Breda University Of Applied Sciences

Technical Report

Team

Nick Belterman	Nick.Belterman@outlook.com
Cristian Stinga	223385@buas.nl
Amyr Lourensz	221264@buas.nl
Gin Li	221136@buas.nl

3-6-2024

Table of Contents

1 Introduction

Natural Language Processing (NLP) is an intersection of linguistics and artificial intelligence that began in the 1950. Nadkarni et al., 2011, in order to model the complexity of human language. The goal of this project is to identifying the kind of emotions found in textual information transcribed from episodes of the reality TV-show Expedition Robinson. This is one of the major difficulties in this field. Even with major progress, classifying emotions is still a challenging task. Not only because of the complex interactions between context, tone, and the perceived emotional content of a sentence, but also due to the subjective nature of emotions and the wide range of ways in which people express and experience them. Because cultural, social, and individual factors often influences emotions. This makes developing a model that effectively captures all emotional subtleties across many contexts and demographics challenging. Paul Ekman, a pioneer in the field of emotion research, published a groundbreaking paper in the early 1970s that proposed six basic emotions: happiness, sadness, fear, anger, surprise, and disgust. Ekman and Friesen, 1971. Numerous research continue to use these six emotions as their foundation. However synonyms and variations may occasionally be used. The following graph 1 shows the relationship between these primary emotions and the secondary and tertiary emotions that arise from them. Following a specified nomenclature is essential when preparing data for tasks involving the classification of emotions.

Figure 1: Relationship between core emotions

2 Dataset Description & Exploration

2.1 Dataset Description

In Table: 4 there is a short description of all the datasets that were used for training in this project. We will follow with a more in-depth description:

The Affect Dataset consists out of 185 fairytales and stories from three different authors, namely; Grimms, Anderson and Potter. The dataset consists of 15292 sentences, annotated by emotion and mood. This dataset was saved as three folders named by the respective author, before mentioned. Each folder had 3 separate folders containing files for each sentence: agree-sent(containing the sentence), emmood(containing the sentence ID, and 2 emotion labeles and mood labels) and Part-of-speech(containing the POS tags). Since we will extract Part-of-Speech (POS) later, we only processed the data in the emmood and agree-sent folders.

The Affective Text (Test Corpus of SemEval 2007) consists of 250 newspaper headlines, saved in two folders: train and trial. The text data is stored as a .xml with id and the sentences. The emotions are stored in a .gold file with a corresponding id, so they could be easily merged. This dataset is does not contain a lot of data. Additionally we decided not to use this dataset for training since most of the language used is not representative of the language used in the Dutch TV-show Expeditie Robinson

The CARER dataset has 416,809 English tweets, not manually annotated, saved as a pickle file, contains only two rows, sentence and emotion. This file already had the desired structure so there were no preprocessing steps left, only making sure that the emotions correspond to the 6 primary ones and dropping duplicates. We did not include this dataset in our training data because the dataset was not manually annotated, because we did not want to introduce additional error into our model.

The GoEmotions dataset is composed of 58K English Reddit Comments. It was stored in three different .csv files. The only problem being that the emotions were one-hot encoded, and not adhering to the 6 primary ones. This dataset had exceptional quality and was one of the main sources for our training data.

The Daily Dialogue is a manually labeled dataset consisting of 13,118 dialogues about daily life topics. Each sentence has additional information such as topic, intention of communication and emotion. The emotions were saved as scores raging from 0 to 100. We included this dataset in our training data because it is representative of the kind of language that is used in the TV-show Expeditie Robinson.

The MELD contains about 13,000 manually annotated utterances from 1,433 dialogues from the TV Show Friends. We also included a part of this dataset because the distribution of emotions balanced out the distrobution of emotions in our dataset.

Additionally we have received two CSV files and 17 episodes of the 22nd season of Expedition Robinson in a video format. One CSV file contains the names and roles of the actors that appear in the show and the other CSV file contains information about the episode and the segments with the corresponding emotions labeled by Banijay.

2.2 Dataset Exploration

In this Section we will discuss the Exploratory Data Analysis (EDA) we conducted. Before we started working on the Robinson dataset we had to train our models on a preset list of datasets that were provided to us, these datasets are:

- **GoEmotions Dataset**: Comprising a rich tapestry of 58K English Reddit comments, each entry is meticulously annotated with a spectrum of over 27 emotions. The challenge initially lay in mapping the one-hot encoded emotions to our specified categories. The access to this dataset was facilitated through the Google Cloud's `gsutil` command, a testament to the dataset's comprehensive nature in our training regime.

- **CARER Dataset**: With its vast trove of 416,809 tweets reflecting the public's sentiment, its structure, already cleansed, greatly reduced our prepossessing workload. The dataset's construction, though not manually annotated, was pivotal for aligning the sentiments to the six primary emotion categories identified for our study.

- **Daily Dialogue Dataset**: A dive into the everyday, this dataset features 13,118 dialogue instances, each rich in annotations detailing emotions, topics, and communicative intentions. Of particular interest was the numerical scoring of emotions, which we converted to fit our categorical model.

- **Affective Text and MELD Datasets**: The former presents a concise assortment of newspaper headlines from the SemEval 2007 task, whereas the latter deepens the exploration with dialogues from the famed TV show *Friends*. The datasets' structural annotations laid the groundwork for our models.

- **Affect Dataset**: Sourced from the enchanting narratives of Grimms, Andersen, and Potter, this dataset casts a magical spell with 15,292 sentences, each annotated for emotion and mood. The format necessitated an adroit extraction process to glean the relevant features.

The intricacies of dataset access varied; for instance, to retrieve the GoEmotions dataset, the `gsutil` command was employed, illustrating the seamless integration of Google Cloud's services. Conversely, the Daily

Dialogue dataset demanded meticulous attention due to its unique formatting. Sentences were demarcated by __eou__ tokens, with newline characters that remained concealed, necessitating careful parsing of the text files.

Figure 2 show the distribution of sentence lengths. There is a clear bias towards certain emotions like happiness or neutral emotions. These emotions are more common in the datasets' distribution. This was particularly clear in the Daily Dialogue and SMILE Twitter Emotion datasets as seen in figure: **??** and figure: 4. For this reason we decided not to continue with the SMILE Twitter Emotion after preprocessing, because the dataset's quality was similarly subpar. To balance the dataset, we choose to remove the extra rows of neutral for the Daily Dialogue. A higher representation of neutral and happiness in the distribution does not have to be a problem if the distribution is similar in the test set. Which we found was the case.

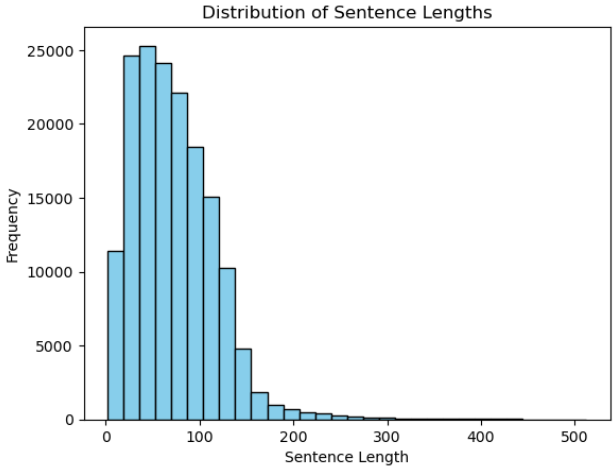


Figure 2: Distribution of sentence lengths

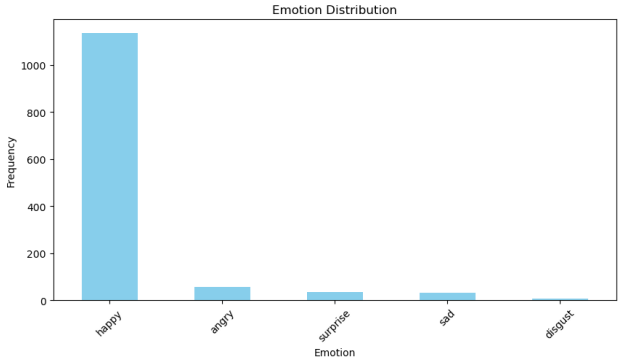


Figure 3: Distribution of emotions in SMILE Twitter Emotion

We also looked at the most frequently used words for each emotions, we visualized this using word clouds. We did this for every primary emotion and neutral. Something interesting to note is that there is a lot of overlap between words. It might proof useful to investigate how to decrease of get rid of this overlap. In the next paragraph we will examine the Expeditie Robinson data

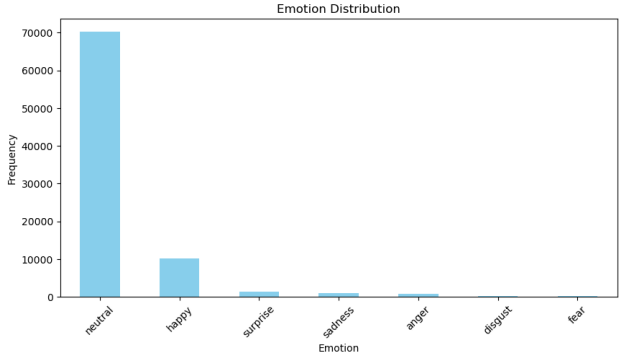
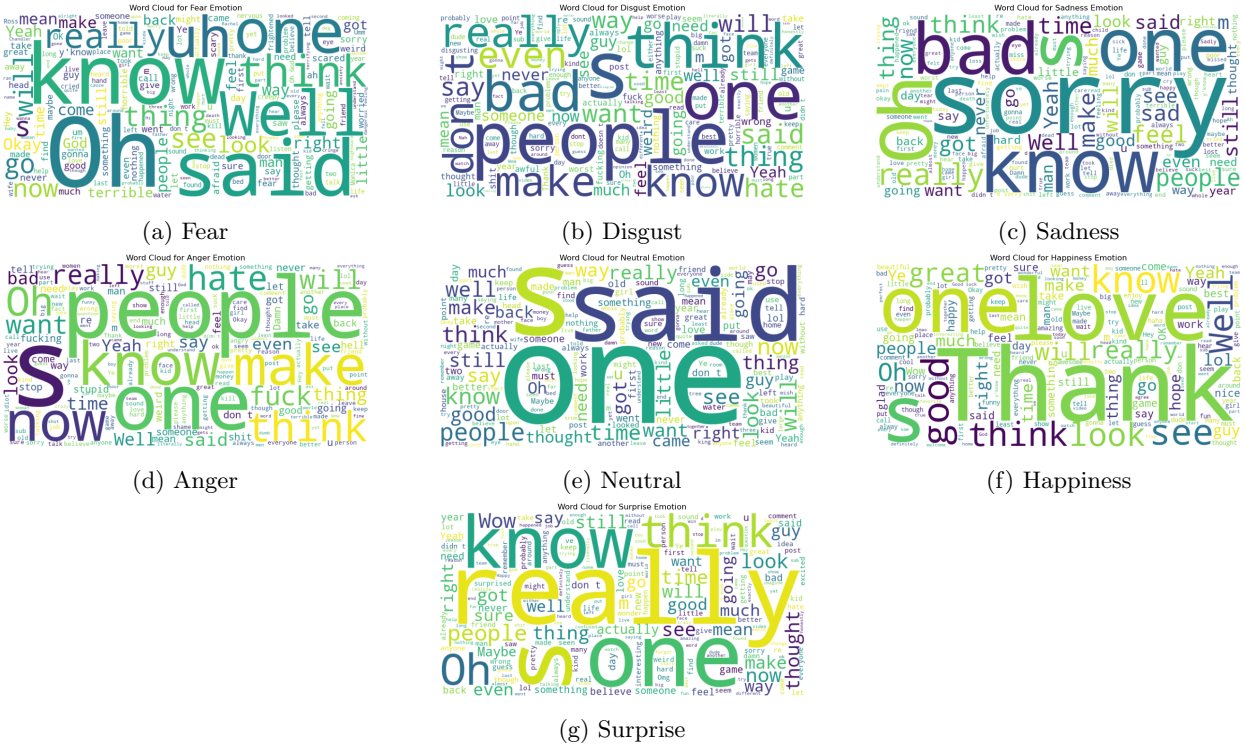


Figure 4: Distribution of emotions in Daily Dialogue



(a) Fear (b) Disgust (c) Sadness

(d) Anger (e) Neutral (f) Happiness

(g) Surprise

Figure 5: Word clouds of different emotions

The Expedition Robinson data that we are interested in is the information corresponding which each segment (e.g, "Joy, Caring, Gratitude, Optimism, Pride, Realization"). In figure: 6 the distribution of the emotions labeled by Banijay Benelux are shown. Interestingly the most common emotions excluding disappointment and annoyance are all positive.

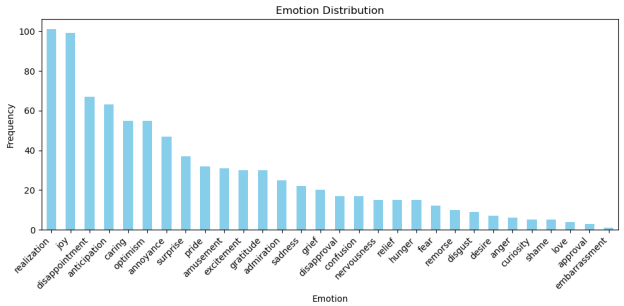


Figure 6: Distribution of emotions in expedition Robinson s22

For our purposes these emotions had to be mapped to the six primary emotions. This was done by assigning all emotions present in the Expedition Robinson data to the six primary emotions, then counting the values and creating a new column with the primary emotion that had the highest representation. This process is further explained in the preprocessing section of this report.

3 Preprocessing & Feature Engineering

3.1 Preprocessing

After the data collection took place and all the datasets were gathered, the next step would be data preprocessing. This involves going over each dataset and applying all sorts of transformations till all of the datasets are compatible and ready for merging into one final dataset. I will describe all the general preprocessing steps now:

Considering the datasets come from different sources, the very first step was to convert the data to a pandas DataFrame, and then to select all the relevant columns: the column containing the labeled text/sentence, and every column that was related to emotions annotations and labels. Thus we only work further with the most important components: the text and it's label, dropping every other columns not relevant for our use case. The next step, a crucial one, would be to make sure all the emotions are labeled to the six core emotions, as described by Ekman & Friesen, in 1971, plus the neutral label. All emotions replaced with their primary counterpart according to the emotion wheel presented in Figure 1. There were cases where there were multiple secondary emotions annotated to the same sentence. To fix this issue, we converted all the secondary emotions to one of the core emotion, and then counted the occurances of each primary emotion, followed by assigning the most occuring emotion as the correct label. In cases were two or more emotions shared the same count, or the label consisted of multiple core emotions, we just dropped the rows in order to assure the quality of the data. For example, since the GoEmotions dataset was one-hot encoded to multiple secondary and primary emotions, it required to be decoded, as well as to be mapped to the 6 primary emotions. In addition, emotion labels such as 'no-label', 'nocode' or 'not-relevant' were deemed as useless and were therefore removed.

At last, duplicate instances of pairs of sentence and the emotion label, as well as NaN values were checked in all datasets, and removed for quality purposes. As a finishing touch for the convenience of modelling, as well as an easy way of filtering the data, we added a new column with the name of the dataset from whom the data is coming from. This was done to easily remove or select certain desired datasets for training our models. After all the preprocessing steps were completed, all the individual dataframes were concatenated into the final file.

Certain datasets, such as the GoEmotions and Daily Dialogue datasets, required more complex preprocessing steps in order to format them into the desired form, a pandas DataFrame. For example, we divided the text into the necessary structure by going through the appropriate Daily Dialogue files iteratively. In addition, we had to convert the string labels to the integer labels. The Affective Text Dataset had scores for each emotion, stored in a different dataset. To ensure a strong presence of the emotion in the sentence, we changed all the scores lower than 20 to 0, and then selected the highest scoring emotion as the correct label.

And finally, the SMILE dataset consisted of annotated tweets, which required additional steps like removing hyperlinks, hashtags, emojis and mentions of another users.

3.2 Feature Extraction

This section outlines the processing steps undertaken to refine our dataset for subsequent analysis and model training. Leveraging a comprehensive processing script, we applied a series of transformations to the merged dataset's sentences, optimizing their compatibility with machine learning algorithms. We used

multiple processing scripts. The first script processes and combines all different datasets into a single CSV. The second processing script is used to extract additional features and transforms the data to be used for machine learning. The last script allows video files to be converted to an audio file and transcribed and translated from Dutch to English and be returned as a CSV. These scripts will be used to construct a pipeline, that allows a video to be used as input for our model to return the predicted emotions from that video. This will make it more user friendly for the client. Below are the key processing steps implemented:

- **Tokenization:**
  - **tokens_sentence:** Tokenizes the raw sentences.
  - **Stopword:** Extracts the stopwords from the sentences.
- **Preprocessed Tokens:**
  - **preprocessed_tokens_sentence:** Removes links, hashtags, and mentions starting with '@' and converts capital letters to lowercase.
- **Stemming and Lemmatization:**
  - **stemming_sentence:** Stems the sentences.
  - **stemming_preprocessed_tokens:** Applies stemming to the preprocessed tokens.
  - **lemma_sentence:** Lemmatizes the sentences.
  - **lemma_preprocessed_tokens:** Removes links, hashtags, and mentions starting with '@', converts capital letters to lowercase, and applies lemmatization to the preprocessed tokens.
- **Tokenization for Machine Understanding:**
  - **sequences_sentence:** Converts sentences into numerical sequences using a tokenizer.
  - **sequences_preprocessed_tokens:** Removes links, hashtags, and mentions starting with '@', converts capital letters to lowercase, and converts tokens into numerical sequences using a tokenizer.
  - **sequences_stemming_preprocessed_tokens:** Removes links, hashtags, and mentions starting with '@', converts capital letters to lowercase, applies stemming to the preprocessed tokens, and converts them into numerical sequences using a tokenizer.
  - **sequences_lemma_preprocessed_tokens:** Removes links, hashtags, and mentions starting with '@', converts capital letters to lowercase, applies lemmatization to the preprocessed tokens, and converts them into numerical sequences using a tokenizer.
- **Padding Sequences:**
  - **pad_sequences_sentence_95:** Pads the sequences of sentences to a length representing 95
  - **pad_sequences_token_95:** Pads the sequences of preprocessed tokens to a length representing 95
  - **pad_sequences_stem_95:** Pads the sequences of stemmed preprocessed tokens to a length representing 95

  - **pad_sequences_lemma_95:** Pads the sequences of lemmatized preprocessed tokens to a length representing 95

- **Named Entity Recognition (NER), POS Tagging, Chunking, Syntax Parsing, and Topic Extraction:**
  - **ner:** Extracts Named Entities (NER).
  - **pos_tags:** Extracts Part-of-Speech (POS) tags.
  - **chunks:** Extracts chunks.
  - **syntax_parsing:** Performs syntax parsing.
  - **topic_tfidf:** Extracts the most relevant topics using TF-IDF, providing scores.
  - **extracted_topic:** Matches topics extracted from 'topic_tfidf' to the current sentences and appends them.

We improved our dataset by using a standardized and structured format that is useful for effective analysis and model training by carrying out these processing processes. Consistent column names make it easier to find and understand information, and they are consistent with the README file of the project.

4 Model Selection & Evaluation

In this section we will discuss our model selection and evaluation. The chosen evaluation metrics to evaluate the performance of our model are; weighted F1 score, recall and precision. This will allow us to assess the effectiveness and robustness of our model before it is applied to the Robinson dataset. Our data will be split up into a training-, validation- and test- set. These before mentioned datasets were used to develop and validate the model. With a split of 65%, 17.5% and 17.5% respectively. Additionally we have a test data from Kaggle consisting out of 1436 unlabeled sentences to negate the risk of data leakage. These sentences were written by second-year students of the ADS&AI program. This ensures that our model is adequately trained and optimized for accurate predictions on unseen data. We fitted and tested a number of models on the dataset to figure out which one worked best for our use case. We started out with Multinomial Naive Bayes which is a linear probabilistic classifier. This model was fitted on matrix of token counts, the hyper-parameters were selected by a randomized grid-search. The weighted F1-score on the test set for this model was: 0.542.

Table 1: Parameter Grid for Naive Bayes Tuning

Parameter	**Values**
`vectorizer__ngram_range`	(1, 1), (1, 2), (1, 3), (1, 4), (1, 5)
`vectorizer__max_features`	10000, 15000, 20000, 25000, 30000
`Classifier`	MultinomialNB(), ComplementNB(), BernoulliNB()

We also experimented with a stacked bidirectional LSTM (BILSTM) model that included an attention layer. This model had multiple iterations and was trained on all the data that was described in 4. We investigated the impact of adding more bidirectional LSTM layers and discovered that doing so resulted in a decrease in the weighted F1-score. Moreover, we observed that raising the dropout did not raise the F1-score. We have observed repeatedly that when a model performs poorly on the test or validation data, it predicts the class that occurs the most frequently. The best performing stacked bidirectional LSTM model had a weighted F1-score of 0.34 on the test set.

Listing 1: Model Architecture Summary

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_35 (Embedding)    (None, 100, 2)            131676
 attention_21 (Attention)    (None, 100, 2)            102
 batch_normalization_47 (Ba  (None, 100, 2)            8
 tchNormalization)
 bidirectional_61 (Bidirect  (None, 100, 256)          134144
 ional)
 dropout_84 (Dropout)        (None, 100, 256)          0
 bidirectional_62 (Bidirect  (None, 100, 256)          394240
 ional)
 batch_normalization_48 (Ba  (None, 100, 256)          1024
 tchNormalization)
 flatten (Flatten)           (None, 25600)             0
 dropout_85 (Dropout)        (None, 25600)             0
 dense_52 (Dense)            (None, 32)                819232
 dropout_86 (Dropout)        (None, 32)                0
 dense_53 (Dense)            (None, 9)                 297
=================================================================
Total params: 1480723 (5.65 MB)
Trainable params: 1480207 (5.65 MB)
Non-trainable params: 516 (2.02 KB)
_________________________________________________________________
```

Additionally we fine-tuned transformer models, namely; bert-uncased-base and roberta-base. These models were initially fine-tuned on combinations of GoEmotions, Daily Dialogue and MELD. These transformer models we tested performed significantly better than all other models we tried. The results of these tests can be found in Table: 3. Based on the results of these test we started hyper-parameter tuning on roberta-base fitted on GoEmotions, Daily Dialogue, Affect & MELD.

4.1 Evaluation Metrics and Results

A problem we encountered during the testing of our Logistic Regression, Naive Bayes and BILSTM models was that there was a significant difference between the reported validation accuracy and the weighted F1-score on the Kaggle test set as seen in Table: 2. These models were all fitted on the entire dataset. There are a few possible reasons for this. The first and most obvious reason is that the models are over-fitting. This might be the case for the BILSTM, but it is less likely to be the case for the Logistic Regression model and the multi-nominal Naive Bayes model. These statistical models are relatively simple compared to deep-learning models, they also have less hyper-parameters to tune. That together with the fact that they are less depended on large datasets makes them less likely to over-fit. Another reason for this might be that there is a discrepancy between the distributions of emotion in the validation set and the Kaggle test set or that there are specific features in the validation set that warrant this response. In figure: 7 can been seen

Table 2: F1-scores & Accuracy for different models

Model	Accuracy	F1-score	Recall	Precision
Logistic Regression	0.543	0.432		
Multinominal Naive Bayes	0.852	0.542		
BILSTM	0.2883	0.327	0.406	0.430

that the BILSTM model was mostly not able to predict emotions besides happiness and neutral. We tried to fix this by creating a function that returns a Dataframe with balanced emotions. The result of this test can be seen in figure: 8. interestingly this model for the most part only predicted the emotion that had the lowest amount of data point in the original dataset.

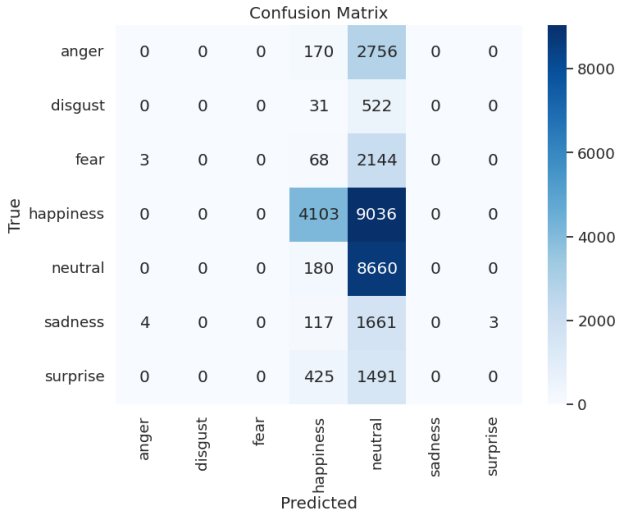


Figure 7: Confusion Matrix BILSTM

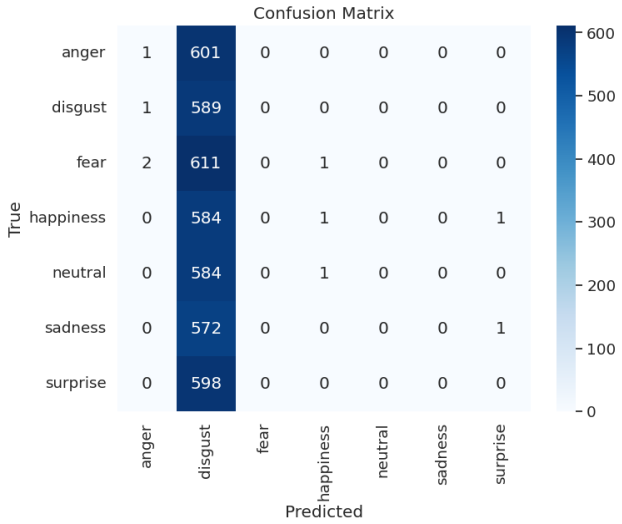


Figure 8: Confusion Matrix BILSTM with a balanced dataset

Table 3: F1-scores for different models on different datasets

Model	Data	F1-score
bert-uncased-base	GoEmotions & Daily Dialogue A	0.781
bert-uncased-base	GoEmotions, Daily Dialogue & MELD	0.697
roberta-base	GoEmotions & Daily Dialogue	0.777
roberta-base	GoEmotions, Daily Dialogue & MELD	0.786
roberta-base	GoEmotions, Daily Dialogue, Affect & MELD	0.805

5 Discussion

In this section, we discuss our main conclusions and the difficulties we ran into while working on the project. Our exploration of different datasets revealed that some datasets provided a rich and diverse distribution of emotions. While others had a strong bias towards emotions like happiness and neutral. This imbalance post a challenge for training the model since it would create a bias towards neutral and happiness. In order for a machine to process and interpret natural language effectively, we applied techniques such as transcribing, translation, tokenisation, padding and encoding. We applied multiple models, including Naive Bayes, Logistic Regression, Bidirectional LSTM, and transformers. However the results varied between the validation set and the Kaggle test set. Which indicates potential over-fitting or the models' sensitivity to specific characteristics of the validation data. However, the significant drop in performance on the Kaggle test set raises questions about the generalizability of the models. This discrepancy could be attributed to differences in the data distribution between the training set and the Kaggle test set. It could also be the case that the models' are in able to capture the subtleties and complexities inherent in emotional expressions across different datasets. There were a few consideration we had to make when choosing our data. We had to consider the resources we had available to us and how much data we would be using to train our models. Especially for fine-tuning the transformer models. So we had to narrow down the amount of data we were going to use. We wanted the dataset to be representative of the data in the expedition Robinson data. Based on the test we conducted and the previously mentioned considerations, we used GoEmotions, Daily Dialogue, Affect & MELD datasets. These dataset contains both monologues and dialogues which is also the case for the Expedition Robinson data.

5.1 The Pipeline

In this subsection we will describe the solution for our client, Banijay Benelux, and how it operates. To start things off, the pipeline works by receiving two parameters, the path to an audio file and a Structure Dataframe, which is optional. Then, it reads that provided autio file and extracts the text by transcribing using Open AI's Whisper medium model. This process takes approximately an hour. After saving the text, it is translated from Dutch to English using Google Translate's Translator feature. We used this translator instead of Whisper's built-in one because it performs slightly better. After translation, all sentences are tokenized and run through our RoBERTa model to generate emotion predictions at the sentence level. The predictions are firstly in logits format, but are turned into probabilities, and those probabilities into one singular emotion per sentence. The following step is optional and occurs when a Structure DataFrame is sent to the pipeline. The Structure DataFrame is expected to contain the episode's transcribed structure, broken down into fragments. The start and end times of each fragment are saved in the columns'start_time' and 'end_time'. If the Structure dataframe is provided, the pipeline will aggregate the sentences and predicted emotions to match the fragments based on the start and end times of each sentence as determined by the transcription. The pipeline will then count all of the emotions in a fragment and return the top three most frequently occurring emotions as emotion predictions. As the final step, the pipeline will return the text translated in English and the emotions predicted. The pipeline was tested on the 1st episode on Expeditie Robinson 22nd season, and it scored a weighted F1 score of 0.5752.

6 Conclusion

6.1 Recommendation

In this section we will discuss possible recommendations for our Banijay. These recommendations are geared towards improving data quality and model performance. In addition to recommendations for more advanced models.

1. **Expanding& Improving the dataset**: Increasing the variety and volume of expressions used for training can significantly increase the models' performance. This increase of variety and volume should not only encompass a wide range of sources but also various cultural contexts, age groups and socio-economic backgrounds. In order to maintain or increase performance across different demographics

2. **Update Preprocessing scripts**: Languages chance overtime, this happens especially quickly on social-media. That is why it is important to update the preprocessing scripts to accommodate new linguistic expressions and slang. This ensures the model stays relevant and effective over time. In addition updating the preprocessing scripts can improve the data quality. For example this can be done by adding checks to verify that each word in the vocabulary is an actual word and not spam. Capping the sentences to a specicfied lenghts could also improve the data quality

3. **Advanced Models**: Algorithms like Hidden Markov models (HMM) or Conditional Random Fields (CRF) can potentially capture sequential patterns in text data more effectively.

4. **Training Model on Dutch Text**: Since the Expedition Robinson data is in dutch it can at least reduce the error in translation and at best improve model performance.

5. **Annotate data on a lower level**: Annotating the emotions on a lower level then the segment level, can provide deeper insights into the nuances of emotional expression in the episodes. This will enable the model to improve its performance.

6. **Investigate to determine the most representative features for emotion expression to optimize training resources**: Investigating features that are representative of emotional expressions can reduce the amount of data that is directly used for training, this will result in faster training time and might result in better model performance

References

Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, *17*(2), 124.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544–551.

A Appendix A

Table 4: Emotion Datasets

Dataset	Description	Emotions	Source	Year
GoEmotions	Manually annotated dataset with 58K English Reddit comments.	27 + Neutral	**augenstein-etal-2020-goemotions**	2020
SMILE Twitter Emotion dataset	Composed of 3,085 tweets mentioning 13 Twitter handles associated with British museums.	5 + nocode and not-relevant	SMILE Twitter Emotion dataset	2016
Friends emotion-labeled dialogues	Composed of 12,606 manually annotated utterances from episodes of the TV Show Friends.	6 + Neutral	**ghosh-etal-2017-arena**	2017
MELD dataset	Contains about 13,000 utterances from 1,433 dialogues from the TV Show Friends.	6 + Neutral	**poria-etal-2019-meld**	2019
CARER dataset	Contains 416,809 English tweets collected through noisy labels and annotated via distant supervision as described in Go et al. (2009).	6	**qadir-etal-2018-carer**	2018
Affective Text	Consisting of 250 newspaper headlines, collected for the SemEval 2007 task.	6	**strapparava2007semeval**	2007
Daily Dialogue	Manually labeled dataset consisting of 13,118 dialogues about daily life topics.	5 + no emotion	**li-etal-2017-dailydialog**	2017
Affect data	Collected and annotated data from 185 fairytales, consisting of 15,292 sentences. It splits on emotions and mood.	6 + Neutral	**strapparava2005learning**	2005