# Introduction

Assignment 4: Text and Sequence Data

The project aims to build a model that classifies movie reviews on the IMDB dataset as positive or negative. The analysis focuses on a subset of the data, considering only the top 10,000 most frequent words. Training will be conducted on various sample sizes (100, 500, 1000, and 100,000), with a dedicated validation set of 10,000 samples. Additionally, reviews will be cut off after 150 words.

## Key Challenge

## The core question is: which word embedding method yields superior performance in sentiment classification?

## Objective

This research aims to identify the most effective approach for sentiment analysis in the IMDB dataset, specifically predicting whether a movie review is positive or negative.

# Data and Preprocessing

* The analysis utilizes the IMDB movie review dataset, which contains sentiment labels (positive or negative).
* Preprocessing involves converting reviews into word embeddings and restricting the vocabulary to the top 10,000 words.
* Reviews are transformed into integer sequences, with each integer representing a unique word.

# Technique: Two word embedding methods are compared

## Custom-trained Embedding Layer:

A separate embedding layer is trained specifically on the IMDB review dataset.

## Pre-trained Word Embedding Layer (GloVe):

* This popular model utilizes a large corpus of text data (Wikipedia and Gigaword 5) to train word embeddings, capturing semantic and syntactic relationships between words.
* The 6B version, with 400,000 words and 6 billion tokens, is employed

# Results summary table

|  |  |  |  |
| --- | --- | --- | --- |
| **Embedding technique** | **Training sample size** | **Accuracy** | **Loss** |
| Custom-trained embedding layer | 100 | 0.498 | 0.697 |
|  | 500 | 0.515 | 0.693 |
|  | 1000 | 0.622 | 0.654 |
|  | 10000 | 0.860 | 0.324 |
|  |  |  |  |
| Pretrained word embedding layer (GloVe) | 100 | 0.511 | 0.716 |
|  | 500 | 0.507 | 0.693 |
|  | 1000 | 0.500 | 0.693 |
|  | 10000 | 0.500 | 0.693 |

## Custom-trained Embedding Layer

* For a training sample size of 100, the custom-trained embedding layer achieved accuracy (0.498) with a loss of 0.697.
* As the training sample size increased, the accuracy improved to 0.860, and the loss decreased gradually to 0.324

## Pretrained Word Embedding Layer (GloVe)

* The pretrained word embedding layer consistently outperformed the custom-trained embedding layer across all training sample sizes.
* For a training sample size of 100, the pretrained word embedding layer achieved accuracy (0.511) with a significantly higher loss of 0.716.
* As the training sample size increased, the accuracy decreased slightly, ranging from 0.511 to 0.500, and the loss decreased gradually from 0.716 to 0.693.

# Recommendations

## Use Pretrained Word Embeddings (GloVe)

Pretrained word embedding layers, such as GloVe, consistently underperform custom-trained embedding layers in terms of accuracy and loss across all training sample sizes. This might be due to limited data. Therefore, it is not recommended to utilize pretrained word embeddings, like GloVe, for text classification tasks, especially when dealing with limited data.

## Consider Data Augmentation Techniques:

Implement data augmentation techniques to increase the size and diversity of the training data. Techniques such as translation, rotation, or adding noise to text data can help improve model generalization and performance, particularly with limited training data.

## Explore Fine-Tuning Pretrained Embeddings:

Consider fine-tuning pretrained word embeddings to adapt them to the specific domain of the dataset. Fine-tuning pretrained embeddings on the target dataset can further enhance model performance by capturing domain-specific semantics.