Multi-Label Legal Document Classification using Legal-BERT

NLP Assignment – ECHR Article Violation Prediction

Abstract

This report presents a complete workflow for building a multi-label classifier for predicting violated articles of the European Convention on Human Rights (ECHR) using the `lex_glue/ecthr_b` dataset. The study includes dataset understanding, preprocessing, exploratory data analysis, model selection, fine-tuning, evaluation, and comparison against a baseline. All experiments were executed in Google Colab using HuggingFace Transformers.

1 Introduction

The task aims to develop a multi-label classifier capable of predicting human-rights violations from factual paragraphs extracted from European Court of Human Rights (ECHR) case documents. Since each case may involve multiple articles, a multi-label approach is required. We fine-tune a domain-specific transformer model (Legal-BERT) and compare its performance against a pretrained baseline.

2 Dataset Description

The dataset used is the **ECTHR-B** subset of the **LEXGLUE** benchmark. It contains:

- **text** – factual case description
- **labels** – list of ECHR articles violated

Each label corresponds to human-rights articles such as:

- Article 2 – Right to Life

- Article 3 – Prohibition of Torture
- Article 6 – Right to Fair Trial
- Article 8 – Right to Private Life

With approximately 11k cases, the dataset is split into:

- 9000 training samples
- 1000 validation samples
- 1000 test samples

3 Approach Overview

The complete pipeline includes the following steps:

1. Dataset loading and inspection
2. Text preprocessing (cleaning, normalization)
3. Multi-label binarization using `MultiLabelBinarizer`
4. Tokenization using Legal-BERT tokenizer
5. Model selection:
    - Baseline: pretrained Legal-BERT without fine-tuning
    - Fine-tuned: Legal-BERT trained on ECHR data
6. Training using HuggingFace Trainer API
7. Evaluation and comparison
8. Exploratory Data Analysis (EDA)

4 Preprocessing Steps

The preprocessing pipeline includes:

4.1 Text Cleaning

- Lowercasing
- Removal of special characters
- Whitespace normalization

4.2 Handling Missing Values

Duplicate or empty text fields were filtered out.

4.3 Label Encoding

Since each case may have multiple violated articles, we use multi-hot encoding:

```
from sklearn.preprocessing import MultiLabelBinarizer

mlb = MultiLabelBinarizer()
mlb.fit(train_ds["labels"])

def encode_labels(example):
    return {"labels": mlb.transform([example["labels"]])
       [0].astype("float32")}
```

5 Tokenization

Legal-BERT tokenizer is applied with:

- max length = 512
- padding = “max_length”
- truncation enabled

```
def tokenize(batch):
    return tokenizer(batch["text"],
                     truncation=True,
                     padding="max_length",
                     max_length=512)
```

6 Model Architecture

6.1 Baseline Model

The baseline uses **Legal-BERT** (`nlpaueb/legal-bert-base-uncased`) without fine-tuning. Only the classifier head is randomly initialized.

6.2 Fine-Tuned Model

The same pretrained Legal-BERT is fine-tuned on the ECHR dataset with:

- Problem type: multi-label classification
- Loss: BCEWithLogitsLoss
- Activation: Sigmoid output layer

7 Training Setup

We use the HuggingFace Trainer with the following parameters:

- **batch size**: 32
- **learning rate**: 2e-5
- **epochs**: 10
- **evaluation**: per epoch
- **optimizer**: AdamW
- **scheduler**: linear warmup

```
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    load_best_model_at_end=True
)
```

8 Evaluation Metrics

Since this is a multi-label problem, we use:

- Micro-F1 Score
- Macro-F1 Score
- Precision and Recall

9 EDA Findings

9.1 Text Length Analysis

Most factual paragraphs are between 50–250 words.

9.2 Label Distribution

Articles 6, 8, and 3 occur most frequently, reflecting real-world ECHR case trends. Rare labels show heavy class imbalance.

9.3 Label Co-occurrence

Certain articles (e.g., Article 6 & 8) appear together frequently, suggesting legal dependency patterns.

10 Results

10.1 Baseline Results

The baseline (untrained classifier head) achieves:

- Micro-F1 0.22

10.2 Fine-Tuned Model Results

After training:

- Micro-F1 0.72

Conclusion: Fine-tuning significantly improves performance.

11 Challenges and Solutions

11.1 Label Shape Mismatch

Problem: Labels had inconsistent lengths (e.g., [6] vs [6,8]). **Solution:** Multi-hot encoding using `MultiLabelBinarizer`.

11.2 Wrong Label Dtype

Problem: Labels were int64 but BCEWithLogitsLoss requires float. **Solution:** Cast labels to `float32`.

11.3 Tokenizer Overwriting Text

Problem: Applying label encoding after tokenization removed original labels. **Solution:** Encode labels **before** tokenization.

12 Conclusion

This project demonstrates that domain-specific fine-tuning of Legal-BERT offers substantial performance gains on multi-label legal classification tasks. The ECHR dataset's complex multi-label structure presents challenges such as label imbalance and dependency, but transformer-based models handle these effectively.