Emotion Classification in Natural Language Processing

Banovac, Lea (224055), Dziechciarz, Michal (225484), Heesters, Stijn (226180), Ribeiro Mansilha, Francisco (220387), van den Berg, Rens (224190)

Artificial Intelligence & Data Science, Breda University of Applied Sciences

Block C: Artificial Intelligence Scientist - Natural Language Processing

Msc. Dean van Aswegen

April 11, 2024

Contents

Introduction

Natural language processing, or NLP, is a significant subject of study and research in the quickly developing field of artificial intelligence (AI), with the goal of bridging the comprehension gap between humans and machines. The job of emotion classification, or recognizing and classifying emotions in text, is a major challenge in this subject. This work requires not just an in-depth knowledge of language syntax and semantics but also an understanding of the nuances of emotion, such as cultural interpretations, tone, and context.

The importance of emotion classification is demonstrated by its diverse applications, which range from improving user interactions with chatbots and virtual assistants to delivering significant insights in social media sentiment analysis, healthcare, and even the entertainment industry. Our project, in collaboration with Banijay Benelux and 3Rivers media consultants, aims to make a significant contribution to this latter domain by using emotion classification to analyze and understand the emotional dynamics in television programming, specifically the series "Expeditie Robinson."

Emotion classification in NLP has its roots in psychology's historical investigation of emotions. Paul Ekman's pioneering work in the early 1970s, which established six primary emotions (happiness, sadness, fear, anger, surprise, and disgust), has helped to guide research and development activities in this field. Even with time and the development of increasingly complex emotional frameworks, these fundamental emotions are still the starting point for most emotion classification tasks.

To ensure the highest quality of our emotion detection algorithm, there was a Kaggle competition among groups of students, utilizing datasets from diverse sources. This competitive approach not only honed the skills of participants in handling real-world data but also led to the development of sophisticated models capable of accurately classifying emotions. These refined models were then applied to textual data extracted from "Expeditie Robinson" episodes through

advanced speech-to-text technology, aiming to accurately identify and categorize the range of emotions expressed in various fragments of the show.

This report will discuss the theoretical foundations of emotion classification, the approaches used for data processing and model training, and the difficulties encountered in bridging the gap between theoretical models and their practical application on real-world data as we get into the specifics of the project. Our goal is to add to the ongoing discussion in the AI field regarding the challenges associated with recognizing human emotions and how natural language processing (NLP) can change the way people engage with media and technology.

Data Processing and Exploration

Overview of Datasets

The project's emotion classification model leverages a diverse range of datasets to capture a wide spectrum of emotional expressions within text. These include GoEmotions, SMILE Twitter Emotion dataset, Friends emotion-labeled dialogues, MELD dataset, CARER dataset, Affective Text, Daily Dialogue, EmoBank, and Affect data, sourced from various platforms such as social media, TV show transcripts, and literature. This broad dataset collection is foundational for understanding and classifying textual emotion expressions comprehensively.

Artificially Generated Data

In addition to the mentioned datasets, the project incorporates an innovative approach by generating 30,000 artificial sentences for each emotion using the OpenAI API. This process involved

dynamic prompting and topic generation to create a wide range of contextually rich sentences that simulate various emotional states. This artificial dataset significantly enhances the model's training data, providing diverse and nuanced examples of each targeted emotion, thereby enriching the model's learning and generalization capabilities.

Training and Test Data

The training dataset is comprised of GoEmotions, SMILE Twitter Emotion, Friends emotion-labeled dialogues, and MELD datasets, which offer a rich array of contexts and emotional expressions. For example, the GoEmotions dataset, consisting of manually annotated Reddit comments, spans 27 emotions plus a neutral category, providing a broad emotional spectrum. The test dataset was creaated by collecting 5 sentences per emotion from students, which were then validated by lecturers of the ADSAI (Applied Data Science & Artificial Intelligence) programme at Breda University of Applied Sciences (BUas).

Use Case Data: Expeditie Robinson

Specifically tailored to the project's use case, the data from Expeditie Robinson, a well-known TV series, comprises manually annotated episode segments that highlight the emotional dynamics among participants. This dataset presents a unique challenge due to the nuanced emotional expressions and complex participant interactions, mirroring the intricate nature of human emotions in real-world social interactions.

Annotation and Reliability

Annotations across the datasets vary, from manual annotations by trained professionals in datasets like GoEmotions and Friends dialogues to distant supervision in the CARER dataset. Manual annotations are deemed more reliable due to the detailed attention to context and adherence to

consistent annotation guidelines. However, datasets annotated through distant supervision might contain noise, as the annotations are derived from indirect cues rather than direct analysis.

The Expeditie Robinson data, annotated by show experts, ensures high relevance and reliability but may introduce some subjectivity, given that emotion perception can vary among individuals.

Sentence Length Analysis

An important aspect of our dataset's structural analysis was the evaluation of sentence lengths. This analysis included calculating the average, minimum, and maximum sentence lengths across the entire dataset and within each emotional category:

Overall average sentence length: 15.83 words

Overall minimum sentence length: 1 word

Overall maximum sentence length: 784 words

A breakdown by emotion yielded the following:

Emotion	**Average Length**	**Min Length**	**Max Length**
Total	15.83	1	784
Anger	15.77	1	261
Disgust	14.51	1	234
Fear	16.70	1	160
Happiness	18.03	1	258
Sadness	17.24	1	406
Surprise	16.05	1	237

Table 1 Average, minimum and maximum lengths of sentences (word counts) for each emotion in the final dataset

These metrics provide insight into the verbosity of emotional expression, indicating a tendency for sentences expressing happiness and sadness to be longer, potentially due to the complex narrative often required to convey these emotions.

Data Distribution and Commonality

The emotion distribution chart reveals a preponderance of 'happiness' and 'neutral' in the dataset, underscoring the presence of more nuanced emotional expressions that align with these categories. This distribution is crucial for understanding model performance, as imbalances can lead to biases in classification.

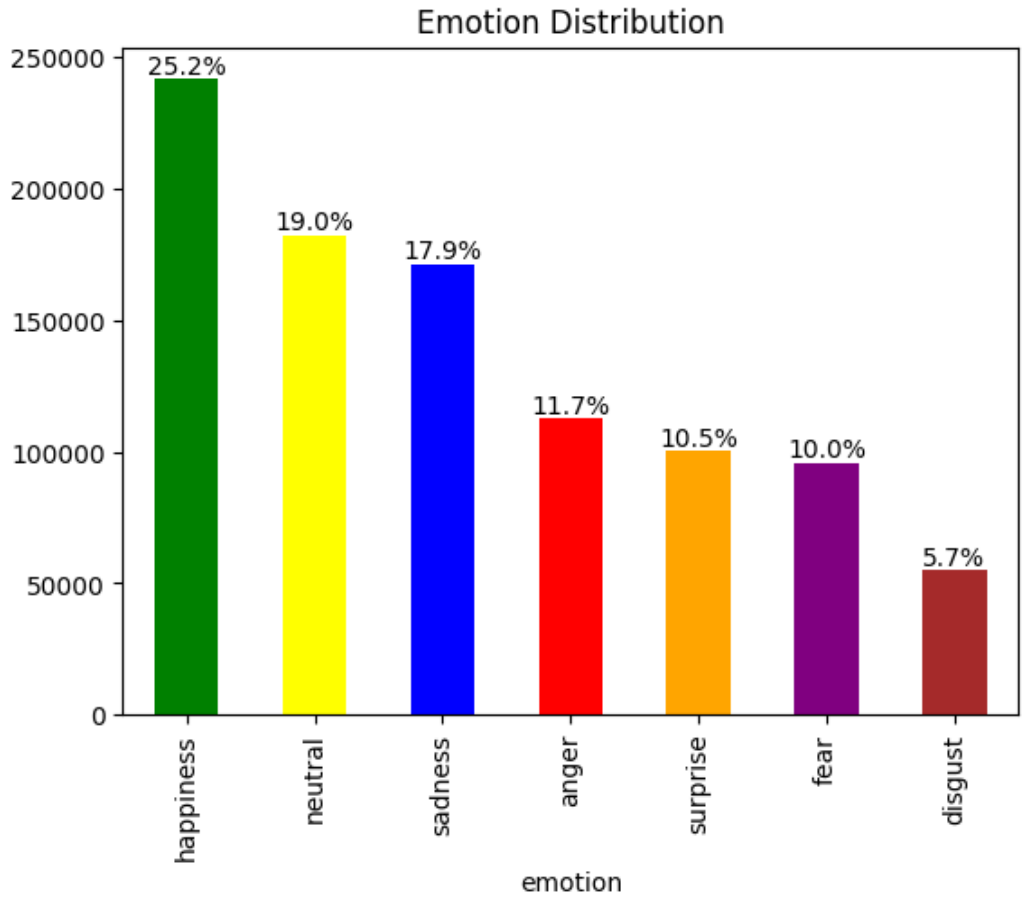


Figure 1 Data distribution between every emotion of the final merged dataset

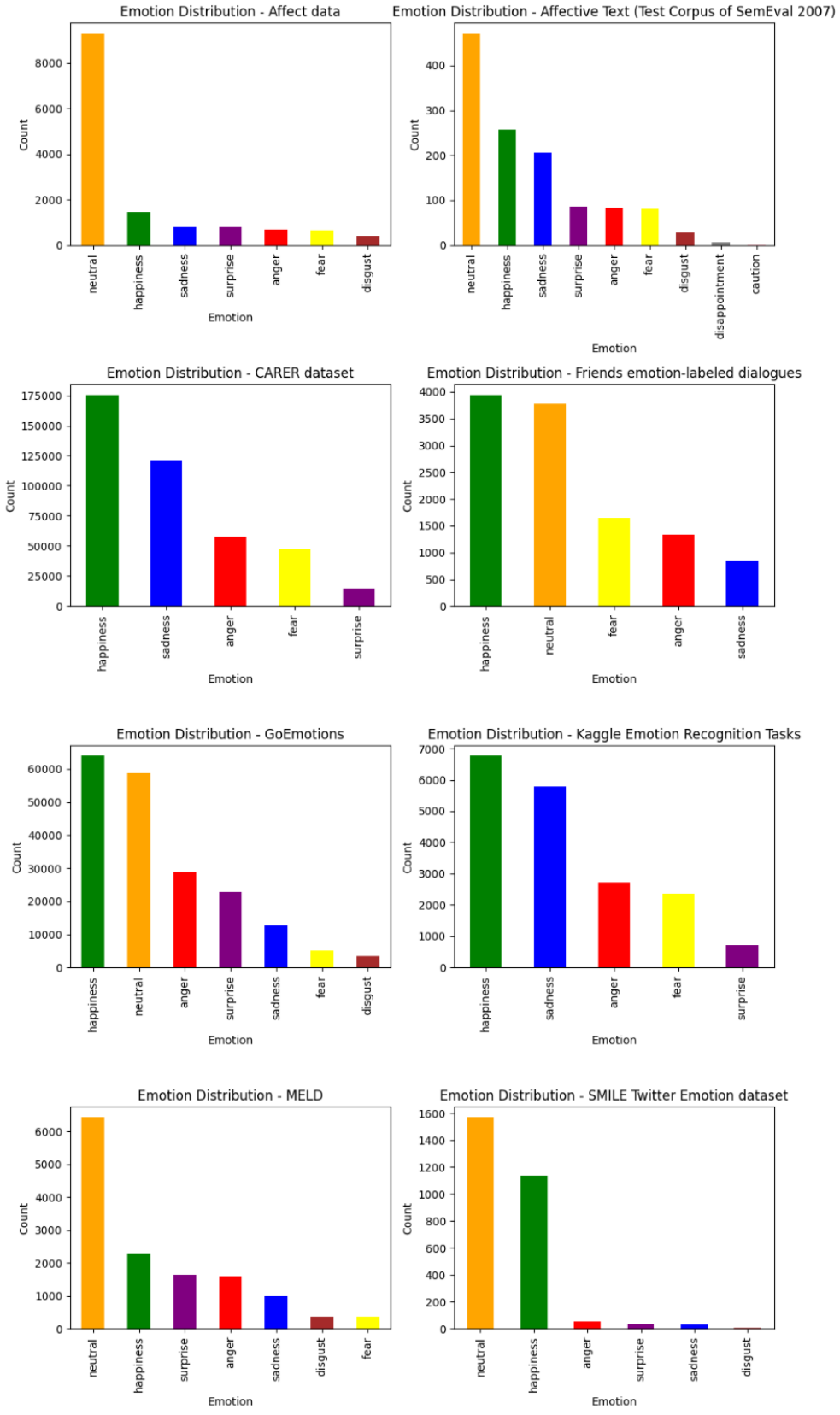
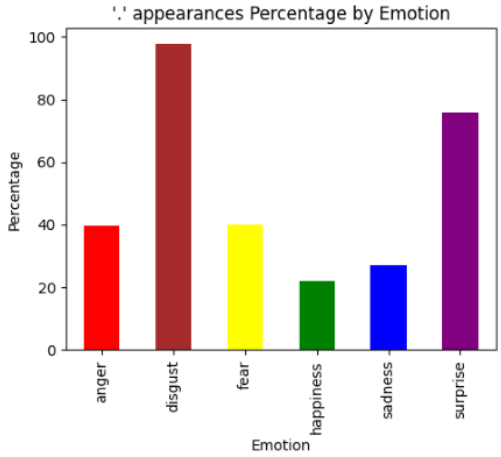
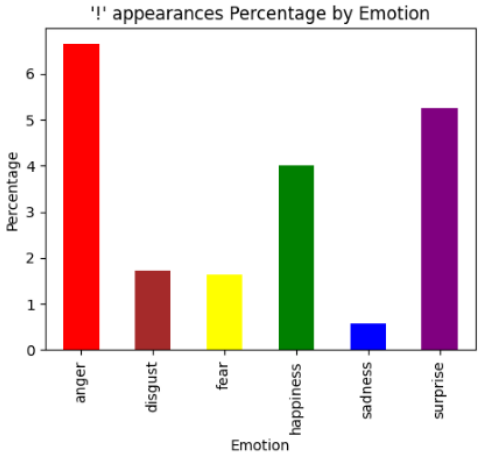
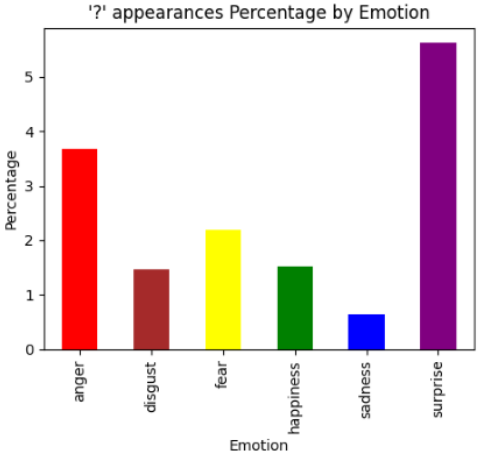


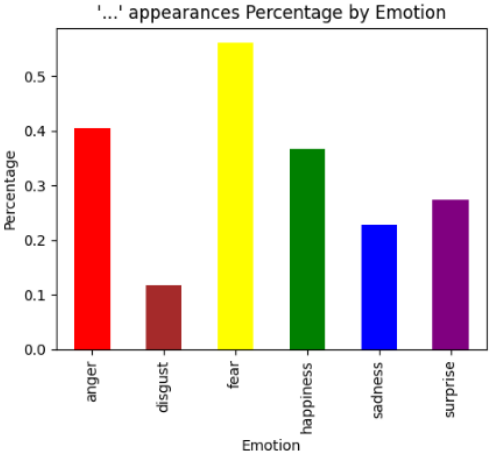
Figure 2 Data distributions for each dataset used

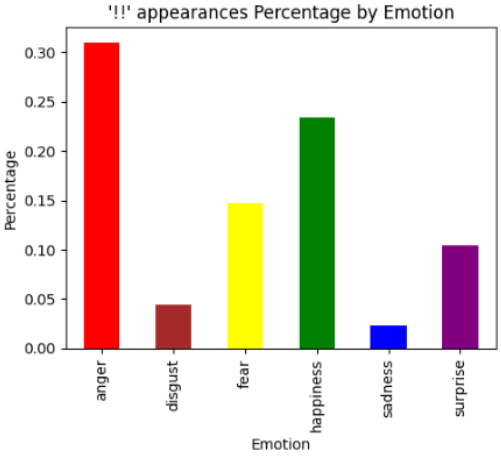
Analysis of sentence endings by emotion showcases the punctuation's role in conveying emotion. For example, exclamation points are predominantly found in sentences expressing 'anger' and 'surprise,' highlighting their function in depicting intensity or shock.

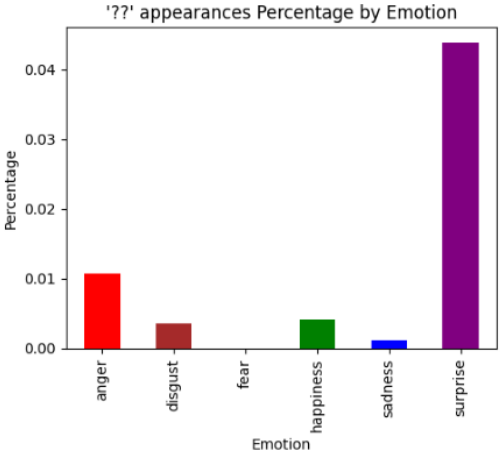

'.' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise

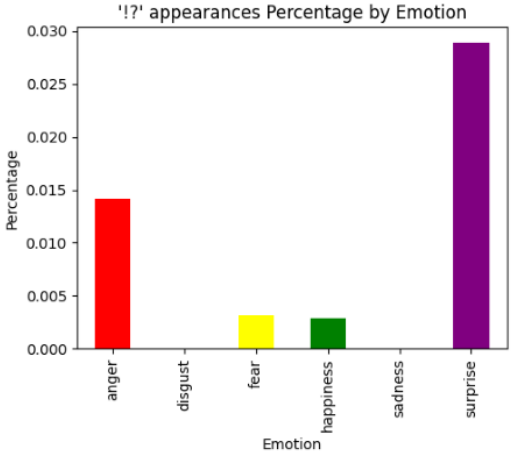

'!' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise


'?' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise


'...' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise


'!!' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise


'??' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise


'!?' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise

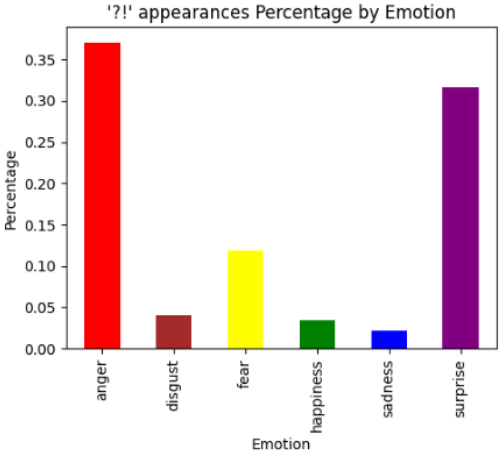

'?!' appearances Percentage by Emotion
Percentage
Emotion
anger
disgust
fear
happiness
sadness
surprise

Figure 3 Percentages of sentences ended with different puntucation marks

Word Frequency and Emotion

Word clouds provide a visual representation of the most common words associated with each emotion, elucidating the lexical choices that are frequently used to express each emotional state. For instance, words like "know," "feel," and "think" are prominent across multiple emotions, indicating their versatile role in constructing emotional narratives.

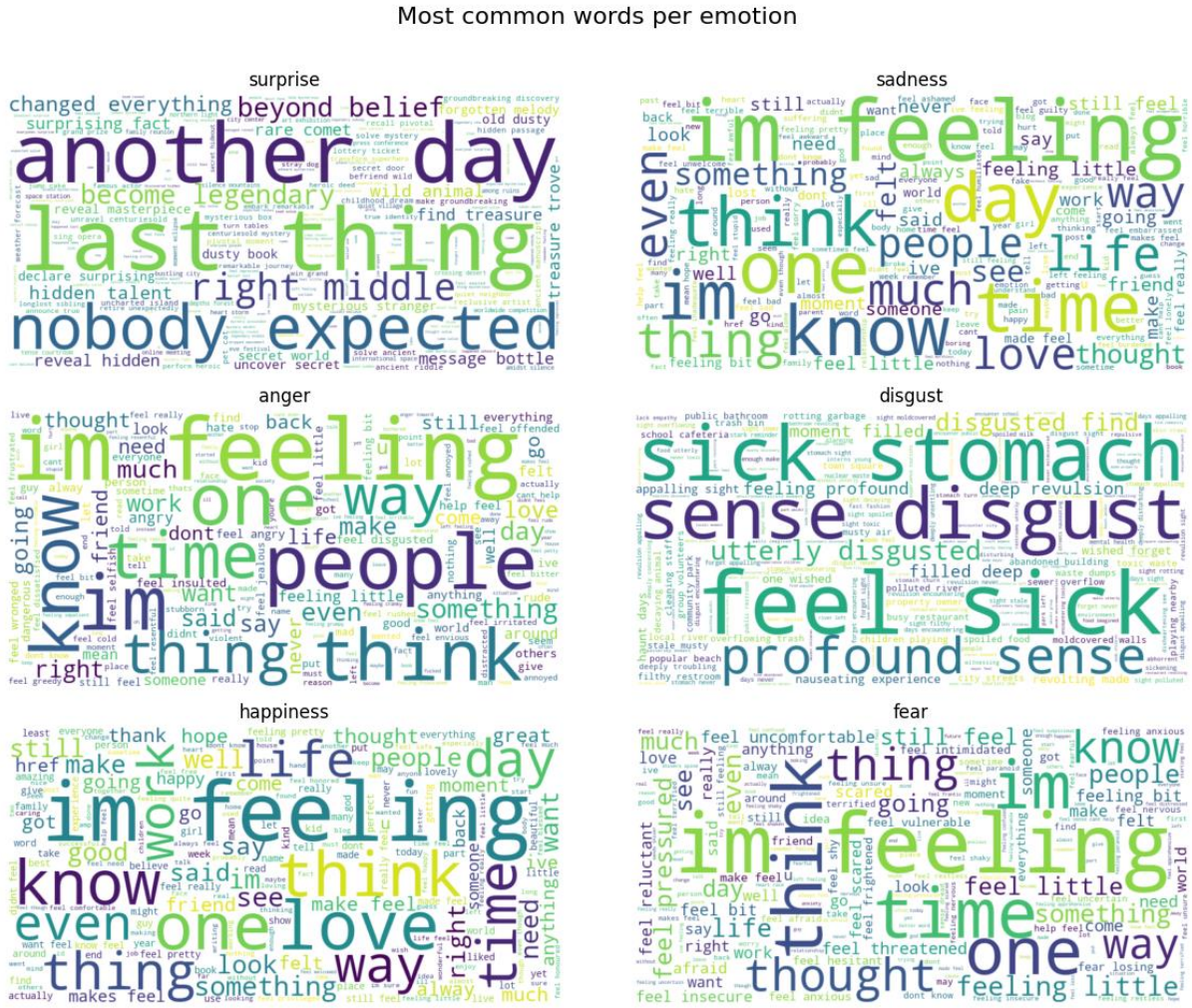


Figure 4 Most frequently used words in sentences of every emotion

Interpretation of Visual Data

The visual data encompassing word clouds and punctuation usage charts offers a multi-dimensional understanding of emotional expressions within our datasets. For example, the prominence of certain words within the 'anger' word cloud, such as "people," "feel," and "know," can be attributed to the personal and relational context in which anger is often expressed. Meanwhile, the frequency of punctuation like '!' in 'surprise' and 'anger' suggests a linguistic pattern where heightened emotions are coupled with more expressive punctuation.

The sentence-ending analysis, coupled with word frequency data, paints a detailed picture of how emotions are textually represented. By understanding these patterns, we can better preprocess the data and engineer features that are indicative of the emotional tone conveyed by sentence structure and word choice.

Preprocessing & Feature Engineering

In the development of our top-performing models, consistency in data preparation is key. The preprocessing steps described here form the backbone of our approach and have been uniformly applied across our three best models. This standardization ensures that the models are comparably trained and validated, allowing us to attribute differences in their performance to the model architecture and hyperparameters, rather than variations in data handling.

Dataset Loading and Initial Cleaning

The preprocessing process begins with loading our dataset, specifically created for our emotion classification task.

```python
# Specify the path to your CSV file
file_path = 'emotion_data_merged_8.csv'

# Load the dataset and drop duplicates
dataset = pd.read_csv(file_path).drop_duplicates(subset=['sentence'])
```

Python

Figure 5: Loading the dataset

After loading the dataset using the **pd.read_csv** function, the first action taken is to eliminate duplicate entries based on the 'sentence' column. This ensures the uniqueness of each data point, preventing overrepresentation of particular sentences and their associated emotions, which could otherwise skew the model's learning process.

Handling Missing Values and Neutral Emotions

Subsequent to deduplication, the dataset undergoes further refinement by dropping rows with missing values in the 'emotion' column. The presence of such entries could introduce ambiguity and dilute the model's ability to discern distinct emotional sentiments. Additionally, entries tagged with a 'neutral' emotion are filtered out as the project's scope is concentrated on analyzing the six basic emotions: happiness, disgust, sadness, fear, anger, and surprise.

```python
# Drop rows with missing values in the 'emotion' column and filter out 'neutral' emotions
dataset = dataset.dropna(subset=['emotion']).query("emotion != 'neutral'")
```

Python

Figure 6: Handling missing values

Label Standardization and Cleanup

An integral part of our data preprocessing is the standardization of emotion labels. The process highlighted a critical need for label mapping when two 'happiness' categories were discovered in the dataset, for reasons that remain elusive. This duplication risked the potential for confusion and error during the model's training phase, as it could inaccurately interpret these as distinct emotions.

```python
emotion_counts = dataset['emotion'].value_counts()
print(emotion_counts)
```

✓ 0.0s Python

```
emotion
happiness      207906
sadness        147572
anger           88599
surprise        76965
fear            62073
disgust         33609
happiness          20
Name: count, dtype: int64
```

Figure 7: 2 happiness classes due to unknown error

To resolve this, we consolidated similar or duplicated emotion labels, such as the two variations of 'happiness', into a single, consistent label. This not only streamlines the label space but also eliminates the ambiguity that could adversely affect the learning algorithm. Such standardization is paramount for maintaining the dataset's integrity and ensuring the reliability of our emotion classification model's outcomes.

```python
# Replace any labels not in the specified list with NaN
valid_labels = ['happiness', 'disgust', 'sadness', 'fear', 'anger', 'surprise']
dataset['emotion'] = dataset['emotion'].apply(lambda x: x if x in valid_labels else pd.NA)

# Map any duplicate labels to one of the specified labels
label_mapping = {
    'happiness': 'happiness',  # Merge 'joy' into 'happiness'
    'disgust': 'disgust',  # Merge 'disgustt' into 'disgust'
    'joy': 'happiness'
    # You can add more mappings if needed
}

dataset['emotion'] = dataset['emotion'].map(label_mapping).fillna(dataset['emotion'])

# Drop rows where emotion is NaN (if any)
dataset = dataset.dropna(subset=['emotion'])
```

Python

Figure 8: Combining labels

Moreover, any labels falling outside the predefined set of valid emotions (i.e., happiness, disgust, sadness, fear, anger, surprise) are replaced with NaN, further purifying the dataset. This is immediately followed by the removal of any rows where the 'emotion' column is NaN, ensuring the dataset's integrity by retaining only entries with clear, valid emotional tags.

Dataset Shuffling

Shuffling the dataset is a critical preprocessing step to ensure that when we create our validation set, it represents a broad spectrum of sentences across all emotions. If we didn't shuffle, the validation set could inadvertently become biased; for example, it might only contain the first 10,000 sentences labeled as happiness. There's a risk that these sentences are not representative of the diversity within the 'happiness' category or the dataset as a whole—they could, hypothetically, be the weakest examples of expressing happiness.

```python
# Shuffle the combined dataset
data = dataset.sample(frac=1).reset_index(drop=True)
```

Python

Figure 9: Dataset shuffling

To counteract this, shuffling randomizes the order of all sentences, ensuring that no single emotion dominates the beginning or end of the dataset. Consequently, when we split the dataset into training and validation sets, the validation set benefits from this randomness.

Data Inspection and Feature Preparation

The final steps in our preprocessing routine involve a comprehensive examination of the dataset, followed by feature extraction. We start by obtaining a glimpse of the dataset with **data.head()**, which displays the first few entries to provide an initial sense of the data structure and content. This snapshot showcases sentences alongside their associated emotion, indicating a successful merge of text with labels.

```
Preview of the dataset:
                                              sentence    emotion
0  i feel very honoured to be asked to write this...  happiness
1  i feel like it was a perfect time to offer som...  happiness
2  i didn t feel very humorous last week therefor...  happiness
3  i came to understand why i feel devoted to my ...  happiness
4  The tale of how a message in a bottle came to ...   surprise
```

Figure 10: First 5 rows of the dataset

Next, **data.describe()** generates summary statistics, giving us an overview of the dataset's composition. It reveals the count of total entries, the number of unique sentences, the most frequent emotion, and its occurrence rate, confirming the data's richness and diversity for all six emotions of interest.

```
Summary statistics of the dataset:
                                                sentence    emotion
count                                             616724     616724
unique                                            616724          6
top     i feel very honoured to be asked to write this...  happiness
freq                                                   1     207906
```

Figure 11: Summary of the dataset

We then ascertain the structure of the dataset using **data.info()**, which confirms that there are no null values in the primary columns of interest, 'sentence' and 'emotion', and that all data types are appropriate for the analysis, ensuring dataset integrity for the next stages.

```
Information about columns in the dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 616724 entries, 0 to 616723
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   sentence  616724 non-null  object
 1   emotion   616724 non-null  object
dtypes: object(2)
memory usage: 9.4+ MB
None
```

Figure 12: Technical info about the dataset

Lastly, we enumerate the unique labels in the 'emotion' column using **data['emotion'].unique()**, which confirms the presence of six distinct emotions—happiness, surprise, sadness, disgust, anger, and fear—thus validating that our dataset is ready for the next phase: model training and validation, where these emotions will be predicted from the sentences.

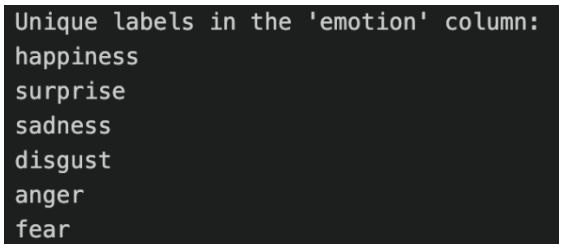


Figure 13: Unique emotion labels

Feature engineering

In the process of enhancing our model's performance, we experimented with integrating various features. Initially, we incorporated non-transformer-generated sentiment analysis, but this addition did not yield an improvement in accuracy. Further, we explored the potential influence of sentence-ending punctuation on model performance. However, due to data imbalance, this feature also did not lead to better results, particularly in improving the accuracy for the surprise class. Additionally, we extracted Part-of-Speech (POS) tags with the expectation of performance enhancement. Nevertheless, this feature did not contribute to any noticeable improvement, and therefore, we decided against incorporating it into our transformer models and rather focus on gathering more data.

Model selection and implementation

Model selection

In this project, our lecturer Myrthe Buckens created a Kaggle competition to serve as a benchmark for evaluating the effectiveness of the emotion classification models built by the different groups. The competition utilized a test set consisting of 1436 sentences, each labelled with one of Ekman's six emotions. These sentences were crafted by the students of the second year of this course, with each student contributing five sentences per emotion. To learn about different techniques for emotion classification as well as obtain the best performance possible we experimented with a wide range of different approaches: from traditional machine learning techniques to more complex Neural Network architectures and even Transformer models.

Following an evaluation across these model categories, it was clear that the pre-trained Transformer models, fine-tuned for our specific dataset (they were not merely applied in their pre-existing state), stood out. Building upon this foundation, we delve deeper into Transformer models, including BERT, XLNet, and DistilBERT and others. This exploration, enabled by the Kaggle competition's structure, underscored RoBERTa's superior efficiency, evidenced by its standout average F1-score on the competition's public leaderboard, thereby making it our chosen model for further development.

With our focus narrowed to the RoBERTa architecture, we fine-tuned and obtained our top 3 performing models:

Model name	Average F1-score
RoBERTa_V3_final 1	0.905
RoBERTa_V3_1	0.872
RoBERTa_V3_2	0.867

Table 2: Performance of the top 3 models

The training process for these models was precisely engineered, however, it involved employing common fundamental NLP techniques for all 3 models, such as tokenization—to segment text data into manageable units for the model, padding—to standardize sequence lengths for efficient batch processing, and attention masks—to direct the model's focus towards relevant data. These steps were fundamental in structuring raw text for model training.

In terms of hyperparameter tuning, notable distinctions among these models include the datasets they were fine-tuned on and training parameters. RoBERTa_V3_1 and RoBERTa_V3_2 were trained on the emotions_merged_4 dataset, while RoBERTa_V3_final 1 utilized the Dataset_final_V1, marking a significant enhancement in data quality. For RoBERTa_V3_final 1, we opted for a batch size of 32, adjusted from the 64 used in RoBERTa_V3_1 and RoBERTa_V3_2. This change was necessary due to computational constraints of our university server, which was at capacity and could not accommodate larger batch sizes—though it did not affect from its performance only its training speed.

Before delving into the implementation of these models on the 'Expeditie Robinson' dataset, it's essential to highlight the differences between them that contributed to their varied performances. RoBERTa_V3_2's shorter training duration of just 3 epochs was a limiting factor in its performance. In contrast, RoBERTa_V3_1 incorporated a dynamic learning rate and early stopping, enhancing its

effectiveness. RoBERTa_V3_final 1 built upon these improvements and was trained on a new dataset, further elevating its performance.

Model implementation

Our task transitioned towards operationalizing our best emotion classification models for direct application on the "Expeditie Robinson" series, Season 22 data. The goal was to provide to the client a Speech-to-Text pipeline capable of processing audio snippets to transcribe them and subsequently determine their underlying emotions. Initially, we were provided with the season's seventeen episodes in .mov format, alongside a CSV file delineating the start and end times of individual audio fragments within these episodes. This file also included emotion labels for certain fragments, assigned by human annotators. While the CSV contained additional columns referring to information such as ratings and actors, only those relevant to our task—fragment timings and emotional labels—were utilized.

The preparatory phase involved converting the video files into audio format, specifically to .mp3, chosen for its balance of compression efficiency and audio quality. Following this, the audio was segmented into fragments according to the 'Start time' and 'End time' specifications detailed in the CSV. For the transcription of these audio segments into text, we employed OpenAI's Whisper Python API, specifically, we used the base version of the model that strikes a good balance between transcription accuracy and processing speed. The transcribed text was then fed into our previously trained models to predict the emotional context, and further compared to the human annotations provided in the CSV. Notably, any labels not aligning with Paul Ekman's six basic emotions were either remapped for compatibility if possible or discarded, such as the 'hungry' label.

To streamline the user experience for our client, the entire workflow—from audio processing to emotion prediction—was encapsulated within a Python backend script. Furthermore, a Jupyter

notebook was developed, offering comprehensive guidance on leveraging the pipeline. This notebook features a short text with instructions on how to use the pipeline as well as a short code cell that imports functions from the backend script and integrates a Gradio interface. This user-friendly interface enables users to upload .mp3 files for real-time transcription of episode fragments, displaying the corresponding emotional prediction, alongside an indication of the prediction's accuracy relative to the human-labelled emotions as well as the option to show transcription of the audio itself.

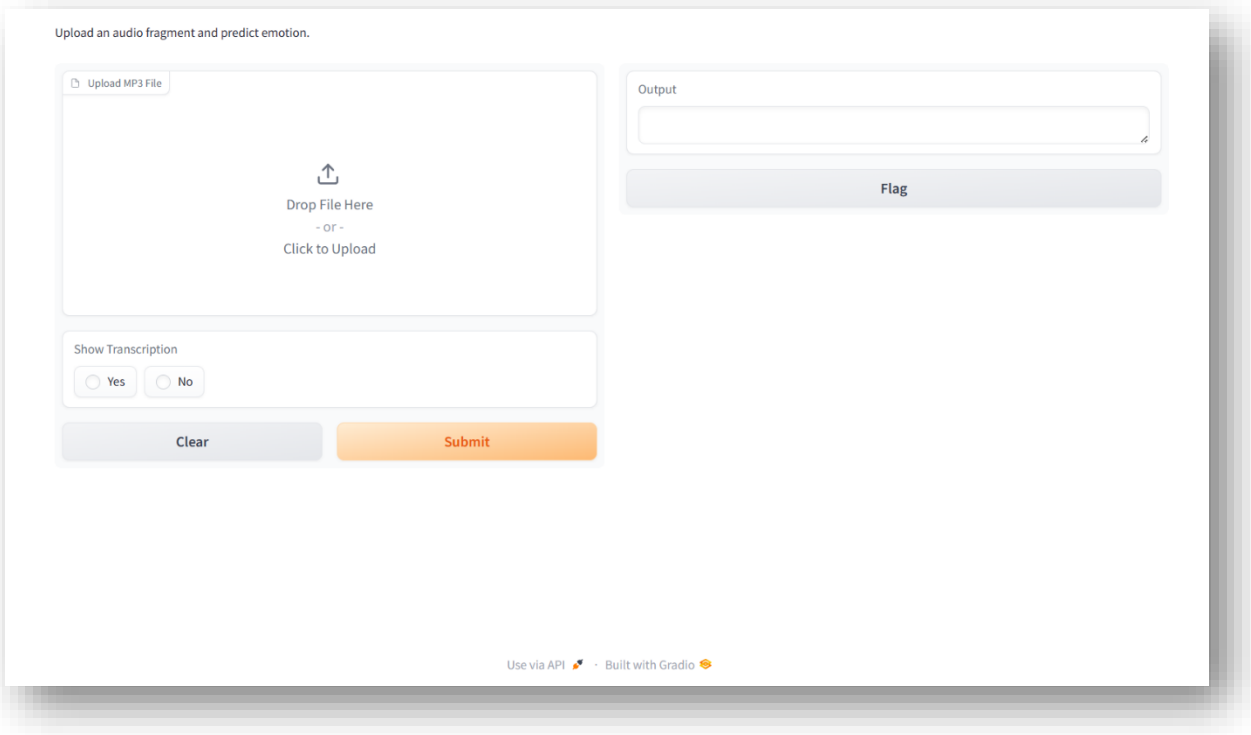


Figure 14 Speech-to-Text pipeline interface

Evaluation Metrics and Results

Reflecting on the model performance and summarizing the results, including accuracy, precision, recall, F1-score, and any other relevant metrics. Take a look at the predictions made by the model in an error analysis to see if there are explainable mistakes in the predictions. Discuss the

strengths and limitations of the chosen evaluation metrics/models. Is there a benefit of model A compared to model B, based on their performance on a specific class?

The "Task 15" notebook focuses on error analysis, presenting a detailed evaluation of the "Roberta_V3_1_task12" model performance through various metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of the model's ability to generalize and its performance across different classes. At the time starting with the error analysis " Roberta_V3_1_task12" was the best performing Kaggle model.

Model " Roberta_V3_1_task12" Performance Insights:

	precision	recall	f1-score
anger 0	0.97	0.98	0.98
disgust 1	0.94	0.98	0.96
fear 2	0.95	0.99	0.97
happiness 3	1.00	0.99	0.99
sadness 4	1.00	0.98	0.99
surprise 5	0.98	0.99	0.98
accuracy			0.99
macro avg	0.97	0.98	0.98
weighted avg	0.99	0.99	0.99

Figure 15 Classification Report for Roberta_V3_1_task12 model

The classification report reveals Model A's performance across different classes with the following metrics:

Precision: Ranges from 0.94 to 1.00, indicating a high level of accuracy in the model's positive predictions, with 'happiness' and 'sadness' classes achieving perfect precision.

Recall: Also high, between 0.98 and 0.99, showing the model's strength in identifying all relevant instances within the dataset, with particularly strong recall for 'fear' and 'surprise'.

F1-Score: The balance between precision and recall is maintained across classes, with scores from 0.96 to 0.99, reflecting the harmonic mean of precision and recall and suggesting a well-calibrated model, especially for 'happiness' and 'sadness'.

Accuracy: The model demonstrates an overall accuracy of 99% across the dataset, signifying exceptional generalization capability. This high level of accuracy indicates that the model is highly effective at making correct predictions across the range of classes presented.

In determining the trade-off between precision and recall for Banijay Benelux's emotion classification, the choice hinges on their objectives. If accurate emotion tagging is crucial to inform content strategy, precision is key, ensuring each identified emotion truly resonates with viewers. However, if capturing every instance of emotion is essential, particularly for subtle but pivotal moments in "Expeditie Robinson," then recall should be prioritized to avoid missing any emotional cues that could inform viewer engagement. However, the focus was on F1-score.

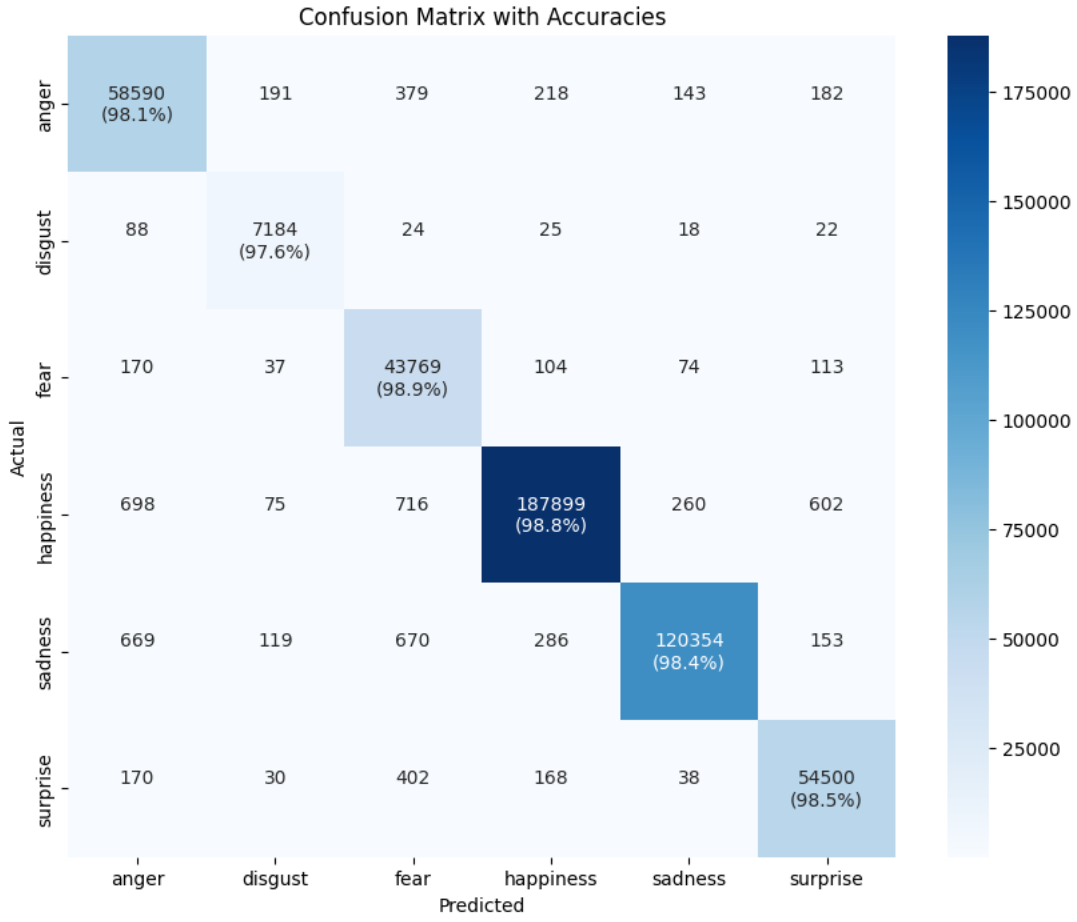


Figure 16 Confusion Matrix with Accuracies for Roberta_V3_1_task12 model

The confusion matrix for Roberta_V3_1_task12 presents a nuanced visualization of its performance across six emotional states: anger, disgust, fear, happiness, sadness, and surprise. The model exhibits high accuracy in classifying each emotion, with over 97.6% accuracy across all classes, indicating a strong ability to correctly identify emotions with minimal confusion. While the model shows slight tendencies to confuse anger with fear and disgust with anger, these instances are relatively low.

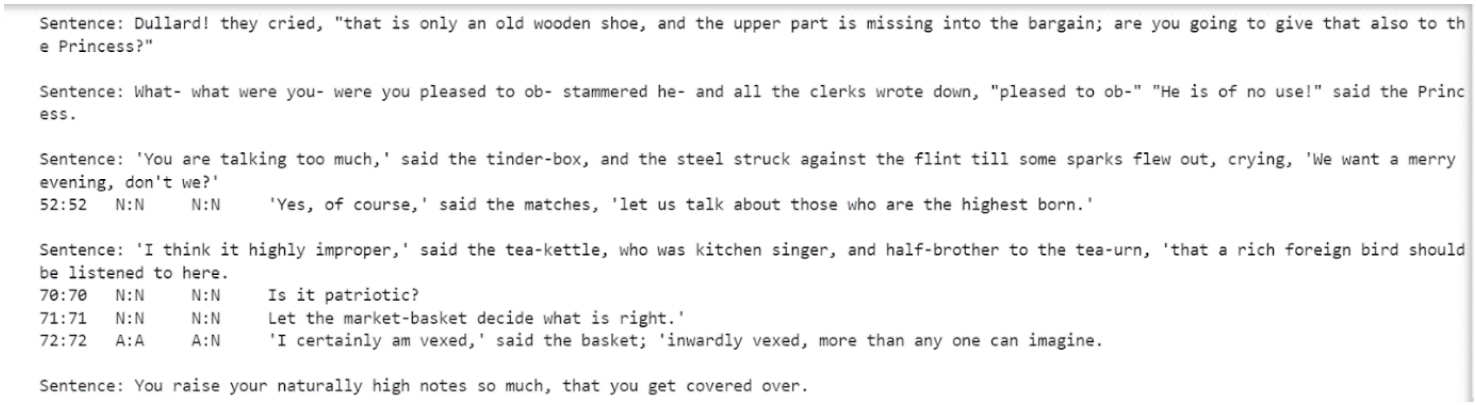


Figure 17 'Disgust' sentences predicted as 'anger'

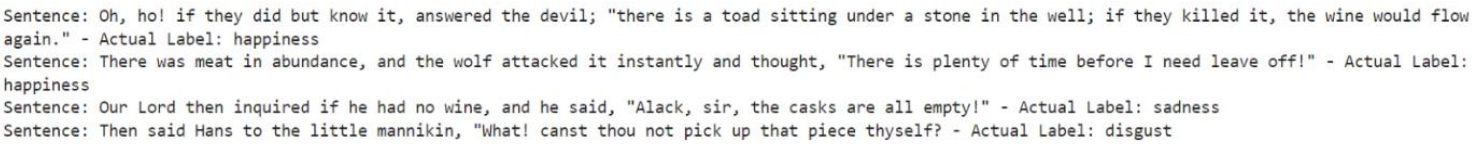


Figure 18 Sentences that contain '!' are misclassified as 'surprise'

After analyzing misclassified sentences, it becomes evident that the dataset itself can cause worse results. There are similarities between classes and many errors in recognizing context or subtle differences in meaning. So, the next decision was to make a new dataset with cleaner data and more like the Kaggle dataset.

The "Roberta_V3_1_task12" model has a public weighted F1-score of 0.872 and our new improved model “Roberta_V1_final” reached 0.905 score. That is because of a new dataset with added synthetic data.

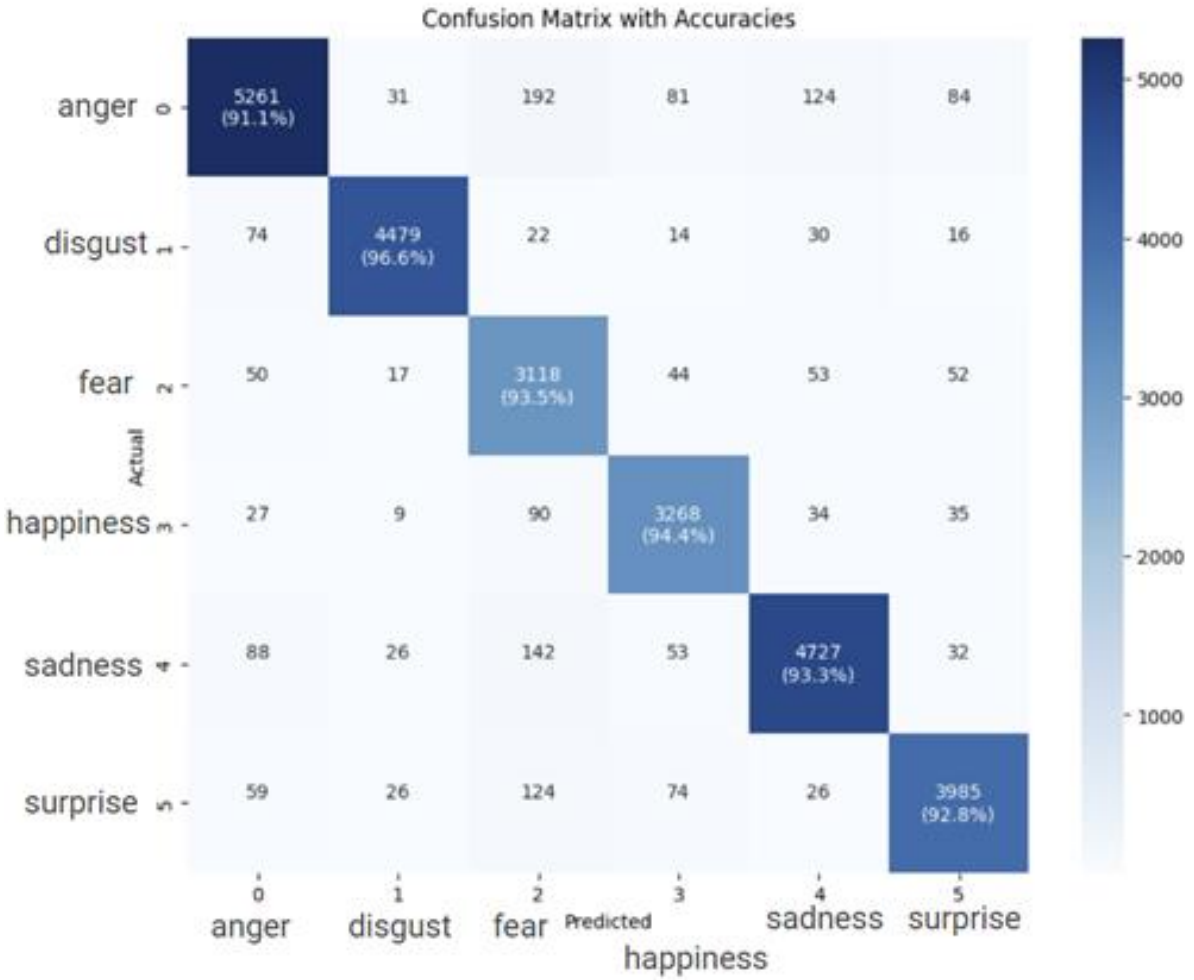


Figure 19 Confusion Matrix with Accuracies for Roberta_V1_final model

The new model, however, has slightly lower accuracy. It performs better on this specific Kaggle dataset but might struggle with other more general tasks, especially on long and speech data.

Discussion

Our project's goal was to create an emotion classification model with high accuracy in textual data. Throughout the experimentation phase, we explored different kind of features, preprocessing techniques, models, and data generation strategies. Key findings revealed that including sentiment analysis, punctuation recognition, and part-of-speech tagging did not significantly improve model performance. Preprocessing methods like expanding contractions, removing punctuation, and applying lemmatization similarly did not improve, and sometimes detracted from, model accuracy. Our exploration spanned various models, from RNNs, LSTMs, and GRUs to advanced transformers such as DistilBERT, XLNet, BERT, and ultimately RoBERTa, with the latter showing the most promise, particularly in its base variant(Roberta-base), achieving an F1 score of 0.872 on Kaggle competitions. Data generation techniques, including backtranslation and sentence generation through the ChatGPT API, played a pivotal role in creating a robust dataset that closely represented the diversity and complexity of the target domain. However, challenges such as the operating cost and supervision requirements of backtranslation, along with the limitations of using another model for sentence generation, were notable.

Our ensemble learning efforts, aimed at combining the strengths of various high-performing models, did not yield the anticipated improvement, suggesting the complexity of emotion classification might surpass the capability of straightforward ensemble approaches.

The adaptation of our model to the Expedite Robinson (ER22) dataset, which consisted of speech-based data, revealed a significant performance drop, attributed to differences in data nature, label mismatches, and the inherent challenges of working with speech-to-text translations.

Interpretation

The experiments highlight how complex it is to classify emotions in written text. Simply relying on surface-level features such as sentiment, punctuation, and parts of speech tagging isn't enough to capture the full emotional context of language. The different strategies used to prepare the data showed varying degrees of success, reinforcing that a model's performance is influenced by both the inherent characteristics of the data and the design of the model itself, rather than just the preprocessing techniques employed.

The success of RoBERTa over other models underscores the importance of choosing an architecture that can deeply understand the context and distinctions of language. The challenges faced in data generation highlight the critical role of diverse, high-quality datasets in training effective models, yet also point to the need for efficient, cost-effective methods to create such datasets.

The difficulties encountered with ensemble methods and the stark performance difference on the ER22 dataset illuminate the complex interplay between model training, the nature of the data, and the specificity of the task, especially when transferring learning from text to speech-derived text.

Implications

Our results matter for several reasons. Firstly, they contribute to the broader understanding of what strategies are effective in emotion classification within textual data, providing valuable insights for future research and practical applications. The effectiveness of advanced transformer models like RoBERTa signals a promising direction for similar tasks.

The exploration of data generation methods, despite their challenges, opens up discussions about innovative ways to augment training datasets, crucial for tasks lacking large, annotated datasets.

The performance disparity on the ER22 dataset raises important considerations for applying NLP models to speech-derived text, emphasizing the need for specialized training strategies and models that can better accommodate the nuances of spoken language.

Limitations

Our study has some limitations that we should acknowledge. We mainly relied on features found in text and preprocessing methods, which did not improve performance much. This suggests there might be better features or techniques we have not explored yet. Also, some data generation methods are really expensive and logistically challenging, which might make them hard to use in places with limited resources.

The ensemble learning strategy didn't perform as well as we hoped, suggesting we might need more advanced methods or a better understanding of how different models can work together.
One big limitation is how well our findings can be applied to other situations. When we tried applying our model to text derived from speech, its performance dropped significantly. This shows there's a gap we need to address if we want models to work well across different types of data.

Recommendations

Future studies could explore cross-validation more extensively to identify relevant data types and filter out irrelevant information, improving model robustness and applicability. Employing sentence generation methods from the outset could enhance the quality and diversity of training datasets. Additionally, exploring more nuanced ensemble learning strategies and expanding the dataset to

include a broader range of sentence types and emotional labels could refine model accuracy and generalizability.

For models intended to work with speech-derived text, training on datasets that more closely mimic the target domain, possibly including multimodal data that incorporates visual cues, could bridge the current performance gap. Finally, experimenting with advanced versions of transformer models or custom architectures tailored to the specific challenges of emotion classification in diverse data types may yield further improvements.

Conclusion

Our journey through the complexities of emotion classification in textual data highlighted the critical role of data quality, model selection, and the adaptability of models to diverse datasets. While we achieved notable success with RoBERTa in a text-based context, the transition to speech-derived text unveiled significant challenges, underscoring the multifaceted nature of NLP tasks and the necessity for models that can traverse the nuances of human language across different mediums. Our exploration opens avenues for further research, particularly in improving model performance on speech-derived texts and in the efficient generation of diverse, high-quality datasets. The undertaking, while challenging, sets a foundation for future advancements in emotion classification and NLP at large, encouraging a continued pursuit of models that can more accurately reflect and interpret the spectrum of human emotions.

References

OpenAI (2024). Chatgpt. Source text was written by the user and then rewritten. Prompt: Act as an NLP lecturer and rewrite, improving readibility and ensuring scientific rigor, 08-04-2024.

Breda University of Applied Sciences. (2024). Year 2 Block C. ADSAI.https://adsai.buas.nl/Year2/BlockC/

Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology, 17*(2), 124–129. https://doi.org/10.1037/h0030377