# Intro

Sentiment analysis, the process of identifying and categorizing opinions expressed in text to determine the writer's attitude, plays a pivotal role in understanding consumer opinions, monitoring brand sentiment, and enhancing customer experience. In recent years, this field has advanced significantly with the adoption of deep learning techniques and pretrained word embeddings, allowing for more accurate classification of emotions and opinions. Pretrained embeddings like Word2Vec and GloVe have played significant roles in these advancements by providing meaningful word representations that capture semantic relationships essential for natural language understanding.

In this assignment, we focus on building a classification system for sentiment analysis, leveraging pretrained Word2Vec embeddings and experimenting with various deep learning models. Our goal is to classify movie review sentences from the dataset introduced by Bo Pang and Lillian Lee (2005), distinguishing positive and negative sentiments. This task is structured to deepen our understanding of natural language processing (NLP) applications and to explore the use of pretrained embeddings in constructing effective sentence classification models.

The assignment is divided into three main parts: dataset preparation, model training with a Recurrent Neural Network (RNN), and enhancements on the RNN. We start with dataset preparation, loading and structuring the movie review data, and initializing word embeddings as input to the models. To address out-of-vocabulary (OOV) words, we also discuss our approach to mitigate their impact on model performance in a subsequent section of the report.

Model training is initially conducted on a basic RNN classifier designed to predict sentiment labels, followed by hyperparameter tuning based on validation performance. To enhance this baseline model, we experimented with bidirectional LSTM (BiLSTM) and bidirectional GRU (BiGRU) architectures, which capture context from both directions in the text. Additionally, we replaced the vanilla RNN with Convolutional Neural Networks (CNNs) to observe their effectiveness in natural language processing tasks. Finally, we applied BERT, a transformer-based model, to further improve performance, achieving significant enhancements in sentiment classification accuracy.

# 

# 

# 

# BiGRU

Hyperparameters

| Hidden Size | 64, 128, 256 |
| --- | --- |
| GRU Layers | 1, 2, 3 |
| Learning Rate | 1e-5, 1e-4, 1e-3, 1e-2, 1e-1 |
| Batch Size | 32, 64, 128 |
| Epochs | 10, 20, 30, 40, 50 |
| Optimizer | Adam, SGD, RMSprop, AdamW |

Keep padding

| Aggregation | Hidden Size | GRU Layers | Learning Rate | Batch Size | Epochs | Optimizer | Val Acc | Test Acc |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Max Pooling | 256 | 3 | 0.0001 | 64 | 40 | RMSprop | 0.8058 | 0.8049 |
| Mean Pooling | 128 | 1 | 0.001 | 64 | 40 | RMSprop | 0.8011 | 0.8021 |
| Max+Mean Pooling | 256 | 2 | 0.0001 | 32 | 10 | RMSprop | 0.8058 | 0.7917 |
| mhsa | 256 | 1 | 0.0001 | 32 | 40 | RMSprop | 0.8124 | 0.8002 |

Mask padding

| Aggregation | Hidden Size | GRU Layers | Learning Rate | Batch Size | Epochs | Optimizer | Val Acc | Test Acc |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Max Pooling | 128 | 3 | 0.001 | 32 | 50 | Adam | 0.8021 | 0.8171 |
| Mean Pooling | 128 | 2 | 0.0001 | 32 | 50 | AdamW | 0.7983 | 0.7880 |
| Max+Mean Pooling | 256 | 2 | 0.001 | 32 | 20 | RMSprop | 0.8077 | 0.8208 |
| mhsa | 256 | 1 | 0.00001 | 32 | 40 | AdamW | 0.7899 | 0.7974 |