Analysis of Reddit Jokes for Humor Identification, Categorization and Classification

Abstract

Humor is a complex and subjective human experience that varies across cultures and individuals. This study analyzes a large dataset of 500,000 jokes sourced from Reddit's r/Jokes subreddit to identify patterns in humor, detect duplicates, classify jokes, and quantify funniness. We employed natural language processing techniques, including text preprocessing, language detection, embedding generation with transformer models, clustering algorithms, and machine learning classifiers. Our findings reveal thematic clusters in jokes, effective methods for duplicate detection, and moderate success in classifying jokes based on content and predicting their popularity. The study contributes to the understanding of computational humor analysis and proposes future work to enhance the accuracy and applicability of the models.

1. Introduction

Humor, an essential aspect of human communication, remains one of the most challenging phenomena for computational systems to comprehend and analyze. The subtle interplay of language, context, and culture in humor presents a formidable challenge for modern artificial intelligence. As natural language processing (NLP) and machine learning advance, there is a unique opportunity to deepen our understanding of humor using data-driven approaches. This paper explores the landscape of humor through a large-scale analysis of one million jokes sourced from Reddit, one of the internet's most expansive platforms for user-generated content.

The primary focus of this research is to unravel the complexities of online humor by addressing several key questions. We aim to distinguish genuine jokes from other forms of content, investigate the prevalence of duplicate or similar jokes, and explore the potential to classify jokes by their content and appropriateness. Additionally, we seek to investigate whether humor can be quantified by identifying linguistic, semantic, or structural features correlated with perceived humor.To address these challenges, we employ a comprehensive methodology combining advanced machine learning techniques. We utilize state-of-the-art natural language processing models to classify jokes into offensive and non-offensive categories, leveraging deep learning architectures like BERT and transformer-based models. By implementing multi-class classification algorithms, we develop a robust system capable of detecting nuanced forms of offensive content. Our approach integrates unsupervised and supervised learning techniques, applying clustering, dimensionality reduction, and sophisticated classification models to analyze the dataset. This interdisciplinary exploration not only contributes to computational linguistics but also provides insights into the complex landscape of online humor, bridging human creativity with advanced artificial intelligence methodologies.

2. Dataset Overview

The dataset comprises one million jokes collected from Reddit, particularly from humor-centric subreddits. Key data fields include: **type**: Indicates whether the entry is a 'post' or 'comment' , **title**: The title of the Reddit post. , **selftext**: The body text of the post **score**: The Reddit score reflecting the post's popularity.

Exploratory Data Analysis

Data Preparation - We combined the 'title' and 'selftext' fields to create a unified 'joke_text' for each entry. To handle missing values, we removed entries where both 'title' and 'selftext' were empty.

Joke Length Analysis

We employed tokenization to calculate the word count of each joke and found interesting outcomes. **The average joke length of the one million data is 35.21 words, and the median is 16 words, with the maximum number of jokes less than ~200 words, which can be seen in Fig 1. Range of words for jokes is from 1 to 9492 words.**

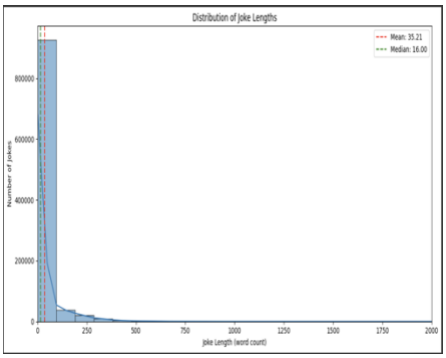

Figure 1: Histogram showing the distribution of joke lengths by word count.

Figure 2: Word Cloud - Cluster of words depicted in the Different sizes. The bigger and bolder the word appears, the more often its mentioned within the jokes.

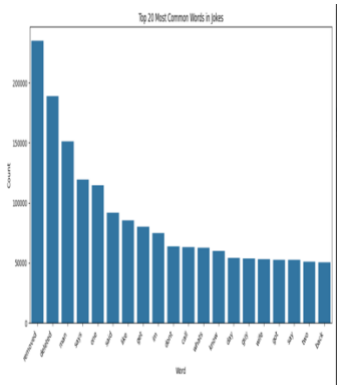

Figure 3:Top 20 most common words in jokes

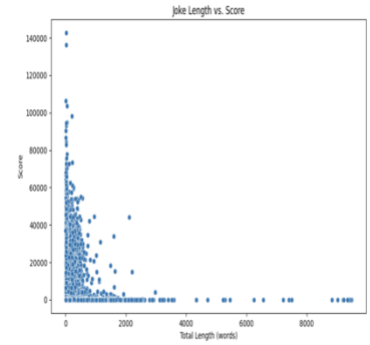

Figure 4: Distribution of score over the length of jokes

Vocabulary Analysis

In this, we identified the frequent words such as 'removed', 'deleted', 'says'. Also stopwords that are most **common English words were removed to focus on the significant terms**. We also performed content filtering to do better analysis. Fig 3 shows the most common words with its frequency.

Sentiment Analysis

In this, we used SentimentIntensityAnalyzer to find the sentiment of the jokes. Most of the sentiment scores are concentrated around the neutral (0), means the dataset has many texts with neutral sentiment. There is some spread towards both positive and negative sentiment values, with heavier concentration on the negative side. Overall average score is 0.0045, indicating that the most jokes have sentiment on the positive side. Figure 5 shows the distribution of the sentiment scores of all the jokes.

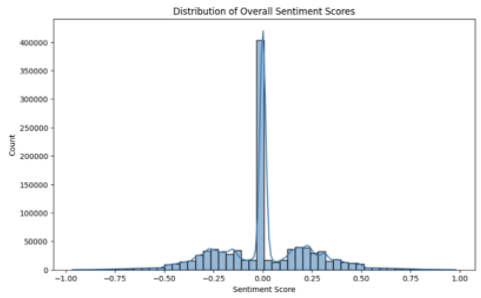

Figure 5: Distribution of Overall Sentiment Scores

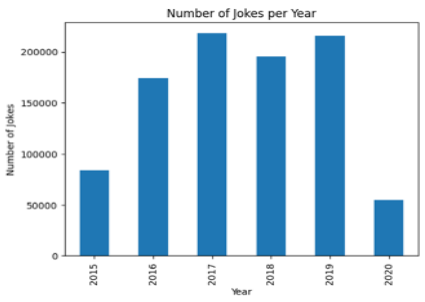

Figure 6: Number of Jokes per Year in bar graph

3. Methodology

3.1 Text Preprocessing

In this text processing pipeline, we started with normalization, a crucial step to maintain consistency and prepare the text for further analysis. All the text was converted to lowercase to maintain uniformity and eliminate the case

sensitivity issues. URLs and HTML tags were stripped from the text to remove irrelevant or noisy data that could interfere with analysis. Non-alphabetic characters, including special characters and numbers, were eliminated to focus solely on meaningful textual content. Also extra whitespaces were condensed into single spaces to standardize the formatting of the text. After normalization, we used **word_tokenise** function from the NLTK library, the text was split into individual words or tokens, enabling more granular analysis. Common English stopwords, such as "the," "and," or "is," were removed using NLTK's predefined stop word list, as these words typically do not contribute significant meaning in natural language processing tasks. To further simplify the words to their base forms, lemmatization was performed using **WordNet's lemmatizer**, which helped reduce inflected forms of words (e.g., "running" to "run") while preserving their semantic meaning.

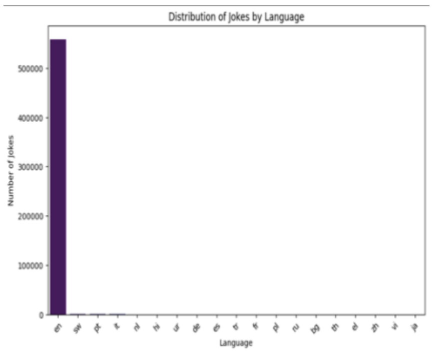

Figure 7: Distribution of of Jokes by Language

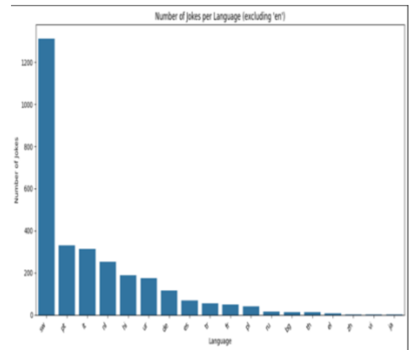

Figure 8: Number of jokes per language except en

Language detection was another critical step in the preprocessing workflow. The transformers library, combined with PyTorch, was used alongside **the pre-trained multilingual model papluca/xlm-roberta-base-language-detection** to accurately identify the language of each joke in the dataset. The detected language was appended as a new column in the DataFrame for each entry. To ensure consistency and relevance for subsequent analysis, only jokes identified as being in English were retained in the dataset. Additionally, entries with null or missing text were excluded during this filtering process, **resulting in a clean and focused dataset of English-language jokes ready** for further processing and analysis.

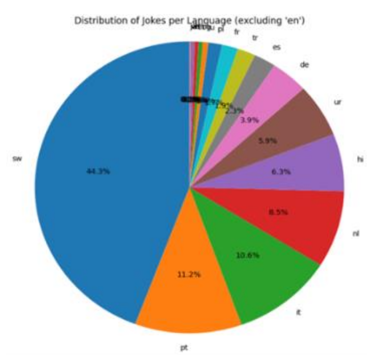

Figure 9: Pie chart showing distribution of jokes per language

```
Samples for language 'es':
                                                   joke_text language
500773  Eng: you should wear the leather's jacket... E...       es
504487       ¿Qué hace el pez cuando está aburrido? Nada.       es
550322            How long is a meter in Spain? A Spañard       es
130734  Mi amigo Sal decidió aprender latín sólo para ...       es
34854   If you're looking for new Elder Scrolls Online...       es

Samples for language 'hi':
                                                   joke_text language
11081   ¡ʎqɐǝ ƃuıʞɔnℲ ˙sʞɔns ʎlǝʇnlosqɐ pɹɐoqʎǝʞ ʍǝu sıɥ┴       hi
466573  f Hunter (Most popular PVE armor, least popula...       hi
117652  ((( কৌতুক :ক্ষমা চাওয়ার পরেও মারলেন|| Best Edu...       hi
508234                                               Me :(       hi
99788   গুড়িয়া যেখানে গুড্ডু সেখানে সিরিয়ালে বড় গু...       hi
```

Few examples of language detection (like es and hi)

3.2 Joke Identification

In this study, we performed a systematic methodology to identify and validate jokes within a large dataset. We used a zero-shot classification approach using **the facebook/bert-large-mnli** model from the transformers library to classify each joke as either "joke" or "not joke" based on the pre-defined candidate labels, facilitating the exclusion of non-joke content. The classification process was conducted on a manageable subset of the data (first 100,000 entries) to optimize computational resources. All the processed and classified data was stored into a new CSV for further analysis and review. This method effectively combined the advanced NLP techniques with efficient data handling practices to accurately identify and validate jokes within the dataset.

After this this identification and classification of jokes within the dataset, a comprehensive analysis was conducted to explore various characteristics and patterns inherent in the data. Initially the distribution of detected language was noted to understand the diversity present to filter out English-language jokes. To find the effectiveness of the classification model, the proportion of text labelled as "joke" versus "not joke" to provide insights into the dataset

composition and models's accuracy in distinguishing humorous content. **Also relationship between joke-length and its classification label** was explored to determine if longer or shorter texts were more likely to be categorised as jokes.These visual and statistical analyses collectively offered a deeper understanding of the dataset's structure, the nature of the jokes it contains, and the performance of the classification methodology employed. This multifaceted approach ensured a thorough examination of the data, facilitating the extraction of meaningful insights relevant to the study of humor in textual form.

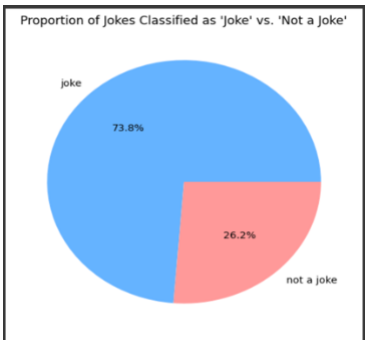

Figure 10: Pie chart showing percentage of jokes vs non jokes

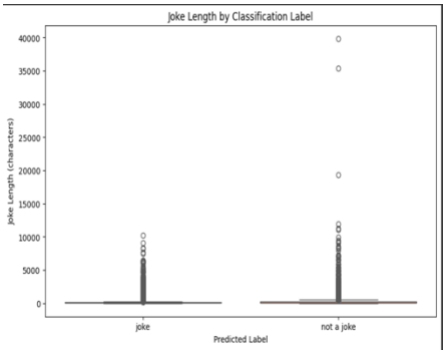

Figure 11: Joke Length by Classification Label

3.2 Duplicate Joke Detection

The pre-processed text is vectorised using **Term Frequency Inverse Document Frequency (TF-IDF),** which converts the text into high dimensional numerical representation based on term importance. To reduce computational complexity, we reduced dimensionality using **Truncated Singular Value Decomposition (SVD)**, which reduces the TF-IDF vectors to 500 components while retaining significant information. The reduced vectors are then indexed in the **FAISS (Facebook AI Similarity Search)** library, leveraging GPU acceleration for efficient cosine similarity computation between joke vectors.

Then we calculated the similarity scores for each joke against its most similar counterpart. Pairs of jokes with similarity scores exceeding **a threshold of 0.9** are identified as near duplicates. For validation, we created a DataFrame containing joke IDs, similarity scores, and corresponding joke texts, enabling manual inspection of high-similarity pairs. The pipeline concludes by flagging duplicates in the dataset and exporting the processed results. This comprehensive approach combines natural language processing techniques **and scalable similarity search to identify duplicate jokes efficiently making it suitable for large-scale datasets.**

Joke_ID	Most_Similar_Joke_I	Similarity_Scor	Joke_Text_Full	Most_Similar_Joke_Text_Full
0	52677	0.9991432	What part of a vegetable can you not eat? The wheelchair.	What is the most difficult part of the vegetable to eat? The wheelchair
6	29723	0.95477474	An engineer, physicist, and a statistician in a hotel room... So an engineer, a physicist, and a statistician are all sleeping in a hotel room, when suddenly an outlet catches fire. The engineer wakes up first and says to himself "this is an electrical fire, water won't work!" And runs to grab a fire extinguisher. The physicist wakes up next and thinks to himself "we have to cut the electricity off!" And runs to the power panel in the basement. The statistician wakes up and looks around,	So an engineer, a physicist, and a statistician are all sleeping in a hotel suddenly an outlet catches fire. The engineer wakes up first and says to himself "this is an electrical fire, water won't work!" And runs to grab a fire extinguisher. The physicist wakes up next and thinks to himself "we have to cut the electricity off!" And runs to the power panel in the basement. The statistician wakes up and looks around, he then screams "we need more data!!" And he sets the curtains
9	79639	0.9999993	How do porcupines have sex? Carefully.....very Carefully	How do porcupines have sex? Very..... carefully.
10	56651	0.9912999	A man and a woman were asleep like two innocent babies. Suddenly, at 3 o'clock in the morning, a loud noise came from outside. The woman, bewildered, jumped up from the bed and yelled at the man 'Holy Crap' That must be my husband!' So the man jumped out of the bed; scared and naked jumped out the window. He smashed himself on the ground, ran through a thorn bush and to his car as fast as he could go. A few minutes later he returned and went up to the bedroom and screamed at the woman, 'I AM your husband!' The woman yelled back, 'Yeah, then why were you running?'	A man and a woman were asleep like two innocent babies. Suddenly, at 3 o'clock in the morning, a loud noise came from outside. The woman, bewildered, jumped up from the bed and yelled at the man "Damn. That must be my husband!" So the man jumped out of the bed; and jumped out the window. He smashed himself on the ground, ran through a thorn bush and to his car as fast as he could go. A few minutes later he returned and went up to the bedroom and screamed at the woman, 'I AM your husband!' The woman yelled back, 'Yeah, then why were you running?'
11	60007	0.99924546	V V *Edit: seems like the ctrl key on my keyboard is not working	v v edit seems like the shift key on my keyboard is not working

Table 1.1: Top 5 most similar jokes

Joke_ID	Least_Similar_Joke_ID	Similarity_Score	Joke_Text	Least_Similar_Joke_Text
32	93590	0.4776415	woman go italy conference husband drive airport woman go italy conference husband drive airport thank honey say would like bring back laugh say italian girl conference meet airport asks honey trip good reply happened present present asks one asked italian girl oh say well could wait nine month see girl	italian woman walk husband giving man golden shower dumbfounded state shocked woman could think one thing say european
58	18263	0.47776377	astronaut first step onto alien planet astronaut first step onto alien planet alien excited change sign english even rename place landmark human place landmark thing astronaut decides first place want go pub see nearby alien asks wheres pub alien gurgle back suit translates astronaut real time alien say around corner astronaut head around corner see labelled keyboard asks bouncer called keyboar bouncer reply bos lowe thing human changed name reflect ask he bartender astronaut enters keyboar go bartender excuse pub astronaut say bartender gurgle back called keyboard man asks well alien gurgle reply since knew human coming updated name astronaut edge seat reason called keyboard space bar	nasa decorate astronaut board spacecraft particularly quickly starship enterprize
107	29213	0.36591297	yesterday met friend slovakia opened trampoline park near border yet seemed saddened something walked looked tired eye asked wrong whats matter asked many people surely business well replied heavy sigh ceiling low young local people around jump high keep hitting ceiling earth supposed happe roof crack answer seemed simple surely money coming pay someone raise ceiling must enough pay replied still disheartened may true weve influx visitor across border prague dont tip well barely even break even come around resigned fate taking deep breath looking said amount local visitor may roof czech keep bouncing	circumcise people livin job doesnt pay well least get keep tip
115	62118	0.49391434	remember teacher telling looking window wouldnt get anywhere boy smug look later handed burger fry drivethru	teacher said whoever answer question go home suddenly boy throw bag window teacher asked threw bag boy replied
133	19286	0.42037964	owner trendy restaurant dump rating poor review say asian inspired menu lack taste one dish particul stand bad way poke bowl help add punch owner enlists help old experienced consultant chef he hear top field organises come turn thing around better arrives quick sample made chef take mouthful look deep thought think asks owner rice nori fine elderly chef continues muttering owner bit fed asks make taste better asks chef eye light blurts one word tuna tuna tuna exclaims owner fierce gust air pick around billowing ancient chef grey hair wind chime outside clang echo mysteriously woman look eye piercing wisdom like speak age old adage tuna flavour bowl	trucker driving random stretch highway happens upon billboard driving read peach engineered taste mile mildly intrigued driver decides mean engineered taste bullshit say driving next couple mile finally come across turnoff restaurant park get enters cozy atmosphere catc peach taste like nearly anything think truck driver doubting bullshit heard far decides ask steak dinner old woman doesnt miss beat ask

Table 1.2: Top 5 moderately similar jokes

Joke_ID	Moderate_Similar_Joke_ID	Similarity_Scor	Joke_Text	Moderate_Similar_Joke_Text
20	93958	0.5436979	recently saw vvitch eye opening experience idea hilary clinton rough childhood	recently applied hokie pokie contest shoein
31	71646	0.5530155	man walk bar man walk bar see beautiful woman go sits next get drink sitting minute chuckle whats funny asks woman gon na tell joke dick long oh say id tell one pussy youll never get	man walk bar get concussion
32	93590	0.4776415	woman go italy conference husband drive airport woman go italy conference husband drive airport thank honey say would like bring back laugh say italian girl conference meet airport asks honey trip good reply happened present present asks one asked italian girl oh say well could wait nine month see girl	italian woman walk husband giving man golden shower dumbfounded state shocked woman could think one thing say european
35	66103	0.5783529	dont boxer sex big match theyre friend	dont baptist sex standing dont want people think theyre dancing
37	93069	0.596709	night time joke bed monkey walked bar bartender said get ya monkey said nothin sat	banana six year old nephew cracked joke nfc game could stop laughing alcohol might played role monkey go bar monkey bartender banana bartender monkey banana bartender monkey banana bartender ask put nail tongue hang wall monkey nail bartender monkey banana

Table 1.3: Least 5 similar jokes

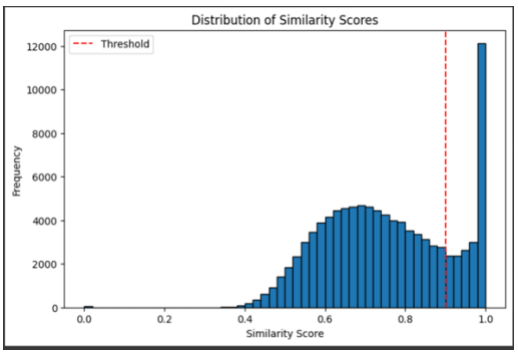

Figure 12: Distribution of similarity of jokes

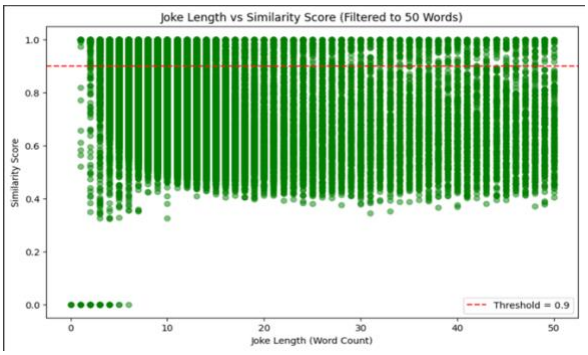

Figure 13: Joke Length vs Similarity Score

3.3 Clustering and Categorization

Moving ahead to the next finding clustering, the embedding generation process was a crucial step in our analysis, employing the **all-MiniLM-L6-v2 Sentence Transformer model**. This is lightweight yet powerful transformer architecture was chosen to generate high quality dense vector representations of text while maintaining computational efficiency. To manage memory constraints and optimize processing speed, we implemented a batch processing approach, allowing us to handle large volumes of data effectively. After the embedding generations, we applied Principal Component Analysis (PCA) to reduce the dimensionality of the embeddings. This step was essential in preparing the data for clustering while preserving the most important information. **By reducing the the embeddings to 50** (**which is ideal number as per the 3rd homework**), we maintained a balance between retaining critical features and facilitating efficient clustering operations. This dimensionality reduction not only improved the computational; efficiency of subsequent steps but also helped in mitigating potential noise in the data. For the **clustering phase, we opted for the MiniBatch K-Means algorithm,** primarily due to its scalability with large datasets. We conducted experiments with various cluster sizes to determine the optimal number of clusters ranging from 5-12, ultimately settling on 5 clusters based on their interpretability and distinctiveness. The cluster analysis involved a careful examination of sampled jokes from each cluster to identify prevalent themes. This process led to the labeling of clusters based on common topics such as politics, technology, relationships, and professions, providing a comprehensive categorization of the jokes. This approach not only allowed us to organize the dataset effectively but also offered valuable insights into the thematic distribution of jokes within our corpus.

The data points (jokes) are projected onto a 2D space using PCA, a method for reducing the dimensionality of datasets while retaining the most important patterns. Each point represents a joke from the dataset. Jokes with similar styles, themes, or sentiments might be grouped into the same cluster. For example, one cluster could group jokes that are politics, another for technology, etc. The colors represent different clusters, which were likely identified using a clustering algorithm Minibatch K-Means The color scale bar on the right indicates the cluster labels.

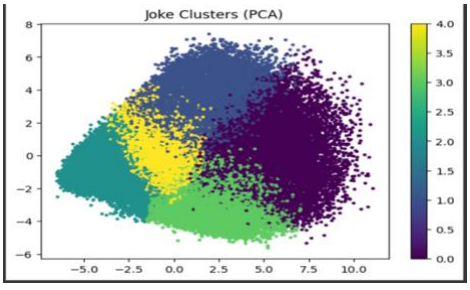

Figure 14: Joke Clusters using PCA

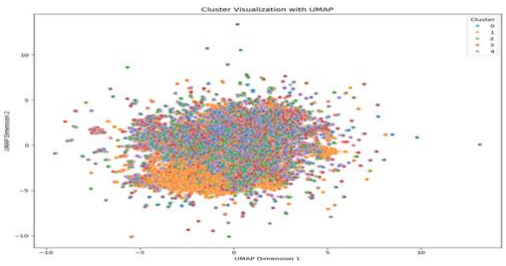

Figure 15: Joke Clusters using UMAP

We employed both Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction and visualization of the joke dataset. PCA, a linear technique, proved valuable for understanding overall variance and identifying broad linear patterns within the data. Its simplicity and computational efficiency made it a quick and effective tool for initial exploration. UMAP's ability to preserve both local and global structure resulted in clearer, more distinct clusters, offering deeper insights into the subtle relationships between different types of humor.

3.4 Classification of Jokes

In this we have implemented a systematic and intelligent workflow for detecting and classifying offensive content in textual data, addressing key challenges like label imbalance. We have created an extensive list of offensive words, covering explicit language, slurs, and sensitive terms by traversing the dataset and identifying adult/offensive words and then expanding it using perplexity and AI to get an exhaustive list. A custom function labels text as either "Offensive" or "Clean" by checking for the presence of these words.

Figure 16: List of offensive words

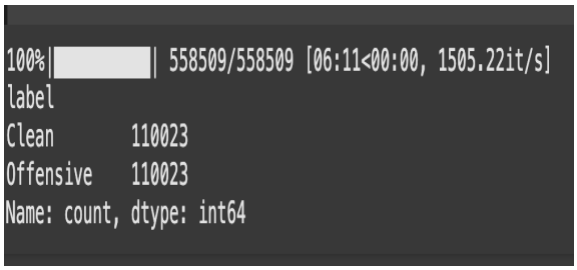

Figure 17: Balanced dataset of clean and offensive jokes

One notable strength of this approach is its thoughtful resolution of the label imbalance problem, which is a common issue in text classification tasks. Imbalanced datasets can lead to biased models that overfit the majority class**. To counter this, we have separated the offensive and clean subsets after labeling, and an equal number of samples from each subset are randomly selected to create a balanced dataset**. This ensures that both classes are equally represented, preventing the model from favoring one class over the other during training. By shuffling the balanced dataset before use, we have also eliminated any potential ordering biases. The balanced dataset is then splited into training and testing sets using stratification to preserve the label proportions in both splits. Text data is transformed into numerical form using TF-IDF vectorization, which assigns weights to words based on their relative importance. A **logistic regression along with XGBoost model** is trained with hyperparameter tuning via grid search to optimize performance. The final model is evaluated using a classification report and a confusion matrix, providing detailed insights into precision, recall, F1-score, and accuracy for each class.

Classification Report:

	precision	recall	f1-score	support
Clean	0.93	0.97	0.95	22005
Offensive	0.97	0.93	0.95	22005
accuracy			0.95	44010
macro avg	0.95	0.95	0.95	44010
weighted avg	0.95	0.95	0.95	44010

Figure 18: Performance of Logistic Regression Model

XGBoost Classification Report:

	precision	recall	f1-score	support
0	0.97	1.00	0.99	22005
1	1.00	0.97	0.99	22005
accuracy			0.99	44010
macro avg	0.99	0.99	0.99	44010
weighted avg	0.99	0.99	0.99	44010

Figure 19: Performance of XGBoost Model

4. Results

4.1 Identification of Jokes

The analysis showed that 73.8% of the entries were classified as "jokes," while 26.2% were labeled as "not a joke." This indicates that the dataset predominantly contained humorous content, but a significant portion was excluded from the "joke" category. The exclusion could be due to the presence of non-humorous content or text deemed offensive or inappropriate, which would not qualify as jokes. A deeper exploration of joke length revealed that both jokes and non-jokes tended to have shorter text lengths, with a few outliers extending to tens of thousands of characters. However, the overlap in text length between the two categories suggests that length alone is not a determining factor for classification. **Additionally, some texts categorized as "not a joke" were likely flagged for containing offensive or harmful language. The model, designed to identify context, sentiment, and tone, excluded such entries to maintain**

ethical standards and align with societal norms, ensuring that the dataset's humor classification was not compromised by inappropriate content. This comprehensive approach effectively combined technical precision and ethical considerations, offering meaningful insights into the structure and nature of humor in textual data.

```
                                   joke_text language predicted_label
I am soooo glad I'm not circumcised! My corona...       en      not a joke
```

Eg - False positive result, It is labelled as not a joke, but actually it is a joke which contains a offensive word

4.2 Duplicate Detection

To better understand patterns in joke content and identify highly similar jokes, we analyzed a dataset of jokes using a similarity scoring algorithm. The analysis revealed a range of similarity scores between jokes, with values close to 1.0 indicating near-identical jokes and lower scores suggesting varying degrees of thematic or textual divergence. For instance, **one pair of jokes had a similarity score of 0.999,** reflecting only minor lexical differences, such as phrasing variations **("What part of a vegetable can you not eat? The wheelchair" vs. "What is the most difficult part of the vegetable to eat? The wheelchair").** Another pair, scored at 0.991, differed subtly in word choice but maintained identical punchlines.

This similarity analysis is valuable for identifying joke rephrasing, redundancies, or creative variations. It provides a framework for understanding joke structures and their impact on comedic effect. High similarity scores suggest that humor is often retained even with slight rewording, while lower similarity scores may indicate more diverse creativity in joke generation. This analysis can also serve as a foundational tool for content moderation or for clustering similar jokes in large datasets, ensuring diversity in presentation or avoiding repetitive humor in public domains. This operation requires more resource computing, so we performed it only on 1 lakh sample data. **The data indicates that all jokes have a similarity score greater than 0.8, with have high similarity i.e. 22,478 jokes, moderately similar jokes are 18,020 jokes and only 3,636 jokes are similar with least similarity jokes that just more than 0.5. Rest all don't have any similar pairs.**

4.3 Classification Performance

Classifying text data into appropriate categories is a critical task in natural language processing, especially when dealing with sensitive content such as offensive jokes. Accurate classification ensures that potentially harmful content can be identified and managed effectively. In this study, we compared the performance of two popular machine learning models, Logistic Regression and XGBoost, to classify jokes as either "Clean" or "Offensive." The evaluation focused on key metrics such as accuracy, precision, recall, and F1-score to assess the models' effectiveness and reliability. Both models utilized TF-IDF vectorization to convert text into numerical representations for classification, enabling a detailed comparison of their predictive capabilities.

The comparison of Logistic Regression and XGBoost for the classification task revealed significant differences in performance. **Logistic Regression achieved an accuracy of 95%, with balanced F1-scores of 95%** for both "Clean" and "Offensive" jokes. However, the model showed some limitations in handling misclassifications, **with 625 false positives (clean jokes mislabeled as offensive) and 1,584 false negatives** (offensive jokes mislabeled as clean). This indicates that while the model performs well overall, it has a slight bias toward predicting offensive jokes correctly at the cost of occasionally misclassifying clean jokes. T**he recall for offensive jokes (93%) was notably lower than for clean jokes (97%)**, highlighting areas for potential improvement.In contrast, XGBoost outperformed Logistic Regression significantly, achieving an accuracy of **99% and F1-scores of 99% for both classes**. The model demonstrated near-perfect **precision for offensive jokes (100%) and perfect recall for clean jokes (100%), with only 617 false negatives and no false positives for clean jokes.** These results highlight XGBoost's superior ability to handle complex relationships in the data, minimizing misclassifications across both classes.

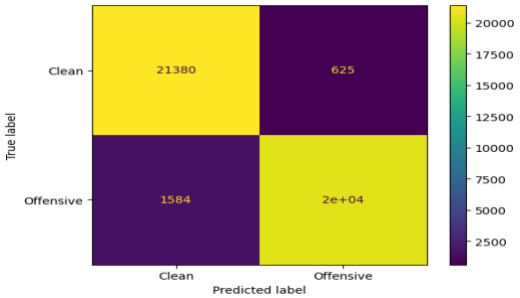

Figure 20: Confusion Matrix of Logistic Regression model

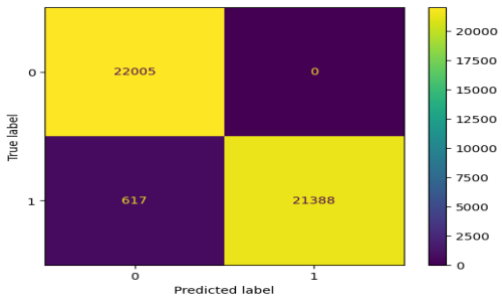

Figure 21: Confusion Matrix of XGBoost Model

In conclusion, while both Logistic Regression and XGBoost demonstrated strong performance in classifying jokes, XGBoost emerged as the superior model with a 99% accuracy and balanced F1-scores across classes. Its ability to minimize misclassifications, particularly achieving perfect precision for offensive jokes and perfect recall for clean jokes, underscores its suitability for sensitive content detection.

5. Limitations

Language Detection and Dataset Purity: The accuracy of language detection algorithms is essential for preserving the quality of multilingual datasets. In humor analysis, misidentifying a language can result in irrelevant content being included or important data being left out. This challenge becomes even more significant when working with short texts or code-switching, both of which are common in humorous content. Poor language detection can add noise to the dataset, distort analysis, and lead to incorrect conclusions about humor trends across languages. To address these issues, researchers should account for these limitations by adding extra validation steps or employing more advanced language identification tools to maintain the dataset's integrity.

Subjectivity of Humor and Popularity Metrics: Humor is inherently subjective, making it a tough nut to crack when it comes to quantitative analysis. What one person finds hilarious might leave someone else completely unimpressed, making it tricky to define universal metrics for funniness. Popularity markers like likes, shares, or upvotes are often used as stand-ins for humor appreciation, but they can be deceiving. External factors like the timing of a post, the poster's social influence, or even current events can skew the popularity of a joke, regardless of its comedic value. Additionally, certain types of humor might be more shareable or likable without actually being funnier. To get a fuller picture of how humor is perceived, researchers should look beyond these surface-level metrics. Combining qualitative assessments, input from diverse focus groups, and more refined engagement metrics could offer deeper insights into the complexities of humor appreciation.

Limitations of Offensive Content Detection: Using predefined lists of offensive words as the sole method for content moderation in humor analysis is far from foolproof. This simplistic approach often misses context-dependent nuances, subtle discriminatory undertones, or culturally specific insensitivities. It also struggles to keep up with the ever-evolving nature of language and societal norms. To address these shortcomings, more advanced techniques are necessary. Researchers should explore machine learning models trained on diverse datasets, integrate context-aware analysis, and continuously update detection methods. Furthermore, incorporating human moderators or diverse focus groups into the review process can help identify nuanced cases of offensive content that automated systems might overlook.

6. Future Work

The future work for humor detection and analysis in natural language processing encompasses several key areas for improvement and expansion:

Enhanced Language Processing: Utilizing advanced contextual embeddings like GPT-4 can significantly improve humor detection by capturing nuanced contexts, idiomatic expressions, and cultural references. Implementing models that enhance semantic understanding will enable systems to interpret complex forms of humor, such as sarcasm and irony, which rely heavily on context.Comprehensive Funniness Metrics: Exploring alternative metrics for assessing humor, such as user comments and laughter reactions, will provide a more holistic understanding of funniness. Integrating human evaluations into the process will validate and enhance predictive models, ensuring they align with real-world perceptions of humor.

Multilingual Analysis: Expanding analysis to include multiple languages is vital for understanding cultural differences in humor. Cross-cultural studies can reveal universal patterns and culture-specific elements, offering insights into how humor varies across societies and contributing to fields like anthropology and psychology.

Ethical Considerations: Addressing ethical concerns is crucial in developing responsible AI systems for humor detection. Creating effective content moderation systems will help filter harmful jokes, while mitigating biases in data and models ensures fair treatment across diverse groups. Prioritizing these considerations will lead to accurate and socially responsible humor detection systems.

7. Conclusion

This study demonstrates the feasibility of analyzing large-scale humor data using NLP techniques and machine learning algorithms. By effectively preprocessing data, detecting duplicates, clustering jokes, and classifying content we have uncovered patterns in humor shared on Reddit. While challenges remain due to the subjective nature of humor and computational limitations, the methodologies applied here lay the groundwork for future research in computational humor analysis and its applications in technology and society.

References

1. Mihalcea, R., & Strapparava, C. (2005). Making Computers Laugh: Investigations in **Automatic Humor Recognition. In Proceedings** of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (pp. 531-538). Association for Computational Linguistics.
2. Potash, P., Romanov, A., & Rumshisky, A. (2017). SemEval-2017 Task 6: #HashtagWars: Learning a Sense of Humor. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017) (pp. 49-57). Association for Computational Linguistics.
3. Yang, D., & Liu, Y. (2019). Understanding the Laughter: A Survey of Computational Humor Recognition. ACM Computing Surveys, 52(6), Article 106.
4. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
5. Chen, L., Lee, S., & Chen, K. (2018). Humor Recognition Using Deep Learning. Proceedings of the 27th International Conference on Computational Linguistics (pp. 113–123). Association for Computational Linguistics.
6. Yang, L., Wang, K., & Chen, Z. (2015). Predicting the Funniest Jokes in Social Media. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1425–1430). Association for Computational Linguistics.
7. Pirkkalainen, H., Pawlowski, J. M., Bick, M., & Hedberg, H. (2018). Engaging Users with Social Media: The Use of Social Media for Work-Related Purposes and Its Impact on Job Performance. Journal of Organizational Computing and Electronic Commerce, 28(2), 97–121.
8. Kao, J. T., Levy, R., & Goodman, N. D. (2016). A Computational Model of Linguistic Humor in Puns. Cognitive Science, 40(5), 1270–1285.
9. Weston, D., & Saint-Amand, H. (2018). Clustering and Unsupervised Classification for Natural Language Processing. In Advances in Neural Information Processing Systems (pp. 1–10).

Appendix

- **Code Repository**: https://drive.google.com/drive/folders/1miG7pVUtuplD4vqP3jCF91yrKrVE2N1b?usp=sharing

- **Data Set - https://www.kaggle.com/datasets/priyamchoksi/1-million-reddit-jokes-rjokes**