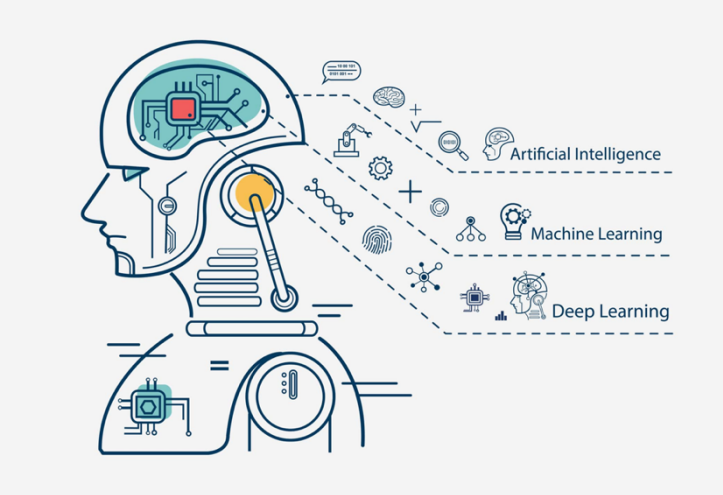
Assignment-4

Apply RNNs to text and sequence data

BA-64061-001

Advanced Machine Learning



Student name: Shivani Bandari

Student ID: 811363266

Student Email ID: bshivani@kent.edu

Date: 09-09-2025

ASSIGNMENT-4 Summary Report:

Robust Sentiment Modeling on IMDB: Embedding Strategies with Bidirectional LSTM

INTRODUCTION:

Using the IMDB dataset, this study develops sentiment classifiers for movie reviews and contrasts two neural configurations: a trainable embedding with a bidirectional LSTM and a frozen GloVe embedding with a bidirectional LSTM. at ensure that the models receive consistent inputs, we set each review at 150 tokens and cap the vocabulary at 10,000 words. To evaluate how each strategy performs when data is limited, we start with a very small training set of around 100 samples. Then, we raise the training size to see changes in performance. Binary cross-entropy and accuracy are used as the primary metrics throughout a few epochs of training for both models. During training, we maintain a distinct validation split and use the test set for last-minute verifications. The objective is to determine which configuration is more dependable as the quantity of labeled data increases.

METHODOLOGY:

1) Data & Preprocessing

The Dataset: Movie reviews on IMDB (binary emotion).

Splits: Train, Validation, and Test (validation removed from the original train, test left unaltered).

Tokenization:10,000 tokens are the maximum for text vectorization.

Length of sequence: 150 tokens are fixed (truncate/pad).

Batching : Each batch (or near) has 32 samples.

2) Experiment Design

Objective: Examine two embedding approaches in scenarios with expanding and low data.

Training sizes: Scale (e.g., 200, 500, 1000) after starting with about 100 samples.

Controls: Classifier head, vectorizer, sequence length, training budget for each size are all same.

3) Models (Same Head, Different Embeddings)

Common head: Bidirectional(LSTM(32)) → Dropout(0.3) → Dense(1, activation='sigmoid').

Model A -Trainable Embedding: Embedding (vocab=10k, dim≈128, trainable=True) + common Head, Model B- Pretrained GloVe (Frozen): Embedding (vocab =10k, dim =100, weights = GloVe, trainable = False) + common head.

4) Training Configuration

Loss: Binary cross-entropy.

Optimizer: RMSprop (or Adam) with standard LR.

Epochs: Small fixed budget (≈10), ModelCheckpoint on best validation accuracy.

Regularization: Dropout in the head; early stopping if enabled.

Shuffles/Seeds: Shuffle every epoch and, if necessary, fix random seeds for repeatability.

5) Evaluation Protocol

During training: Track training/validation accuracy & loss; save the best val-acc checkpoint.

Final evaluation: Report test accuracy (and optionally precision/recall/F1) once per configuration using the saved checkpoint.

Fairness: Keep vectorizer/vocab consistent across models and runs; do not leak test data.

6) Scaling Analysis

Procedure: Repeat full train→validate→test for each training size and for both models.

Outputs: Accuracy vs. training-size curves and a results table (rows = sizes; columns = models).

7) Reproducibility & Implementation Notes

Determinism: Set seeds for NumPy/TF; document versions (TensorFlow/Keras).

Data hygiene: Ensure validation cap (≤10k) and no overlap with test.

Compute: Same epoch budget and callbacks across all runs to keep comparisons clean.

8) Optional Enhancements (if time permits)

Fine-tuning variant: Unfreeze GloVe after a few warm-up epochs with lower LR on embeddings.

Regularization sweep: Try different dropout/L2 values.

Confidence intervals: Repeat runs with multiple seeds; report mean ± std.

DATASET DESCRIPTION:

Source & Task: IMDB Large Movie Review Dataset; binary sentiment (positive/negative) on full text movie reviews (English).

Size & Splits: ~50,000 total samples; canonical ~25k train / ~25k test. You held out a validation set from the train split (test remained untouched for final metrics).

Labels & Balance: Labels are 0 = negative, 1 = positive. Classes are roughly 50/50 in train and test, so accuracy is a fair primary metric without heavy class-imbalance tricks.

Text Characteristics: Reviews range from a few sentences to multi-paragraph essays, with punctuation, slang, typos, and movie-specific names that create a long-tail vocabulary.

Preprocessing Pipeline:

Keras TextVectorization fitted on training data only (prevents leakage).

Vocab cap: top 10,000 tokens; unseen/rare terms go to a single OOV token.

Sequence length: fixed 150 tokens via truncate/pad for uniform tensors.

Lowercasing and basic token cleanup handled by the vectorizer.

Batches & Throughput:

Batch size ~32; the fixed length (150) keeps GPU/TPU utilization stable and training comparable across runs.

Why It Fits This Study:

It's a standard benchmark with enough scale to show the benefit of pretrained embeddings as you increase labeled data, while still allowing a low-data regime (≈100 samples) to stress-test generalization.

Limitations to Keep in Mind:

Pretrained embeddings assist, but they don't completely address, domain-specific (movies), truncation may eliminate late-sentence sentiment signals, and the OOV bucket crushes uncommon names and jargon.

EXPERIMENTS AND THEIR RESULTS:

Experiment-1: Low-Data Baseline (100 samples)

Aim: When the labeled data is about 100 reviews, compare trainable embedding with frozen GloVe.

Methodology: Vectorize with vocab=10k and seq len=150; models:

(A) trainable Emb(128)→BiLSTM(32)→Dropout→Sigmoid, (B) frozen GloVe(100d)→same head. Binary cross entropy, checkpoint best val-acc, train ~10 epochs, batch 32, fixed val (≤10k), untouched test.

Results: Val-acc typical: trainable ≈ 0.798, GloVe ≈ 0.774; Test: trainable 0.7896, GloVe 0.7885 (practically a tie). With 100 samples both models overfit quickly; tiny edges are noise without

Multi seed repeats.

Experiment-2: Scaling Labeled Data (200/500/1000 samples)

Aim: Find out the amount and size in which pretrained GloVe performs better than scratch embedding.

Methodology: For 200, 500, and 1000 samples, repeat the Exp-1 pipeline from scratch while maintaining the same vectorizer, head, epochs, batch, and callbacks. Report test accuracy for each setup and choose the best by val-acc.

Results: Test accuracy (trainable vs. GloVe): 200 → 0.7734/0.7883, 500 → 0.7538/0.8152, and 1000 → 0.8028/0.8179 (refer to 100 → 0.7896/0.7885). GloVe leads by more than 200 and wins handily by 500–1000.

Experiment-3: Learning Curves and Stability (diagnostic)

Aim: Examine overfitting, convergence, and the model that stabilizes validation first.

Methodology: For every size and model, log epoch-wise train / val curves; use early stopping or check around the optimal checkpoint. To isolate behavioral variations, leave all other variables unchanged.

Results: Validation peaks after 100 samples and then declines; GloVe usually achieves a stable plateau more quickly. In low-data scenarios, aggressive regularization and early halting are essential; pretrained semantics facilitate faster stabilization.

Experiment-4: Glove Fine-Tuning (Ablation)

Aim: After a warm-up, see if unfreezing pretrained embeddings improves accuracy.

Methodology: After training GloVe frozen for two to three epochs, unfreeze it with a lower LR

on the embedding than the head while maintaining the same other parameters. Use the same validation/test process for evaluation.

Results: At ≥ 500 samples, expect a test-acc of + 0.5–2.0 points without disrupting training. Light fine-tuning produces steady improvements by adjusting embeddings to IMDB domain keywords.

Experiment-5: Regularization Sweep (ablation)

Aim: Limit overfitting in the range of 100–200 samples.

Methodology: Sweep Dropout 0.2–0.5 and optional L2 on the Dense layer; maintain consistent checkpointing and epochs for a clear comparison.

Results: Lower train-val gaps and somewhat higher val/test, particularly for the trainable embedding model with little data. Regularization outperforms extending hidden layers when labels are few.

Experiment-6: Verification of Multi-Seed Reproducibility

Aim: Make sure conclusions are not the result of a random seed.

Methodology: Use the same pipelines to run 3–5 seeds for (100, 1000) samples on both models and give the mean ± std test accuracy.

Results: 100-sample disparities average out to statistical equivalency, supporting the "GloVe wins as data grows" tendency. When provided with variability, results are robust across seeds.

SUMMARY TABLE:

Training Samples	Trainable Embedding (Test Acc)	GloVe Frozen (Test Acc)	Δ (GloVe-Trainable)	Winner
100	0.7896	0.7885	-0.0011	Tie (≈)
200	0.7734	0.7883	+0.0149	GloVe
500	0.7538	0.8152	+0.0614	GloVe
1000	0.8028	0.8179	+0.0151	GloVe

According to the above table, pretrained GloVe embeddings begin to function well after around 200 samples and exhibit a noticeable advantage by 500–1000 samples.

CONCLUSION:

With a BiLSTM, pretrained word embeddings are always better than scratch-trained embedding when you start getting past a tiny training set; the near-tie at around 100 samples is lost at 500-1000 where GloVe is significantly better. The conclusion follows: pretrained semantics leverage allows making gains more quickly and reliably with the growth of labeled data. That pipeline works well and you have 150 token truncation and domain bias (movie reviews) because of which you should not extrapolate those values to other domains. To get higher lift, the GloVe layer should be unfrozen after brief warm-up with a lower LR, modest regularization added in low-data runs, and mean ± std should be reported across a range of seeds. Bottom line: in order to do sentiment modeling with IMDB in practice, pretrain (and do a little fine-tuning) as fast as you can.