Sentiment Analysis of Movie Reviews

Classifying movie reviews as positive or negative

Dan-Cosmin Dăgădiță — dagadita.dan.v7d@student.ucv.ro

January 2026

Presentation Overview

Introduction & Motivation

Why sentiment analysis?

- Massive amounts of unstructured data (social media, reviews).
- Automation of opinion categorization for business insights.

Inherent challenges:

- **Nuances:** Sarcasm, irony, and context-dependent meanings.
- **Long-term dependencies:** Sentiment often depends on words far apart in a text.

Goal: Compare different deep learning architectures to find the most efficient solution for movie review classification.

Deep Learning Methodology

The sequence-to-vector approach

- Text is processed as a varying-length sequence of words.
- The network compresses the sequence into a single probability score.

RNN architectures tested:

1. **Simple RNN:** The baseline recurrent model.
2. **LSTM (Long Short-Term Memory):** Gated memory cells.
3. **GRU (Gated Recurrent Unit):** A lighter, simplified LSTM.
4. **Bidirectional:** Processing text forward and backward simultaneously.

The Vanishing Gradient Problem

Definition

A phenomenon where signals (gradients) used to update network weights decay exponentially as they travel back through long sequences.

- Standard RNNs “forget” the beginning of long reviews.
- **LSTM/GRU solution:** They use **gates** to regulate information flow, effectively maintaining a “memory” of important sentiment-carrying words over long distances.

Dataset & Preprocessing

The IMDB dataset [1]

- **Size:** 50,000 highly polar reviews.
- **Split:** 80% Training (40,000) / 20% Testing (10,000).
- **Labels:** Binary (Positive = 1, Negative = 0).

Preprocessing pipeline:

- **Vectorization:** Mapping top 10,000 words to unique IDs [2].
- **Padding:** Making all reviews in a batch (size 128) equal length.
- **Masking:** Ensuring the RNN ignores the “0” padding tokens.

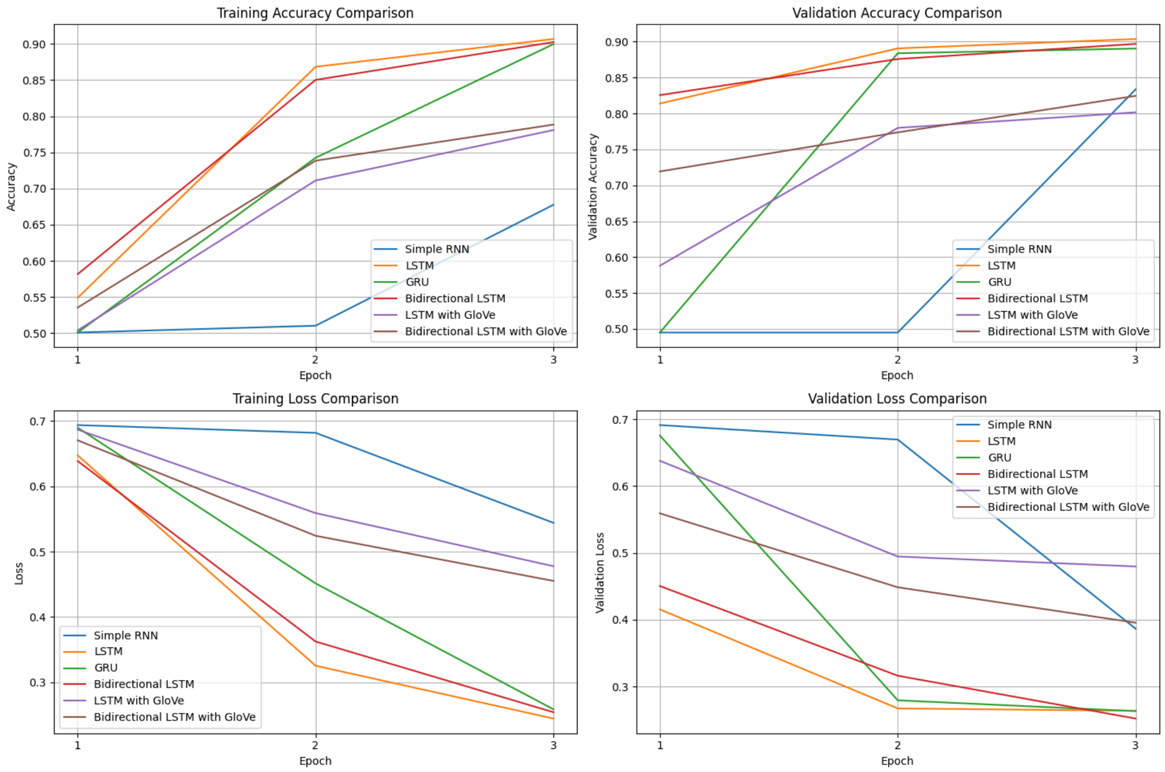
Word Embeddings: Random vs GloVe

- **Random initialization (128d):** Weights learned from scratch on the IMDB data.
- **Pre-trained GloVe (100d):** Learned from billions of words on Wikipedia [3].

Hint

The hypothesis: We expected GloVe to provide a superior semantic foundation for the model to understand word relationships out of the box.

Experimental Results



Analysis:

- **Winner:** Standard LSTM provided the best balance of speed and accuracy.
- **The surprise:** GloVe performed worse. Task-specific random embeddings captured the critics' language better than general-purpose GloVe vectors.

Technical Implementation

Reproducibility & Environment

- **Framework:** TensorFlow 2 / Keras API.
- **Infrastructure:** Dockerized environment with Nvidia GPU support.
- **Architecture:** Modular `build_model` factory for fair comparison.

> **Task**
>
> All code and local runtime setup details are available on GitHub:
> https://github.com/DanDagadita/movie-sentiment-analysis

Conclusion

- Gated architectures (LSTM/GRU) are essential for long-text sentiment analysis.
- Local, domain-specific training can outperform general pre-trained embeddings.
- Bidirectionality adds complexity but provided marginal gains for this specific task.

Thank you for your attention!

References

[1] N. Lakshmi, “IMDB Dataset of 50K Movie Reviews.” [Online]. Available: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[2] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Online, 2023. [Online]. Available: https://d2l.ai/chapter_recurrent-neural-networks/index.html

[3] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation.” [Online]. Available: https://nlp.stanford.edu/projects/glove/