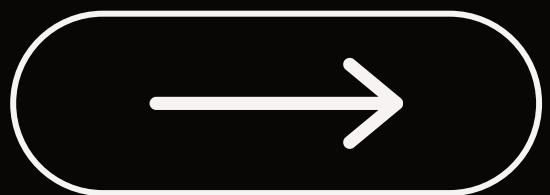




AI & DS Experiential Learning



Introduction to NLP & Sarcasm



What is NLP?

Natural Language Processing (NLP) allows computers to understand, interpret, and generate human language for tasks like sentiment analysis and translation.

Why Sarcasm is Challenging

Sarcasm often expresses the opposite of literal meaning, making it difficult for basic models to detect intent accurately.

Impact on Sentiment Analysis

Undetected sarcasm can mislead sentiment classifiers, especially in reviews, social media, and public opinion mining.

Our Goal

To use a Hierarchical BERT architecture to improve sarcasm detection accuracy by combining language modeling with deep contextual understanding.



Rise of Transformer Models in NLP

Rise of Transformer Models in NLP

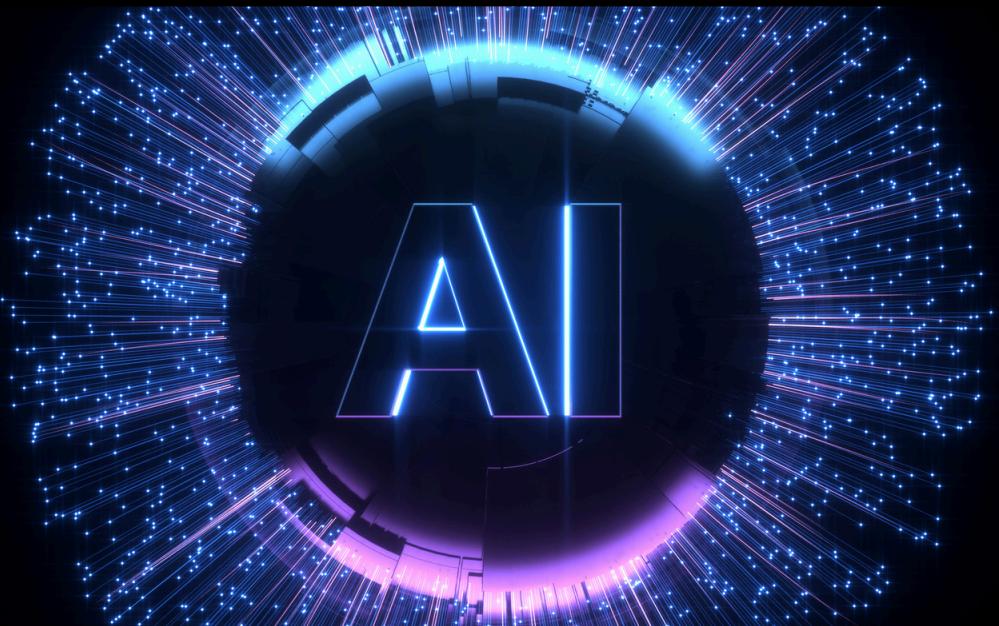
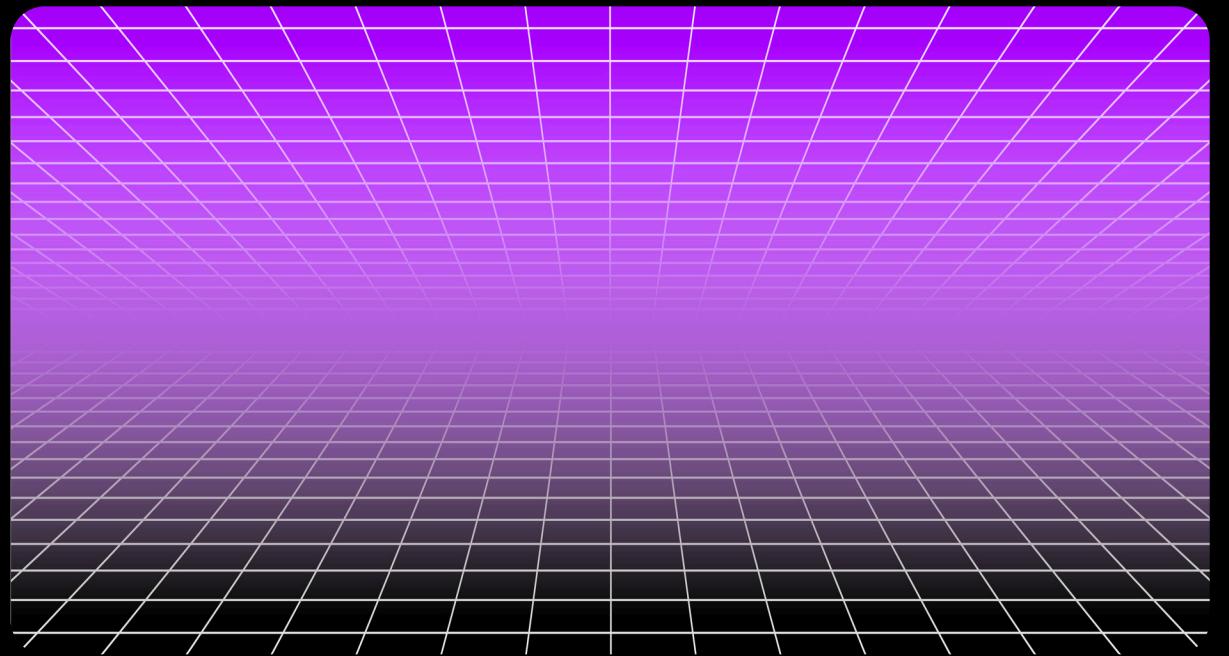
Transformers have transformed how machines understand human language, powering advanced NLP applications like sarcasm detection.

Contextual Understanding

Transformer models like BERT understand context by analyzing relationships between all words in a sentence simultaneously, improving language comprehension.

Self-Attention Mechanism

Self-attention allows the model to focus on key parts of the input, enabling better interpretation of complex, nuanced language like sarcasm.



Understanding the Dataset

01 REDDIT COMMENTS

The dataset comprises over 1.3 million Reddit comments collected using the “\s” tag to identify sarcasm in online conversations.

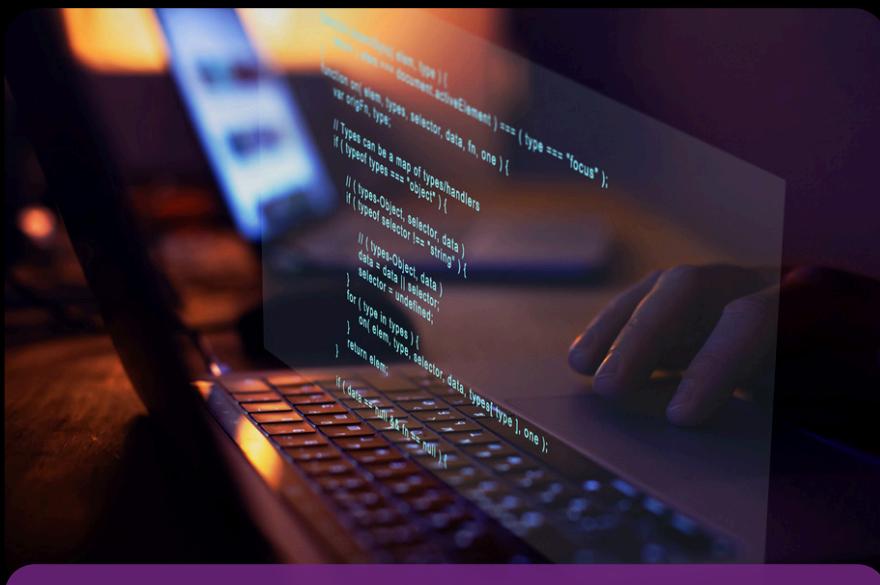
02 BALANCED VERSION

For training, a balanced subset is used with an equal number of sarcastic and non-sarcastic samples to avoid bias during learning.

03 ADDITIONAL CONTEXTUAL DATA

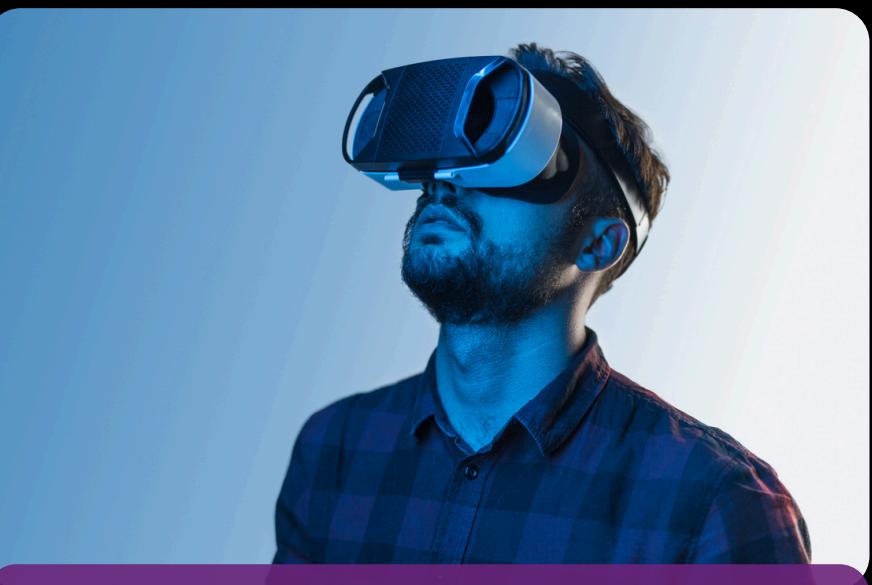
Includes fields like parent comment, author, score, and subreddit — enabling deeper analysis and future context-aware extensions.

Data Preprocessing



Text Cleaning

Understanding the skills required for AI-driven jobs is crucial for tailoring training programs and addressing workforce needs.



Tokenization with BERT

Used BERT's word-piece tokenizer to break down input text into tokens that match its pre-trained vocabulary.



Null & Noise Handling

Dropped rows with null values or unusable data fields (e.g., empty comments, invalid characters).



Train-Test Split

Dataset split into 80% training and 20% testing to ensure a robust evaluation of model performance.

Hierarchical BERT Architecture

BERT EMBEDDINGS

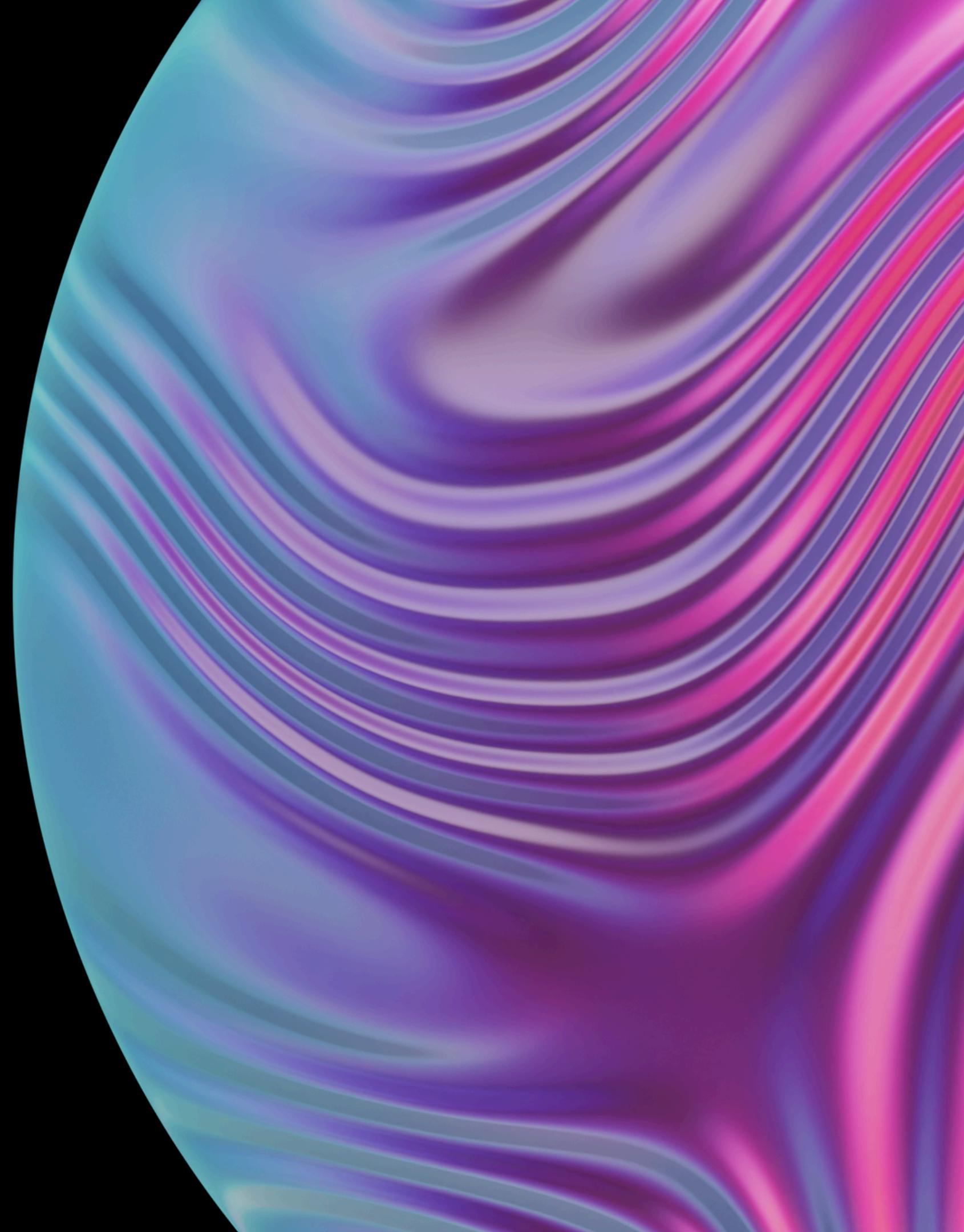
Pretrained BERT provides contextual word embeddings that understand both the meaning and position of words in a sentence.

BILSTM LAYER

Bidirectional LSTM captures forward and backward sequential dependencies, enhancing temporal understanding of text flow

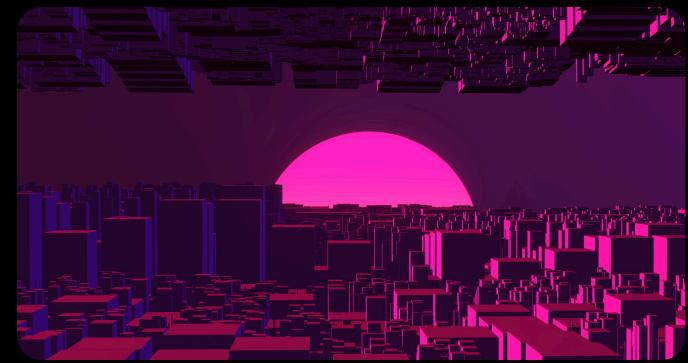
CNN LAYER

A Convolutional Neural Network identifies local features and phrase-level sarcasm patterns not captured by sequence models alone.



Model Training and Parameters

This project combines cutting-edge NLP and deep learning methods, including BERT embeddings and sequential models like BiLSTM, to detect sarcasm effectively.



Training Framework

Built using TensorFlow and HuggingFace Transformers, combining flexibility with powerful pretrained models.

Loss Function

Binary Crossentropy was used, ideal for two-class classification problems like sarcasm detection.

Data Split & Epochs

Data split into 80% training and 20% testing. Epochs tuned using validation performance to prevent overfitting.

Deployment using Streamlit

Why Streamlit?

Streamlit is an open-source Python framework that allows fast deployment of machine learning models through a user-friendly web interface.

Model Integration

The trained BERT-based model is loaded using transformers and connected to Streamlit with a simple API, enabling live predictions.

Interactive UI

Users can input custom text and get real-time sarcasm predictions with just one click, making the model accessible even to non-tech users.

Hosting Options

The app can be deployed on Streamlit Cloud, Render, or Heroku, allowing anyone to access the sarcasm detection tool from a browser.

Demonstration

Introducing ChatGPT | OpenAI

Sarcasm comment examples

Sarcasm Detection using Hierarchical BERT

Enter a parent comment and a reply to detect sarcasm.

Parent Comment

"This player is really struggling this season."

Comment

"Yeah, he's totally the MVP this year. Just wow."

Predict

Prediction: Not Sarcastic 😊

Sarcasm Detection using Hierarchical BERT

Enter a parent comment and a reply to detect sarcasm.

Parent Comment

This player is really struggling this season.

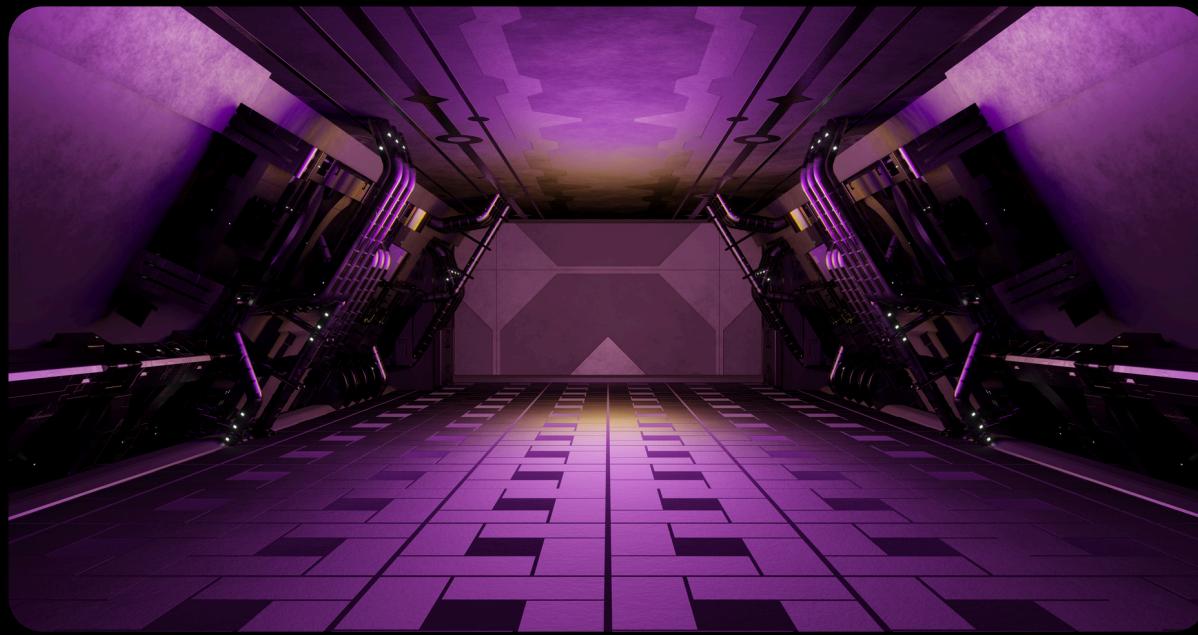
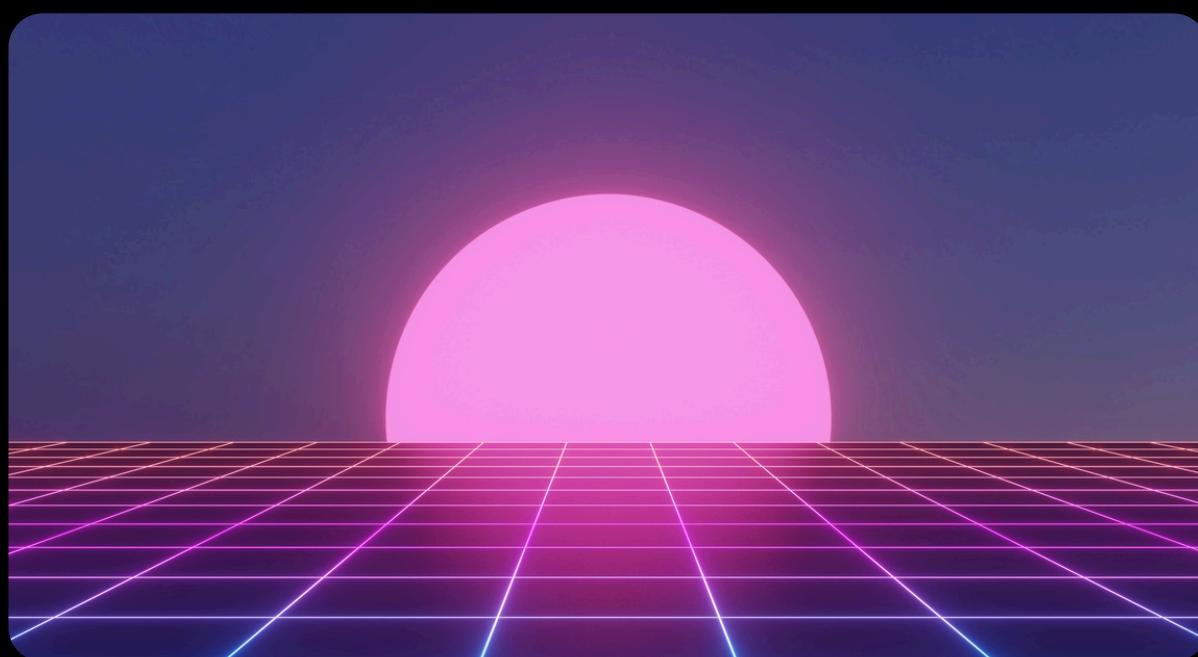
Comment

Yeah, he's totally the MVP this year. Just wow.

Predict

Prediction: Sarcastic 😬

Future Enhancements



01 CONTEXT-AWARE MODELING

Incorporate parent and thread-level comment history to give the model more conversational context when detecting sarcasm.

02 EXPLORE OTHER ARCHITECTURES

Experiment with transformer variants like RoBERTa, DeBERTa, or ALBERT for potential improvements in accuracy and efficiency.

03 WEB/API INTEGRATION

Deploy as a plug-in or public API that can be integrated into content moderation systems, social media tools, or comment platforms.

Conclusion :

Navigating the Transforming Sarcasm Detection with Advanced AI Techniques -Driven Job Landscape

This project demonstrates a Hierarchical BERT model for sarcasm detection in online text. By integrating BERT with BiLSTM and CNN layers, it captures nuanced meanings and significantly outperforms traditional methods. Real-time deployment via Streamlit allows user interaction, enhancing sentiment analysis and content moderation capabilities.

