AMRITA VISHWA VIDYAPEETHAM

21AIE314

AI in Natural Language Processing

End Semester Project Report

Topic: Offensive Humour Classification

B-13 TEAM MEMEBERS:

S.NO	NAME	ROLL.NO
1.	Bhaswanth Reddy I	CB.EN.U4AIE21103
2.	Karthik MS	CB.EN.U4AIE21121
3.	Sai Praneeth Reddy	CB.EN.U4AIE21103
4.	Abhinav. V	CB.EN.U4AIE21177

Index

Abstract

This project tackles the classification of offensive humor, leveraging a dataset categorized into four levels of offensiveness. We applied traditional machine learning models and parallel neural networks, using TF-IDF, GloVe, and Word2Vec embeddings to capture the jokes' semantic content. Comprehensive preprocessing, data balancing, and model evaluation were performed. The results show that combining traditional models with neural networks enhances the accuracy and reliability of offensive humor classification, providing valuable insights for automated content moderation and sentiment analysis.

Introduction:

Classifying offensive humour presents a significant challenge in the field of natural language processing (NLP) due to the subjective and nuanced nature of humour and offensiveness. This project aims to address this challenge by leveraging a diverse dataset categorized into four levels of offensive humor. By applying traditional machine learning models and parallel neural network architectures, we explore various feature extraction methods, including TF-IDF, GloVe, and Word2Vec embeddings. These techniques enable us to capture the semantic intricacies of the jokes, providing a robust foundation for our classification tasks. Through comprehensive preprocessing, data balancing, and model evaluation, this project seeks to enhance the accuracy and reliability of offensive humour classification, contributing to advancements in automated content moderation and sentiment analysis.

Dataset:

The Offensive Humor Dataset from Kaggle is a collection of text samples categorized into four levels of offensiveness: mildly offensive, moderately offensive, highly offensive, and extremely offensive. Each entry in the dataset includes a joke or humorous statement, labeled according to its perceived level of offensiveness. This dataset is designed to aid in the development and evaluation of models aimed at detecting and classifying offensive content in humor. It is particularly useful for tasks in natural language processing, such as sentiment analysis, content moderation, and automated text classification.

Total of 92,153 jokes across 4 categories

Categories:

- Clean Jokes (7,450 examples)
- Dark Jokes (79,230 examples)
- Dirty Jokes (5,473 examples)
- News articles as non-jokes (10,710 examples)

Pre-Processing:

Data Balancing

- One of the primary challenges encountered was the imbalance in the dataset. Initially, the dataset had a skewed distribution of classes, which could lead to biased model performance. To address this, the following balancing techniques were employed:
- Under-sampling: Reducing the number of samples from over-represented classes.
- Over-sampling: Increasing the number of samples in under-represented classes.

Preprocessing Techniques

- Preprocessing is crucial in NLP to ensure the dataset is clean and standardized. The following general NLP preprocessing techniques were applied:

- Null Value: Dropping the jokes having null values.
- Tokenization: Splitting text into individual tokens (words or phrases).
- Lowercasing: Converting all characters to lowercase to maintain consistency.
- Removal of Punctuation: Eliminating punctuation marks to focus on the words.
- Stop Words Removal: Removing common words (e.g., "and", "the") that do not contribute significant meaning.
- Stemming and Lemmatization: Reducing words to their base or root form to handle variations of the same word.
- Normalization: Standardizing text to handle variations in spelling and formatting.
- Removal of URLs and Emojis: Eliminating https links and emojis to ensure the text focuses on linguistic content without distractions.

Methodology:

Approach – 1 : Traditional Models

1.Word Embeddings

1.1Tf-idf:

- Building the Corpus: The entire preprocessed dataset was used as the corpus for TF-IDF calculation.
- The Term Frequency-Inverse Document Frequency (TF-IDF) technique was applied to the preprocessed data. TF-IDF measures the importance of a term within a document relative to a collection of documents (corpus). The following steps were undertaken:
- TF-IDF Vectorization: Each document (joke) was transformed into a vector representation using TF-IDF weighting. This process assigns higher weights to terms that are frequent within the document but rare across the entire corpus.
- Average TF-IDF Calculation: After TF-IDF vectorization, the average TF-IDF value for each joke was computed. This provides a compact representation of the text's content based on the importance of terms within it.

- CSV File Creation: The TF-IDF values for each joke were saved in a CSV file, allowing for further analysis and model training.

1.2 Word2Vec Embeddings Extraction

- The Skip-Gram model of Word2Vec with 300 dimensions and a window size of 5 was utilized to generate dense vector representations for each word in the jokes. The process involved:
- Training Word2Vec Model: The Skip-Gram model was trained on the preprocessed joke dataset to learn word embeddings. Skip-Gram focuses on predicting context words given a target word, capturing semantic relationships between words.
- Generating Joke Embeddings: For each joke in the dataset, the individual word embeddings were averaged to create a 300-dimensional vector representation for the joke. This step captures the overall semantic content of the joke based on the Word2Vec embeddings.
- CSV File Creation: The 300-dimensional Word2Vec embeddings for each joke were saved in a CSV file for further analysis and model training.

1.3 Glove Embeddings

- The Global Vectors for Word Representation (GloVe) embeddings trained on the Wikipedia and Gigaword corpus were utilized to represent each joke as dense vectors. The process involved:
- Loading GloVe Embeddings: The pre-trained GloVe embeddings (glove-wiki-gigaword-100) were loaded into memory. These embeddings capture semantic relationships between words in a high-dimensional vector space.
- Generating Joke Embeddings: For each joke in the dataset, the individual word embeddings were averaged to generate a 100-dimensional vector representation for the joke. This step captures the overall semantic content of the joke in the GloVe embedding space.
- CSV File Creation: The 100-dimensional GloVe embeddings for each joke were saved in a CSV file, facilitating further analysis and model training

2. Model Training

- Following the extraction of Word2Vec embeddings, machine learning models were trained to classify offensive humor. The models included:
- Support Vector Machine (SVM)
- Random Forest (RF)
- XGBoost (XG-B)
- The training process involved:
- Data Splitting: The dataset was split into training and testing sets for model evaluation.
- Model Training: The SVM, RF, and XG-B models were trained using the training data, with the Word2Vec embeddings serving as features.
- Model Evaluation: The trained models were evaluated on the testing data using metrics such as accuracy, precision, recall, and F1-score to assess their performance.

Approach – 2 : Parallel Neural Networks

In this report we propose a parallel neural network architecture to detect offense in humor. Our method leverages several pathways in the network to analyze both individual sentences and the entire text.

In the first stage, sentences are separated and transformed into numerical representations using BERT sentence embedding. This embedding process is applied to each sentence independently and also to the complete text. Parallel hidden layers within the network then extract mid-level features from each sentence's embedding, capturing aspects like context and sentence type. Additionally, a dedicated pathway analyzes the entire text's embedding, identifying word-level connections that might influence congruity, such as the presence of synonyms or antonyms. The output of the sentence pathway, a vector of size 20, and the whole-text pathway, a vector of size 60, that are obtained after passing through 3 layers of neural network parallelly are concatenated in the fourth layer and continued in a sequential manner to predict the target value. After the parallel substructure of the design, the model integrates the outputs from all pathways through three sequential neural

network layers. This combined analysis allows the network to predict the final outcome.

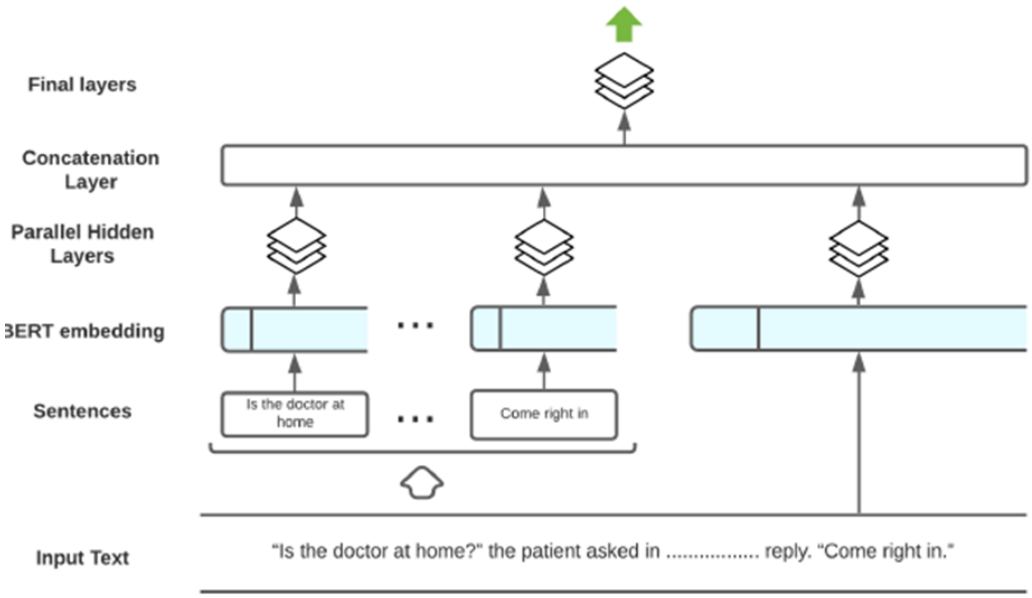
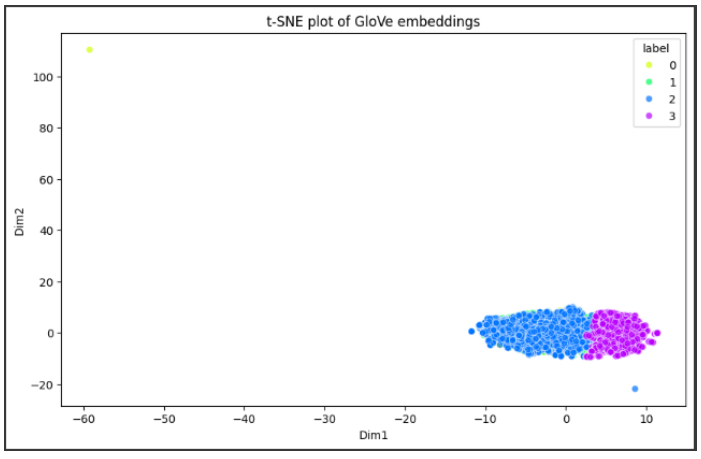


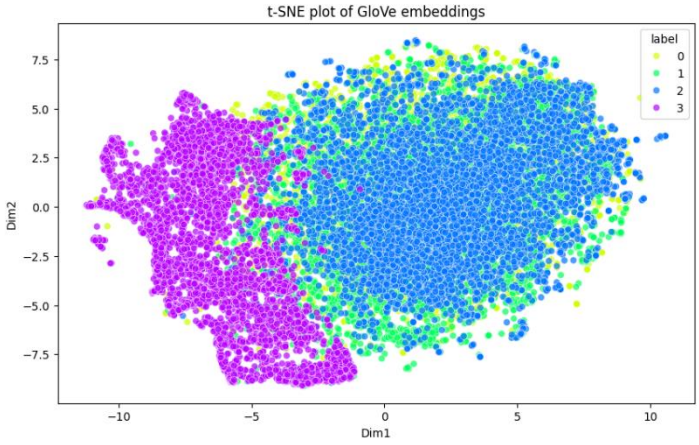
Figure 1: Components of the proposed method

Results:

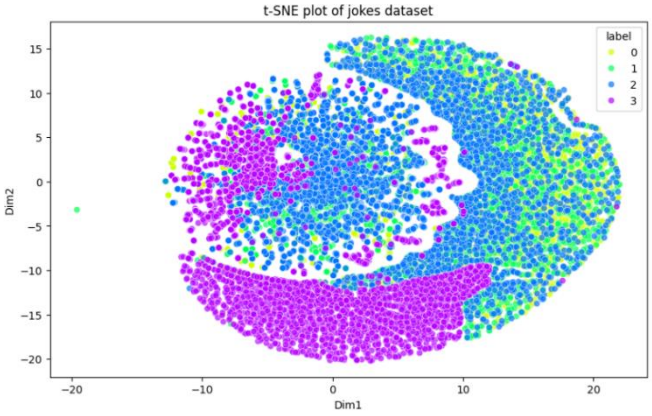
TSNE Plot



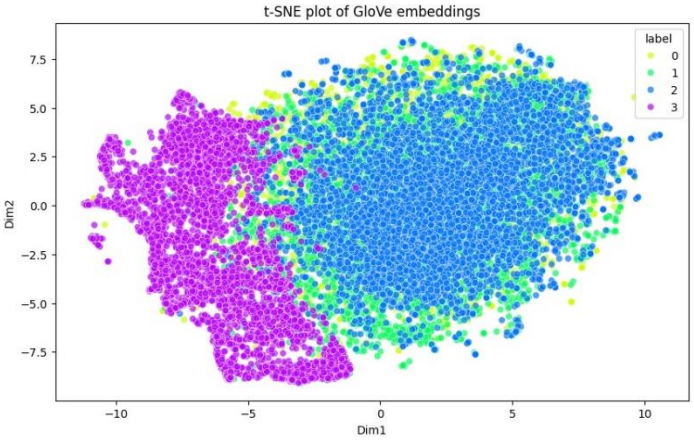
Glove Emb



Word2Vec



Tf-idf



n-gram

Accuracy :

SVM	Accuracy
3-gram	0.59
Tf-idf	0.56
Glove	0.72
Word2Vec	0.73

Random Forest	Accuracy
3-gram	0.59
Tf-idf	0.56
Glove	0.71
Word2Vec	0.71

XG-Boost	Accuracy
3-gram	0.60
Tf-idf	0.55
Glove	0.72
Word2Vec	0.73

NTK	Accuracy
3-gram	0.58
Tf-idf	0.50
Glove	0.72
Word2Vec	0.73

Conclusion

In this project, we explored the classification of offensive humor using both traditional machine learning models and a parallel neural network approach. By leveraging TF-IDF, n-gram, GloVe, and Word2Vec embeddings as feature representations, we trained models such as SVM, Random Forest, and XGBoost, achieving notable performance improvements through comprehensive preprocessing and data balancing techniques. Additionally, the parallel neural network, which effectively captured intricate semantic nuances, further enhanced the classification accuracy. The comparative analysis underscored the strengths of traditional models in handling structured feature sets, while the neural network demonstrated superior capability in modeling complex patterns within the offensive humor dataset. This dual approach highlights the complementary nature of traditional machine learning and deep learning techniques in advancing NLP tasks.

References:

https://huggingface.co/datasets/metaeval/offensive-humor

https://arxiv.org/abs/2211.14369

https://arxiv.org/abs/2402.01759

https://www.sciencedirect.com/science/article/pii/S0957417424005517