**Social Media Sentiment Analysis**

**Supervised by:** Maysa Khalil**cheiter  
Name:** Hussein Maher

**1. Introduction**

**Background:**  
The rapid growth of user-generated content on platforms like YouTube has created vast opportunities—and challenges—for understanding public opinion. Sentiment analysis, a subfield of Natural Language Processing (NLP), uses machine learning (ML) and deep learning (DL) techniques to automatically classify text as positive, negative, or neutral. This project leverages recent advances in deep learning (specifically Conv1D and BiLSTM architectures) to analyze sentiment in Social media comments and provide insights into audience reactions.

**Problem Statement:**  
Existing tools for analyzing Social Media comments often rely on rule-based or shallow ML methods, which struggle with informal language, slang, emojis, and varied comment lengths. This project aims to build a robust DL-based sentiment classifier that handles the nuances of Social Media comment data.

**Objectives:**

* Develop a data pipeline to collect, clean, and preprocess YouTube comment data.
* Implement a Conv1D + BiLSTM model for sentiment classification.
* Compare the model's performance against baseline algorithms (e.g., logistic regression, SVM).
* Visualize results and conduct error analysis to uncover strengths and limitations.

**Scope:**

* **Included:** Comments from a curated Social Media dataset (via Kaggle); (positive/negative/neutral) sentiment classification; deep learning implementation in TensorFlow.
* **Excluded:** Multilingual processing; aspect-level sentiment; real-time deployment.

**2. Literature Review**

**Previous Work:**

* Rule-based lexicon approaches (VADER, TextBlob) for social media sentiment.
* ML classifiers (Naive Bayes, SVM) on short texts.
* DL architectures: CNN, LSTM, BiLSTM.

**Comparison:**

* Lexicon methods fail on sarcasm/slang and lack adaptability.
* Traditional ML needs extensive feature engineering, underperforms on unstructured comments.
* Pure LSTM models capture sequence but miss local n-gram patterns; CNN captures n-grams but misses long dependencies.

**Contribution:**  
This project fuses Conv1D to capture local phrase patterns and BiLSTM to model temporal dependencies, providing improved accuracy and robustness on Social Media comment sentiment.

**3. Methodology**

**Data Collection:**

* Source:

- “YouTube Comments Dataset” from Kaggle (publicly available).  
 -” Sentiment Analysis Dataset - Twitter” from Kaggle (publicly available).  
 - “YouTube Statistics” from Kaggle (publicly available).

* Contents: ~135,000 comments labeled positive/negative/neutral.
* Merger: merge the three dataset in one dataset, after simple edit on it to prepare it to  
   merge.

**Data Preprocessing:**

* Removal of punctuation, stop words, and emojis.
* Tokenization using TensorFlow's TextVectorization.
* Padding to fixed sequence length (e.g., 100 tokens).
* Word embeddings

**Model Selection:**

* **Justification:** Conv1D layers are effective at extracting local n-gram features; BiLSTM captures bidirectional context, beneficial for sentiment nuances.

**Model Architecture:**

Input -> Embedding Layer (160D) -> Conv1D (224 filters, kernel=5) -> BiLSTM (128 units) -> BiLSTM (32 units) -> Dropout(0.2)-> Dropout(0.3)

-> Dense (3 units, softmax).

Hyperparameters chosen based on literature, testing, and preliminary tuning.

**Training Strategy:**

* Loss: Categorical cross-entropy.
* Optimizer: Adam
* Metrics: Accuracy, Precision, Recall, F1-score.

**Validation Strategy:**

* Split: 80% train, 20% validation,
* Early stopping on validation loss (patience=3).
* checkpoint with monitor = 'val\_accuracy',

**4. Experiments**

**Experimental Setup:**

* Hardware: Google Colab
* Software: Python 3.8, TensorFlow 2.x, scikit-learn, Pandas, NumPy.

**Hyperparameter Tuning:**

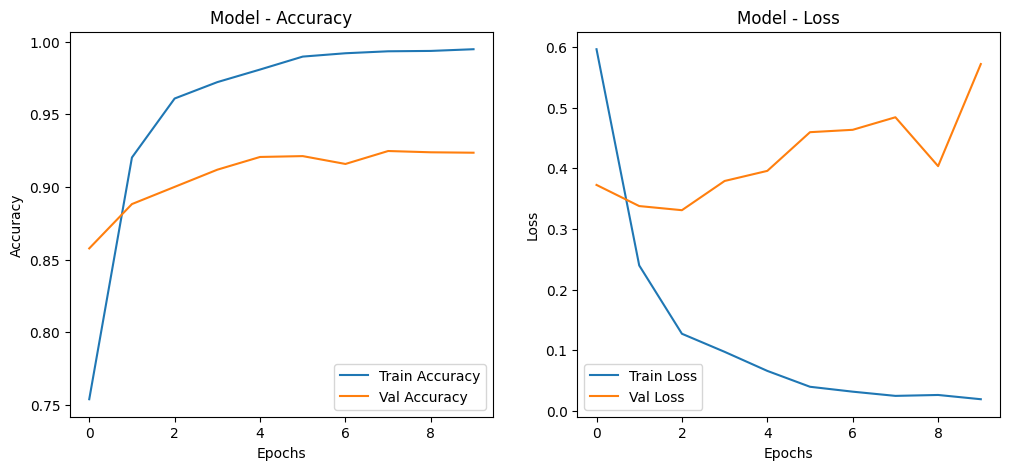
* Grid search over learning rates, Embedding(64 - 256), Conv1D(64 - 256), Bidirectional(32-128), Bidirectional(16-64), Dropout(0.1 - 0.5), Dropout(0.1 - 0.5), batch sizes (32).

**5. Results and Discussion**

**Performance Metrics:**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| --- | --- | --- | --- | --- |
| Logistic Regression (TF-IDF) | 0.85 | 0.89 | 0.78 | 0.83 |
| SVM (TF-IDF) | 0.93 | 0.95 | 0.86 | 0.90 |
|  |  |  |  |  |
|  |  |  |  |  |

**Visualizations for the best model:**



**Error Analysis:**

* Common errors in short comments with sarcasm/emojis.
* Misclassifications when comments contain mixed sentiments (e.g., "Great video, but audio was bad").

**6. Conclusion and Future Work**

**Summary of Findings:**  
The Conv1D + BiLSTM model achieved 93% accuracy

**Limitations:**

* Struggles with sarcasm, code-switching, and nuanced comments.

**Future Directions:**

* Extend to multiclass sentiment.
* Incorporate attention mechanisms or Transformers.
* Deploy as a real-time sentiment monitoring dashboard.

**7. References**

1. Kaggle YouTube Comments Dataset. URL: [click](https://www.kaggle.com/datasets/atifaliak/youtube-comments-dataset/data)
2. Pennington, J. et al. (2014). GloVe: Global Vectors for Word Representation. EMNLP.
3. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. EMNLP.
4. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.
5. Kaggle Sentiment-analysis-dataset . URL : [click](https://www.kaggle.com/datasets/pythonafroz/sentiment-analysis-dataset)

6. Kaggle إoutube-statistics. URL : [click](https://www.kaggle.com/datasets/advaypatil/youtube-statistics)