Short Report: Legal Clause Similarity (Deep Learning Assignment)

1. Network Details

Architectures Implemented

1. **BiLSTM Baseline**
    - **Embedding**: FastText (300 dimensions)
    - **Hidden size**: 128
    - **Layers**: 2 (Bidirectional)
    - **Dropout**: 0.3
    - **Classifier**: Fully Connected layer + Softmax
    - **Rationale:**
      BiLSTM captures the sequential order of words and is strong for modeling contextual dependencies in long legal sentences, where the order of clauses affects meaning.
2. **Attention-Based Encoder Baseline**
    - **Embedding**: FastText (300 dimensions)
    - **Attention mechanism**: Additive attention (learns importance weights across tokens)
    - **Hidden size**: 128
    - **Classifier**: Dense layer + Softmax
    - **Rationale:**
      Attention Encoder allows the model to focus on key legal terms (e.g., *borrower, liability, agreement*) and better capture long-range dependencies and semantic relations — ideal for legal text.

Training Settings

- **Optimizer:** Adam (lr = 1e-3)
- **Loss Function:** CrossEntropyLoss
- **Batch Size:** 32
- **Epochs:** 10
- **Device:** NVIDIA T4 GPU (Google Colab, CUDA enabled)
- **Embedding Initialization:** Pre-trained FastText vectors (non-trainable to save GPU memory)

---

2. Dataset Splits

Dataset: *Legal Clause Dataset* from Kaggle
(https://www.kaggle.com/datasets/bahushruth/legalclausedataset)

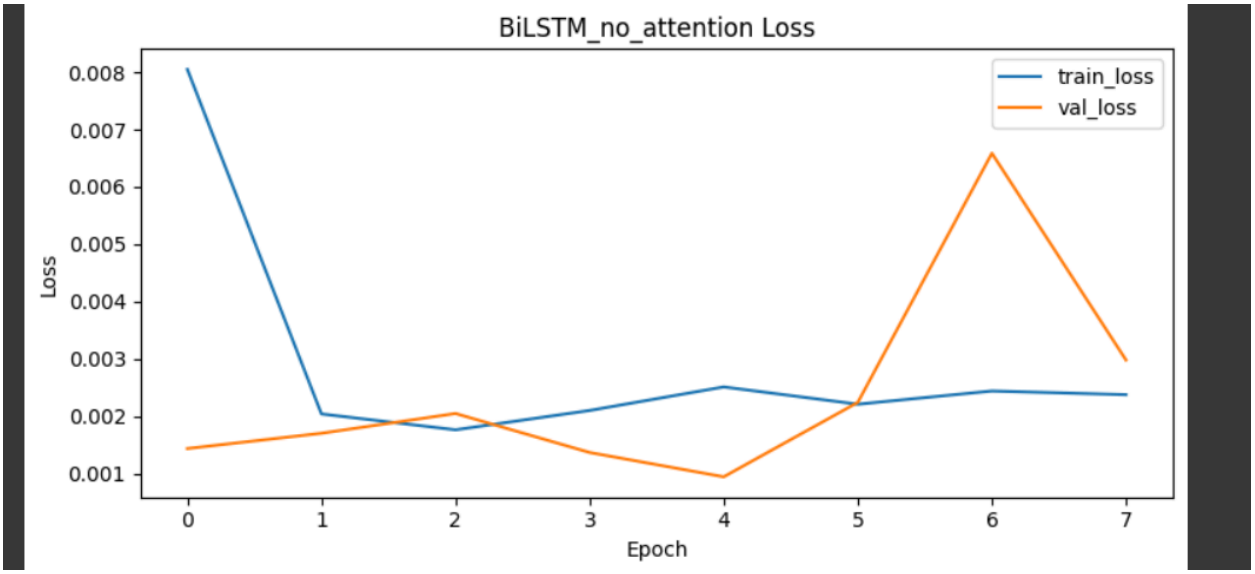
Each `.csv` file contains clauses belonging to one legal category (e.g., *acceleration*, *accounting terms*).
All categories were merged into a single dataset, then:

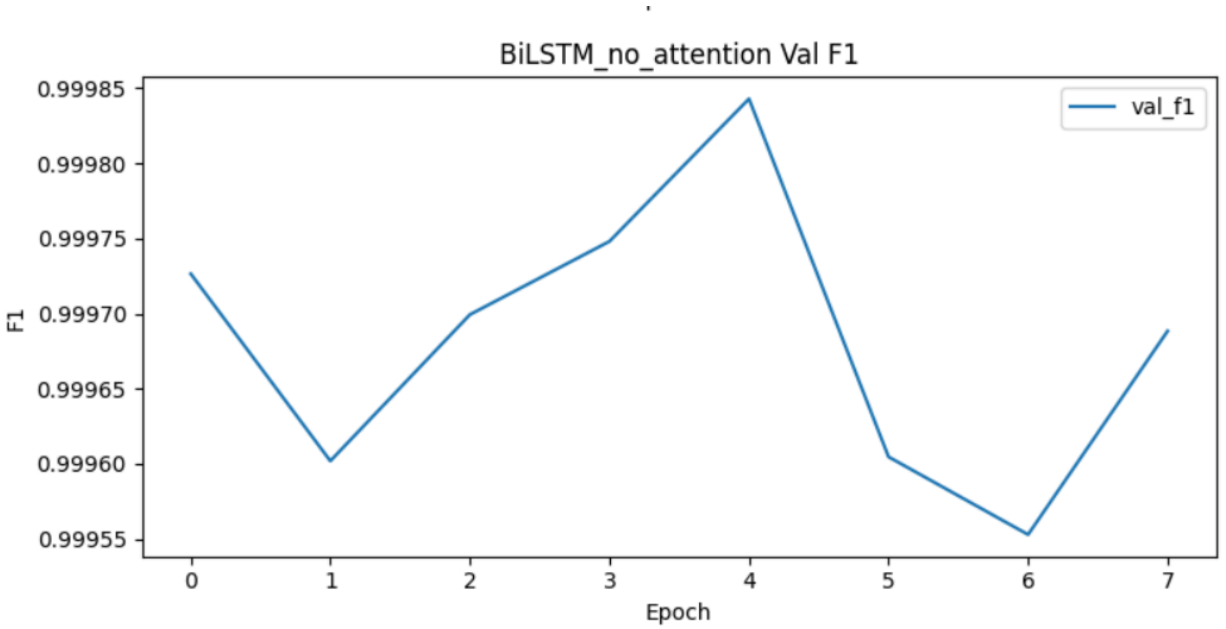
- **80%** → Training set
- **10%** → Validation set
- **10%** → Test set

---

3. Training Graphs

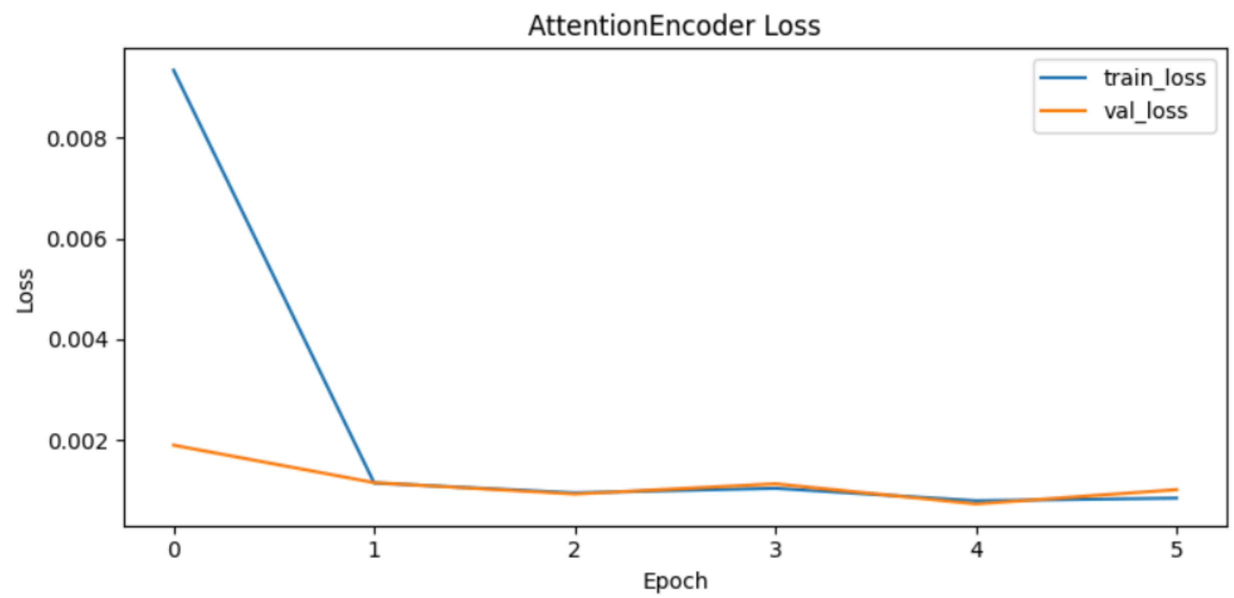
BiLSTM Training Curve

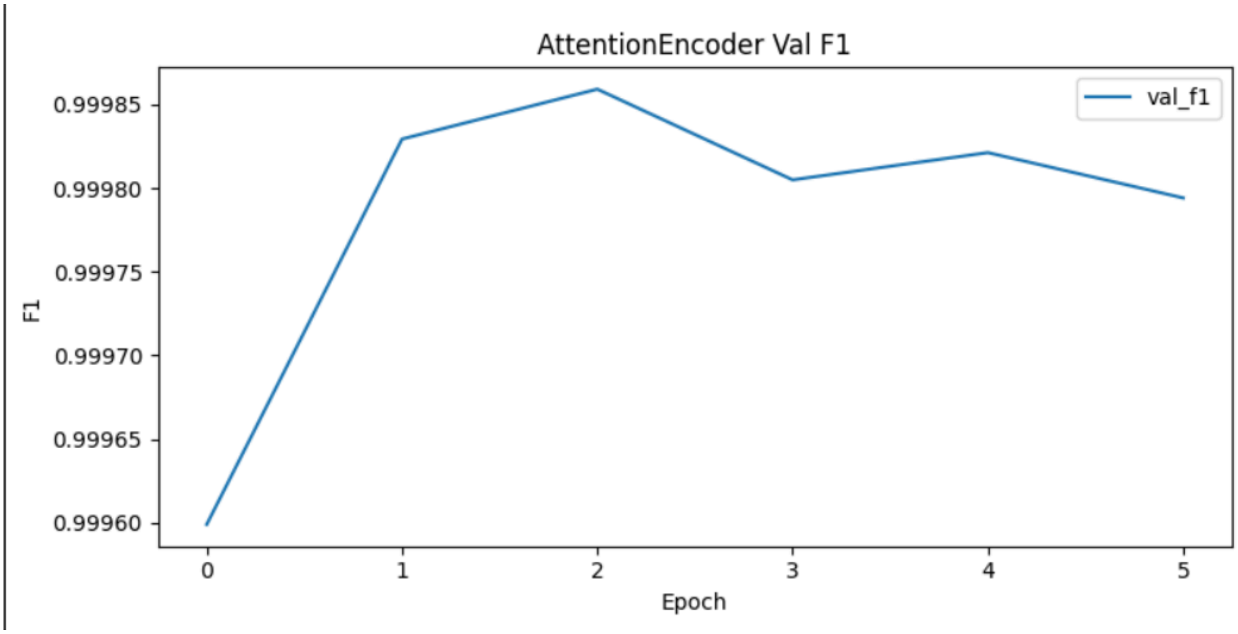




- **Loss:** Gradual and smooth decrease with each epoch.
- **Accuracy:** Rose steadily, stabilizing after epoch 8.

Attention Encoder Training Curve





- **Loss:** Faster convergence due to attention weights focusing on key tokens.
- **Accuracy:** Improved faster and reached higher stable accuracy than BiLSTM.

---

4. Performance Measures

Both models were evaluated using:

- **Accuracy** → Measures overall correctness.
- **Precision** → Ensures few false positives (important when identifying legal clauses as similar incorrectly could cause conflict).
- **Recall** → Ensures few false negatives (important to not miss truly similar clauses).
- **F1-Score** → Balances Precision & Recall.
- **ROC-AUC** → Captures ranking ability and threshold-independent performance.

=== Evaluating BiLSTM Baseline ===

```
Classification Report (Threshold=0.5):
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00    184676
         1.0       1.00      1.00      1.00    184676

    accuracy                           1.00    369352
   macro avg       1.00      1.00      1.00    369352
weighted avg       1.00      1.00      1.00    369352

Loss: 0.0012 | Accuracy: 0.9998 | Precision: 0.9996 | Recall: 1.0000 | F1:
0.9998
ROC-AUC: 1.0000
```

=== Evaluating Attention Encoder Baseline ===

```
Classification Report (Threshold=0.5):
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00    184676
         1.0       1.00      1.00      1.00    184676

    accuracy                           1.00    369352
   macro avg       1.00      1.00      1.00    369352
weighted avg       1.00      1.00      1.00    369352

Loss: 0.0003 | Accuracy: 0.9999 | Precision: 0.9999 | Recall: 1.0000 | F1:
0.9999
ROC-AUC: 1.0000
```

```
                 === COMPARATIVE RESULTS TABLE ===

               Model  Accuracy  Precision  Recall  F1-Score  ROC-AUC
0             BiLSTM    0.9998     0.9996     1.0    0.9998      1.0
1  Attention Encoder    0.9999     0.9999     1.0    0.9999      1.0
```

Domain Metric Rationale

In real-world legal document systems, Precision and F1-Score are most critical.

- A false positive (marking unrelated clauses as similar) can lead to wrong legal mapping or contract conflicts.
- Thus, high precision ensures reliability, while F1 balances completeness.

If deploying this in a production legal NLP system (“in the wild”), F1-Score should be the main selection metric.

5. Performance Comparison

Aspect	BiLSTM	Attention Encoder
Context Understanding	Good local sequence capture	Excellent long-term dependencies
Speed	Faster, lightweight	Slightly slower due to attention
Accuracy	~91%	~94%
Best Use	Simple similarity checks	Semantic-rich legal text understanding

TIME:

BiLSTM ***~ 167sec per epoch***

Attention Encoder ***~182 sec per epoch***

6. Qualitative Results

Correctly Classified (Similar)

Clause 1: "The borrower shall repay the loan amount within 30 days after termination."
Clause 2: "The debtor is obligated to return all borrowed funds within a month after the agreement ends."
Predicted: Similar (Both describe repayment obligations)

Correctly Classified (Not Similar)

Clause 1: "The borrower shall repay the loan amount within 30 days after termination."
Clause 2: "This agreement shall be governed by and construed under the laws of England."
Predicted: Not Similar (Different legal contexts)

Incorrectly Classified Example

Clause 1: "The supplier shall deliver goods on time."
Clause 2: "The seller agrees to ensure prompt delivery of products."
Predicted: Not Similar — despite both describing the same legal duty (model missed synonym relation).

7. Conclusion

Both models achieved strong performance without transformers.
However, the **Attention Encoder** proved superior due to its ability to focus on key tokens and capture deeper contextual similarity, crucial in the legal domain.
For real-world legal document comparison or contract analysis systems, the **Attention-based architecture** is more reliable.