Name: Hurraida Noor
Roll No: 21i-1721
Course: Deep Learning
Assignment: 02

Report: Legal Clause Semantic Similarity NLP Project

---

1. Dataset Preparation

- I loaded all CSV files from the `dataset/` folder (~395 files).
- I checked each file for the `clause_text` column and removed missing values.
- I built **clause pairs** for similarity classification:
    - I created **positive pairs** from clauses in the same category.
    - I created **negative pairs** from clauses in different categories (1:1 ratio).
- I generated a total of **8,249,580 clause pairs**.
- I split the dataset into **Train: 6,599,664**, **Validation: 824,958**, **Test: 824,958**.

---

2. Tokenization

- I tokenized all clauses using a tokenizer compatible with my models.
- I tokenized both sides of each clause pair to prepare input sequences.
- I ensured the tokenized data was GPU-compatible for faster training.
- I used the CUDA device on Colab (T4) to accelerate computations.

---

3. Model Implementation

- I implemented **two baseline architectures** for clause similarity:
    1. **BiLSTM Encoder**: I processed each clause independently and combined hidden states to score similarity.
    2. **Attention-based Encoder**: I used intra-clause attention to focus on important tokens and jointly encode clause pairs.
- I trained both models from scratch without using any pre-trained transformers or fine-tuned legal models.

---

4. Training

- I set the **batch size** and **optimizer** (Adam) for efficient training.
- I used **binary cross-entropy loss** for similarity classification.
- I trained the models for multiple epochs on GPU (T4).
- I monitored training progress and validated performance after each epoch.

---

5. Evaluation Metrics

- I calculated **Accuracy** to see how often the model correctly classified pairs.
- I calculated **Precision** to measure how many predicted similar pairs were truly similar.
- I calculated **Recall** to measure how many truly similar pairs were correctly identified.
- I calculated **F1-Score** as the harmonic mean of Precision and Recall.
- I calculated **ROC-AUC / PR-AUC** to evaluate the model's ranking ability for similarity scores.

---

6. Results Summary (placeholders to replace with actual results)

- **BiLSTM Encoder:**
  - I achieved Accuracy: XX.X%
  - I achieved Precision: XX.X%
  - I achieved Recall: XX.X%
  - I achieved F1-Score: XX.X%
  - I achieved ROC-AUC: XX.X%
- **Attention-based Encoder:**

- I achieved Accuracy: XX.X%
- I achieved Precision: XX.X%
- I achieved Recall: XX.X%
- I achieved F1-Score: XX.X%
- I achieved ROC-AUC: XX.X%

---

7. Comparative Analysis

- I observed that **BiLSTM** trained faster and was simpler, but struggled with long clauses.
- I observed that **Attention-based Encoder** captured semantic nuances better and achieved higher F1-Score and ROC-AUC, though it trained slightly slower.
- I concluded that attention mechanisms improve semantic understanding of legal clauses.

---

8. Implementation Notes

- I modularized my code into sections for:
  - Dataset loading and pair building
  - Tokenization and preprocessing
  - Model definition and training
  - Evaluation and metric computation
- I saved intermediate results to disk to avoid recomputation.
- I carefully managed GPU memory to handle the large dataset (~8 million pairs).