset that we had was too small to fine-tune the model on. Therefore, we were not able to implement DeBERTa for this task.

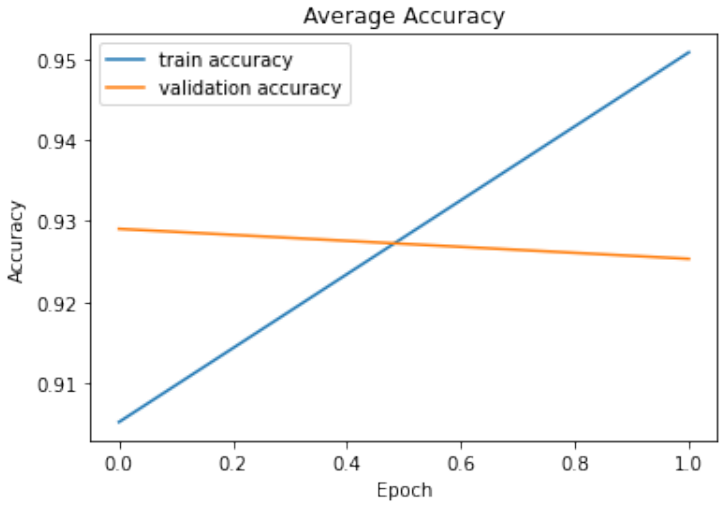

Figure 9: Average accuracy of training vs. validation data on XLM-RoBERTa.

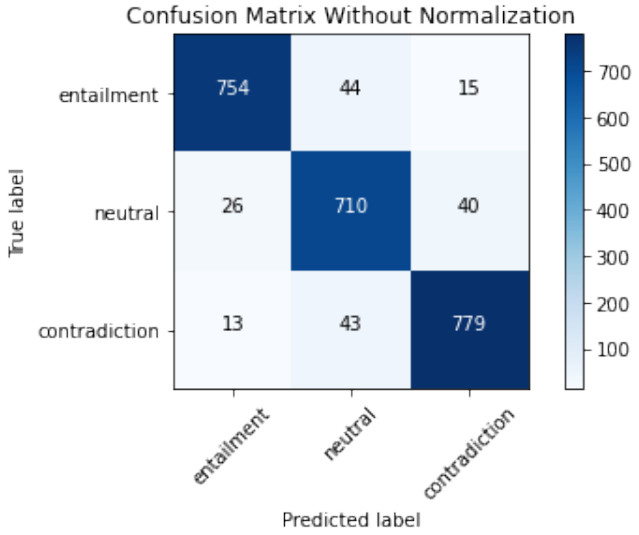

Figure 10: Confusion Matrix of XLM-RoBERTa.

```
Accuracy of the model per language
Accuracy of Arabic is 98.0
Accuracy of Bulgarian is 100.0
Accuracy of Chinese is 95.0
Accuracy of English is 88.0
Accuracy of French is 98.0
Accuracy of German is 100.0
Accuracy of Greek is 97.0
Accuracy of Hindi is 97.0
Accuracy of Russian is 100.0
Accuracy of Spanish is 100.0
Accuracy of Swahili is 96.0
Accuracy of Thai is 96.0
Accuracy of Turkish is 100.0
Accuracy of Urdu is 97.0
Accuracy of Vietnamese is 100.0
```

Figure 11: Accuracy of XLM-RoBERTa per language.

10 Conclusion

One of the major motivations for us to choose this problem statement was working on an NLP project, which was an area neither of us had worked on before.
We learnt about the general workflow of NLP tasks and got to study various tokenization techniques in detail, including WordPiece and SentencePiece [7]. We also got the opportunity to explore latest developments in the world of NLP and the architecture of different models proposed by leading researchers at Google, Facebook and Microsoft.

Natural Language Inference is one of the many new frontiers on which the development in Natural Language Processing in booming and is benefiting from the advancements in the field, be it new architectures, tokenization techniques or better and better datasets. We studied quite a few papers written in recent years on this topic – the latest of which published just a few months ago in January 2021 [6] – and applied two state-of-the-art solutions to this task. Out of which, we were able to get pretty good results in terms of accuracy across multiple languages.

We also evaluated the performance of M-BERT and XLM-RoBERTa on a number of criteria such as

1. Average accuracy,
2. Accuracy per language,
3. Precision and recall in each class,
4. impact of tokenization techniques in overall performance, etc.

As we can see from the results, Facebook's XLM-RoBERTa model, which is the state-of-the-art in cross-lingual natural language inference classification tasks, far exceeded the expectation and outperformed Google's M-BERT model with more than 95% average accuracy.

11 GitHub Repository

Here is the link to the GitHub repository for this project: https://github.com/shuklashwin/natural-language-inference.git

References

[1] Sawan Kumar and Partha Talukdar. *NILE : Natural Language Inference with Faithful Natural Language Explanations*. 2020. https://arxiv.org/abs/2005.12116

[2] Telmo Pires, Eva Schlinger and Dan Garrette. *How multilingual is Multilingual BERT?*. 2019. https://arxiv.org/pdf/1906.01502

[3] Yonghui Wu et al. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. https://arxiv.org/pdf/1609.08144v2

[4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer and Veselin Stoyanov. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. https://arxiv.org/pdf/1911.02116

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. https://arxiv.org/abs/1907.11692

[6] Pengcheng He, Xiaodong Liu, Jianfeng Gao and Weizhu Chen. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. https://arxiv.org/pdf/2006.03654

[7] Taku Kudo and John Richardson. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing*. 2018. https://arxiv.org/pdf/1808.06226