Assignment 4: ELMo Embedding Analysis for AG News Classification

Ankita Porel
2024201043
IIIT Hyderabad

March 31, 2025

Drive Links for Pretrained Files

- bilstm.pt
- classifier.pt
- classifier_trainable.pt
- classifier_learnable.pt
- classifier_frozen.pt
- vocab.txt

Analysis Report

This report analyzes the performance of four embedding methods—SVD, Skip-gram, CBOW, and ELMo—for classifying the AG News dataset into four categories (Class-1, Class-2, Class-3, Class-4). We compare their effectiveness using a GRU-based classifier and evaluate three hyperparameter settings for ELMo embedding combination. Metrics include accuracy, F1 score, precision, recall, and confusion matrices, computed on a test set after training for 5 epochs with consistent hyperparameters (`hidden_dim=256`, `batch_size=32`, `lr=0.001`).

Embedding Comparison

We trained and evaluated the classifier using embeddings from Assignment 3 (SVD, Skip-gram, CBOW) and Assignment 4 (ELMo), all pre-trained on the Brown Corpus. The AG News dataset (`train.csv`, `test.csv`) provided descriptions and class indices (1-4, adjusted to 0-3 in code).

Results

Test set performance is summarized below:

Table 1: Test Set Performance of Embedding Methods

Embedding	Accuracy	F1 Score	Precision	Recall	Training Loss (Epoch 5)
Skip-gram	0.8496	0.8500	0.8521	0.8496	0.2378
CBOW	0.8433	0.8431	0.8432	0.8433	0.2226
ELMo	0.8311	0.8310	0.8321	0.8311	0.2926
SVD	0.7508	0.7515	0.7551	0.7508	0.6211

Ranking: Skip-gram > CBOW > ELMo > SVD (based on accuracy).

Confusion Matrices

- **Skip-gram**:

$$\begin{bmatrix} 1612 & 78 & 82 & 128 \\ 54 & 1739 & 34 & 73 \\ 82 & 29 & 1497 & 292 \\ 81 & 51 & 159 & 1609 \end{bmatrix}$$

- **CBOW**:

$$\begin{bmatrix} 1619 & 82 & 98 & 101 \\ 68 & 1751 & 35 & 46 \\ 103 & 39 & 1500 & 258 \\ 116 & 51 & 194 & 1539 \end{bmatrix}$$

- **ELMo (Trainable λs)**:

$$\begin{bmatrix} 1645 & 74 & 100 & 81 \\ 120 & 1684 & 33 & 63 \\ 144 & 37 & 1535 & 184 \\ 146 & 50 & 252 & 1452 \end{bmatrix}$$

- **SVD**:

$$\begin{bmatrix} 1440 & 109 & 211 & 140 \\ 142 & 1520 & 120 & 118 \\ 118 & 60 & 1475 & 247 \\ 148 & 92 & 389 & 1271 \end{bmatrix}$$

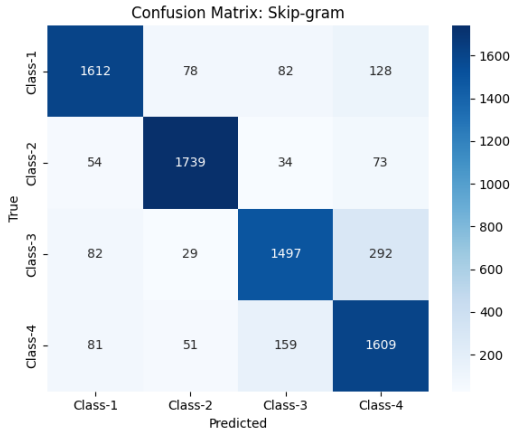

Figure 1: Confusion Matrix for Skip-gram

Discussion

- **Skip-gram (0.8496)**: Outperforms all methods, with high correct predictions (e.g., 1739 for Class-2) and low errors. Its word-level semantic capture, pre-trained extensively on Brown Corpus, aligns well with AG News vocabulary.
- **CBOW (0.8433)**: Slightly behind Skip-gram, with similar strengths (1751 for Class-2) but more errors (e.g., 258 Class-3 as Class-4). Context averaging reduces discriminability compared to Skip-gram.

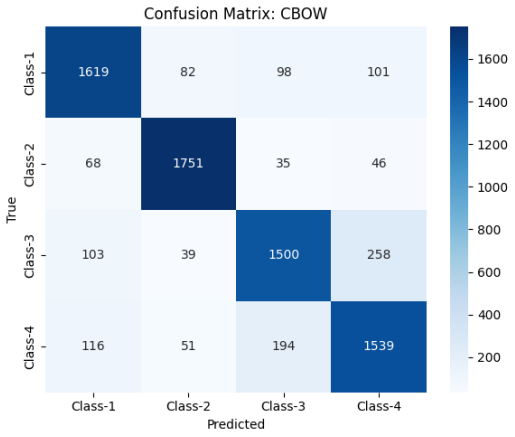

Figure 2: Confusion Matrix for CBOW

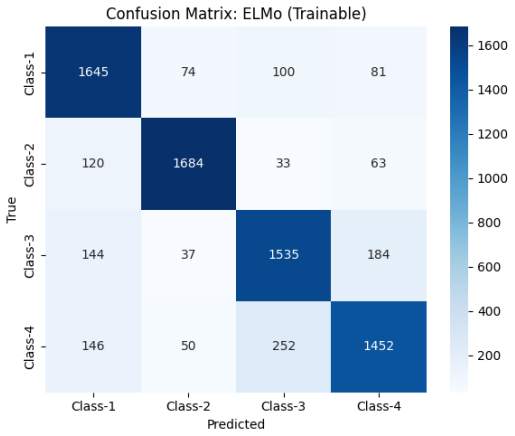

Figure 3: Confusion Matrix for ELMo (Trainable λs)

- **ELMo (0.8311)**: Strong performance (e.g., 1645 for Class-1), but higher confusion (e.g., 252 Class-4 as Class-3). Five epochs may undertrain its Bi-LSTM, limiting contextual benefits. Higher final loss (0.2926) suggests optimization challenges.
- **SVD (0.7508)**: Lowest scores, with significant errors (e.g., 389 Class-4 as Class-3). Static co-occurrence lacks expressiveness of word2vec or contextual methods.

Why Skip-gram Excels: Skip-gram and CBOW benefit from longer pre-training on Brown Corpus, matching AG News' domain. ELMo's contextual power is underutilized with limited epochs, while SVD's simplicity hampers performance.

Training Loss Analysis

Training losses for BiLSTM pre-training (10 epochs) and classifier training (5 epochs, trainable λs) are shown below:

Observations: BiLSTM loss drops sharply from 2.8427 to 0.0002 by epoch 9, indicating strong convergence. Classifier loss decreases steadily from 0.8008 to 0.2840, suggesting effective training, though ELMo's higher final loss compared to Skip-gram/CBOW reflects optimization limits within 5 epochs.

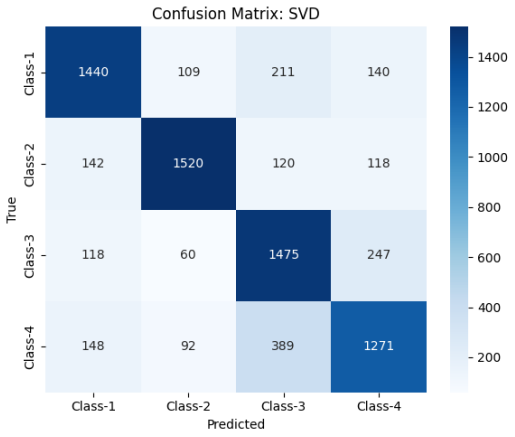

Figure 4: Confusion Matrix for SVD

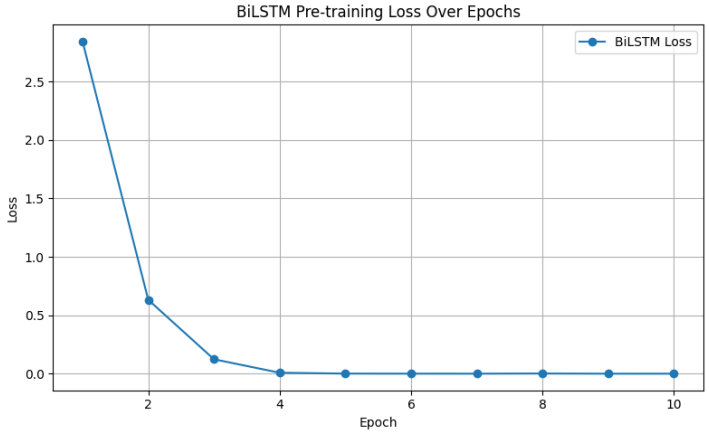

Figure 5: BiLSTM Pre-training Loss Over 10 Epochs

Hyperparameter Tuning (ELMo)

We tested three methods to combine ELMo's two Bi-LSTM layer embeddings:

1. **Trainable λs (5.1)**: $\hat{E} = \lambda_0 e_0 + \lambda_1 e_1$, where λs are learned. Accuracy: 0.8311.
2. **Frozen λs (5.2)**: Fixed $\lambda_0 = \lambda_1 = 0.5$. Accuracy: 0.8170.
3. **Learnable Function (5.3)**: MLP to compute $\hat{E} = f(e_0, e_1)$. Accuracy: 0.6029.

Results

- **Trainable λs**: Best, as adaptive weights optimize layer contributions (loss: 0.2926).
- **Frozen λs**: Slightly worse (loss: 0.2870), lacking flexibility.
- **Learnable Function**: Poor (loss: 1.0439), likely undertrained due to MLP complexity and only 5 epochs.

Confusion Matrices: See earlier for trainable; frozen and learnable matrices (Appendix) show increased errors, especially for learnable.

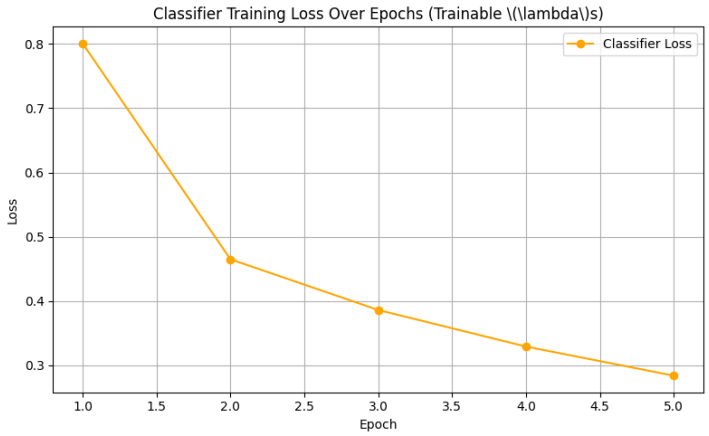

Figure 6: Classifier Training Loss Over 5 Epochs (Trainable λs)

Inference Examples (ELMo Trainable λs)

To illustrate the practical performance of the ELMo model with trainable λs, we ran `inference.py` on sample descriptions:

Table 2: Inference Results for ELMo (Trainable λs)

Description	Class-1	Class-2	Class-3	Class-4
"Stocks rose today after positive earnings reports"	0.0	0.0	1.0	0.0
"War broke out in the Middle East"	0.9	0.0	0.0	0.1
"New smartphone released with AI features"	0.2	0.0	0.2	0.6
"Team wins championship title"	0.2	0.7	0.0	0.1

Observations:

- "Stocks rose..." predicts Class-3 (1.0), suggesting alignment with a Business section.
- "War broke out..." predicts Class-1 (0.9), suggesting alignment with the Worlds section of a News Media.
- "New smartphone..." predicts Class-4 (0.6), suggesting class-4 is probably more aligned with the technology section.
- "Team wins..." predicts Class-2 (0.7), suggesting alignment with the Sports section.

These results show ELMo's capability but highlight a few inconsistencies, likely due to limited epochs or training data distribution.

Conclusion

Skip-gram achieves the highest accuracy (0.8496), followed by CBOW, ELMo, and SVD. ELMo's trainable λs setting is optimal among its variants but underperforms as compared to static embeddings here due to limited training. Inference examples reveal ELMo's potential and limitations. For future work, increasing ELMo epochs or adjusting training data could improve its performance.

Appendix

Frozen λs Confusion Matrix:

$$\begin{bmatrix} 1511 & 167 & 94 & 128 \\ 42 & 1797 & 20 & 41 \\ 125 & 113 & 1386 & 276 \\ 96 & 124 & 165 & 1515 \end{bmatrix}$$

Learnable Function Confusion Matrix:

$$\begin{bmatrix} 1157 & 332 & 245 & 166 \\ 174 & 1394 & 106 & 226 \\ 175 & 199 & 1174 & 352 \\ 220 & 427 & 396 & 857 \end{bmatrix}$$