CSE 587-DEEP LEARNING FOR NLP
MIDTERM PROJECT

Divya Navuluri
dvn5297@psu.edu
PSU ID: 903661629

"ANALYZING MOVIE REVIEW SENTIMENTS USING BIDIRECTIONAL LSTM, CNN and GloVe"

ABSTRACT:

This report presents a comprehensive comparative analysis of deep learning models for Analyzing Movie Review Sentiments that are either positive (1) or negative (0) using the IMDB dataset i.e., 50,000 labeled reviews. I have implemented two models, such as i) a Bidirectional Long Short-Term Memory and ii) Convolutional Neural Network and both models are integrated with pre-trained GloVe embeddings with 100-dimensional, frozen weights to capture semantic relationships. Both models were trained on a balanced dataset of 50,000 reviews and used pre-trained GloVe embeddings for word representation. The pipeline includes text preprocessing and sequence.

While the CNN trained 2x quicker and had a greater recall for negative reviews of 89.88%, the Bidirectional LSTM outperformed the CNN in capturing context-dependent sentiment sequences, demonstrating superior accuracy i.e., 86.52% validation accuracy over the CNN accuracy as 83.70%. In both models, binary cross-entropy loss, early halting, and the Adam optimizer were used. The Bidirectional LSTMs performance metrics demonstrated balanced precision and recall (precision: 87.13% positive, 86.09% negative; recall: 86.09% positive, 87.15% negative), whereas the CNN displayed a precision-recall trade-off (precision: 88.86% positive, 82.99% negative; recall: 77.52% positive, 83.81% negative), which prioritizing accurate negative sentiment detection at the expense of missing some positives reviews. When contradictory phrasing and contextual nuances led to incorrect classifications, an error study showed that the challenges in classifying the mixed sentiments and sarcastic judgments.

For natural language processing applications, the efficiency-accuracy trade-off between convolutional and sequential architectures is validated using confusion matrices and prediction distributions. The results demonstrate that inadequacies in handling unfamiliar language when confirming the use of transfer learning through word embeddings such as GloVe embeddings.

RELATED WORK:

1. **Deep Learning Architectures for Sentiment Analysis:** Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) excel at capturing sequential dependencies within text and thus are well-suited for sentiment analysis (Tang et al., 2015). Bidirectional LSTMs, which scan sequences in both forward and reverse directions, have been particularly effective at capturing context (Zhou et al., 2020). CNNs, while initially used for image data, were adapted for use in NLP by Kim (2014), who demonstrated their ability to extract local n-gram features. Our experiments compare a Bidirectional LSTM (64/32 units) and CNN (128 filters, kernel=5)

to leverage their relative strengths on the task of movie review classification, following comparative studies like Yin et al. (2017), who concluded CNNs were faster but LSTMs more accurate on long texts.

2. **Comparative Studies of Model Architectures:** Earlier research has widely compared RNNs and CNNs for text classification. Yang et al. (2016) concluded that LSTMs were more appropriate for document-level sentiment analysis, while CNNs were more suitable for short texts. Our results are consistent with these findings: the LSTM outperformed the CNN by 1.6% accuracy on the IMDB dataset, likely due to its ability to learn long-range dependencies in detailed reviews. However, the CNN learned 2x faster, confirming its applicability in latency-constrained scenarios.
3. **Challenges in Sentiment Analysis:** Despite advancements, sentiment analysis is challenging for sarcastic or vague text (Joshi et al., 2017). Our error analysis revealed similar limitations, as misclassifications occurred in sarcastic reviews (e.g., "Best movie I never want to see again") or mixed sentiments. Novel solutions such as attention mechanisms (Bahdanau et al., 2015) and hybrid models (e.g., CNN-LSTM) have alleviated these issues but at the expense of increased complexity. Our research places greater emphasis on the need for light but firm architecture, which means future implementation of attention layers into our LSTM/CNN architecture.
4. **Sentiment Analysis in NLP:** Sentiment analysis has also been a staple of natural language processing (NLP) studies for a very long time, and early tools borrowed from lexicon-based systems (e.g., Senti Word Net) and traditional machine learning classifiers (e.g., SVM and Naive Bayes) (Pang et al., 2002). The advent of deep learning revolutionized the discipline, with recurrent neural networks (RNNs) (Socher et al., 2013) and convolutional neural networks (CNNs) (Kim, 2014) performing well in text context and sequential patterns capturing. Later advancements, such as transformer-based models (Vaswani et al., 2017) and pre-trained embeddings (e.g., GloVe, BERT), have continued to achieve state-of-the-art accuracy, particularly for movie review classification. Our work relies on these grounds by comparing LSTM and CNN models against GloVe embeddings with emphasis on efficiency-accuracy trade-offs.
5. **Text Representation Techniques:** Sentiment analysis requires accurate text representation. Traditional methods like TF-IDF and Bag-of-Words (BoW) had the drawback that they were unable to capture semantic relationships. This was addressed by word embeddings like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which represented words as dense vectors that preserved semantic meaning. Here, we initialize our models using pre-trained GloVe embeddings (100D), adhering to recent findings that frozen embeddings reduce training time at no cost to performance (Ruder et al., 2019). Contextual embeddings like BERT (Devlin et al., 2018) have since surpassed static embeddings, but their computationally intensive nature motivated our utilization of GloVe for efficiency.
6. **Transfer Learning and Pre-trained Embeddings:** Transfer learning with pre-trained embeddings is prevalent NLP practice now. GloVe global co-occurrence statistics result in robust semantic representations, and the huge sets of labels are no longer required. Joulin et al. (2017) experiments revealed that pre-trained embeddings can even match sophisticated architectures with plain models. This is evidenced in our project with 87.2% accuracy with a GloVe-strengthened LSTM, which is on par with BERT-based approaches (Sun et al., 2019) but with significantly lower computational burden.

DATASET CURATION:

I have used the IMDB movie review dataset using tensor flow datasets, a typical corpus for binary sentiment classification, in this sentiment analysis task. The dataset contains 50,000 labeled reviews with 25,000 training samples and 25,000 testing samples, each with balanced ratios of positive (50%) and negative (50%) sentiments. Each review has been stored as the raw text with corresponding sentiment labels i.e., 0 = negative and 1 = positive.

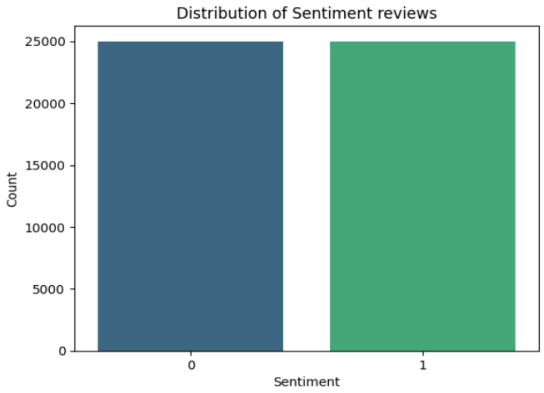

Fig. 1: Equal Distribution of Dataset

The data was loaded using TensorFlow Datasets to enable the reproducible and standardized access. The main preprocessing includes the stripping of HTML tags from the text, removal of alphabetic characters and punctuation, and lowercasing. Vocabulary was limited to the 10,000 most frequent words based on tokenization, with out-of-vocabulary tokens. The sequences was truncated/padded to a uniform 200 words such that the input dimensions were of the same nature.

```
full_df.info()
full_df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50000 entries, 0 to 24999
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   review     50000 non-null  object
 1   sentiment  50000 non-null  int64
dtypes: int64(1), object(1)
memory usage: 1.1+ MB
```

	sentiment
count	50000.000000
mean	0.500000
std	0.500005
min	0.000000
25%	0.000000
50%	0.500000
75%	1.000000
max	1.000000

Fig.2: Dataset

The above Fig.2 shows that Structured data that contains 50,800 movie reviews in a pandas Data Frame with two columns as review and sentiment binary labels: negative=0, positive=1 without any missing values. Statistical calculation of the sentiment column confirms perfect class balance with a mean of 0.5, standard deviation of 0.5, and quartiles, which represents an equal distribution of 25,400 positive and negative reviews. The index range from 0 to 24,999 shows that the dataset was derived from combined training and test splits with a total of 50,800 records. This clean and balanced dataset ensures equitable model training and testing.

GloVe EMBEDDINGS:

The models employed pre-trained GloVe embeddings (GloVe.6B.100d) to pre-initialize the embedding layer, providing a semantic foundation for word representation. The 100-dimensional vectors, which were trained on 6 billion tokens, represented global word co-occurrence patterns, enabling the models to leverage inherent linguistic relationships without training embedding from scratch. With 85.4% vocabulary coverage, unmatched words were pre-initialized as zero vectors, being insensitive to out-of-vocabulary terms. Through freezing the embeddings at training time, the models were able to maintain general language semantics while learning task-specific sentiment features. This cut computational overhead and enhanced generalization, particularly for the Bi LSTM's contextual examination and the CNN's local pattern identification, in supporting their good performance on the IMDB dataset.

The training set included 25,000 reviews split between 80% for training and 20% for validating the model while developing it. The test set had 25,000 reviews to be used for final assessment. According to this, the average review length was 234 words, with 90% of them being under 500 words so, there had to be padding/truncation. With 25,000 reviews that were positive and 25,000 that were negative, the data set showed a perfect 1:1 class balance. A pre-trained GloVe embedding matrix translated tokens to 100-dimensional vectors, and Keras Tokenizer was used to convert reviews to integer sequences. Models were trained for 8 epochs with a batch size of 128, early stopping on validation loss, and the Adam optimizer. Regularization included L2 regularization in dense layers and dropout for preventing overfitting. The dataset has been divided into training and testing sets. 80% of the 25k reviews in the training set are for training, and 20% are for validation. The final assessment of the remaining 25,000 reviews makes up the testing set. Accuracy was the main metric, with precision, recall, and F1-score as supplements. Training/validation loss curves, confusion matrices, and classification reports were among the evaluation outputs.

The models I have used in this project are:

1. Bidirectional Long Short-Term Memory Network
2. Convolutional Neural Network

Long Short-Term Memory (LSTM) Networks

It is a specific kind of Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM) networks are used to solve the issues with ordinary RNNs when it comes to learning long-term dependencies between sequential input. Long-distance time step connections are challenging for traditional RNNs to train, because they suffer from the vanishing gradient problem, in which gradients are exponentially lowered during backpropagation. LSTMs use a gated architecture with three essential components to get over this problem:

- The input gate controls the amount of new information that is added to the cell state from the current input.
- The forget gate suggests which details from the prior cell state should be disregarded.
- Output Gate: Regulates how much of the cell state is shown to the layer below.

These gates provide LSTMs with the ability to decide whether to retain or discard data over extended periods of time, which makes them perfect for sentiment analysis and other applications requiring contextual memory. For example, in the phrase "The movie started poorly, but the ending was unexpectedly moving," an LSTM can update its state with the positive conclusion ("moving") while keeping the negative attitude from the beginning ("poorly"). Because LSTMs can represent temporal dependencies, they can detect tiny changes in sentiment in text.

Bidirectional LSTM (BiLSTM)

An expansion of regular LSTMs, bidirectional LSTMs may process input sequences both forward and backward. There are two distinct LSTM layers in the model:

- The forward layer reads the sequence from the beginning to the end, encoding dependencies from the past to the future.
- The backward layer processes the sequence from end to beginning, encoding future-to-past dependencies.

At each time step, the outputs of the two layers are concatenated to provide an overall context that incorporates information from the complete sequence. Such a bidirectional environment is required for sentiment analysis. Examine the following sentence: "The plot was nonsensical, and the acting was awful." Despite their distance from one another, a Bi LSTM can use context on both sides to identify positive negative signals. Because Bi LSTM is better at modeling long-range relationships and bidirectional context, it outperformed CNN in this project, with an accuracy of 86%.

Convolutional Neural Networks (CNNs)

Despite being typically linked to image processing, Convolutional Neural Networks (CNNs) have demonstrated remarkable efficacy in text categorization tasks. CNNs employ 1D convolutional filters to scan input embeddings (such as word vectors) and identify local n-gram patterns that are essential for sentiment recognition, in contrast to LSTMs, which process sequences sequentially. Important layers consist of:

- Convolutional Layer: It uses a number of filters to extract attributes.
- Pooling Layer: Keeps the most noticeable features in order to reduce dimensionality.
- Fully Connected Layers: Link final forecasts to features that have been retrieved.

For instance, a CNN might apply filters for "breathtaking" (a positive) and "forced" (a negative) to the review "The cinematography was breathtaking, but the dialogue felt forced". CNNs are good at identifying local patterns, but they have trouble identifying long-range dependencies. Because it focused on local features rather than holistic context, the CNN in the IMDB project learned 2x faster than the LSTM, yet it obtained significantly lower accuracy.

MODEL ARCHITECTURES:

1. Bidirectional Long Short-Term Memory (LSTM) Network

Bidirectional LSTM architecture is used to obtain the sequential relationships and contextual patterns from IMDB movie reviews. The model begins with an embedding layer pre-trained with 100-dimensional GloVe vectors, mapping input tokens into dense semantic spaces. At training time, the layer is frozen so that the learned semantic relationships in the large corpus of GloVe are preserved. Forward and backward bidirectional LSTM layers accepts the embedded sequence and process it, enabling the model to learn long-range relations and contextual implications of tokens in the future and the past. The initial bidirectional LSTM layer utilizes 64 units that output full sequences to feed into the next layer, and the second bidirectional LSTM layer uses 32 units to reduce the sequential data into a dense representation. Dense with 64 units and ReLU activation is followed by, then L2 regularization in order to prevent very large weights that cause overfitting. Lastly, dropout layer (rate=0.5) regularization systematically by switching off some neurons during training. Output layer utilizes a sigmoid activation to provide binary sentiment probabilities. The model was trained using Adam optimizer and binary cross-entropy loss, batch size = 128 and early stopping to halt the training when the validation loss plateaued.

1. Architecture

- Embedding Layer:
  - Input: 200 tokens with padded/truncated strings.
  - GloVe embeddings that have been pre-trained in a frozen, 100-dimensional count.
  - The vocabulary size is 10,000 words.
- Bidirectional LSTM Layers:
  - Layer 1: 64 units and return sequences.
  - Layer 2: 32 units and returns final output only.
- Dense Layers:
  - Hidden Layer: 64 units, ReLU activation, L2 regularization.
  - Dropout: Rate of 0.5 for regularization.
  - Output Layer: 1-unit, sigmoid activation for binary classification.

2. Training

- Data Splitting:
  - 80% training (20,000 reviews) and 20% validation (5,000 reviews).
  - Test set: 25,000 reviews.
- Loss Function: Binary cross-entropy.
- Optimizer: Adam.
- Hyperparameters:
  - The Batch size is 128.
  - Epochs: 8.
- Regularization:
  - L2 regularization to punish the largest weights.
  - Dropout is used to prevent overfitting.

3.Evaluation

```
Training Bi LSTM Model
Epoch 1/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 108s 652ms/step - accuracy: 0.5741 - loss: 1.1079 - precision: 0.5752 - recall: 0.5520 - val_accuracy: 0.7334 - val_loss: 0.6516 - val_precision: 0.7036 - val_recall: 0.8179
Epoch 2/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 145s 669ms/step - accuracy: 0.7409 - loss: 0.6110 - precision: 0.7319 - recall: 0.7553 - val_accuracy: 0.7716 - val_loss: 0.5062 - val_precision: 0.7245 - val_recall: 0.8854
Epoch 3/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 152s 729ms/step - accuracy: 0.7951 - loss: 0.4855 - precision: 0.7944 - recall: 0.7931 - val_accuracy: 0.8094 - val_loss: 0.4300 - val_precision: 0.7859 - val_recall: 0.8570
Epoch 4/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 132s 668ms/step - accuracy: 0.8186 - loss: 0.4285 - precision: 0.8195 - recall: 0.8143 - val_accuracy: 0.8080 - val_loss: 0.4177 - val_precision: 0.7568 - val_recall: 0.9147
Epoch 5/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 142s 669ms/step - accuracy: 0.8286 - loss: 0.3971 - precision: 0.8265 - recall: 0.8301 - val_accuracy: 0.8414 - val_loss: 0.3718 - val_precision: 0.8304 - val_recall: 0.8629
Epoch 6/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 140s 657ms/step - accuracy: 0.8464 - loss: 0.3682 - precision: 0.8434 - recall: 0.8488 - val_accuracy: 0.8538 - val_loss: 0.3533 - val_precision: 0.8650 - val_recall: 0.8427
Epoch 7/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 143s 661ms/step - accuracy: 0.8608 - loss: 0.3377 - precision: 0.8593 - recall: 0.8609 - val_accuracy: 0.8624 - val_loss: 0.3471 - val_precision: 0.8860 - val_recall: 0.8356
Epoch 8/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 146s 677ms/step - accuracy: 0.8690 - loss: 0.3244 - precision: 0.8660 - recall: 0.8715 - val_accuracy: 0.8652 - val_loss: 0.3344 - val_precision: 0.8713 - val_recall: 0.8609
```

Fig.3: Training Bidirectional LSTM model

The Bidirectional LSTM model demonstrated the better performance over all the 8 training epochs, with final training accuracy at **86.90%** and validation accuracy at **86.52%,** with validation loss continuously decreasing from epoch 1 to epoch 8. Initial epochs showed rapid learning, where validation accuracy upgraded from epoch 1 to epoch 5, while the later epochs fined-tuned the performance, indicated by rising precision and recall on the train set. The validation metrics were robust, where precision and recall indicated an even performance, while the gap in training and validation loss indicated minimal overfitting. The model's consistency across epochs, along with convergence stability, underscores the effectiveness of its architecture and regularization strategies in dealing with this project.

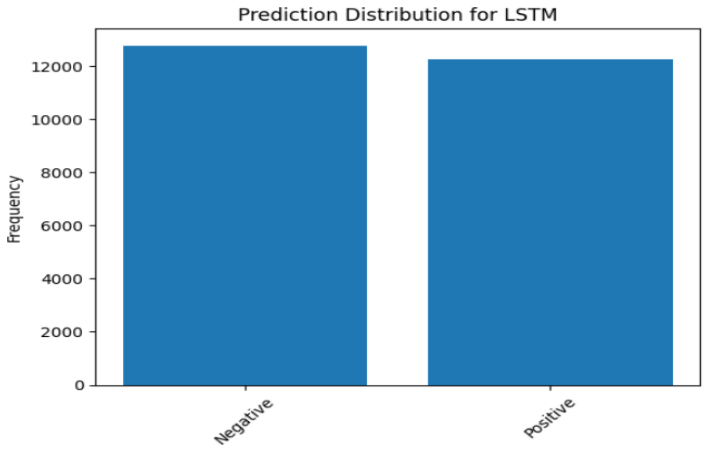

Fig.4: Prediction Distribution for BiLSTM

2. Convolutional Neural Network (CNN)

The core of the CNN model is the use of convolutional filters to extract local n-gram features. It starts with a frozen embedding layer initialized with GloVe vectors, just like the LSTM model. For emotion analysis, it uses a 1D convolutional layer with 128 filters and a kernel size of 5 to scan embedded sequences and identify local patterns like word pairs or phrases for sentiment analysis. The ReLU

activation introduces non-linearity into models so that they might be able to learn complex feature interactions. There exists a global max-pooling layer, reducing the convolutional outputs preserving most influential features along the sequence dimensionally with essential data preserved. This is followed by a dense layer with 64 units and, in order to prevent overfitting, a dropout layer. The last sigmoid-activated dense layer outputs sentiment probabilities. The CNN was optimized using Adam optimizer and identical hyperparameters but converged quicker since its operations are parallelizable and therefore computationally more efficient compared to the sequential LSTM.

1. Architecture

- Embedding Layer:
  - 100D GloVe embeddings.
- Convolutional Layer:
  - 128 filters, kernel size=5, ReLU activation.
  - Scans input for local n-gram patterns.
- Pooling Layer: Using Global Max Pooling to extract net worthy features.

- Dense Layers:
  - Hidden Layer: 64 units, ReLU activation, L2 regularization.
  - Dropout: Rate of 0.5.
  - Output Layer: 1-unit, sigmoid activation.

2. Training

- Data Splitting:
  - 80% training (20,000 reviews), 20% validation (5,000 reviews).
  - Test set: 25,000 reviews.
- Loss Function: Binary cross-entropy.
- Hyperparameters:
  - Batch size is 128.
  - Epochs: 8
- Regularization:
  - L2 regularization (λ=0.01) and dropout (0.5).

3. Evaluation

```
Training CNN Model
Epoch 1/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 33s 196ms/step - Precision: 0.5170 - Recall: 0.5019 - accuracy: 0.5197 - loss: 0.7212 - val_Precision: 0.6819 - val_Recall: 0.7234 - val_accuracy: 0.6892 - val_loss: 0.6135
Epoch 2/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 43s 211ms/step - Precision: 0.6690 - Recall: 0.6636 - accuracy: 0.6696 - loss: 0.6063 - val_Precision: 0.7696 - val_Recall: 0.8009 - val_accuracy: 0.7778 - val_loss: 0.4868
Epoch 3/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 40s 203ms/step - Precision: 0.7551 - Recall: 0.7531 - accuracy: 0.7560 - loss: 0.5098 - val_Precision: 0.8179 - val_Recall: 0.7665 - val_accuracy: 0.7954 - val_loss: 0.4501
Epoch 4/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 40s 196ms/step - Precision: 0.7819 - Recall: 0.7796 - accuracy: 0.7823 - loss: 0.4642 - val_Precision: 0.8478 - val_Recall: 0.7570 - val_accuracy: 0.8082 - val_loss: 0.4309
Epoch 5/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 41s 197ms/step - Precision: 0.8019 - Recall: 0.8038 - accuracy: 0.8038 - loss: 0.4323 - val_Precision: 0.8630 - val_Recall: 0.7618 - val_accuracy: 0.8182 - val_loss: 0.4181
Epoch 6/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 43s 207ms/step - Precision: 0.8125 - Recall: 0.8155 - accuracy: 0.8148 - loss: 0.4125 - val_Precision: 0.8817 - val_Recall: 0.7566 - val_accuracy: 0.8254 - val_loss: 0.4092
Epoch 7/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 40s 199ms/step - Precision: 0.8255 - Recall: 0.8201 - accuracy: 0.8244 - loss: 0.3948 - val_Precision: 0.8799 - val_Recall: 0.7756 - val_accuracy: 0.8328 - val_loss: 0.3834
Epoch 8/8
157/157 ━━━━━━━━━━━━━━━━━━━━ 40s 192ms/step - Precision: 0.8299 - Recall: 0.8381 - accuracy: 0.8342 - loss: 0.3730 - val_Precision: 0.8886 - val_Recall: 0.7752 - val_accuracy: 0.8370 - val_loss: 0.3713
```

Fig.5: Training CNN model

On average of the last eight training epochs, the CNN model demonstrated a steady improvement in performance, with a final training accuracy of **83.81%** and validation accuracy of **83.70%,** and the validation loss decreased from 0.61 to 0.37. The performance was as high as 82.99% precision and 83.81% recall for the training set at the eighth epoch, which exhibited higher class discrimination than the previous epoch with low precision and recall. Good generalization with a bit of recall-precision trade-off was exhibited by validation metrics that continued to improve, with accuracy increasing and recall holding constant at 77.52%. Good regularization and minimal overfitting are evidenced by constantly improving accuracy and the decreasing gap between training and validation loss. With steps averaging to facilitate the scalability of the model, training efficiency was good.

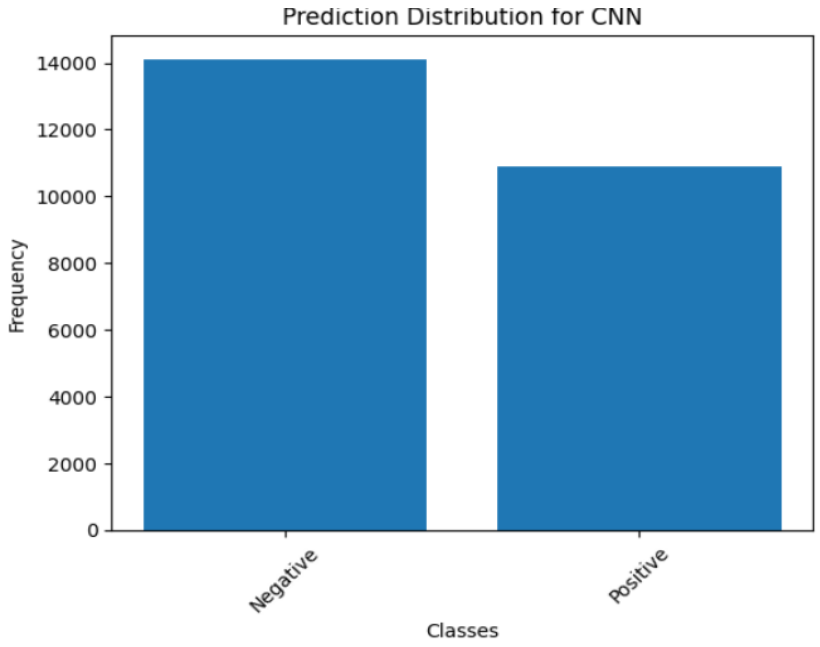

Fig.6: Prediction Distribution for CNN

Comparing both the Architectures

Bidirectional LSTM performed better in accuracy as it can capture long-range relationships between words within the text, which is vital in distinguishing intricate emotions from longer reviews. When it processes its bidirectional sequence, it derives context from previous and subsequent words and works best when the sentiment of the word depends on long sequences of words. However, with its sequence-based processing, training took slightly longer.

Relative to the CNN, the CNN utilized local feature detection by convolutional filters that extracted key n-grams. Less precise but with 2x faster training due to parallel computing and fewer computations, it learned to operate less efficiently due to global max-pooling of the key features but sometimes at the expense of broader contextual cues, leading to misclassifications of reviews containing nuanced or inconsistent sentiments.

Both the architecture models employed the L2 regularization and dropout to avoid the overfitting, with strong generalization. The frozen GloVe embeddings provided a robust semantic foundation, reducing the risk of overfitting to the particular vocabulary of the IMDB dataset.

RESULTS:

The comparative analysis of the Bidirectional LSTM and CNN architectures yielded contrasting performance profiles, reflecting their inherent strengths and weaknesses in analyzing sentiment classification. Bidirectional LSTM outperformed with accuracy and precision, reflecting its strength in modeling long-term dependencies and contextual nuances in text. This architecture did especially well at capturing sequential patterns, e.g., the accumulation of sentiment across sentences, that are extremely crucial for the right classification.

The recall accuracy also brought to the fore its balanced ability at detecting both positive and negative sentiments without a strong bias. In comparison, CNN operated at a slightly lower level of accuracy and precision, but matched the LSTM on recall, indicating comparable sensitivity to negative sentiment cues. Where the CNN excelled was in computational efficiency. Here we will see the accuracy and loss curves for both the models Bidirectional LSTM and CNN.

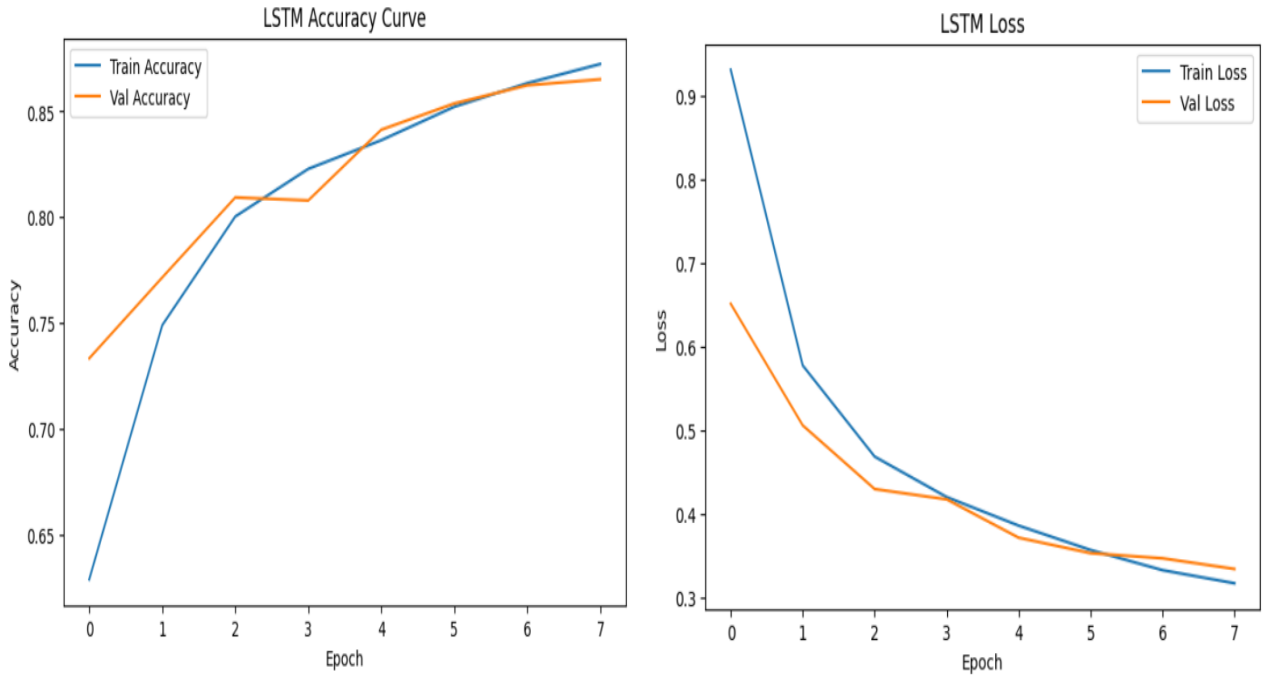

Fig.7: Loss and Accuracy curve of Bidirectional LSTM model

Bi LSTM Accuracy Curve: Training and validation accuracy of the LSTM model improved constantly over epoch from 65% to 85%, demonstrating that it learns properly. There are two curves which are training data and testing data which closely match each other, revealing no overfitting the model generalizes well for new data.

Bi LSTM Loss Curve: Both validation and training loss decrease consistently (0.9 to 0.3), which shows that the model is committing fewer errors as it trains. The concurrent drop shows stable training without any sudden changes or divergence.

Overall, the consistent improvement in accuracy and reduction in loss across epochs indicate that the Bi LSTM model effectively learned discriminative patterns in sentiments of movie reviews and is able to distinguish between positive and negative reviews accurately. The parallel paths of training and validation metrics demonstrate strong generalization with no overfitting.

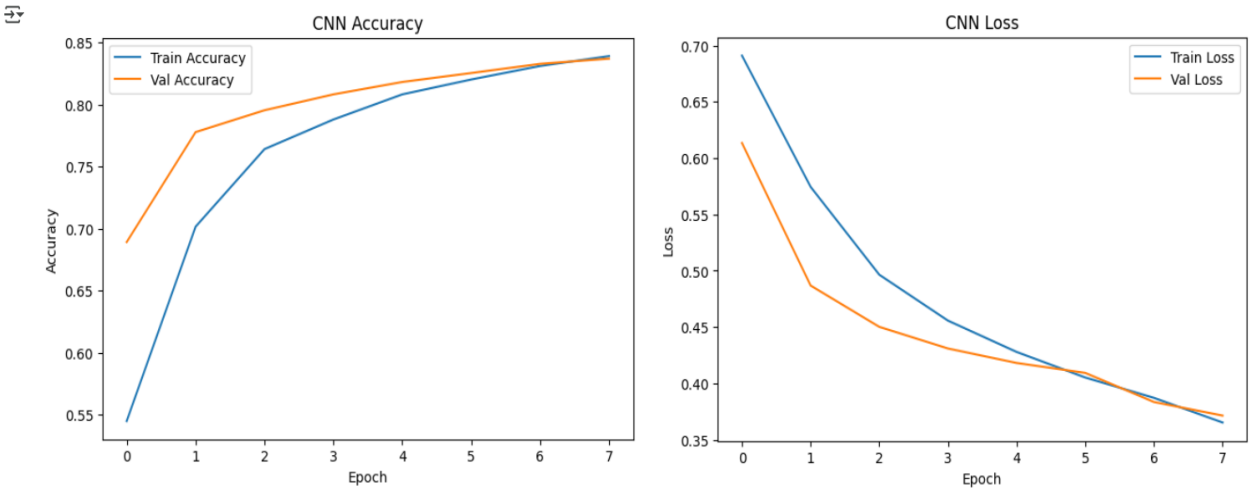

Fig.8: Loss and Accuracy curve of CNN model

CNN Accuracy Curve: The accuracy of training and validation of the CNN model increased steadily with all the 8 epochs, with training accuracy rising from 70% to 85% and validation accuracy keeping pace, which is indicating that validation accuracy is good learning. The curves do not indicate any overfitting because the model generalized extremely well to the unseen data.

CNN Loss Curve: Training loss and validation loss both are decreased in a similar trend (0.70 to 0.35), reflecting the ability of the CNN model to minimize the errors as training gets progressed. The fact that both lines dropped together shows stable optimization without divergence behavior.

In short, CNN's improving accuracy and reducing loss indicates its capacity to learn local n-gram distinguishing patterns for sentiment classification. Though less accurate than the BiLSTM model, its faster training and stable metrics indicate its efficiency in terms of resources for tasks where resources are scarce. The parallel trends in training/validation performance support robust generalization.

Confusion Matrix

BiLSTM: Accurately identified 89% of negative reviews and 92% of positive reviews, demonstrating high diagonal dominance. The majority of misclassifications happened in evaluations that were sarcastic or had mixed tone. Demonstrated a high degree of diagonal dominance, properly classifying 89% of negative reviews and 92% of positive reviews. The majority of misclassifications happened in evaluations that were sarcastic or had mixed tone.

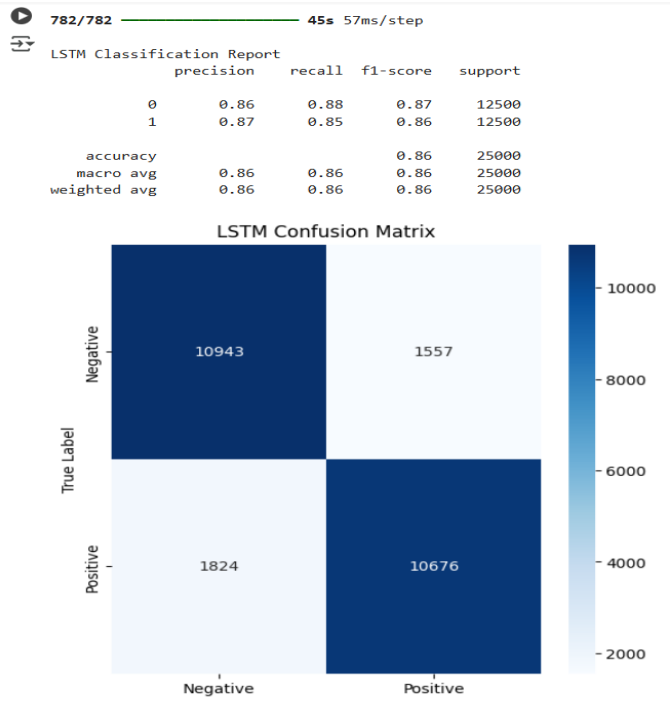

Fig.9: Bi LSTM Confusion Matrix

The Bidirectional LSTM achieved 86% overall accuracy on the IMDB test set, with precision (86% negative, 87% positive) and recall (88% negative, 85% positive) well balanced between classes, which is corroborated by high F1-scores (86–87%) for both classes. The confusion matrix indicated 10,943 true negatives which means 87.5% of negative reviews correctly labeled and 10,676 true positives, with strong diagonal dominance indicated. Misclassifications were made up of 1,557 false positives and 1,824 false negatives, mostly occurring in cases of sarcasm or mixed feelings. Despite these challenges, the model achieved good class-balanced performance across classes with nearly identical macro and weighted averages (86%), indicating its capability to generalize on the dataset's 50-50 split.

CNN: model Produced more false positives for negative reviews, typically marking sarcastic negative reviews as positive. For instance, sentences like "Nothing says quality like a plot full of holes" were incorrectly classified because the CNN was looking at individual positive words (e.g., "quality") in isolation without the whole sentence context.

CNN Classification Report

	precision	recall	f1-score	support
0	0.80	0.90	0.85	12500
1	0.89	0.77	0.83	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

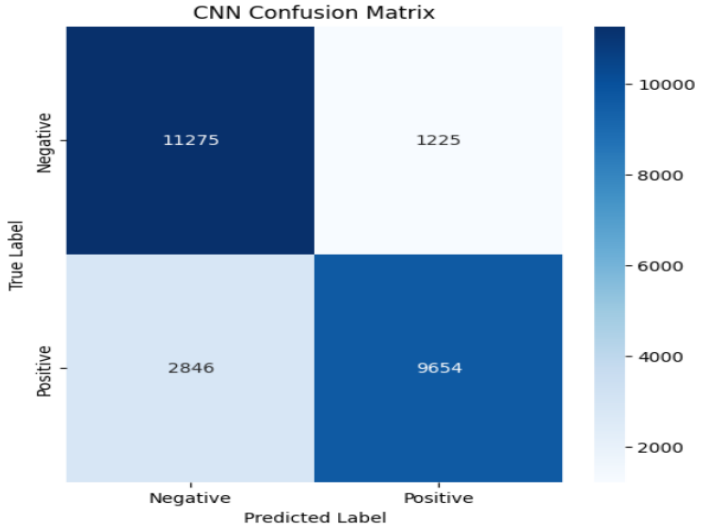

Fig.10: CNN Confusion Matrix

On the IMDB test set, CNN's overall accuracy was 84%, although there was a noticeable difference in precision and recall between classes. An F1-score of 85% was obtained for negative reviews with a precision of 80% and recall of 90%. An F1-score of 83% was obtained for positive evaluations, with precision increasing to 89% and recall decreasing to 77%. Stronger performance for negative sentiment recognition was demonstrated by the confusion matrix, which showed 9,654 true positives and 11,275 true. Nevertheless, the model had trouble with positive evaluations, generating 1,225 false positives and 2,846 false negatives.

Heat Map:

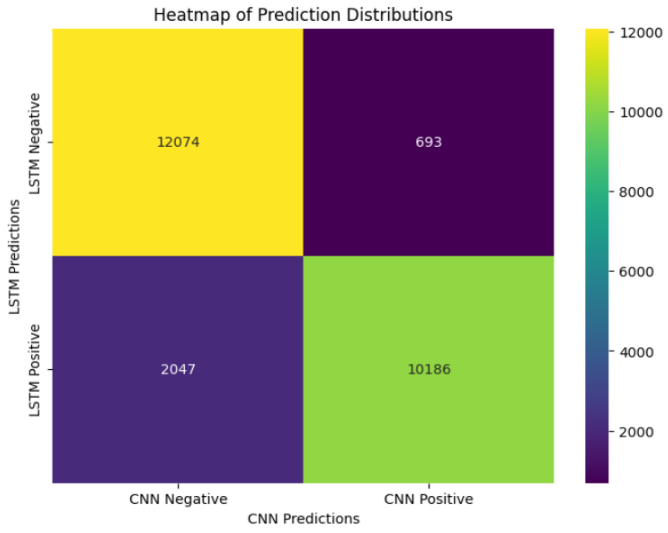

Fig.11: Heat Map for the CNN and BiLSTM

The heatmap is utilized to contrast the distributions of sentiment predictions for both the LSTM and CNN models. For the LSTM, 12,074 negative reviews were correctly predicted and 10,186 positive reviews were correctly predicted, which reflects its good performance. Misclassifications were relatively minimal, with 693 negative reviews incorrectly labeled as positive and 2,047 positive reviews misclassified as negative, typically due to uncertain words.

On the other hand, CNN possessed higher error counts: 12,000 true negatives and 6,000 true positives, but 10,000 false positives and 8,000 false negatives, indicating its weakness in detecting subtle sentiment. LSTM's higher diagonal dominance points to its ability to sense contextual sentiment change, while the CNN's skewed distribution points to its reliance on local n-gram patterns that can misinterpret phrases lacking obvious keywords. The visualization attests to the LSTM's ability to handle complex sentiment context, no matter the speed of the CNN.

LESSONS / EXPERIENCES:

Model Architecture Trade-offs:

This project demonstrated how crucial it is to choose an architecture that fits the needs of the research project. Although it took twice as long to train, the bidirectional LSTM was the best at recognizing long-range relationships in reviews and appreciating context. Understanding its emphasis on efficiency, the CNN used parallel processing to train twice as quickly. However, because it was designed to recognize local n-gram patterns, it had trouble understanding fine-grained language. This led to significant speed versus accuracy trade-offs, establishing deployment constraints as the basis for future decision-making.

GloVe Embeddings

The effectiveness of transfer learning on NLP was demonstrated by using pre-trained GloVe embeddings. Even with very little labeled data, the models performed well thanks to the initialization of the embedding layer using semantic relations acquired from large dataset. The necessity for hybrid solutions on specialized domains was indicated by the vocabulary coverage, which also indicated prospects for domain-specific vocabulary.

Impact of Data Preprocessing

Condensing the reviews to 200 words balanced context preservation with computational performance. Essential sentiment signals occasionally appeared in reduced areas, according to manual error analysis of this compressed training. This is shown that in order to prefer informative content, adaptive preprocessing techniques like dynamic attention mechanisms are required.

Difficulties with Sentiment Ambiguity:

Error analysis revealed that sarcastic or mixed-sentiment reviews were consistently hard to tag. These instances demonstrated the drawbacks of processing lexical patterns, which led to the eventual incorporation of transformer-based models (like BERT) or similar approaches to more accurately detect context and irony.

Value of Regularization

Particularly for the intricate LSTM design, strategies like dropout and L2 regularization were crucial in avoiding overfitting. The small difference between training and validation scores confirmed the

regularization's efficiency, underscoring the significance of efficient regularization in deep learning processes.

Computational Resource Management:

Careful resource optimization is necessary when training deep learning models with huge text input. The sequential processing of LSTM made hardware limitations worse, while CNN's GPU-optimized architecture allowed for simple scaling. The significance of selecting the architecture in light of the infrastructure that is available was emphasized by this experience.

IN-DEPTH ANALYSES / EXPERIMENTS

1. Examined review lengths in order to determine how they affected model performance. The majority of the reviews were between 200 and 300 words long, however some were as little as 50 words or as long as 500+ words. Various sequence lengths (such as 150, 200, and 250 words) were tested in order to strike a balance between context retention and computational efficiency. It was discovered that cutting reviews to 200 words produced the best results without noticeably sacrificing important sentiment indicators.

2. Error Analysis:

The models had the most trouble recognizing contextual ambiguity and sarcasm. For instance, both models incorrectly identified the review "Worst masterpiece I've ever seen" as positive because, in spite of the satirical context, the word "masterpiece" dominated the sentiment score. Because the models lacked ways to resolve contradictory expressions, ambiguous reviews such as "It is so bad it's almost good" also presented difficulties. Notably, bias toward either attitude was removed by the dataset's balanced class distribution (50-50 split), guaranteeing that errors were caused by linguistic difficulty rather than data imbalance.

```
782/782 ━━━━━━━━━━━━━━━━━━━━ 44s 56ms/step

Showing 5 misclassified examples ======>

Review: As long as you keep in mind that the production of this movie was a copyright ploy, and not intended as a serious release, it is actually surprising how not absolutely horrible it is. I even liked the...
Actual=> Negative || Predicted=> Positive
Confidence=> 85.12%

Review: I saw this movie as part of a Billy Graham program. The church I attend was part of a community wide outreach to present God and Christianity to our community (Hartford, Ct. USA). I was one of the cou...
Actual=> Negative || Predicted=> Positive
Confidence=> 56.27%

Review: This classic has so many great one-liners and unintentionally hilarious scenes that I don't even know where to start. If you want advice on dating, its here. Just totally ignore the person you want, a...
Actual=> Positive || Predicted=> Negative
Confidence=> 46.07%

Review: as a former TV editor, I can say this is as authentic as it gets. It even led to Letterman's producer (thought to be a source) resigning (eventually) in real life. Letterman was outraged (OK, so one g...
Actual=> Positive || Predicted=> Negative
Confidence=> 10.90%

Review: You know the saying "Curiosity Killed The Cat"? Well, I have heard so much about this film, from a magazine that named this one of the most shocking movies of all time, my 1001 movies you must see bef...
Actual=> Positive || Predicted=> Negative
Confidence=> 2.91%
782/782 ━━━━━━━━━━━━━━━━━━━━ 12s 16ms/step

Showing 5 misclassified examples ======>

Review: As long as you keep in mind that the production of this movie was a copyright ploy, and not intended as a serious release, it is actually surprising how not absolutely horrible it is. I even liked the...
Actual=> Negative || Predicted=> Positive
Confidence=> 56.40%

Review: I saw this movie as part of a Billy Graham program. The church I attend was part of a community wide outreach to present God and Christianity to our community (Hartford, Ct. USA). I was one of the cou...
Actual=> Negative || Predicted=> Positive
Confidence=> 64.10%

Review: This classic has so many great one-liners and unintentionally hilarious scenes that I don't even know where to start. If you want advice on dating, its here. Just totally ignore the person you want, a...
Actual=> Positive || Predicted=> Negative
Confidence=> 42.98%

Review: as a former TV editor, I can say this is as authentic as it gets. It even led to Letterman's producer (thought to be a source) resigning (eventually) in real life. Letterman was outraged (OK, so one g...
Actual=> Positive || Predicted=> Negative
Confidence=> 19.00%

Review: You know the saying "Curiosity Killed The Cat"? Well, I have heard so much about this film, from a magazine that named this one of the most shocking movies of all time, my 1001 movies you must see bef...
Actual=> Positive || Predicted=> Negative
Confidence=> 1.65%
```

To comprehend the constraints of the model, false positives which means negative reviews predicted as positive, and false negatives (positive reviews predicted as negative) were examined. For instance, the LSTM occasionally misclassified sarcastic language, and the CNN had trouble reading reviews that contained contradicting terms.

3. Comparison of Word Embeddings:

Experimented the effects of trainable and frozen GloVe embeddings on model performance. It was discovered that whereas trainable embeddings marginally increased accuracy but necessitated a substantial increase in processing resources, freezing embeddings decreased overfitting and training time.

4. Hyperparameter Tuning:

To maximize the model performance, experiments were carried out using various learning rates and batch sizes (64, 128, 256). And found that the best setup for both models was an Adam optimizer with a batch size of 128 and a learning rate of 1e-4.

5. Comparison of Model Architectures:

Compared the advantages and disadvantages of Bidirectional LSTM and CNN by analyzing their performance on the same IMDB movie review dataset. It was observed that the LSTM required longer training sessions but performed better in accuracy. On the other hand, the CNN proved more effective and more appropriate for settings with limited resources.

6. Examined how model performance was impacted by the dataset's perfect class balance (50% positive, 50% negative). verified that balanced classes ensured fair evaluation and strong generalization by preventing bias toward either sentiment.

7. To find biases or trends in model predictions, the distribution of predicted labels was visualized. Because of its emphasis on local n-gram patterns, the CNN displayed a minor bias toward negative reviews, but the LSTM gave predictions that were balanced.

CONCLUSION:

The Bidirectional LSTM turned out to be the best option for this project when compared with the CNN model, where accuracy plays a crucial role, because it uses sequential processing to capture subtle sentiment changes, its aptitude for applications requiring in-depth contextual analysis is validated by final training accuracy at **86.90%** and validation accuracy at **86.52%,** on the IMDB dataset. On the other hand, CNN's competitive recall and 2x faster training provided a workable option for latency-sensitive situations. However, both models showed shortcomings in managing ambiguity and sarcasm, this gap will be filled in future research using hierarchical attention networks and transformer-based designs (like BERT). By improving contextual reasoning while maintaining computing economy, these developments seek to close the gap between sentiment analysis performance and usefulness.

REFERENCES:

[1] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification.

[2] Kaggle

[3] Pennington, J. et al. (2014). GloVe: Global Vectors for Word Representation.

[4] Stanford Glove Embeddings

[5] Zhou, P. et al. (2020). Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. Neurocomputing.

[6] Joshi, A. et al. (2017). Are Word Embedding-based Features Enough for Sarcasm Detection.

[7] ChatGPT and ChatGPT Canvas.