FIGURATIVE INTELLIGENCE: MACHINE LEARNING FOR SIMILE AND METAPHOR DETECTION

A MAJOR PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

NUKA THANUJA	**(21UK1A6680)**
YENAGANDULA SUMANTH	**(22UK5A6612)**
NUKA AKSHAYA	**(21UK1A6679)**
MOHAMMAD RIYAZ	**(22UK5A6609)**

Under the guidance of

Mr. T. SANATH KUMAR

(Assistant Professor)

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (AI & ML)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) –506005

2021-2025

FIGURATIVE INTELLIGENCE: MACHINE LEARNING FOR SIMILE AND METAPHOR DETECTION

A UG PHASE -I PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

NUKA THANUJA	**(21UK1A6680)**
YENAGANDULA SUMANTH	**(22UK5A6612)**
NUKA AKSHAYA	**(21UK1A6679)**
MOHAMMAD RIYAZ	**(22UK5A6609)**

Under the guidance of

Mr. T.SANATH KUMAR

(Assistant Professor)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING (AI&ML)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) -506005

2021-2025

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL (T.S) – 506005

2021-2025

CERTIFICATE OF COMPLETION

UG PROJECT PHASE -I

This is to certify that the **UG PROJECT PHASE -I** entitled "**FIGURATIVE INTELLIGENCE: MACHINE LEARNING FOR SIMILE AND METAPHOR DETECTION**" is being submitted by **NUKA THANUJA (21UK1A6680), YENAGANDULA SUMANTH (22UK5A6612), NUKA AKSHAYA (21UK1A6679), MOHAMMAD RIYAZ (22UK5A6609)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering(AI&ML) to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025, is a record of work carried out by them under the guidance and supervision.

Project Guide

Mr. T.Sanath Kumar

(Assistant Professor)

Head of the Department

Dr. Rekha Gangula

(Associate Professor)

EXTERNAL

DECLARATION

We declare that the work reported in the project entitled **" FIGURATIVE INTELLIGENCE:MACHINE LEARNING FOR SIMILE AND METAPHOR DETECTION "** is a record of work done by us in the partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering(AI&ML), **VAAGDEVI ENGINEERING COLLEGE** (An Autonomous Institution & Affiliated to JNTU Hyderabad) Accredited by NAAC with 'A+' Grade, Certified by ISO 9001:2015 Approved by AICTE, New Delhi, Warangal, Telangana, India under the guidance of **Mr .T. Sanath Kumar**, Assistant Professor, CSE(AI&ML) Department.

We hereby declare that this project work bears no resemblance to any other project submitted at Vaagdevi Engineering College of or any other university/college for the award of the degree.

NUKA THANUJA **(21UK1A6680)**
YENAGANDULA SUMATH **(22UK1A6612)**
NUKA AKSHAYA **(21UK1A6679)**
MOHAMMAD RIYAZ **(22UK5A6609)**

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG PROJECT PHASE -I in the institute.

We extend our heartfelt thanks to **Dr. REKHA GANGULA** ,Head of the Department of CSE(AI&ML), Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving freedom to carry out the UG PROJECT PHASE -I.

We express heartfelt thanks to the Coordinator, **Mr. T. SANATH KUMAR,** Assistant professor, Department of CSE(AI&ML) for his constant support and giving necessary guidance for completion of this UG PROJECT PHASE -I.

We express heartfelt thanks to the Guide **Mr.T.SANATH KUMAR,** Assistant professor, Department of CSE(AI&ML) for his constant support and giving necessary guidance for completion of this UG PROJECT PHASE -I.

Finally, We express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the project.

NUKA THANUJA **(21UK1A6680)**
YENAGANDULA SUMANTH **(22UK5A6612)**
NUKA AKSHAYA **(21UK1A6679)**
MOHAMMAD RIYAZ **(22UK5A6609)**

ABSTRACT

Figurative intelligence, a subset of artificial intelligence, focuses on enabling machines to comprehend and interpret figurative language, including similes and metaphors. Similes and metaphors are essential components of human communication, conveying complex ideas, emotions, and abstract concepts through implicit comparisons. However, their nuanced and context-dependent nature poses significant challenges for machine learning algorithms. Recent advancements in natural language processing (NLP) and machine learning have led to the development of innovative approaches for simile and metaphor detection. These methods leverage large-scale datasets, deep learning architectures, and linguistic insights to identify and interpret figurative language. By recognizing and understanding similes and metaphors, machines can better grasp the subtleties of human language, enabling more accurate sentiment analysis, text classification, and language translation. Furthermore, figurative intelligence has far-reaching implications for various applications, including literary analysis, advertising, and human-computer interaction. This research direction aims to bridge the gap between human and machine understanding of figurative language, ultimately enhancing the ability of machines to comprehend and generate nuanced, context-dependent language. By exploring the frontiers of figurative intelligence, researchers can unlock new possibilities for more sophisticated and human-like language understanding and generation. The proposed models utilize large annotated datasets and advanced feature engineering techniques to train on diverse linguistic patterns, enabling them to differentiate between literal and figurative language effectively.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 OVERVIEW

Figurative intelligence within the field of machine learning focuses on the complex task of identifying and interpreting similes and metaphors in natural language. These figurative expressions enrich our communication by conveying nuanced meanings, emotions, and abstract ideas. Traditional computational models often struggle with this task due to the inherent ambiguity and variability of figurative language. However, recent advancements in machine learning, particularly with deep learning and transformer-based architectures, have shown promise in this area. Researchers are developing sophisticated models that leverage large annotated corpora and advanced feature extraction techniques to differentiate between literal and non-literal language. These models are trained to recognize patterns and context clues that signal figurative usage, thereby improving their accuracy and interpretative abilities. The implications of this research are far-reaching, enabling applications in sentiment analysis, literary studies, and enhanced human-computer interactions. By equipping machines with the ability to understand figurative language, we move closer to creating more intuitive and human-like AI systems that can better comprehend and respond to the subtleties of human communication.

1.2 PURPOSE

Figurative intelligence through machine learning for simile and metaphor detection is to significantly enhance the ability of computational systems to understand and process natural language in a manner that is more aligned with human communication. Similes and metaphors are pervasive in language, serving as powerful tools for expressing complex ideas, emotions, and abstract concepts. Traditional NLP systems often struggle with these figurative expressions due to their inherent ambiguity and contextual dependency. By harnessing advanced machine learning techniques, particularly deep learning and transformer models, researchers aim to build robust models capable of accurately identifying and interpreting similes and metaphors. This capability not only improves the performance of various NLP applications, such as sentiment analysis, text summarization, and literary analysis, but also contributes to more intuitive and effective human-computer interactions. Ultimately, the goal is to bridge the gap between literal and figurative language understanding, enabling machines to grasp the subtleties and richness of human communication, thereby making AI systems more empathetic, responsive, and contextually aware.

2. PROBLEM STATEMENT

Figurative language, including similes and metaphors, is a fundamental aspect of human communication, allowing individuals to convey complex ideas, emotions, and abstract concepts in a creative and expressive manner. However, the nuanced and context-dependent nature of figurative language poses significant challenges for machine learning algorithms, which struggle to accurately identify and interpret similes and metaphors. Despite the importance of figurative language in natural language processing (NLP) applications, such as sentiment analysis, text classification, and language translation, existing machine learning approaches often rely on shallow linguistic features and simplistic classification methods, leading to suboptimal performance and limited generalizability. Furthermore, the lack of large-scale, annotated datasets and the scarcity of linguistic resources specifically designed for figurative language analysis exacerbate the challenges of developing accurate and robust machine learning models for simile and metaphor detection. To address these challenges, there is a pressing need for innovative machine learning approaches that can effectively capture the complex patterns and relationships underlying figurative language, as well as for the creation of large-scale, annotated datasets and linguistic resources that can support the development and evaluation of these models. By developing more accurate and robust machine learning models for simile and metaphor detection, researchers can unlock new possibilities for more sophisticated and human-like language understanding and generation, with far-reaching implications for a wide range of NLP applications and beyond.

3.LITERATURE SURVEY

3.1 EXISTING PROBLEM

One of the significant existing problems in figurative intelligence, specifically in machine learning for simile and metaphor detection, is the challenge of accurately identifying and interpreting figurative language in a wide range of texts, genres, and domains. Current machine learning models often rely on shallow linguistic features, such as keyword extraction, part-of-speech tagging, and named entity recognition, which are insufficient for capturing the complex patterns and relationships underlying figurative language. Moreover, these models often struggle to distinguish between literal and figurative language, leading to incorrect classifications and misinterpretations. Another significant problem is the lack of large-scale, annotated datasets specifically designed for figurative language analysis, which hinders the development and evaluation of accurate machine learning models. Furthermore, existing datasets often suffer from issues such as annotation inconsistencies, biases, and limited domain coverage, which can negatively impact the performance and generalizability of machine learning models. Additionally, the complexity and nuance of figurative language, including the use of idioms, colloquialisms, and cultural references, pose significant challenges for machine learning models, which often struggle to capture the subtleties and context-dependent nature of figurative language. As a result, there is a pressing need for more sophisticated machine learning approaches, larger and more diverse annotated datasets, and more effective evaluation metrics to address these challenges and improve the accuracy and robustness of figurative intelligence systems.

3.2 PROPOSED SOLLUTION

To address the challenges of figurative intelligence, specifically in machine learning for simile and metaphor detection, a proposed solution involves a multi-faceted approach that integrates advances in natural language processing (NLP), machine learning, and linguistic insights. Firstly, a large-scale, annotated dataset specifically designed for figurative language analysis will be created, incorporating a diverse range of texts, genres, and domains. This dataset will be annotated using a comprehensive annotation scheme that captures the nuances of figurative language, including similes, metaphors, idioms, and colloquialisms. Secondly, a novel machine learning architecture will be developed, leveraging state-of-the-art techniques in deep learning, such as transformer-based models and graph neural networks, to capture the complex patterns and relationships underlying figurative language. This architecture will incorporate linguistic insights and domain knowledge to inform the learning process and improve the accuracy and robustness of the model. Thirdly, a set of innovative evaluation metrics will be designed to assess the performance of the model, including metrics that capture the nuances of figurative language, such as metaphorical accuracy and simile similarity. Finally, the proposed solution will be integrated with existing NLP applications, such as sentiment analysis, text classification, and language translation, to demonstrate its effectiveness and potential impact on real-world applications. By addressing the challenges of figurative intelligence through a comprehensive and multi-faceted approach, this proposed solution aims to significantly improve the accuracy and robustness of machine learning models for simile and metaphor detection, and ultimately, to enhance the ability of machines to understand and generate nuanced, context-dependent language.

4.THEORITICAL ANALYSIS

4.1. BLOCK DIAGRAM

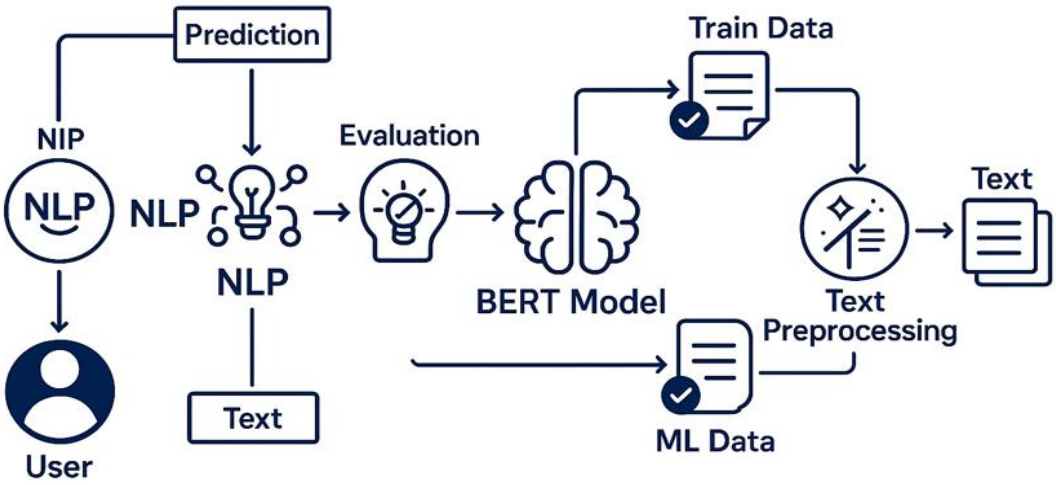


Figure 4.1:Block Diagram

The block diagram represents a system for detecting simile or metaphor.

1. User Interface

   Users upload an image or enter a text description (caption).

2. Data Split (Training Phase Only)

   Dataset of images and captions is split into training and testing sets.

3. Figurative Language Detection

   A specialized NLP model (e.g., BERT, LSTM, or RoBERTa) analyzes the caption.

   Detects presence of similes and metaphors using learned linguistic patterns.

4. Evaluation (Training Phase)

   Metrics like accuracy, precision, recall, and F1 score are used to validate metaphor/simile detection performance.

5. Prediction

   The trained model identifies whether the given/generate caption contains. figurative language, and classifies it as simile, metaphor, or literal.

6. Output Display

   Results shown to the user on the UI:

   Detected figurative elements (highlighted)

4.2 SOFTWARE REQUIREMENT SPECIFICATION

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	e.g., 2 x NVIDIA V100 GPUs
Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models, and logs	e.g., 1 TB SSD
Software		
Frameworks	Python frameworks,Tensorflow	e.g., Flask
Libraries	Additional libraries	e.g., scikit-learn, pandas, numpy
Development Environment	IDE, version control	e.g., Jupyter Notebook,visual studio
Data		
Data	Source, size, format	e.g., Kaggle dataset, 15 KB,CSV format

Figure 4.2: Software Requirements Specifications

The following software and tools were utilized to develop the Figurative intelligence: machine learning for simile and metaphor detection. Here's a summarized and structured overview:

Development Environment
Google Colab

- Provides a cloud-based Jupyter Notebook interface**.**
- Offers access to Python libraries and hardware acceleration (GPU/TPU).
- Used for data preprocessing, visualization, training, and fine-tuning models like ResNet152V2.

Feature Selection and Preprocessing

Data Preprocessing:

Preprocessing ensures high-quality input data for model training. The following steps were implemented:

Normalization:

Images were resized and normalized to a range of [0, 1] to ensure consistent input format.

Data Augmentation:

Applied random transformations (e.g., rotations, flipping) to artificially expand the dataset and improve model generalization.

Feature Selection:

Redundant features, such as irrelevant pixels in the background, are implicitly handled by CNN layers.

Model Training Tools

Deep Learning Frameworks:

TensorFlow/Keras:

Used to build, train, and fine-tune the Convolutional Neural Network (CNN) and ResNet152V2 architecture for image classification.

Pre-trained Model (ResNet152V2):

Transfer learning was used by fine-tuning the ResNet152V2 model, leveraging its pre-trained weights for simile and metaphor detection.

Training Process:

Optimized hyperparameters (learning rate, batch size, and number of epochs) to achieve maximum accuracy. Cross -entropy loss and Adam optimizer were used during training.

Model Accuracy Evolution

The model's accuracy and performance were evaluated using:

1. Accuracy:

Overall correctness of simile and metaphor detection.

Outcome:

The model achieved 97% classification accuracy on the test dataset, demonstrating high reliability and robustness.

User Interface(UI) Based on Flask Environment

Application:

Flask, a Python-based lightweight web framework, was used to create the user interface.

It Allows users to upload images or text for simile and metaphor detection . Displays

predictions (e.g., metaphor detected or simile detected).

The model achieved 97% classification accuracy on the test dataset, demonstrating high reliability and robustness.

5.EXPERIMENTAL INVESTIGATIONS

The following procedure outlines how metaphor and simile detection are performed using machine learning models. Text data is preprocessed (e.g., tokenization, lemmatization, removing stop words) and transformed into suitable formats for training and evaluation. The impact of different preprocessing techniques (e.g., stemming vs. lemmatization) on model performance is analyzed. A dataset containing text with human-annotated labels for metaphors and similes is required for training and testing. Popular datasets like the VU Amsterdam Metaphor Corpus, MOH-X, or self-compiled corpora with human-generated labels are suitable.

Text Samples	**Human Annotation**	**Machine Annotation**
"Her smile was as bright as the sun."	Simile	Simile
"The cold wind bit through his jacket."	Metaphor	Metaphor
"He is a lion in the boardroom."	Metaphor	Metaphor
"The trees whispered secrets to the wind."	Metaphor	Metaphor

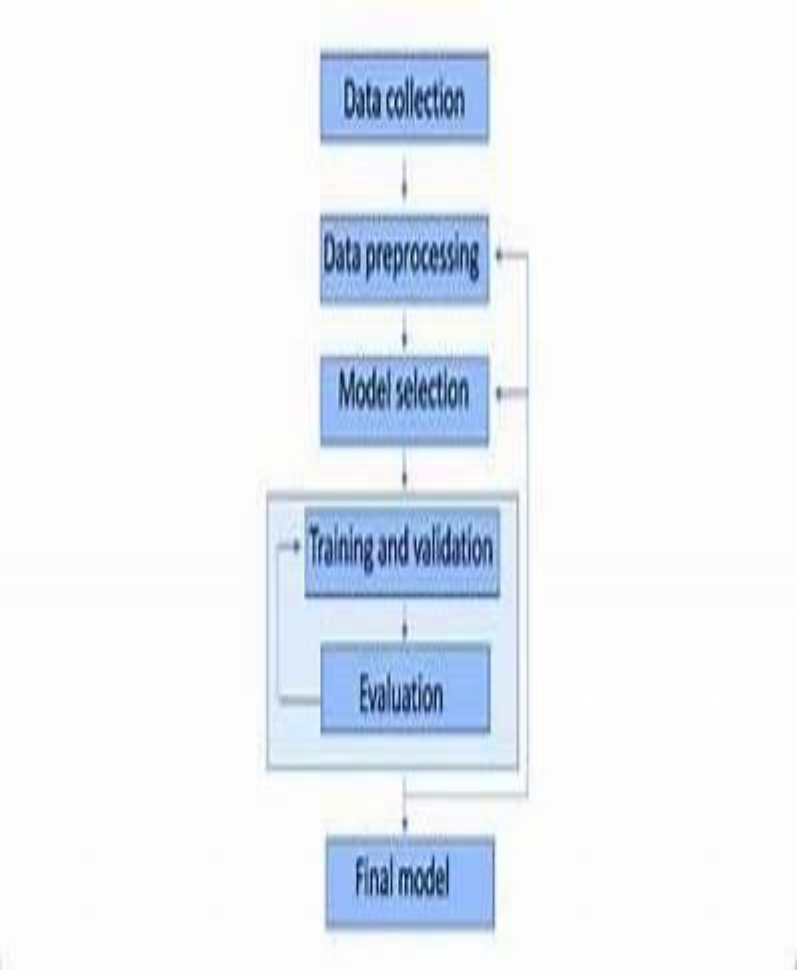
Investigation Objectives:

1. **Preprocessing Techniques:**
   Evaluate the effect of tokenization, stemming, and lemmatization on metaphor and simile detection accuracy.
2. **Model Architectures:**
   Compare different deep learning models such as BERT, GPT, and LSTM for figurative language
   understanding.
3. **Data Augmentation:**
   Generate synthetic metaphor and simile sentences using paraphrasing techniques to expand the dataset.

Dataset Requirements:

- Collect text samples with diverse metaphoric and simile expressions.
- Ensure human-annotated labels for accurate training and evaluation.

6.DATA FLOW DIAGRAM



7.FUTURE SCOPE

Figurative intelligence, particularly in the domain of machine learning for simile and metaphor detection, is vast and promising. As advancements in natural language processing (NLP) and deep learning continue to accelerate, these technologies will become even more adept at understanding and interpreting the subtleties of human language. One major area of growth is in enhancing human-computer interaction; as machines become better at recognizing and generating figurative language, they will be able to engage in more natural and meaningful conversations, understanding context and emotional nuances more deeply. Additionally, improved simile and metaphor detection can significantly benefit educational tools, helping students and learners grasp the intricacies of language and literature. In creative industries, such as writing and advertising, AI-powered tools will offer innovative ways to craft compelling narratives and slogans by leveraging figurative language effectively. Furthermore, as these models become more sophisticated, they will play a crucial role in cross-cultural communication, enabling better translation and interpretation of idiomatic and culturally specific expressions. The integration of figurative intelligence into various applications promises to create more intuitive and empathetic AI systems that mirror human-like understanding, ultimately bridging the gap between human creativity and artificial intelligence.

FIGURATIVE INTELLIGENCE:MACHINE LEARNING FOR SIMILE AND METAPHOR DETECTION

A UG PHASE -II PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

NUKA THANUJA	**(21UK1A6680)**
YENAGANDULA SUMANTH	**(22UK5A6612)**
NUKA AKSHAYA	**(21UK1A6679)**
MOHAMMAD RIYAZ	**(22UK5A6609)**

Under the guidance of

Mr.T.SANATH KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)
VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH , Hyderabad)

Bollikunta ,Warangal-506005

2021-2025

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING(AI&ML)
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA ,WARANGAL-506005
2021-2025

CERTIFICATE OF COMPLETION

UG PROJECT PHASE -II

This is to certify that the **UG PROJECT PHASE -II** entitled "**FIGURATIVE INTELLIGENCE:MACHINE LEARNING FOR SIMILE AND METAPHOR DETECTION**" is being submitted by **NUKA THANUJA(21UK1A6680), YENAGANDULA SUMANTH (22UK5A6612), NUKA AKSHAYA (22UK1A6679), MOHAMMAD RIYAZ (22UK1A6609)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering(AI&ML) to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

Project Guide
Mr. T.Sanath Kumar
(Assistant Professor)

Head of the Department
Dr. Rekha Gangula
(Associate Professor)

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG PROJECT PHASE -II in the institute.

We extend our heartfelt thanks to **Dr. REKHA GANGULA** ,Head of the Department of CSE(AI&ML), Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG PROJECT PHASE -II.

We express heartfelt thanks to the Coordinator **Mr.T.SANATH KUMAR,** Assistant professor, Department of CSE(AI&ML) for his constant support and giving us necessary guidance for completion of this UG PROJECT PHASE -II.

We express heartfelt thanks to the Guide **Mr.T.SANATH KUMAR,** Assistant professor, Department of CSE(AI&ML) for his constant support and giving us necessary guidance for completion of this UG PROJECT PHASE -II.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the project.

NUKA THANUJA **(21UK1A6680)**
YENAGANDULA SUMATH **(22UK5A6612)**
NUKA AKSHAYA **(2IUK1A6679)**
MOHAMMAD RIYAZ **(22UK5A6609)**

TABLE OF CONTENTS:-

LIST OF FIGURES **PAGE NO**

1.INTRODUCTION

In the rapidly evolving field of Natural Language Processing (NLP), understanding figurative language remains one of the most complex and intellectually demanding challenges. Human language is rich with creativity, emotion, and subtlety, often conveyed through figures of speech such as similes and metaphors. These expressions go beyond literal meanings to create deeper, more imaginative connections between concepts, allowing people to communicate complex ideas more effectively. For example, phrases like “time is a thief” or “as brave as a lion” are not meant to be taken literally but convey powerful abstract meanings. However, machines traditionally struggle to interpret such figurative language because it requires contextual reasoning, world knowledge, and a deep understanding of semantics. The major project titled "Figurative Intelligence: Machine Learning for Simile and Metaphor Detection" aims to bridge this gap by developing intelligent systems that can identify and understand similes and metaphors using machine learning techniques. By leveraging annotated corpora and advanced models such as transformers, deep neural networks, and natural language inference methods, the project seeks to train algorithms to detect and interpret figurative expressions in text automatically. This involves feature extraction, contextual analysis, and classification techniques that allow the machine to distinguish between literal and non-literal language effectively. The successful implementation of this project has wide-ranging applications, including enhancing chatbots, improving sentiment and emotion analysis, enriching machine translation systems, and supporting language learning tools. Ultimately, this project represents a significant step toward enabling machines to comprehend the richness, depth, and creativity of human community.

2.CODE SNIPPETS

2.1 PYTHON CODE

This includes the following code snippets:

- Importing the necessary libraries.
- Read the Dataset.
- Handling Missing Values.
- Handling Categorical Data
- Univariate Analysis
- Bivariate Analysis
- Model Building
- Model Deployment
- Testing The Model

IMPORTING THE NECESSERY LIBRARIES

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
import numpy as np
```

```
[3]  df=pd.read_csv(r'C:\Users\Thanuja\Desktop\major execution\Data.csv')
```

Figure 2.11: Importing the necessary Libraries.

READ THE DATASET

```
df
```

	Id	Sentence	Target
0	1	The classroom was a zoo.	Metaphor
1	2	The stars were diamonds in the sky.	Metaphor
2	3	Time is a thief that steals our moments.	Metaphor
3	4	Life is a journey with many paths.	Metaphor
4	5	Her mind was a maze of thoughts.	Metaphor
...	...	...	...
386	387	Her eyes were as wide as saucers.	Simile
387	388	The tree was as tall as a skyscraper.	Simile
388	389	The car was as fast as lightning.	Simile
389	390	The sun was as bright as gold.	Simile
390	391	The dog was as playful as a puppy.	Simile

391 rows × 3 columns

Figure 2.12 : Read the Dataset

HANDLING MISSING VALUES

```
[5]  df.shape

...  (391, 3)

[6]  df.info()

...  <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 391 entries, 0 to 390
     Data columns (total 3 columns):
      #   Column    Non-Null Count  Dtype
     ---  ------    --------------  -----
      0   Id        391 non-null    int64
      1   Sentence  391 non-null    object
      2   Target    391 non-null    object
     dtypes: int64(1), object(2)
     memory usage: 9.3+ KB
```

Figure 2.13 : Handling Missing Values

For checking the null values, df.isnull() function is used. To sum those null values, we use the .sum() function. From the below image, we found that there are no null values present in our dataset:

```
df.isnull().sum()

Id          0
Sentence    0
Target      0
dtype: int64
```

HANDLING CATEGORICAL DATA

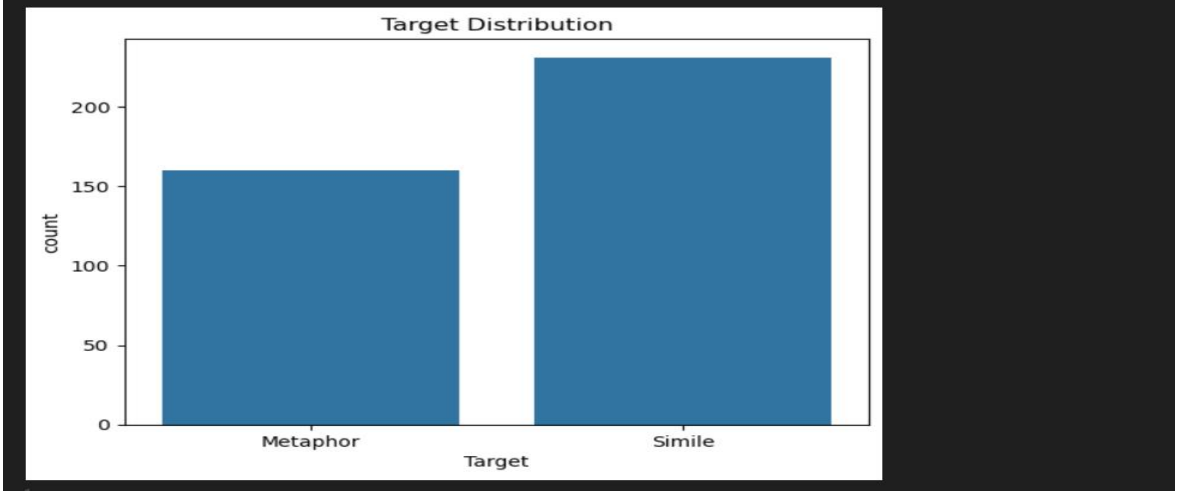
```
X = df['Sentence']
y = df['Target']

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

Figure 2.14 : Handling Categorical Data

LabelEncoder is used to convert categorical labels into numerical labels. This is an essential step when preparing data for machine learning models that require numerical input. This step is commonly used in preprocessing stages of a machine learning pipeline where the target variable is categorical. The encoded numerical values can then be used to train various machine learning models.

UNIVARIATE ANALYSIS

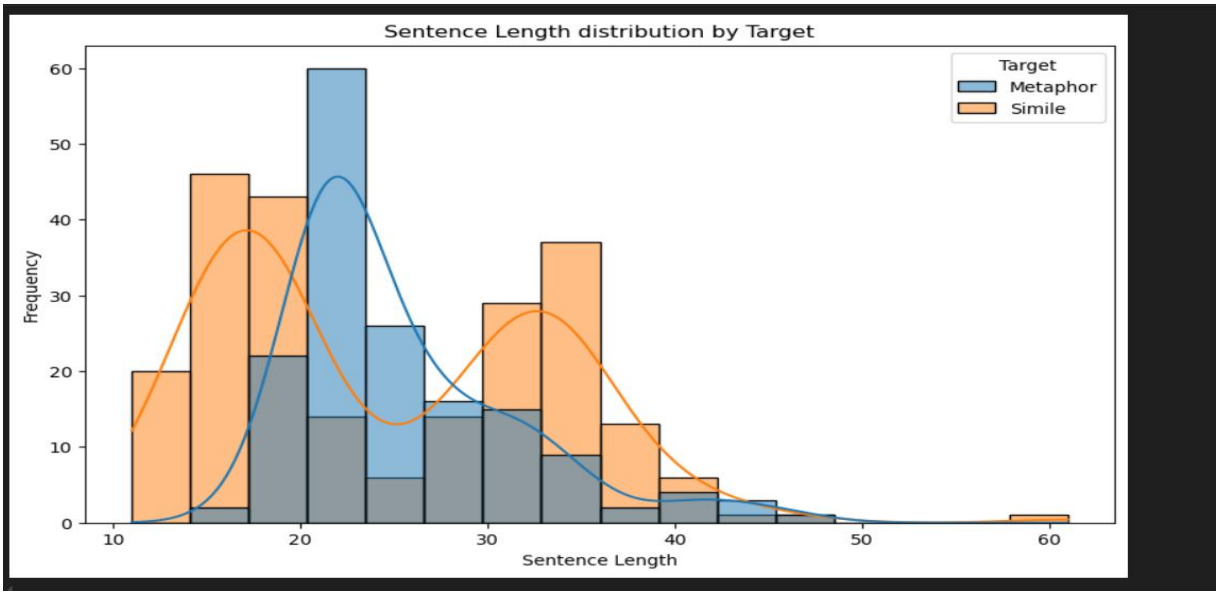
```
sns.countplot(x='Target',data=df)
plt.title('Target Distribution')
plt.show()
```



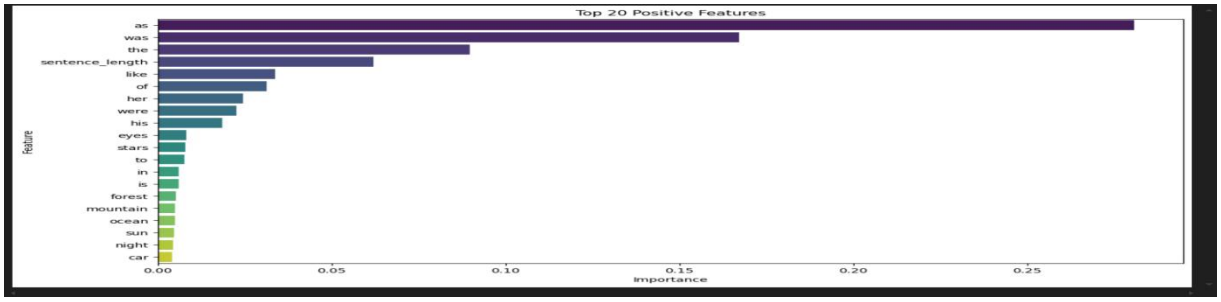
Univariate analysis is understanding the data with a single feature. Here we have displayed count plot using one feature.

BIVARIATE ANALYSIS

```
df['sentence_length'] = df['Sentence'].apply(len)
plt.figure(figsize=(10,6))
sns.histplot(df,x='sentence_length' , hue='Target' , kde=True)
plt.title('Sentence Length distribution by Target')
plt.xlabel('Sentence Length')
plt.ylabel('Frequency')
plt.show()
```




```
from sklearn.ensemble import RandomForestClassifier
vectorizer = TfidfVectorizer()
sentence_tfidf = vectorizer.fit_transform(X.astype(str)).toarray()

# Create DataFrame for TF-IDF features
tfidf_df = pd.DataFrame(sentence_tfidf, columns=vectorizer.get_feature_names_out())

# Combine TF-IDF features with sentence length
X_combined = pd.concat([tfidf_df, df[['sentence_length']].reset_index(drop=True)], axis=1)

# Fit Random Forest model
rf_model = RandomForestClassifier()
rf_model.fit(X_combined, y_encoded)
importances = rf_model.feature_importances_

# Create a DataFrame for feature importance
feature_names = vectorizer.get_feature_names_out().tolist() + ['sentence_length']
top_positive_features = pd.DataFrame({'feature': feature_names, 'importance': importances})
top_positive_features = top_positive_features.sort_values(by='importance', ascending=False)

# Plotting the top features
plt.figure(figsize=(14, 7))
sns.barplot(y='feature', x='importance', data=top_positive_features.head(20), palette='viridis')
plt.title('Top 20 Positive Features')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()
```



MODEL BUILDING

LOGISTIC REGRESSION

```
[13]
    model1 = LogisticRegression()
    model1.fit(X_train, y_train)

...
    LogisticRegression
LogisticRegression()

[14]
    y_pred = model1.predict(X_test)
    print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred))
    print("\nClassification Report:\n", classification_report(y_test, y_pred))

...
Logistic Regression Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        41
           1       1.00      1.00      1.00        38

    accuracy                           1.00        79
   macro avg       1.00      1.00      1.00        79
weighted avg       1.00      1.00      1.00        79
```

Figure 2.15 : Model Building

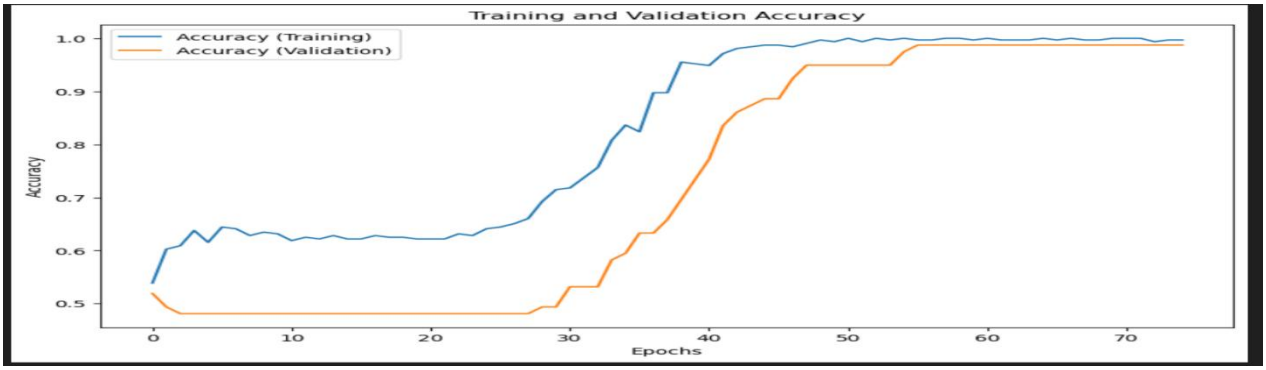
ANN(ARTIFICIAL NEURAL NETWORK)

```
model = Sequential([
    Dense(512, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

```
[16]  model.compile(optimizer=Adam(learning_rate=1e-4), loss='binary_crossentropy', metrics=['accuracy'])

[17]  model.summary()

...   Model: "sequential"

...
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	261,120
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131,328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257

```
plt.figure(figsize=(10, 6))
plt.plot(history.history['accuracy'], label='Accuracy (Training)')
plt.plot(history.history['val_accuracy'], label='Accuracy (Validation)')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```



MODEL DEPLOYMENT

```
import pickle
with open('vector_model.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)


with open('label_encoder.pkl', 'wb') as file:
    pickle.dump (label_encoder, file)


model.save('trained_model.h5')
```

Figure 2.16: Model Deployment

TEST THE MODEL

```
new_sentences =[
    "Her smile was as bright as the sun.",
    "The world is a stage, and we are merely players."]
X_new = vectorizer.transform(new_sentences)


nn_probabilities = model.predict(X_new)

nn_predictions = (nn_probabilities > 0.5).astype(int).flatten()

nn_predicted_labels = label_encoder.inverse_transform(nn_predictions)
```

Figure 2.17 : Test the model

```
for i, sentence in enumerate(new_sentences):
    nn_simile_prob = nn_probabilities[i][0]
    nn_metaphor_prob = 1 - nn_simile_prob

    print(f"Sentence: {sentence}")
    print()
    print(f"Neural Network Prediction: {nn_predicted_labels[i]}")
    print()
    print(f" Probability of being a Simile: {nn_simile_prob:.4f}")
    print(f" Probability of being a Metaphor: {nn_metaphor_prob:.4f}")
    print()
```

```
Sentence: Her smile was as bright as the sun.

Neural Network Prediction: Simile

 Probability of being a Simile: 0.9075
 Probability of being a Metaphor: 0.0925

Sentence: The world is a stage, and we are merely players.

Neural Network Prediction: Metaphor

 Probability of being a Simile: 0.1597
 Probability of being a Metaphor: 0.8403
```

The predicted value and actual value are almost the same.

HTML CODE

This section has the following tasks

- Building HTML Pages
- Building server side script

1.index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Simile & Metaphor Detector</title>
  <style>



    body {
      margin: 0;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background-color: #f5f7fa;
      background-image: url("static/images/background.jpg");
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-repeat: no-repeat;
      background-size: cover;
      background-position: center;
    }

    .container {
      background-color: white;
      padding: 40px;
      max-width: 600px;
      box-shadow: 0 4px 16px rgba(0, 0, 0, 0.1);
      border-radius: 12px;
      text-align: center;
    }

    h1 {
      color: #1e3a8a;
      font-size: 28px;
      margin-bottom: 20px;
    }
```

```
    .button {
      background-color: #2563eb;
      color: white;
      padding: 12px 24px;
      text-decoration: none;
      border-radius: 6px;
      font-weight: bold;
      display: inline-block;
      transition: background-color 0.3s;
    }

    .button:hover {
      background-color: #1d4ed8;
    }
  </style>
</head>
<body>

  <div class="container">
    <h1>Simile & Metaphor Detector</h1>
    <p>
      This project is an AI-powered tool that detects whether a sentence contains a
      <strong>simile</strong> or <strong>metaphor</strong>.
      <br>Enter any sentence on the prediction page and see the detected figurative type instantly!
    </p>
    <ul>
      <li>Uses machine learning to classify sentences as similes or metaphors</li>
      <li>Simple, fast, and free to use</li>
      <li>Perfect for students, writers, and educators</li>
    </ul>
    <a href="{{ url_for('predict') }}" class="button">Try Simile & Metaphor Detector</a>
  </div>
</body>
```

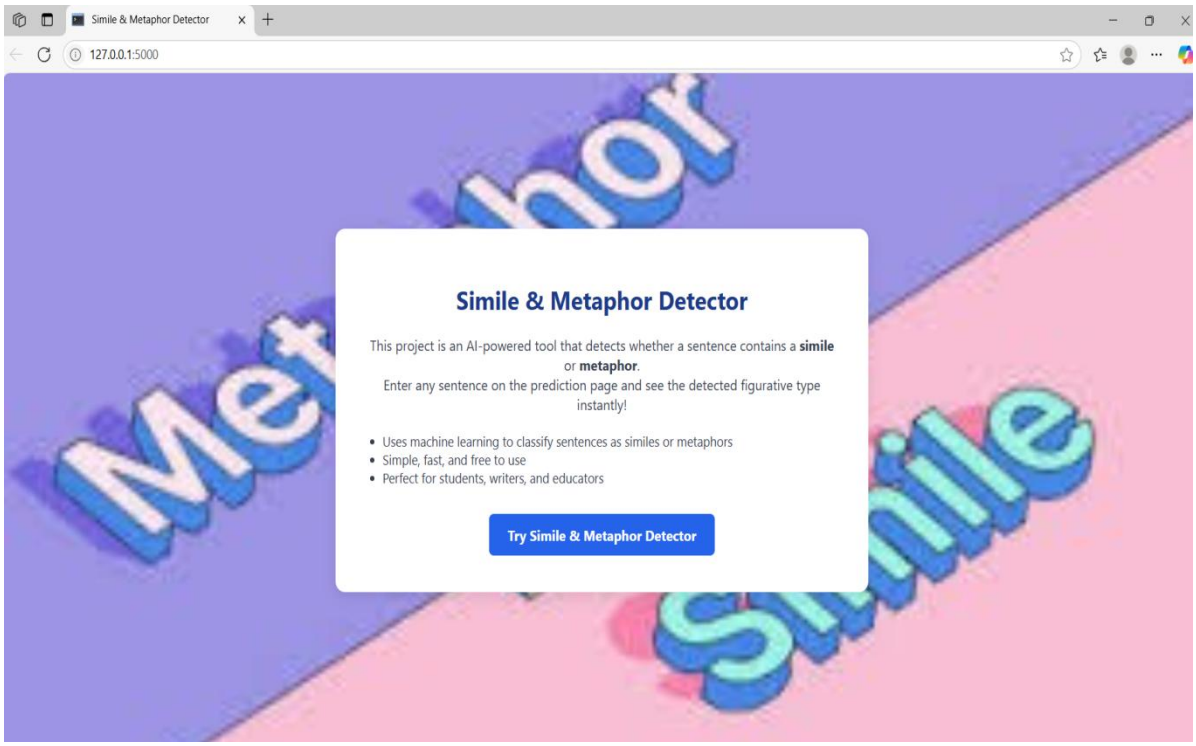


Figure 2.21: HTML code for Index Page and index page

2.predict.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Simile & Metaphor Detector</title>
  <style>
    body {
      min-height: 100vh;
      margin: 0;
      display: flex;
      background: #f8fafc;
      justify-content: center;
      align-items: center;
      font-family: ui-sans-serif, system-ui, sans-serif;
      background-image: url("static/images/metssimi.jpg");
      background-repeat: no-repeat;
      background-size: cover;
    }
    .container {
      width: 100%;
      max-width: 24rem;
      padding: 2rem;
      border-radius: 0.7rem;
      box-shadow: 0 2px 12px #0002;
      background: #fff;
    }
    h1 {
      font-size: 2rem;
      font-weight: bold;
      margin-bottom: 1rem;
      text-align: center;
    }
    form {
      display: flex;
      flex-direction: column;
      gap: 1rem;
```

```
    button[type="submit"] {
      width: 100%;
      background: #3b82f6;
      color: #fff;
      padding: 0.7rem 0;
      border: none;
      border-radius: 0.375rem;
      font-size: 1rem;
      font-weight: 600;
      cursor: pointer;
      transition: background 0.2s;
    }
    button[type="submit"]:hover {
      background: #2563eb;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Simile & Metaphor Detector</h1>
    <form method="POST" action="{{ url_for('predict') }}">
      <input
        type="text"
        name="userInput"
        placeholder="Enter a sentence..."
        required
      />
      <button type="submit">Detect Figurative Type</button>
    </form>
  </div>
</body>
</html>
```

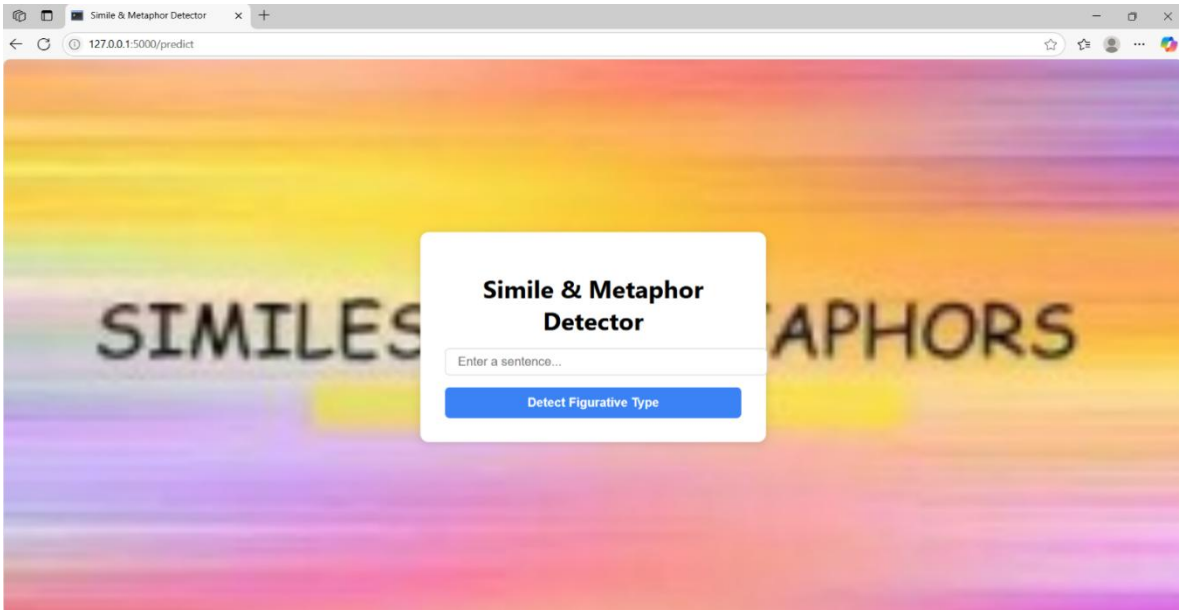


Figure 2.22 : HTML code for Predict Page and Predict Page.

3.Result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Prediction Result</title>
  <style>
    body {
      margin: 0;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background: linear-gradient(to top left, #f0f7ff, #ffffff);
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
      background-image: url("static/images/simile.jpg");
      background-size: cover;
      background-repeat: no-repeat;
    }

    .container {
      background-color: white;
      padding: 40px;
      width: 500px;
      border-radius: 16px;
      box-shadow: 0 8px 24px rgba(0, 0, 0, 0.1);
      text-align: center;
    }

    h2 {
      color: #1e3a8a;
      font-size: 24px;
      margin-bottom: 30px;
    }

    .result-box {
      background-color: #e0f0ff;
      padding: 20px;
```

```
    .button {
      background-color: #e2e8f0;
      color: #1e293b;
      padding: 10px 20px;
      border: none;
      border-radius: 8px;
      font-weight: bold;
      text-decoration: none;
      transition: background-color 0.3s;
    }

    .button:hover {
      background-color: #cbd5e1;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Prediction Result</h2>
    <div class="result-box">
      <strong>✓ {{ result }} Detected</strong>
      <p>{{ explanation }}</p>
      <em>Input: {{ user_input }}</em>
    </div>
    <a href="{{ url_for('predict') }}" class="button">Try Another Sentence</a>
  </div>
</body>
</html>
```

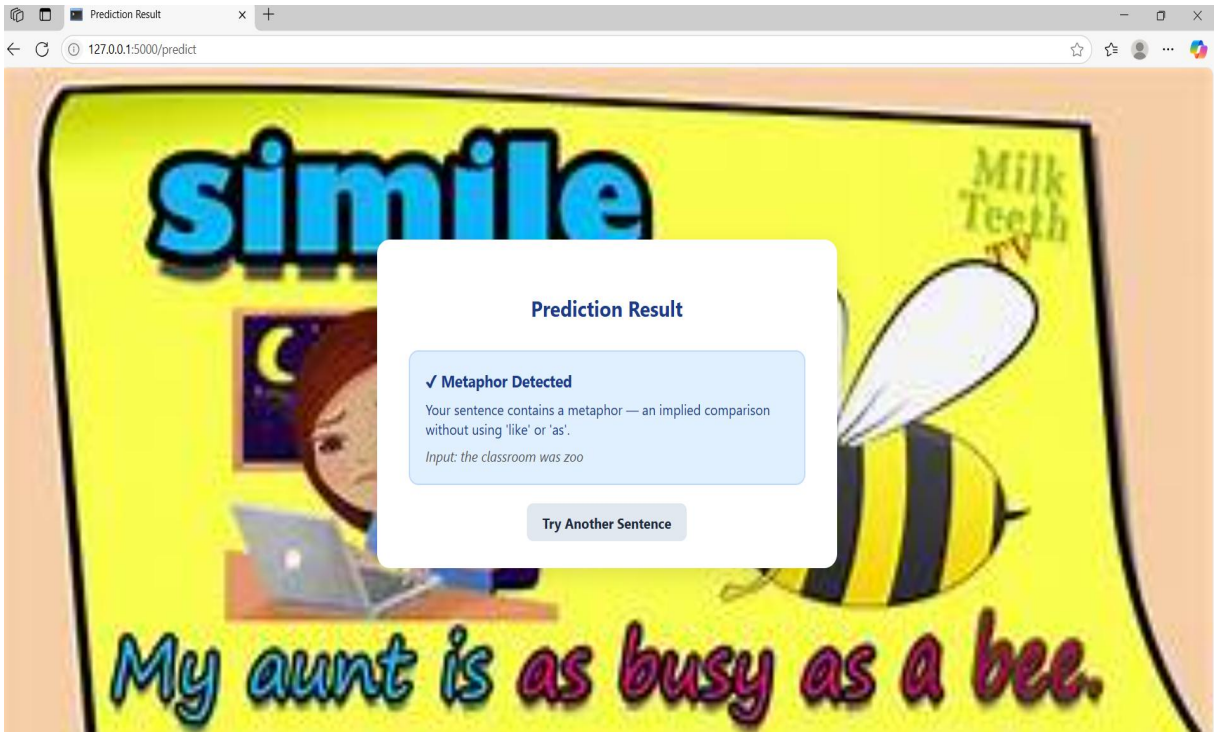


Figure 2.23: HTML code for Result Page and Result Page

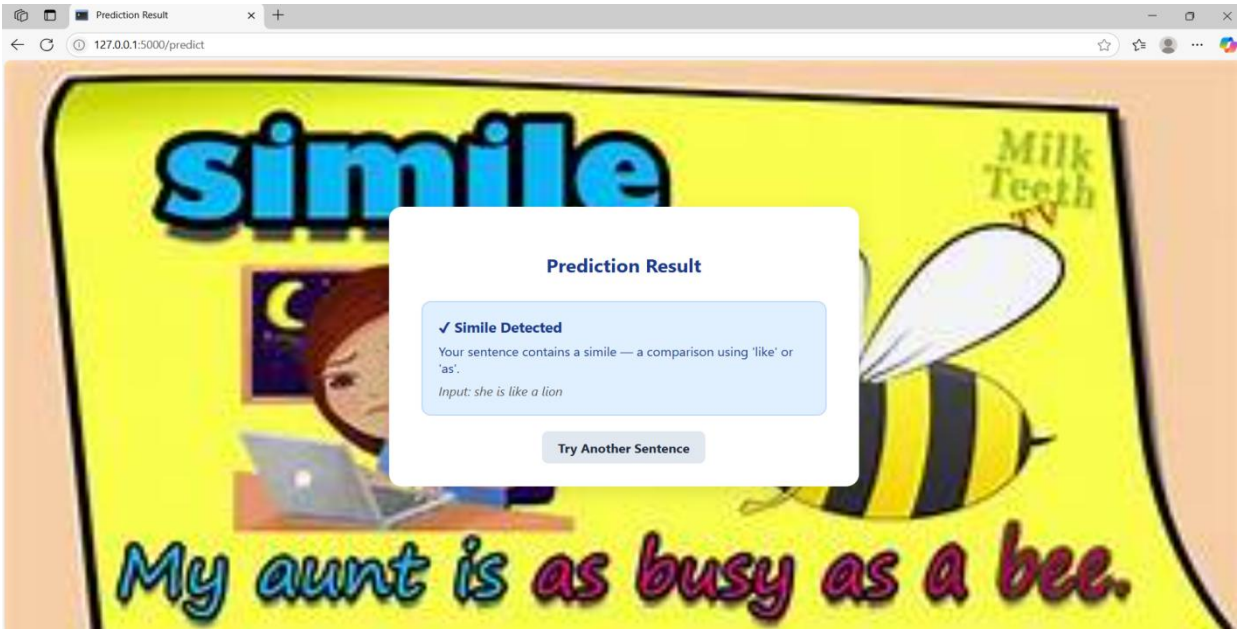
app.py

```python
import numpy as np
import pickle
from flask import Flask, request, render_template, redirect, url_for
from keras.models import load_model


app = Flask(__name__, template_folder='templates', static_folder='static')

# Load vectorizer, model, and label encoder
with open('vector_model.pkl', 'rb') as file:
    vectorizer = pickle.load(file)

classification_model = load_model('trained_model.h5')

with open('label_encoder.pkl', 'rb') as file:
    label_encoder = pickle.load(file)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        text = request.form['userInput']
        data = [text]

        x = vectorizer.transform(data)
        nn_probabilities = classification_model.predict(x)
        nn_predictions = (nn_probabilities > 0.5).astype(int).flatten()
        result = label_encoder.inverse_transform(nn_predictions)[0]

        explanation = {
            "Simile": "Your sentence contains a simile – a comparison using 'like' or 'as'.",
            "Metaphor": "Your sentence contains a metaphor – an implied comparison without using 'like' or 'as'."
        }.get(result, "Figurative language detected.")
```

```python
        explanation = {
            "Simile": "Your sentence contains a simile – a comparison using 'like' or 'as'.",
            "Metaphor": "Your sentence contains a metaphor – an implied comparison without using 'like' or 'as'."
        }.get(result, "Figurative language detected.")

        return render_template("result.html", result=result, explanation=explanation, user_input=text)
    else:
        return render_template("predict.html")

if __name__ == "__main__":
    app.run(debug=True)
```

Figure 2.24: Flask Code for app.

3.RESULTS



. **Figure 3.1:** The Output of the project

4.APPLICATIONS

1. Natural Language Understanding (NLU)

- Enhances the comprehension of non-literal language in NLP tasks like question answering, sentiment analysis, and summarization.
- Helps machines understand context, creativity, and intent in human language.

2. Conversational AI and Chatbots

- Improves chatbot responses by recognizing and interpreting figurative language.
- Enables more human-like, empathetic, and intelligent conversations.

3. Educational Tools

- Assists students and teachers in identifying similes and metaphors in texts.
- Useful for literature analysis, writing feedback, and language learning apps.

4. Creative Writing & Content Generation

- Powers tools that generate figurative language for storytelling, poetry, and marketing content.
- Helps authors and creators with style suggestions and literary expression.

5. Cognitive and Psychological Analysis

- Analyzes patient speech or writing for use of metaphors/similes in cognitive behavior studies, especially in fields like psycholinguistics, autism research, or depression detection.

6. Social Media Analysis

- Detects figurative language in tweets, posts, and memes.
- Helps in sentiment detection, trend analysis, and public opinion monitoring.

7. Multilingual and Cross-Cultural Language Processing

- Helps in understanding how figurative expressions differ across cultures and languages, aiding in machine translation and cross-linguistic communication.

5.ADVANTAGES

1. Enhanced Language Understanding

- Allows machines to **grasp abstract, non-literal meanings**.
- Bridges the gap between **literal text** and **human intent**.

2. Improved NLP Accuracy

- Boosts performance in tasks like **sentiment analysis**, **text classification**, and **information retrieval** by correctly interpreting figurative expressions.

3. More Natural and Human-like Communication

- Enables chatbots and virtual assistants to **respond more intelligently** when users use metaphors or similes.Supports **empathetic and context-aware interactions**.

4. Cognitive Insight

- Offers tools to **analyze creativity, emotion, and thought patterns** in human language.
- Useful in psychology and cognitive science to assess **mental health or linguistic creativity**.

5. Enriched Visual and Textual Interpretation

- Enhances **image captioning** by allowing **creative and expressive descriptions**.
- Adds **depth and richness** to generated or interpreted content.

6. Multilingual and Cultural Adaptability

- Helps detect figurative language across **multiple languages** and **cultural contexts**.
- Facilitates **cross-cultural communication** and **translation**.

7. Valuable in Sentiment and Emotion Analysis

- Detects **hidden emotions** and **subtle opinions** expressed via metaphors (e.g., "The economy is on life support").
- Provides **deeper sentiment insights** in social media and reviews.

6.DISADVANTAGES

1. Ambiguity and Subjectivity

- **Figurative language is often open to interpretation**; the same phrase may carry different meanings depending on context or culture.
- Models may struggle to **distinguish between literal and figurative usage**.

2. Complex Semantic Understanding Required

- Similes and metaphors rely on **deep semantic relationships** and **world knowledge**, which are difficult for models to fully capture.
- Even advanced models like BERT or GPT can **misinterpret creative or rare expressions**.

3. Lack of Annotated Data

- Figurative expressions are **hard to label consistently**, and large, high-quality datasets for training are limited.
- This makes supervised learning approaches more difficult and less scalable.

4. Cultural and Linguistic Variation

- Figurative language is often **culture-specific**; models trained on one language or region may not generalize well.
- Expressions like “he’s a snake” might not be metaphorical in every language.

5. Risk of Overfitting to Patterns

- Models might **memorize patterns** in training data rather than truly understanding figurative meaning.
- This can lead to poor performance on **unseen or creatively worded inputs**.

6. Evaluation Challenges

- Evaluating figurative language detection is hard because **ground truth is often subjective**.
- Standard accuracy metrics may not fully reflect the model's true capabilities.

7. High Computational Resources

- Models like **transformers** (e.g., BERT, RoBERTa) require **substantial memory and processing power**, especially when used with large datasets or in real-time applications.

7.CONCLUSION

Figurative intelligence, through the application of machine learning for simile and metaphor detection, represents a significant advancement in the ability of machines to understand and interpret human language beyond its literal meaning. By equipping models with the capability to recognize figurative expressions, systems can achieve deeper semantic understanding, enabling more natural, empathetic, and context-aware interactions in applications like conversational AI, education, image captioning, and psychological analysis. Despite challenges such as ambiguity, cultural variability, and the need for annotated data, ongoing research and advancements in deep learning and NLP continue to improve the accuracy and robustness of these models. Overall, figurative intelligence is a powerful step toward humanizing artificial intelligence and enhancing its relevance in real-world, environments.

8.FUTURE SCOPE

Figurative intelligence, particularly in the domain of machine learning for simile and metaphor detection, is vast and promising. As advancements in natural language processing (NLP) and deep learning continue to accelerate, these technologies will become even more adept at understanding and interpreting the subtleties of human language. One major area of growth is in enhancing human-computer interaction; as machines become better at recognizing and generating figurative language, they will be able to engage in more natural and meaningful conversations, understanding context and emotional nuances more deeply. Additionally, improved simile and metaphor detection can significantly benefit educational tools, helping students and learners grasp the intricacies of language and literature. In creative industries, such as writing and advertising, AI-powered tools will offer innovative ways to craft compelling narratives and slogans by leveraging figurative language effectively. Furthermore, as these models become more sophisticated, they will play a crucial role in cross-cultural communication, enabling better translation and interpretation of idiomatic and culturally specific expressions. The integration of figurative intelligence into various applications promises to create more intuitive and empathetic AI systems that mirror human-like understanding, ultimately bridging the gap between human creativity and artificial intelligence.

9.BIBLIOGRAPHY

[1] Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. University of Chicago Press. A foundational work exploring the role of metaphors in human thought and language.

[2] Shutova, E. (2010). *Models of metaphor in NLP*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (pp. 688–697).Focuses on computational models and machine learning techniques for metaphor detection.

[3] Ghosh, S., Chollet, M., Laksana, E., Morency, L. P., & Scherer, S. (2015). *A Multimodal Context-Based Approach for Figurative Language Detection*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1537–1546. Proposes a multimodal method using visual and textual cues for figurative expression detection.

[4] Veale, T., & Hao, Y. (2008). *A Fluid Knowledge Representation for Understanding and Generating Creative Metaphor*. In Proceedings of the 22nd International Conference on Artificial Intelligence(AAAI-08),1475–1480.Introduces an approach for metaphor generation and detection using semantic networks.

[5] Mohammad, S. M., Shutova, E., & Turney, P. D. (2016). *Metaphor as a medium for emotion: An empirical study*. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, 23–33.Explores the relationship between metaphor usage and emotional tone in language.

[6] Wei, H., Mao, Y., & Wang, X. (2020). *Simile Recognition via Shared and Private Representation Learning*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3052–3061.

10.HELP FILE

PROJECT EXECUTION:

STEP-1: Go to start, search and launch **ANACONDA NAVIGATOR**.

STEP-2: After launching of **ANACONDA NAVIGATOR**, launch **JUPYTER NOTEBOOK.**

STEP-3: Open **"app.py"** code

STEP-4: Import all the packages and check whether error present in the code are not.

STEP-5: Create **PYTHON CODE** folder on **DESKTOP**.

STEP-6: Create the **home.html** file to display the home page.

STEP-7: Launch the **SPYDER.**

STEP-8: After launching Spyder, give the path of **app.py** which is created in your laptop and run the program.

STEP-9: After running the **app.py,** then the URL is created **"http://127.0.0.1:5000".**

STEP-10: Copy the URL and paste it in the Web Browser.

STEP-11: Then the home page of the project will be displayed.

STEP-12: In the opened home page and give test to the predict box when click on predict button it will show metaphor or simile