Analysis of a Large Joke Corpora

***Abstract* —** *Humor detection is a challenging task in NLP due to its subjective nature, as well as cultural and contextual nuances. Detecting offensive or NSFW jokes, in particular, poses a unique challenge with numerous practical applications in content moderation, user safety, and compliance. While recent studies have shown promising results in this domain, we adopted a novel approach by introducing GPT-generated non-offensive (SFW) counterparts to human-authored jokes. We then evaluated the performance of benchmark humor detection models on a dataset comprising only human-authored jokes. In this project, we demonstrate that augmenting the dataset with GPT-generated pairs improves model performance and accuracy while presenting the augmented dataset as a valid option for further exploration and future research. Our findings suggest that GPT-augmented datasets improve NSFW joke classification, with DistilBERT reaching 83.08% accuracy and an F1-score of 82.93%, and RoBERTa increasing recall by 4.04%. Models trained solely on GPT-generated data achieve lower accuracy (57.75%), emphasizing the importance of mixing human-authored and GPT-augmented data for reliable classification.*

I. Introduction

Among many researches done on the Reddit Jokes dataset [5] or in the context of detecting offensive, inappropriate, or sensitive humor in text, a few have obtained promising results and/or approached the problem from interesting angles. For example, Horvitz, Mayfield, and Seppi (2024) experimented on LLMs to generate non-humorous counterparts for existing jokes within the 1 Million Reddit Jokes (r/Jokes) dataset [1]. This process, often referred to as “unfunning,” involves modifying jokes to create paired data in which each humorous entry is accompanied by a corresponding non-humorous version, resulting in improved humor detection capabilities. Additional research in humor analysis includes Turano and Beatric’s creation of SCRIPTS which utilizes stand-up comedy transcripts for creating Random Forest based models [7], while Chaudhary and Goel used Firefox discussion forums to fuel their neural networks [8]. With specific relation to NSFW, Tang, Wang, and Wang (2022) address the challenge of detecting offensive humor by introducing The Naughtyformer, a transformer-based model fine-tuned specifically for this purpose [2], achieving strong performance in identifying offensive humor over existing benchmark RoBERTa-base model. Furthermore, Qiu et al. (2023) introduce CensorChat, a dataset for detecting NSFW content in open-domain dialogues, using knowledge distillation with models like GPT-4 for annotation [3], which enhanced the detection of explicit language. This approach enhances the detection of explicit language, contributing a valuable resource for safer interactions in conversational AI systems. For the NSFW humor detection task, existing benchmarks primarily rely on datasets and models fine-tuned for offensive language and content moderation. One of them is the TweetEval Offensive Language Detection : A widely-used benchmark that includes a RoBERTa-base model fine-tuned on Twitter’s offensive language data [4][9]. Collectively, these studies underscore the effectiveness of transformer-based models and synthetic data augmentation in enhancing humor and NSFW content detection. Despite these endeavors, no prior work has been done that explores the impact of GPT-generated data for classifying NSFW jokes. Based on the results of the “unfunning” approach mentioned earlier, we believe that this is a worthwhile approach to explore for the task. Therefore, building on recent advancements, this project aims to enhance NSFW humor detection by integrating robust benchmarks with synthetic data to test its impact on model performance. The main contributions of this study are as follows:

1. **The authors introduce** GPT-generated Safe for Work (SFW) and Not Safe for Work (NSFW) joke pairs, creating a unique augmented dataset to enhance humor classification.
2. **This study demonstrates** that GPT-augmented datasets improve classification performance, achieving the highest gain of +3.34% for accuracy, +3.13% for F1-score, and +4.04% on recall with RoBERTa.
3. **The study compares** DistilBERT, RoBERTa, and SVM across three datasets (Jokes-GPT-Augmented, Jokes-wo-GPT, and Jokes-GPT-only), providing a robust performance analysis using accuracy, precision, recall, and F1-score metrics.
4. **Finally, the authors emphasize** the importance of combining human-authored and GPT-augmented data for robust humor classification, paving the way for advancements in NSFW content moderation and automated humor detection.

II. Data

**A. Data Overview:** The dataset we used contains one million jokes collected from the Reddit subreddit [5]. Each data entry has multiple fields, including a title, main joke content, timestamp, and score, among other details. These domains provide a solid foundation for humor analysis and natural language processing (NLP) investigations, specifically in humor detection, sentiment analysis, and clustering. However, for our research purposes, many of these fields were not important which is why preprocessing was necessary

**B. Preprocessing**

We first knew we needed to create a “Joke” field in our dataset which represented the concatenation of the title + self text fields. This is because the reddit jokes were formatted to have the title represent the set up while selftext was the build up and punch line. As jokes were the only thing necessary for our problem, we removed all other columns from the set.

We then made sure all rows with missing values for text or selftext were removed as a joke could not be created from null values. We also removed rows with [deleted] or [removed] as these were the indicators used by Reddit for invalid data. Additional preprocessing included dropping duplicate jokes, ensuring all rows were of value string, eliminating special characters, and only focusing on jokes of 175 characters or less to limit computation overhead.

**C. Annotated Data (Data Set 1)** [Jokes-wo-GPT] [10] : Our first dataset used for training purposes was an 10,000 jokes annotated sample of the Reddit data. The annotator we used was Michelle Li’s “NSFW\_text\_classifier” [6], a fine-tuned transformer model based on DistillBert sentient analysis capabilities. The purpose of this annotator was to add Not Safe for Work (NSFW) or Safe for Work (SFW) labels next to each Reddit joke in our sample to provide the appropriate fields for the models that will be tested in our experiment.

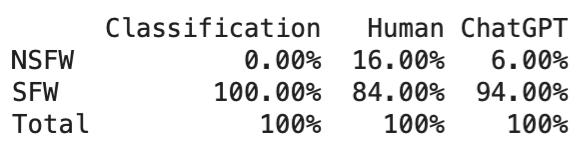
To ensure fairness and an uniformly distributed sample, we aimed to create a sample with an equal amount of NSFW and SFW jokes. WE started with a sample of 20,000 which would be further sampled once the annotations were made.

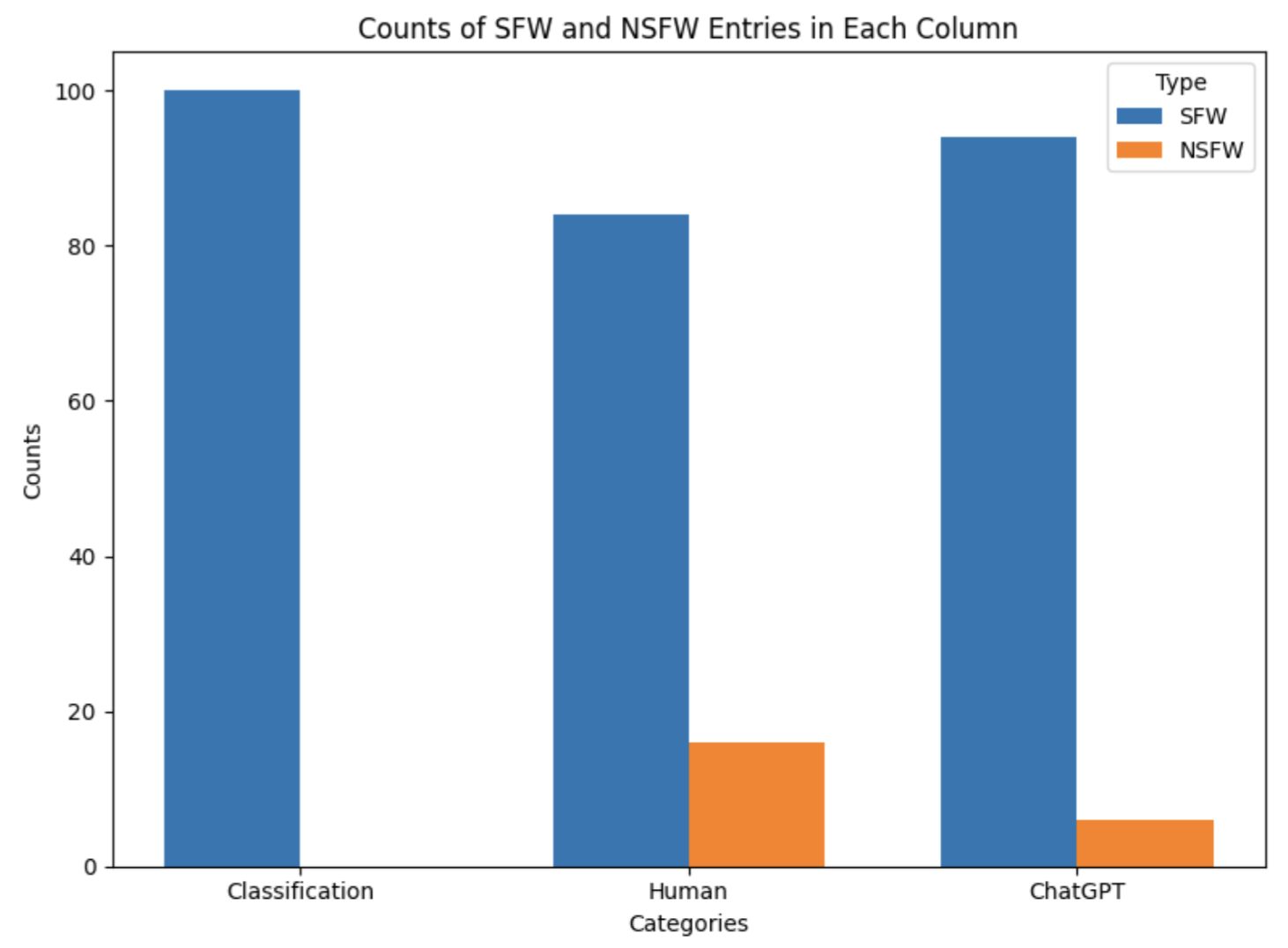
These 20,000 jokes were passed through the annotator and a new column titled “Classification” was created in the process. From there, 5,000 SFW and 5,000 NSFW jokes were extracted and combined to create a fully human curated data set of jokes.

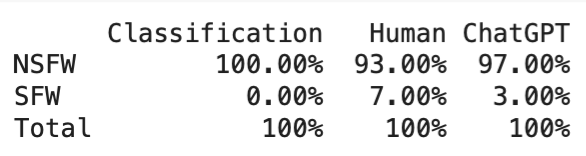
**Ensuring Quality of Annotated Data** : We will use two approaches to verify the accuracy of our annotator, manual and ChatGPT labeling.

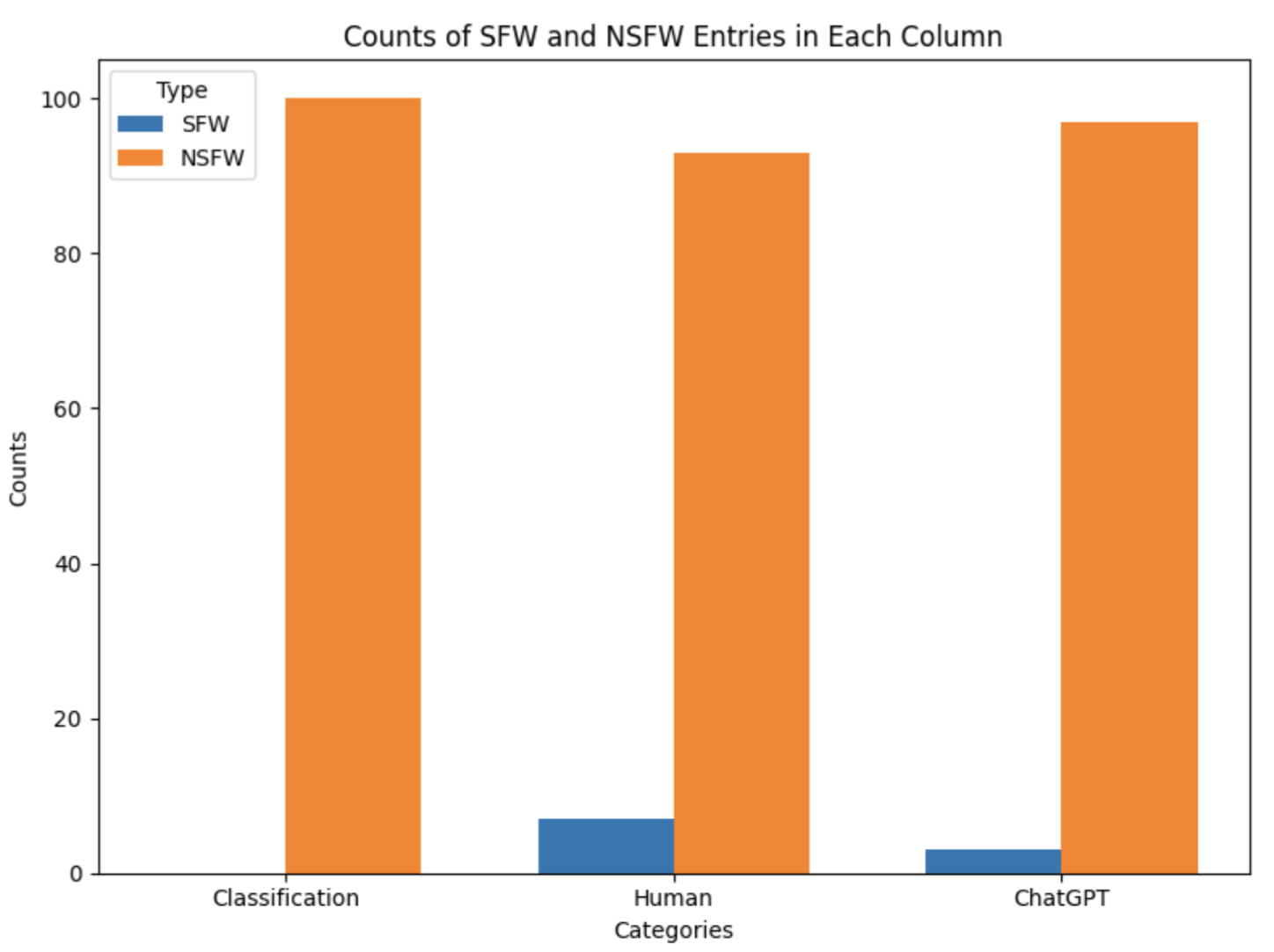
*Manual* ***:*** Out of the 5,000 SFW/NSFW jokes, 100 SFW jokes and 100 NSFW jokes were randomly sampled and labeled with manual human effort while being unknown to the labels produced by the annotator. This was to ensure there was no bias in our evaluation and to test an effective portion of the sample so that we can speak on the quality of the annotator.

*ChatGPT :* The jokes were also passed into OpenAI’s ChatGPT without the annotated labels to compare and contrast the labels produced by ChatGPT, a highly trained model that can further prove the annotator’s value.









**Fig. 1**: Table/Bar Graph of percentages for SFW Joke (TOP) and NSFW Joke (Bottom) sample derived from Annotator compared to Human and ChatGPT labeling

*Results* : As shown by the tables and visual graphs, **Figure 1**, the Annotator sees 84% accuracy in determining SFW classification and 93% accuracy in NSFW classification. This is from a human perspective who have the ability to perform high level analysis of jokes and view them as clean/friendly or offensive/rude due to personal experience or world knowledge not particularly present in a transformative model. This also shows why ChatGPT has higher compatibility with the Annotator as both depend on natural language processing and neural networking. However, overall, the annotator will be expected to be reliable and of high quality as the sample produced acceptable results. Additionally, ChatGPT was not used as our annotator due to its inability to accept at least 10,000 queries at once in an effective and timely manner. ChatGPT is forced to break the prompts down into sections and give the labeling batch by batch waiting for an explicit prompt to continue each time – ineffective.

**D. ChatGPT Generated Jokes (Data Set 2)** [Jokes-GPT-only] [11] **:** In our second data set, we have produced 10,000 AI-generated jokes that were prompted to be either NSFW or SFW using the ChatGPT 4o model.

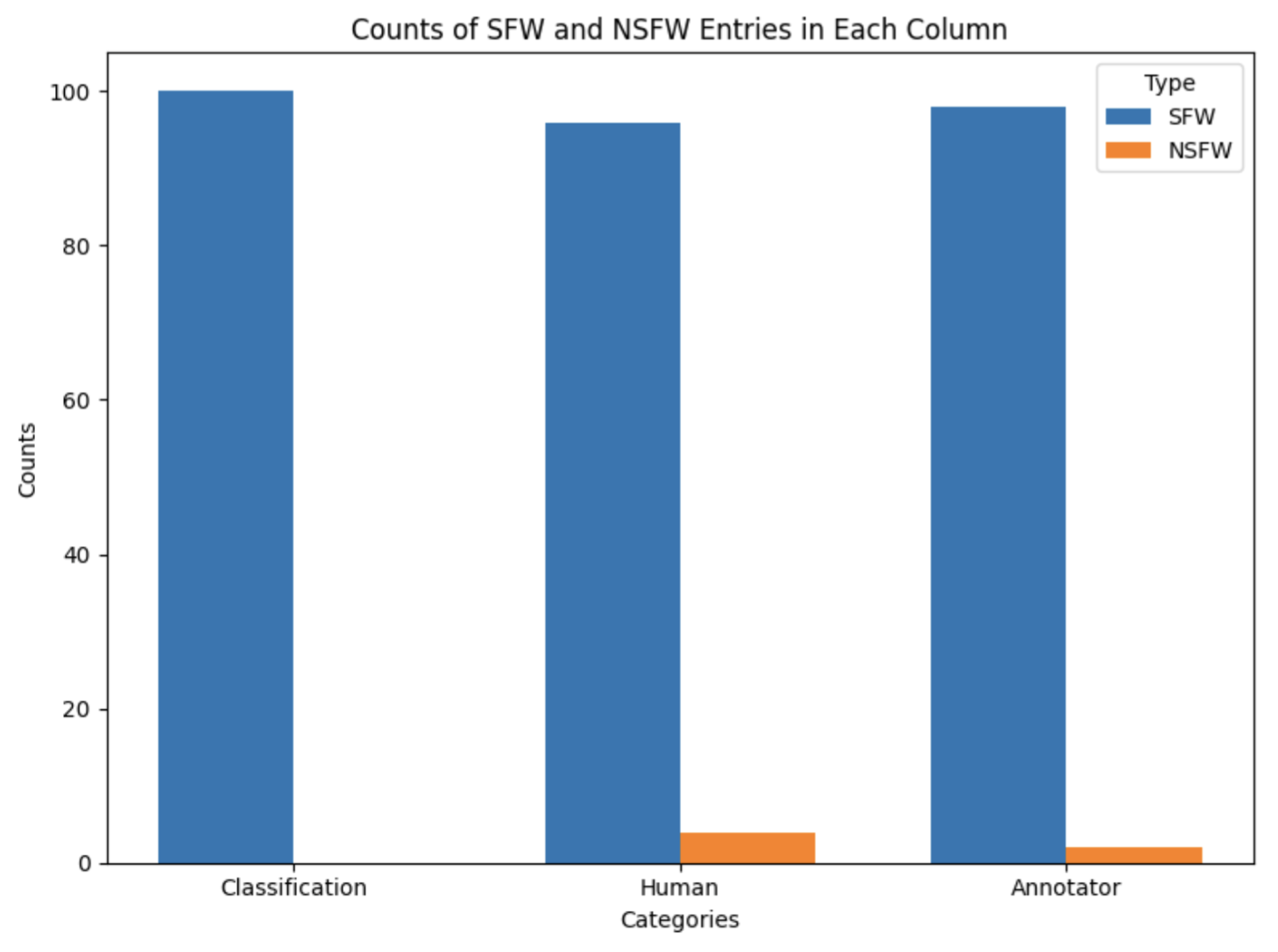
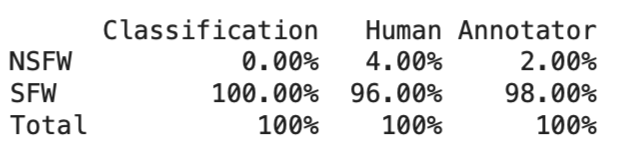
SFW – we prompted ChatGPT to produce SFW jokes that were unique and clean. At first, we gave a prompt for 5,000 jokes which failed immediately. We tried 1,000, 750, 500, and even 200 jokes at a time. Due to the computation limits, we could only prompt for 100 jokes at a time. This presented a need for multiple prompts (about 50), to receive a dataset of 5000 SFW jokes. To make sure duplicates were not created, we added certain topics to our prompts like animals, food, travel, etc. to force ChatGPT to create fresh jokes for each query.

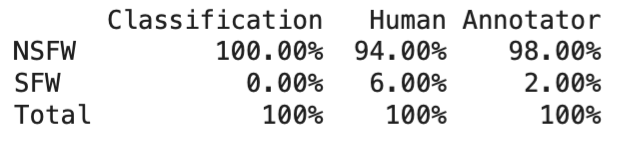
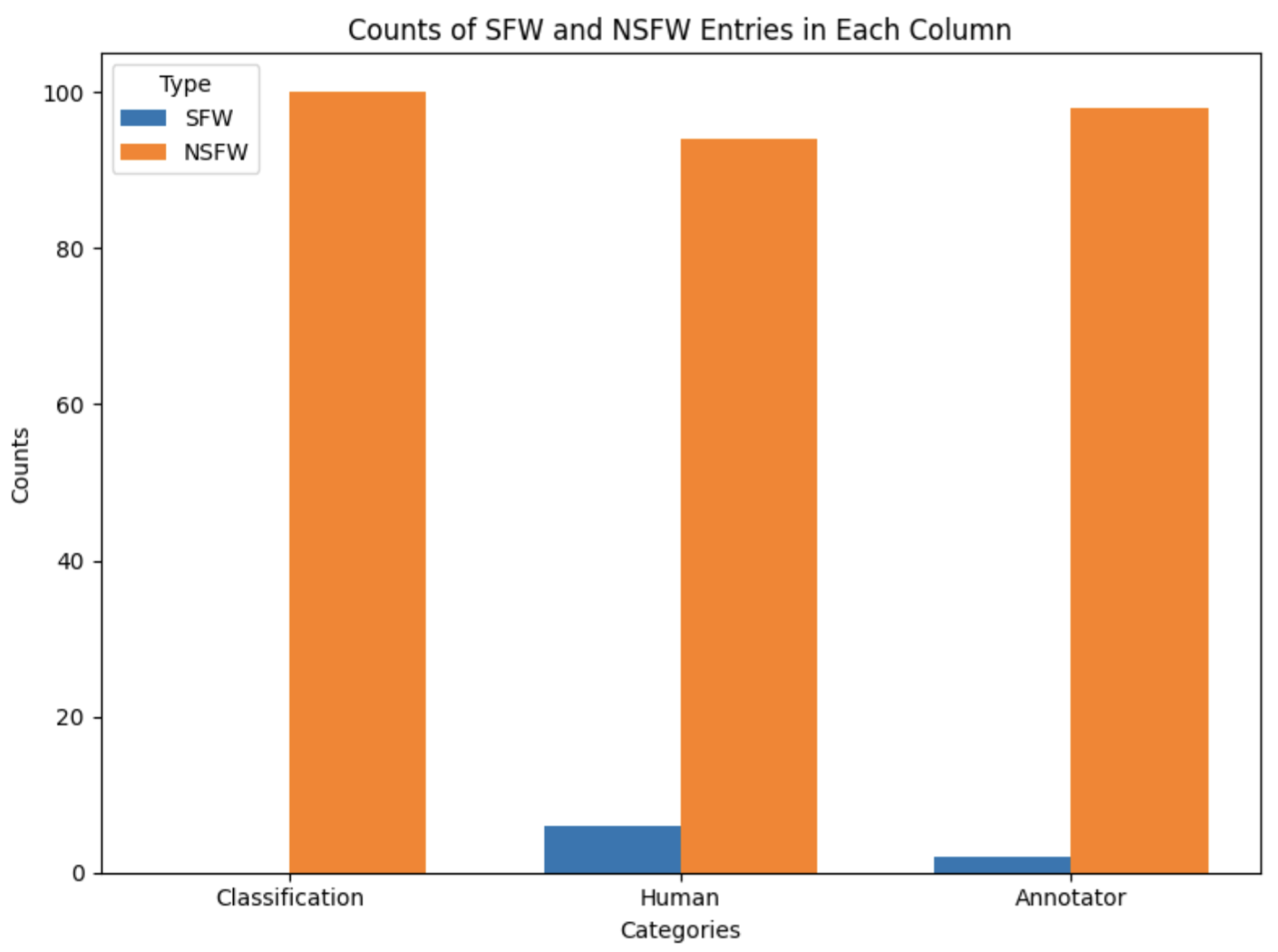
NSFW – we prompted ChatGPT to bypass its restrictive guidelines for not producing offensive content by explaining that the data was for research purposes. This gave us the ability to prompt for NSFW jokes as needed. From here the process matched the SFW approach, and we obtained 5000 NSFW jokes of various topics.

**Ensuring Quality of GPT Generated Data** : We will use two approaches to verify the accuracy of our annotator, manual and annotation.

*Manual* ***:*** Similar to our approach with data set 1, out of the 5,000 SFW/NSFW jokes, 100 SFW jokes and 100 NSFW jokes were randomly sampled and labeled with manual human effort

*Annotation :* As opposed to using ChatGPT to verify ChatGPT data, which would be redundant, the jokes were also passed into the annotator used for dataset 1, which had been proven to be reliable.





**Fig. 2**: Table/Bar Graph of percentages for SFW Joke (TOP) and NSFW Joke (Bottom) sample derived from ChatGPT compared to Human and Annotator labeling

*Results* : As shown by the tables and visual graphs, **Figure 2**, we can determine that ChatGPT is greatly successful at creating NSFW and SFW jokes from scratch as it sees 96% human accuracy for SFW and 94% human accuracy for NSFW. Once again, similar to the first dataset, NSFW is less accurate then SFW by a small amount which can be due to certain words like race having a double meaning that flags it as offensive when in reality, it was clean. We can once again also see the correlation between the Annotator and ChatGPT is stronger than the human perspective which is explained by how the jokes are broken into tokens with language processing using similar approaches and machine learning.

**E. Augmented Data w/ Synthetic Pairs (Data Set 3) [**Jokes-GPT-Augmented] [12] : In our third and final dataset, we have produced 10,000 synthetic jokes using ChatGPT based on 5,000 SFW jokes and 5,000 NSFW jokes. The purpose of this set is to test if ChatGPT has the ability to alter a joke into its opposite version. In other words can it correctly clean a human created NSFW joke and create a SFW joke that has a similar concept. And vice versa as well.

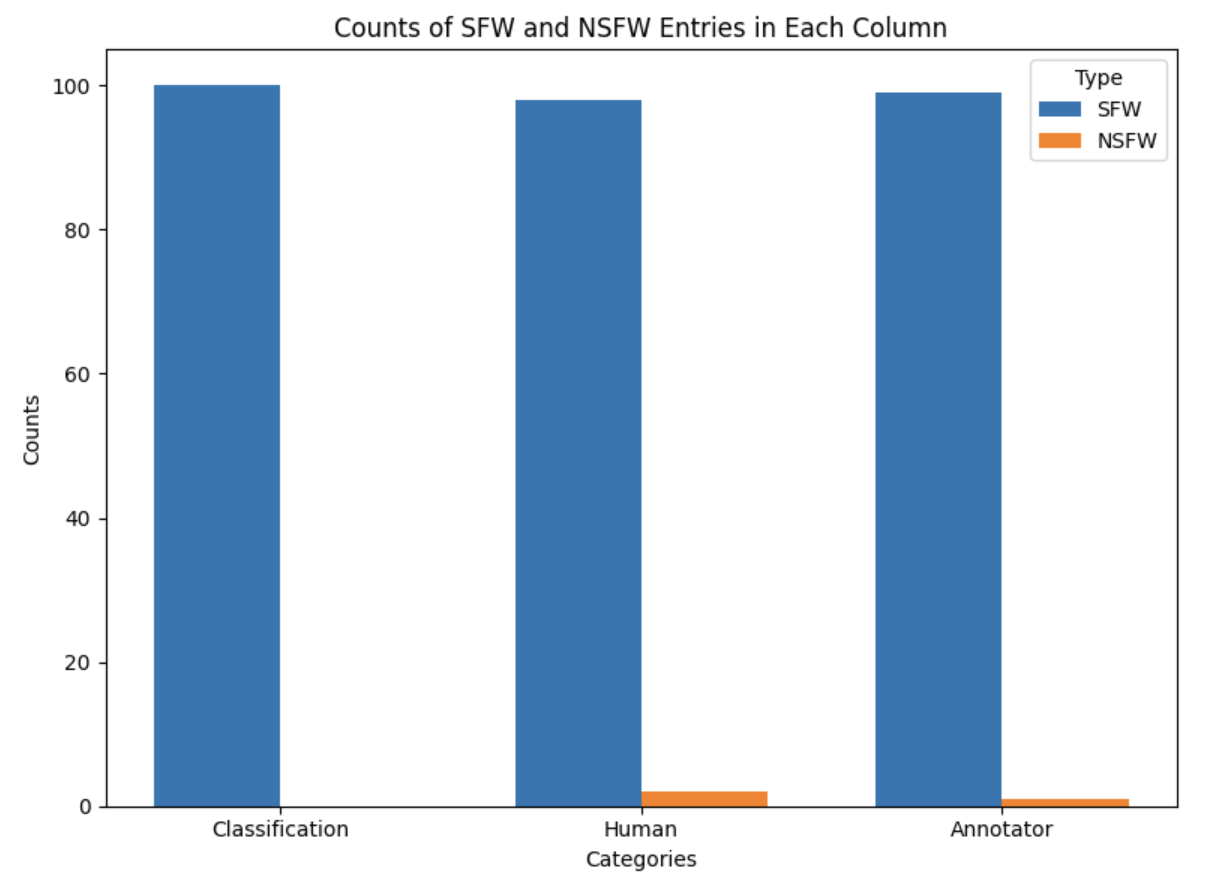
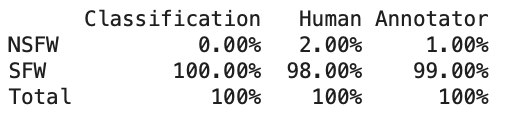
SFW – Similarly to data set 2, we prompted ChatGPT to produce NSFW jokes that were unique based on 5,000 randomly sampled human made SFW jokes from the Reddit dataset. These jokes were the same as the first dataset we did since SFW is the testing data that should match for fair evaluations by our models. Due to the computation limits, we prompted 100 jokes at a time. This presented a need for multiple prompts (about 50), to receive a dataset of 5000 SFW jokes. Despite the time and effort, 100 jokes at a time allowed us to make sure the topics of the outputted jokes were matching our inputted jokes effectively creating synthetic pairs of SFW,NSFW versions.

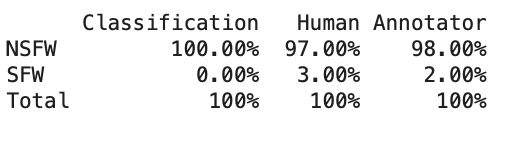
NSFW – As done for SFW jokes, this time we prompted ChatGPT to create clean SFW versions of 5,000 randomly sampled human made NSFW jokes from the Reddit dataset. From here the process mirrored the SFW approach, and we were able to confirm that the concepts of the pairs being created matched as intended. Another 5,000 pairs were created.

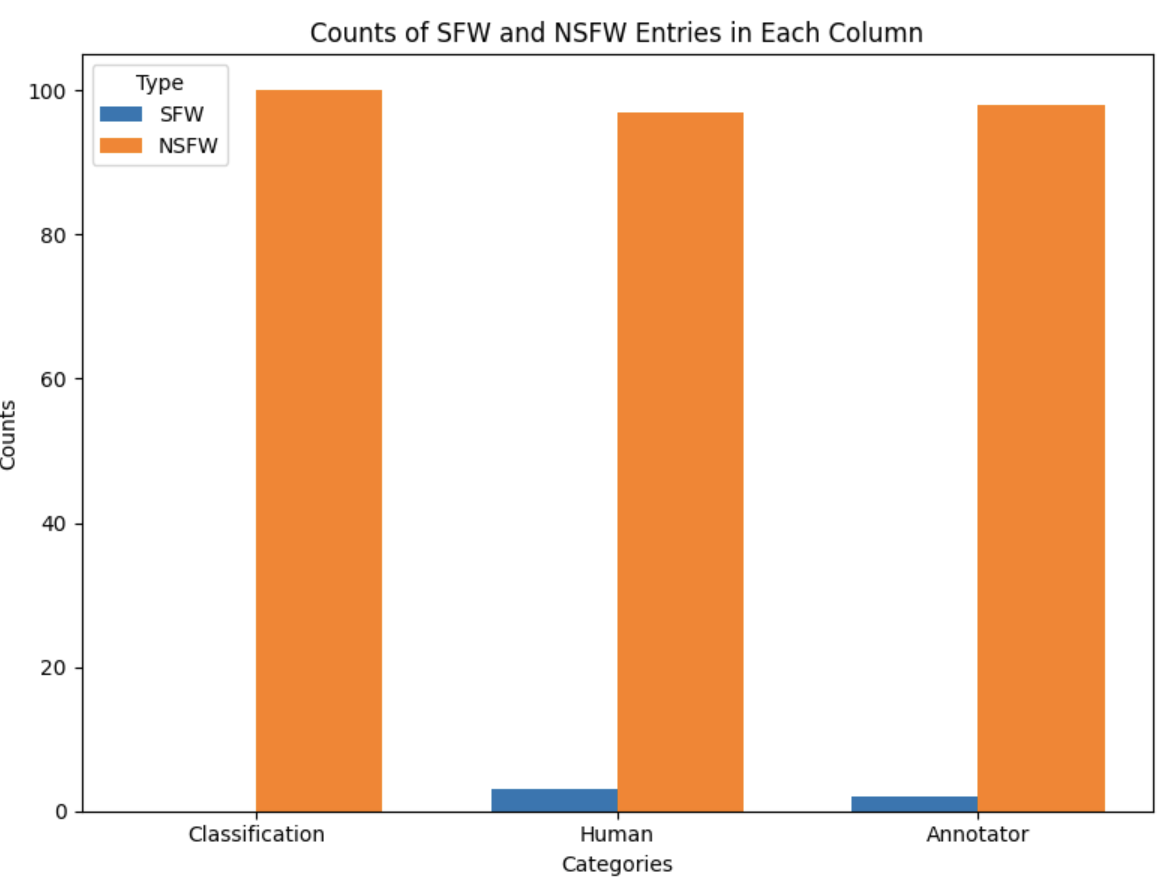
**Ensuring Quality of GPT Generated Synthetic Pairs** : We will use two approaches to verify the accuracy of our annotator, manual and annotation. This time we only tested the synthetic half of each pair, whether it was built to be a NSFW version or SFW version of the original joke.

*Manual* ***:*** Similar to our approach with data set 1 and 2, out of the 5,000 SFW/NSFW jokes, 100 SFW jokes and 100 NSFW jokes were randomly sampled and labeled with manual human effort

*Annotation :* Once again, as done with data set 2, the jokes were also passed into the annotator used for dataset 1, which had been proven its effectiveness in our last two uses.







**Fig. 3**: Table/Bar Graph of percentages for SFW Joke (TOP) and NSFW Joke (Bottom) sample derived from ChatGPT synthetic pairs half compared to Human and Annotator labeling

*Results* : As shown by the tables and visual graphs, **Figure 3**, we see an improvement in human classification for ChatGPT produced data. This is efficiently explained by the fact that ChatGPT was given seed inputs to work with. The model became aware of what made a joke NSFW and removed those portions effectively without changing the concept of the jokes. Similarly for SFW, ChatGPT inputted curse words and rude language to turn the same joke into its NSFW version. This is evident by higher correctly classified NSFW/SFW jokes as well as a higher correlation with the annotator.

III. Models

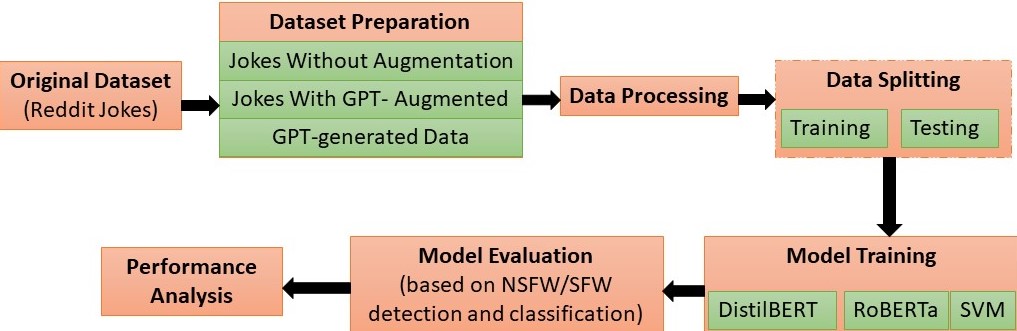
The models we will use for our analysis are **DistilBERT, RoBERTa, and SVM**. DistilBERT is a lighter, distilled version of BERT that has achieved competitive performance while considerably reducing the computational cost, Thus, this model is very suitable for large-scale experiments over augmented datasets. RoBERTa is a robustly optimized variant of BERT, efficient at capturing subtle contextual representations, for understanding the subtleties of offensiveness in text. Complementing these models with transformer models, classic machine learning methods like SVM are strong baselines for the task of text classification

IV. Methods

**A. Approach :** This series of experiments explores the impact of GPT-generated data augmentation on the performance of machine learning models tasked with classifying jokes as either NSFW (Not Safe For Work) or SFW (Safe For Work). **Figure 4** illustrates the structure of the procedure. The experiments were structured to evaluate three scenarios:

1. **Jokes Without Augmentation:** Training and testing with human-authored jokes.
2. **Jokes with GPT-Augmented Data:** Training on GPT-augmented datasets where NSFW jokes were paired with GPT-generated SFW counterparts, and testing with human-authored.
3. **GPT-generated data generalize to human-authored jokes:** Training with GPT Data, Testing with Human Data: Evaluating how well models trained exclusively on GPT-generated data generalize to human-authored jokes.

We employed SVM, and two pre-trained language models — DistilBERT, and RoBERTa— to compare performance across these setups.

**Fig. 4:** Overall Methodology for NSFW vs. SFW Joke Classification.

**B. Experimental Setup:**

1. **Jokes Without Augmentation**:
   * **Dataset**: Jokes-wo-GPT: Subset created from the reddit jokes dataset.
   * **Training Dataset**: Balanced to include 3000 NSFW and 3000 SFW jokes.
   * **Testing Dataset**: 1200 unique jokes (600 NSFW and 600 SFW).
2. **Jokes with GPT-Augmented Data**:
   * **Dataset**: Jokes-GPT-Augmented: GPT- augmented datasets where NSFW jokes were paired with GPT-generated SFW versions.
   * **Training Dataset**: Balanced with 3000 NSFW and 3000 SFW pairs.
   * **Testing Dataset**: A balanced subset of 1200 unique GPT-augmented jokes.
3. **Training with GPT Data, Testing with Human Data**:
   * **Dataset**: Jokes-GPT-only: GPT-generated training dataset; human-authored test dataset.
   * **Training Dataset**: 5000 NSFW and 5000 SFW GPT-generated jokes.
   * **Testing Dataset**: 1200 unique human-authored jokes from Reddit dataset (600 NSFW and 600 SFW).

V. Evaluation Metrics

The metrics for evaluation that we are focusing on is : **Accuracy, Precision, Recall, and F1-Score.** These metrics provide a comprehensive understanding of the performance of a model, especially in tasks like humor detection, where nuanced predictions are common. Accuracy gives an overall measure of the model's ability to correctly classify jokes and non-jokes, offering a straightforward snapshot of performance. However, it may not reflect true performance in imbalanced datasets, making Precision and Recall essential complements. Precision in this regard measures how the model does not overclassify, giving false positives. This aspect is crucial when detecting offensive humor, as too much classification will result in censorship of benign humor. Recall calculates how many instances the model has captured relevantly; this ensures that the offensive jokes are suitably flagged. The F1-Score is the harmonic mean of Precision and Recall, balancing out the two, and in turn, it becomes one score for both false positives and false negatives. Thus, it will prove valuable for model evaluation on a dataset with subjective distinctions.

VI. Results And Discussion

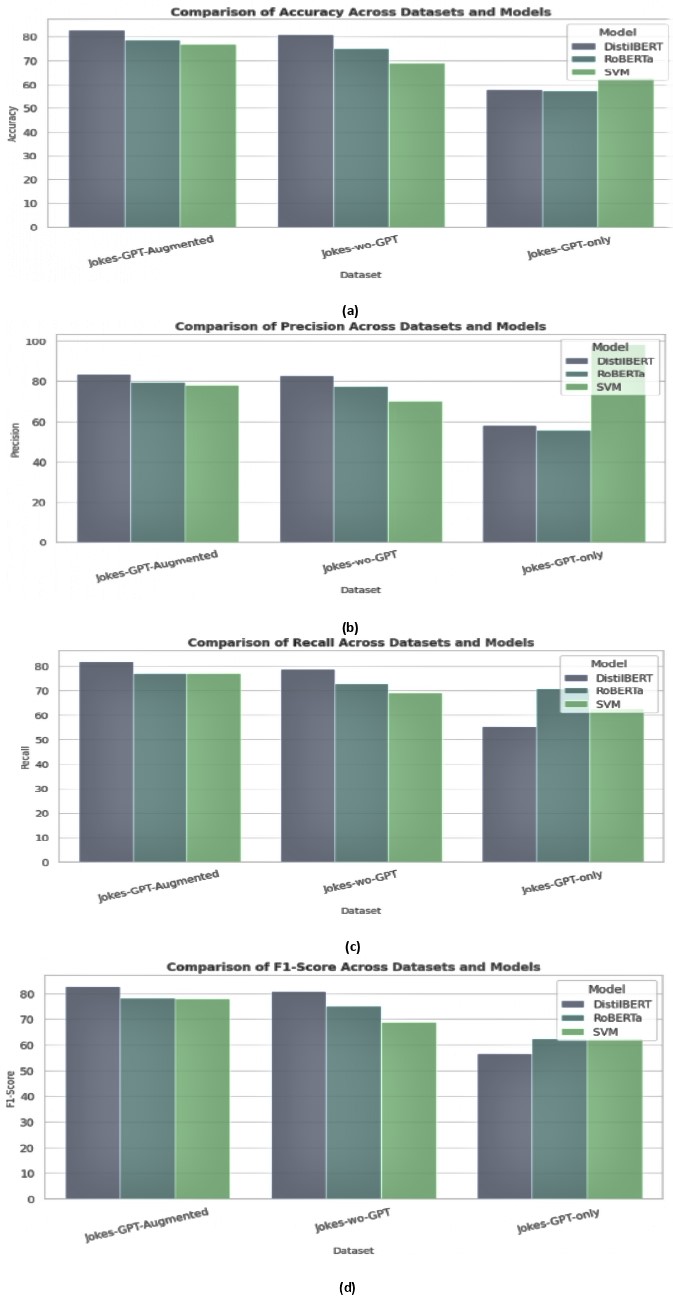
We observe that the models trained on the GPT-augmented dataset demonstrated improvements in accuracy, precision, and F1 scores across DistilBERT, RoBERTa, and SVM compared to models trained on the dataset without augmentation. The complete experimental results are shown in **Table 1**.

**Table 1.** Model Evaluation Results Across Datasets

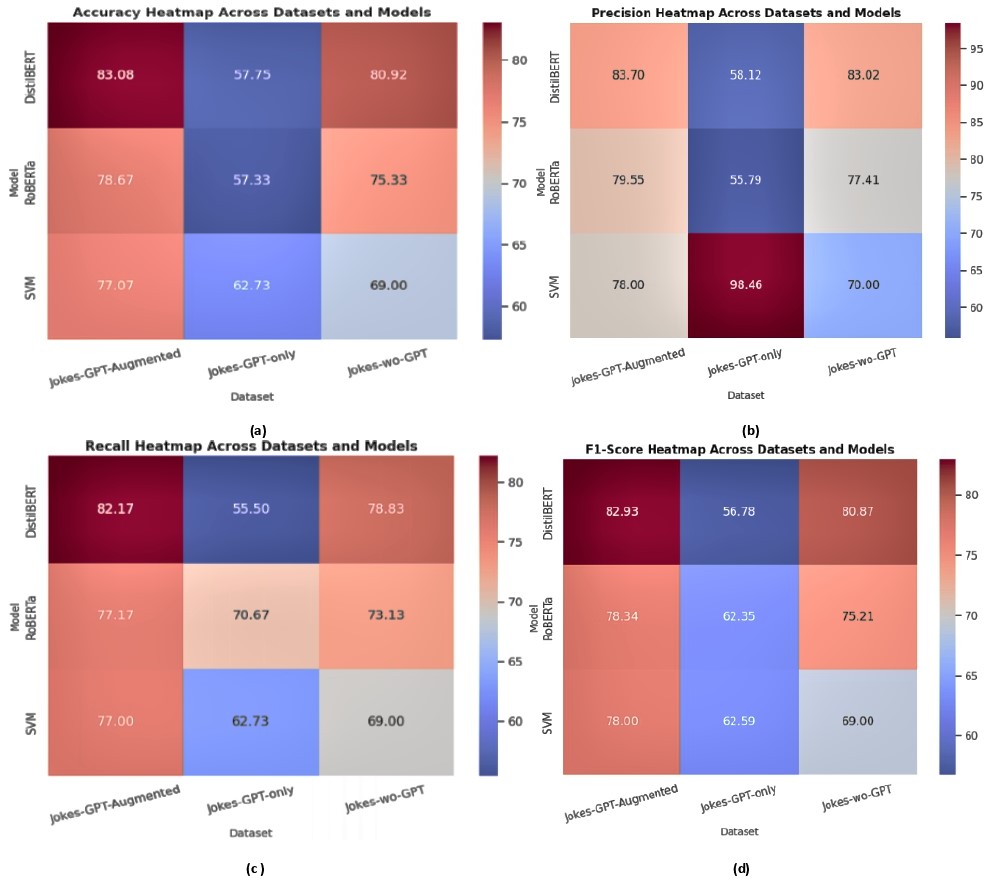
| **Dataset** | **Model** | **Accuracy**  **(%)** | **Precision**  **(%)** | **Recall**  **(%)** | **F1-Score**  **(%)** |
| --- | --- | --- | --- | --- | --- |
| **Jokes-**  **GPT-**  **Augme-nted** | **DistilBERT** | **83.08** | **83.70** | **82.17** | **82.93** |
| **RoBERTa** | 78.67 | 79.55 | 77.17 | 78.34 |
| **SVM** | 77.07 | 78.00 | 77.00 | 78.00 |
| **Jokes-**  **wo-**  **GPT** | **DistilBERT** | 80.92 | 83.02 | 78.83 | 80.87 |
| **RoBERTa** | 75.33 | 77.41 | 73.13 | 75.21 |
| **SVM** | 69.00 | 70.00 | 69.00 | 69.00 |
| **Jokes-**  **GPT-**  **only** | **DistilBERT** | 57.75 | 58.12 | 55.50 | 56.78 |
| **RoBERTa** | 57.33 | 55.79 | 70.67 | 62.35 |
| **SVM** | 62.73 | 98.46 | 62.73 | 62.59 |

Below is a summary of the impact of data augmentation across the performance metric for the models:

The GPT-augmented data improves accuracy for all three models, with a larger increase for RoBERTa (+3.34%) compared to DistilBERT (+2.16%), and the highest for SVM (+8.07). RoBERTa also benefits more from GPT-augmentation in precision (+2.14%) compared to DistilBERT (+0.68%). SVM acquires a higher precision as well. For recall, all three models have the most positive effect on recall indicating better detection of true positives with GPT-augmented data. RoBERTa shows a higher improvement (+4.04%) than DistilBERT (+3.34%). Additionally, the F1 score, which balances precision and recall, also sees improvement with GPT-augmentation, especially for RoBERTa (+3.13%) and SVM (+9%).



**Fig. 5:** Comparison of Model Performance Across Datasets and Metrics



**Fig. 6:** Heatmaps of Model Performance Metrics Across Datasets and Model

**Figure 5** compares different models (DistilBERT, RoBERTa, and SVM) across three datasets (Jokes-GPT-Augmented, Jokes-wo-GPT, and Jokes-GPT-only) using four evaluation metrics: (a) Accuracy, (b) Precision, (c) Recall, and (d) F1-Score. Additionally, **In Figure 6**, we present a heatmap of our experimental results. These figures show that, among the top-performing models, DistilBERT and RoBERTa the largest gain is +4.04%. Additionally, RoBERTa exhibited about 1% more improvement in accuracy compared to DistilBERT. We reason that these improvements across the metrics are likely due to the fact that augmented pairs show subtle variation in words, phrasing, and expressions, with somewhat similar sentence structure and context. This diversity helps models learn to identify patterns of offensiveness or NSFW content based on semantic nuances rather than overfitting to specific word patterns. Additionally, GPT-augmented pairs may also include slight grammatical or syntactic differences, mimicking real-world variations in how jokes are written or delivered. However, both DistilBERT and RoBERTa struggle when tested on human-generated jokes after being trained solely on GPT-generated data (Jokes-GPT-only). This implies that it does not capture all the nuances and variations as human-authored jokes. Nevertheless, GPT generated jokes are helpful when used as a counterpart for human-authored jokes for training. In addition to these analyses, in **Table 2**, we aggregate the results of relevant background studies done on the field of humor-detection and show that the results are on par with the current standards.

**Table 2.** Comparison of the proposed study with existing state-of-the-art methods

| **References** | **Methodology Used** | **Dataset** | **Accuracy**  **(%)** |
| --- | --- | --- | --- |
| Horvitz et al. [1] | GPT, Mistral, RoBERTa | Unfun Corpus (~11,000 pairs), Code-mixed English-Hindi dataset (~591 tweets) | 76.50 |
| Tang et al. [2] | DeBERTa, BERT, RoBERTa, and Longformer | Reddit dataset with 92,153 jokes | 87.69  (Humor subtype) |
| Turano et al. [7] | Random Forest, Logistic Regression, Naive Bayes | SCRIPTS (19137 samples: 9647 humorous, 9490 non-humorous) | 72.00 |
| Qiu et al. [3] | GPT, BERT | CENSORCHAT dataset containing real-life human-machine dialogues | 91.00 |
| Gupta et al. [4] | BERT, RoBERTa, ERNIE-2.0, XLNet, and DeBERTa | SemEval-2021 Task 7 dataset with 8,000 training and 1,000 public-dev labeled examples | 62.02 (Humor subtype) |
| Chaudhary et al. [8] | BERT, LSTM, and Hybrid approach | 200k jokes from Reddit and Firefox discussion forum (non-humorous data) | 63.00 |
| Barbieri et al. [9] | RoBERTa | TWEETEVAL benchmark with 7 tweet classification tasks | 78.50 |
| **Proposed Study** | GPT, SVM, DistilBERT, RoBERTa | **Jokes-GPT-Augmented (10000)** | **83.08** |
| Jokes-wo-GPT (10000) | 80.92 |
| Jokes-GPT-only (10000) | 62.73 |

VII. Limitations

While the outcomes of this research seem encouraging, certain elements remain unexplored due to computational constraints that prevented testing on bigger and more diversified datasets. Models trained solely on GPT-generated data encountered difficulties in generalizing to human-authored jokes, suggesting areas for improvement. Furthermore, the subjective nature of humor and offensiveness may result in profound differences, guiding additional testing and adjustments in future work.

VIII. Future Work

Future work can aim to explore several paths to further improve NSFW joke classification. First, we would like to see what effect can be produced by the usage of a more diverse pre-trained language model, such as Gemini or Anthropic. Second, we can work to train a model to identify sarcasm within humor using the richness of the joke dataset and test it on an external dataset for generalization. Sarcasm can be tested independently or interconnected with NSFW classification as improper sarcasm often leads to rude or dark jokes. Third, the integration of context-aware sentiment analysis could bring better detection as the models will have additional information to classify the entries. Finally, label reliability that is processed at the start can be further improved by increasing annotation from multiple annotators.

IX. Conclusion

This study shows that GPT-augmented datasets can improve NSFW joke classification by attaining significant performance gains, such as an F1-score of 82.93% with DistilBERT and a 4.04% recall boost with RoBERTa. The study emphasizes the importance of combining human-authored and synthetic data in developing strong humor detection methods. While models trained simply on GPT-generated data struggled to generalize, supplementary datasets effectively closed performance imbalances. These findings demonstrate the potential of GPT-augmented data as a supplemental resource for improving moderation of content and humor detection algorithms, paving the door for more complex, context-aware AI applications in this sector.

X. Data Availability

The data and proposed approaches are publicly available onGitHub:<https://github.com/Tanoy004/Analysis-of-a-Large-Joke-Corpra/>

XI. References

[1] Z. Horvitz, L. Mayfield, and K. Seppi, "Getting Serious about Humor: Crafting Humor Datasets with Unfunny Large Language Models," arXiv preprint arXiv:2403.00794, 2024.

[2] L. Tang, W. Wang, and B. Wang, "The Naughtyformer: A Transformer Understands Offensive Humor," arXiv preprint arXiv:2211.14369, 2022.

[3] H. Qiu, L. Zhou, M. Li, Y. Zheng, and X. Wang, "Facilitating NSFW Text Detection in Open-Domain Dialogue Systems via Knowledge Distillation," arXiv preprint arXiv:2309.09749, 2023

[4] Gupta, Aishwarya, et al. "Humor@ IITK at SemEval-2021 task 7: Large language models for quantifying humor and offensiveness." *arXiv preprint arXiv:2104.00933* (2021).

[5] Priyam Choksi, “1 Million Reddit Jokes (r/jokes).”, Kaggle, Available at:<https://www.kaggle.com/datasets/priyamchoksi/1-million-reddit-jokes-rjokes>.

[6] Michelle Li’s NSFW\_text\_classifier, Hugging Face, Available at: :https://huggingface.co/michellejieli/NSFW\_text\_classifier?not-for-all-audiences=true

[7] Turano, Beatrice, and Carlo Strapparava. "Making People Laugh like a Pro: Analysing Humor Through Stand-Up Comedy." *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 2022.

[8] Chaudhary, Tanishq, Mayank Goel, and Radhika Mamidi. "Towards conversational humor analysis and design." *arXiv preprint arXiv:2103.00536* (2021).

[9] Barbieri, Francesco, et al. "Tweeteval: Unified benchmark and comparative evaluation for tweet classification." *arXiv preprint arXiv:2010.12421* (2020).

[10] Dataset 1, *Jokes-wo-GPT*, GitHub Repository: <https://github.com/Tanoy004/Analysis-of-a-Large-Joke-Corpra/blob/main/JOKES-wo-GPT.csv>.

[11] Dataset 2, *Jokes-GPT-Augmented*, GitHub Repository:<https://github.com/Tanoy004/Analysis-of-a-Large-Joke-Corpra/blob/main/Jokes-GPT-Augmented.csv>.

[12] Dataset 3, *Jokes-GPT-Only*, GitHub Repository: https://github.com/Tanoy004/Analysis-of-a-Large-Joke-Corpra/blob/main/Jokes-GPT-Only.csv.