## SENTIMENT ANALYSIS ON IMDB DATA USING AN EMBEDDED LAYER AND PRE-TRAINED EMBEDDING

Rishitha Reddy Muddasani

# Summary: The objective of the binary classification project on the IMDB dataset is to predict if a movie review will be positive or negative. The dataset is made up of 50,000 reviews, of which we evaluate just the top 10,000 words, limit training samples to 100, 500, 1000, and 100000, validate on 10,000 samples, and cutoff reviews after 150 words. Pre-Processing is done on the data. Afterwards, we feed data to a pretrained embedding model as well as the embedding layer, and we test various strategies for evaluating performance.

# Problem: The core challenge is to determine which approach yields superior performance in predicting sentiment in the IMDB dataset specifically, whether a movie review is positive or negative.

# Technique

**Dataset Description:** The IMDB dataset contains movie reviews with sentiment classifications (positive or negative).

**Preprocessing:** Each review is transformed into word embeddings, where each word is represented by a fixed-size vector. The vocabulary size is limited to 10,000 words. Reviews are converted into sequences of integers, with each integer representing a distinct word. To facilitate input into the neural network, integers are converted into tensors by ensuring consistent length through padding.

**Approach:** In our study, we studied two methods for generating word embeddings for our IMDB review dataset: a pretrained word embedding layer based on the GloVe model and a custom-trained embedding layer. Large volumes of text data are utilized to train the popular pretrained word embedding model GloVe, which we employed in our work. It is a well-liked option for natural language processing tasks because to its ability for capturing the syntactic and semantic links between words.

We trained the 6B version of the GloVe model on a corpus of Wikipedia data and Gigaword 5; it has 6 billion tokens and 400,000 words. We created two distinct embedding layers using the IMDB review dataset: one with a custom-trained embedding layer and the other with a pre-trained word embedding layer to assess the efficiency of various embedding approaches.

We examined the accuracies of the two models using training sample sizes that varied, namely 100, 500, 1000, and 10,000. First, using the IMDB review dataset, we developed a custom-trained embedding layer. We used a testing set to gauge each model's accuracy after training it on various dataset samples. Subsequently, we compared these precisions with a model that included a pre-trained word embedding layer, which was also evaluated on various sample sizes.

# Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Embedding technique** | **Training**  **sample size** | Accuracy (%) | **Embedding technique** | **Training**  **sample size** | **Accuracy(%)** |
| **Custom-trained embedding layer** | 100 | 100 | Pretrained word embedding layer (GloVe) | 100 | 100 |
| **Custom-trained embedding layer** | 500 | 97.5 | Pretrained word embedding layer  (GloVe) | 500 | 98.4 |
| **Custom-trained embedding layer** | 1000 | 98.1 | Pretrained word embedding layer (GloVe) | 1000 | 98.2 |
| **Custom-trained embedding layer** | 10000 | 97.9 | Pretrained word embedding layer (GloVe) | 10000 | 84.0 |

**Custom-trained embedding layer:** Depending on the size of the training sample, the accuracy achieved with the custom-trained embedding layer varied from 97.5% to 100%. A training sample size of 100 produced highest accuracy. Since the embedding layer is particularly trained for the task at hand (IMDB review sentiment classification), it is likely that this technique's high accuracy might be attributed to more effective text data representations.

**Pretrained word embedding layer (GloVe):** Depending on the training sample size, the accuracy achieved using the pretrained word embedding layer (GloVe) varied from 84% to 100%. With 100 training samples, the greatest accuracy was attained. The pretrained embeddings can be useful even with minimum training data since they capture a large amount of the underlying semantic information in the text, which could account for the high accuracy with a small training sample size. Even so, the pretrained embeddings might not be equally effective at capturing the minute details of the specific task at hand as the training sample size grows, which might result in lower accuracy. Furthermore, employing the pretrained embeddings with larger training sample sizes causes the model to rapidly overfit, as mentioned in the prompt, which lowers accuracy. These findings make it challenging to say with certainty which method is the "best" to employ because it relies on the requirements and limitations of the work at hand. In this experiment, however, the custom-trained embedding layer performed better overall than the pretrained word embedding layer, especially when training with higher training sample sizes. Although there is a risk of overfitting, the pretrained word embedding layer could be a ‘better choice’ if computational resources are restricted and a short training sample size is required.