Technical Report: Natural Language Processing for Emotions Classification

Fedor Chursin, Kornelia Flizik, Matey Nedyalkov, Panna Blanka Pfandler

1 Introduction & Scope of Work

In our eight-week journey through Natural Language Processing (NLP), we delved into an exploration with a particular focus on the task of emotion classification. Discerning emotions from text remains a challenging task for NLP, as context, tone, and interpretation all influence the perceived emotion from a sentence. Despite these challenges, it opens avenues for understanding human communication on a deeper level and can have significant implications across various domains such as customer feedback analysis, sentiment analysis in social media, mental health monitoring through text, and more.

Our client, Banijay Benelux, a prominent producer of television content, seeks to harness the power of NLP for content classification. By understanding the emotional dynamics within TV shows, Banijay aims to figure out viewer engagement and preferences, thus optimizing content creation and delivery. So, they approach us with this assignment.

1.1 Objective

The objective was to construct an effective pipeline that utilizes speech-to-text technology to annotate emotions expressed in the series Expedition Robinson. Leveraging the six core emotions (happiness, sadness, fear, anger, surprise, and disgust) proposed by Ekman and Friesen, we aimed to categorize emotional content within the series accurately and help Banijay to be able to make data-driven decisions. (Ekman and Friesen (1971))

2 Data Processing & Exploration - EDA

This section describes all the data and decisions considered for collecting and processing the data.

2.1 The Datasets

Firstly, the data used for training the model will be detailed. The final dataset contains 5 different sources displayed in Table 1: GoEmotions, Smile Twitter emotion, Friends emotion-labelled dialogues, MELD dataset, and CARER dataset. While these datasets have various columns, the similarity among them are sentences and corresponding emotions, represented as strings. Only these two columns were extracted from all datasets, as they were the most relevant to the project. All the datasets were merged.

Table 1. Overview of Emotion Analysis Datasets

Dataset name	Source	Data size	Data Quality
GoEmotions	Reddit comments	58,000 rows, 27 emotions	No missing values, outliers detected
Smile Twitter emotion	Twitter mentions	3,085 rows, 5+ emotions	Spelling errors, different languages
Friends Emotion	“Friends” TV show	12,606 rows, 6+ emotions	Consistency across annotators
MELD Dataset	“Friends” TV show	13,000 rows, 6+ emotions	Clean, labelled data
Carer Dataset	English tweets	416,806 rows, 6+ emotions	Human and machine annotations

The final dataset reached over 600,000 entries. The distribution of the data is demonstrated with a bar chart in Figure 1 It can be noticed there is a big imbalance in the classes. With a percentage of roughly 48.6% of the dataset, happiness is found to be the most common emotion. Anger and Sadness come next, with respective percentages of approximately 23.9% and 15.4%. The emotions of surprise and fear each account for almost 10% of the dataset, whereas disgust makes for the least amount of data with only 5,314 instances or 0.9%.

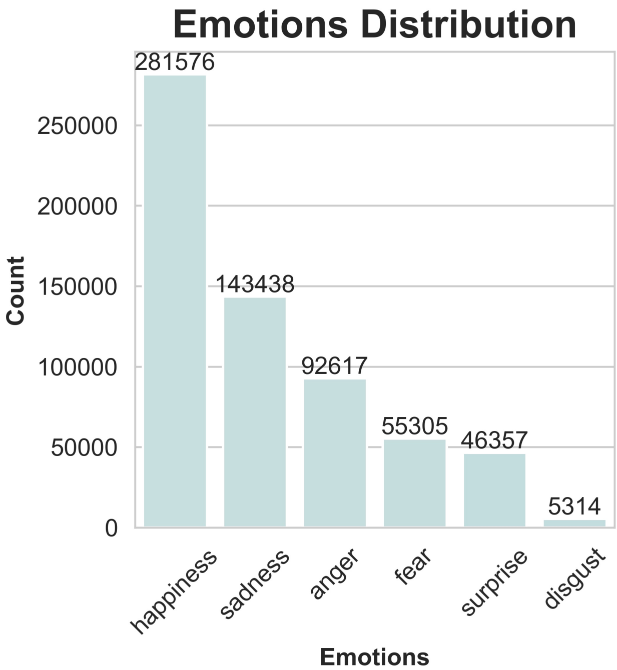

Figure 1. Emotions Distribution

The imbalance in class distribution can lead to risks in training a model. Specifically, there is a risk of bias towards the category with the highest number of instances, which, in our case, is happiness. This can possibly lead to poor performance of the model since it will not learn effectively from the underrepresented classes.

Another dataset used for the project was designed for testing purposes. This dataset was collectively obtained, with each student from the course contributing 5 sentences for each of the 6 emotions, resulting in a total of 30 sentences per person.

The last dataset incorporated in the project consists of episodes from the TV show “Expeditie Robinson”. In this show, participants compete for the Robinson title on a desert island. The dataset includes 16 episodes, each approximately an hour long.

2.2 Preprocessing

For the training set, the emotion column of each dataset was reviewed as some of them contained more than 6 classes (For instance, GoEmotions dataset). Since the project requires only 6 emotions - happiness, sadness, anger, fear, surprise and fear - some classes were either removed or reassigned to the closest emotion based on an emotion wheel. Ekman and Friesen (1971) This narrowed the scope to six emotions.

Another crucial step in the preprocessing is the tokenization. This involves splitting the text in smaller pieces, which are called tokens. This makes it easier for the model to understand and process the data. This process was exclusively implemented on the sentences since the emotion columns consist of one word per row.

Lastly, we encoded the labels. The Label Encoder function from SciPy was used, which provided each emotion with a specific number for representation. For instance, happiness is represented by 0, sadness by 1, and so forth. This encoding process ensures that the model can better grasp the categorical features.

The training data only went through tokenization, preparing it for input into the model for predicting labels.

The “Expeditie Robinson” data was passed to the speech-to-text model Whisper, so the audio could be transcribed. Afterward, the transcriptions were translated to English from Dutch. Finally, similar to the other datasets, the text was tokenized to make it ready to be used as input for the model.

2.3 Feature Engineering

This part will dive deeper into all the features that were extracted to improve the model's performance. Three main features were considered:

1. Part-of-speech tagging
2. Sentiment analysis
3. Word embeddings

The first feature to be discussed is the part of speech. In this case, the data is analysed to determine the part of speech for each token as they get labelled for their corresponding type – nouns, verbs, adjectives, etc. This helps the model with the understanding of the grammatical structure of the sentence for emotion classification. Also, it can be used by providing contextual information, and that way it can be connected to a particular sentiment. For instance, if a verb expressing happiness is identified, this sentence can be classified as happiness.

Another feature that was considered is sentiment analysis. This feature can be valuable, because it attempts to detect the sentiment expressed in the sentences, categorizing them into three types: positive, negative, and neutral. Unfortunately, it was observed that the accuracy of sentiment analysis was not satisfying as some of the sentences were misclassified. Therefore, this feature was not included in the models.

Lastly, word embedding was implemented. This is an NLP technique, which represents tokens in a continuous vector space as it tries to capture the relationship between words by capturing semantic similarity. This technique allows the model to comprehend more the contextual relationship between words or phrases. To complete this task, we utilised Word2Vec with a pre-trained model. Chen et al. (2013) Additionally, a custom word embedding was implemented using the training dataset.

However, not all extracted features were used during different training iterations. For instance, sentiment analysis was initially used for logistic regression, but since better models were identified, where this feature was not useful, it was excluded from the final training data. Similarly, word embeddings were used in the LSTM model, which was later replaced with transformers, that do not require any additional features.

Every step that was covered helped to better choose the decisions during the modelling phase and improved the performance of the final model. (Wambsganss et al. (2021))

3 Model Selection & Implementation

3.1 Model Selection Process

3.1.1 ML Models

In our quest to accurately classify six core emotions using NLP techniques, we carefully evaluated various traditional machine learning models tailored for text classification. We considered Logistic Regression (LR), Support Vector Machine (SVM), and Multinomial Naive Bayes (MNB).

To prepare our text data for classification, we utilized two common text vectorization techniques: TF-IDF and CountVectorizer. These techniques transform text into numerical representations, facilitating the models' ability to extract insights from the data. We then fed the vectorized text and previously identified features into models

3.1.2 Neural Networks

In our exploration of deep learning approaches, we evaluated two models: Multi-Layer Perception (MLP) and Long Short-Term Memory (LSTM). These models stood out for their capacity to comprehend sequential dependencies inherent in textual data. MLP, known for its ability to handle non-linear relationships, provided an additional perspective in our investigation of emotion classification techniques. LSTM, a variant of Recurrent Neural Networks (RNNs), is particularly adept at processing sequential data by preserving information over time. (Vaswani et al. (2017)) By training these models on our emotion classification dataset, we aimed to harness its ability to grasp contextual cues and subtle nuances in the text, thereby improving classification accuracy.

3.1.3 Transformers

Transformers are a type of deep learning architecture, that have revolutionized the field of natural language processing (NLP) in recent years. (Hoque (2023)) The transformer architecture is composed of an encoder and a decoder, each of which is made up of multiple layers of self-attention and feedforward neural networks. The self-attention mechanism is the heart of the transformer, allowing the model to weigh the importance of different words in a sentence based on their affinity with each other. This is similar to how a human might read a sentence, focusing on the most relevant parts of the text rather than reading it linearly from beginning to end. In addition to self-attention, the transformer also introduces positional bias, which allows the model to keep track of the relative positions of words in a sentence. This is important because the order of words in a sentence can significantly impact its meaning.

We employed three cutting-edge transformer models, namely Roberta, Bert, and Albert, sourced from the Hugging Face transformers library. Hugging Face serves as a collaboration platform for the machine learning community, with its Hub acting as a central repository where researchers and practitioners can share, explore, discover, and experiment with open-source machine learning models.

3.1.4 Results

In our pursuit of model evaluation, we prioritized the F1 score as a pivotal metric for assessing predictive performance. This metric measures the balance between precision and recall and is crucial for assessing classification performance accurately.

Our experimentation revealed that Roberta Model transformer with a sequence classification head on top emerged as the top-performing model for emotion classification from the text. It builds on Bert and modifies key hyperparameters, removing the next-sentence pre-training objective and training with much larger mini-batches and learning rates. Roberta is pre-trained on a combination of five massive datasets resulting in 160 GB of text data. In comparison, BERT large is trained only on 13 GB of data. (Liu et al. (2019))

3.2 Adaptations & Modifications Made to the Models

To prepare our text data for input into the Roberta transformer model, we employed the Roberta tokenizer and tokenized our text data for both training and validation purposes.

The model architecture was designed to predict 6 distinct emotion labels, namely happiness, sadness, anger, fear, surprise, and disgust, using SoftMax predictions. This configuration enabled the model to output probabilities for each of the 6 emotion labels, facilitating multi-class classification. The number of trainable parameters is more than 124 million.

3.3 Hyperparameter Tuning

The Adam optimizer was employed with a learning rate of 5e-5, a key hyperparameter that determines the step size during gradient descent. Additionally, the sparse categorical cross-entropy loss function was utilized to compute the loss, ensuring efficient optimization of the model parameters.

3.4 Model Training Process

The Roberta model was trained over 4 epochs with a batch size of 128 using the training dataset, with performance evaluated on a validation set after each epoch. To prevent overfitting, an early stopping technique monitoring validation loss value was applied. Figure 2 represents the learning curves of our best model

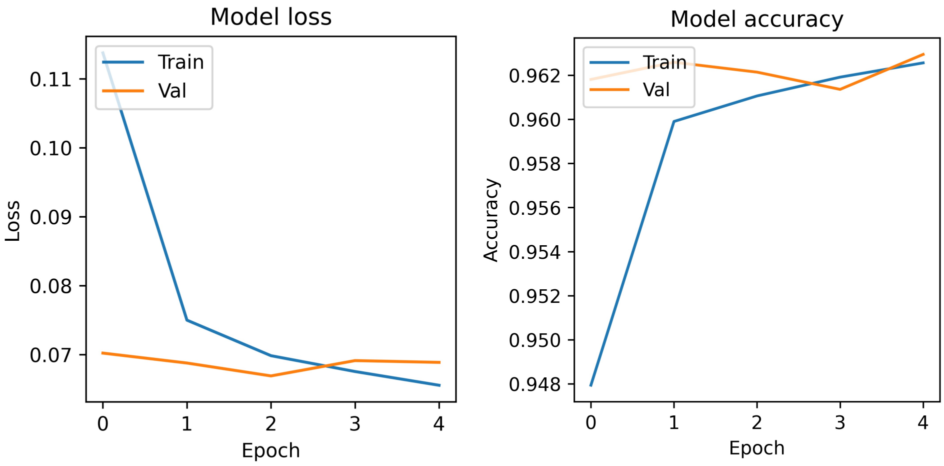

Figure 2. Learning Curves

3.5 Speech-to-Text Model

In the subsequent stages of our pipeline, we integrated a pre-trained model for speech processing. Specifically, we opted for Whisper, an open-source model developed by OpenAI in 2022. This model is a transformer model that processes audio chunks into log-Mel spectrograms. Leveraging Whisper, we transcribed and translated data from the Expedition Robinson series into English, facilitating further analysis and processing within our framework. (Radford et al. (2023))

4 Evaluation, Error Analysis & Results

This part of the report describes the results achieved during the project and the performance of the final pipeline by quantifying the performances of the trained models using metrics chosen for evaluation.

4.1 Model Performance Evaluation

For the project, we tried deploying a variety of models, ranging from logistic regression and naïve bayes to pre-trained transformer-based models. When choosing the metrics for the evaluation of the model performance it is vital to consider their advantages and disadvantages. The chosen metrics together with their strengths and weaknesses are listed below:

- Loss:
  - Advantages
    - Directly optimised during the training process, providing an improvement objective.
  - Disadvantages
    - Hard to interpret in terms of how good or bad the model performs in general.
- Accuracy:
  - Advantages
    - Easy to understand.
    - Offers a quick assessment of overall performance.
  - Disadvantages
    - Vulnerable to imbalanced and biased data.
- F1 Score:
  - Advantages
    - Combines precision and recall, making it robust to biased and unbalanced datasets.
    - Provides a better overview of false positives and false negatives predictions of the model making it easier to interpret the results.

- Disadvantages
  - Equal importance for precision and recall can lead to misleading or unrelated results

We believe that such a combination makes our evaluation process robust and easy to interpret.

While loss and accuracy metrics were measured while training the model based on the training data, the F1 score was calculated on the unseen data that was provided for the group challenge.

Table 2 shows the best-achieved results for each type of model used.

Table 2. Model Performance Summary

Model Used	Loss	Accuracy	F1 Score
Naive Bayes	0.62	80%	0.565
Logistic Regression	0.8	65%	0.521
Multilayer Perceptron	1.09	76%	0.459
LSTM	1.07	63%	0.573
roBERTa	0.34	87%	0.750

The transformer-based roBERTa model is the top performer on the emotion categorisation task. After four training epochs, it obtained a loss of 0.34, an accuracy of 87%, and an F1-score of 0.73. This result can easily be explained, in comparison to other models, roBERTa has a state-of-the-art performance across a range of NLP tasks, as well as benefits from pre-trained knowledge acquired on a large corpus, capturing a deeper understanding of the language. This model is considered a ”black box” which is why it’s difficult to understand the inner processes and logic behind its decision-making, especially when compared to more transparent algorithms like logistic regression or Naive Bayes. Training and fine-tuning roBERTa demands significant computational power making it challenging to tweak the model for optimal performance.

4.2 Error Analysis and Explainable Mistakes

When evaluating a trained model it is vital to understand why certain results are achieved and what can be done to improve the performance that is why we performed an error analysis on our model several times and iterated through the results.

To better grasp the model’s predictions, we created a confusion matrix Figure 3 that visualises the ratios of all the predictions among all 6 emotions. From the matrix, we concluded that the model struggled with the classification of surprise, as well as confusing anger and disgust. The cause of this is the data set we used for the training, the distribution of emotions in this set is imbalanced, and the classes that model have the most issues are underrepresented. Biased data led to a biased model that did not learn enough about such emotions as disgust and surprise and did not generalise well as a result.

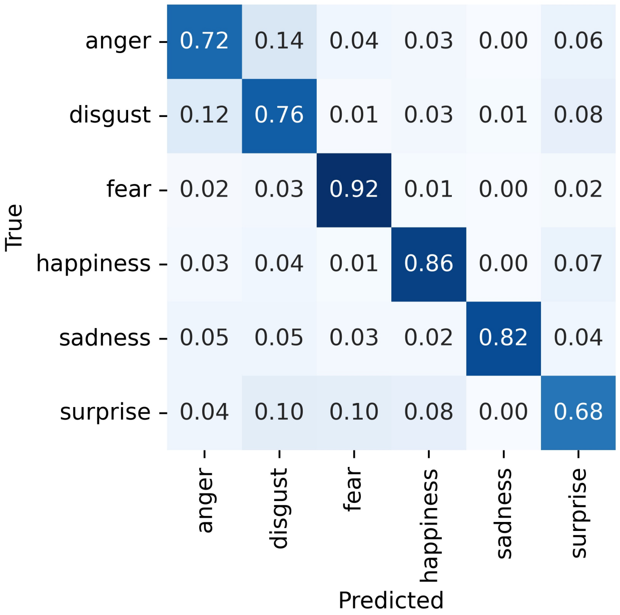

Figure 3. Predictions Confusion Matrix

Another crucial factor is the quality of the training data. Our initial dataset was heavily loaded with slang, primarily sourced from platforms like Twitter and Reddit. This dataset was wicked by grammatical errors, slang terms, and extraneous information, including tags, links, and mentions, all of which were eliminated in the preprocessing phase. However, the removal of these elements introduced gaps in the data, as they often contained context-related information. Consequently, this forced the model to guess in the lack of full context, a phenomenon known as ”hallucination,” which resulted in lower performance. (Mittal et al. (2024))

To tackle these two problems we decided to alter the data we used and the training process itself. We removed all the data that came from social media and populated the dataset with more sentences containing traditional and formal language. For the training processes, we defined weights for each class, balanced them, and passed them to the model. While the training model gave higher priority to the data with higher weight. We assumed this approach would improve the data quality and mitigate the bias in the dataset leading to a desired performance.

Table 3 below represents the results before and after we changed the data and added the weights. We can see that the accuracy improved indeed but the F1 Score got lower.

Table 3. Model Performance Before and After Alterations

	Accuracy	F1 Score
Before Alterations	87%	0.750
After Alterations	96%	0.580

To understand what is happening we created one more confusion matrix shown in Figure 4.

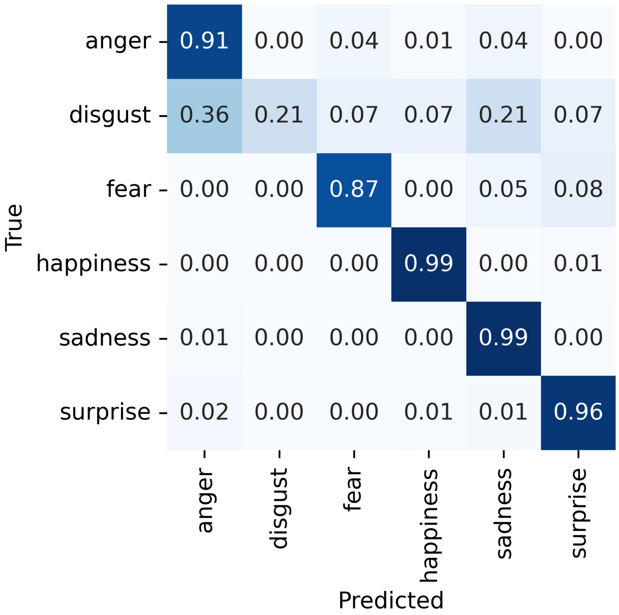

Figure 4. Predictions Confusion Matrix

From the matrix, we can conclude the model almost perfectly identifies 5 emotions and has no understanding of what disgust is. This is explained by the data once again, after removing all tweets and Reddit posts the imbalance increased drastically and the weighted approach was not able to eliminate this imbalance.

From these two iterations, we understood that to improve the performance it is more valuable to focus on the quality and balance of the data that is used, as it is the most important aspect of the whole training process. Also, Roberta - the transformer that we used is already considered a state-of-the-art model, and fine-tuning it would not improve the performance significantly.

It is worth mentioning that loss and accuracy were evaluated on the training set and F1 Score was calculated based on unseen data from the Kaggle competition. These two datasets have two major differences while the training set is real-world data, the kaggle one was created synthetically by students just entering several sentences for each emotion. For a more robust evaluation, it would be necessary to work with unseen real-world data as well.

4.3 Summary of Results

The outcome of the project was the pipeline that we provided to the client that can be used for emotion classification in the fragment of any given show episode.

The solution that we provided for the given problem can be broken into several parts:

- Data Preprocessing
- Speech To Text
- Emotion Classification

To better understand the results and limitations of the provided solution it is crucial to understand every part of it.

4.3.1 Data Preprocessing

The data was provided to us in the form of several recorded episodes of the show. The language of the recordings is Dutch. The speech-to-text model that we used has two major requirements:

- The length of the file should not exceed 10 minutes
- It should be an audio file

To satisfy these requirements we applied several preprocessing steps to the provided data. We broke each recording into several chunks that were shorter than 10 minutes and extracted audio.

4.3.2 Speech To Text

Whisper, an advanced model created by OpenAI, is applied for transcription in the proposed solution. This model was chosen due to its ease of use, high performance, and built-in translation feature. To utilise this model, an audio segment and the preferred language need to be inputted, resulting in a translated transcription of the audio file.

4.3.3 Emotion Classification

The best-performing model - Roberta-based transformer was incorporated into the final solution to classify emotion in the given segment. The model expects text as input and outputs the encoded emotion label that later is decoded and provided to the user. The model can detect any of the following six emotions: happiness, sadness, anger, surprise, fear, disgust.

By combining all three parts we get a fully working pipeline. However, it's crucial to acknowledge that this pipeline has certain technical constraints. These limitations come from various sources; some are related to the emotion classification model previously discussed, while others are associated with the preprocessing and speech-to-text processes.

An episode must be divided into multiple parts during preprocessing to be loaded into the speech-to-text model. Logically full sentences can be divided into distinct parts during this process, changing the sentiment and meaning that is expressed. In rare cases, it might influence the output of the pipeline and mislead the user. Although this method is outside the scope of this project, we recommend using a more sophisticated methodology to preprocess audio recordings to find the most ideal break points that result in the least amount of context loss.

An additional constraint of the method offered is the language prerequisite for the emotion classification model. The model was trained on the English language corpus and is not supposed to work with Dutch, that is why the transcriptions are translated to English and passed to the model after. Translation results in a substantial loss of context and reduces the precision of the classification. A new model should be trained on the Dutch language corpus to overcome this constraint, this will lead to a significant leap in performance.

Despite the fact of the limitations present in the solution it can be deployed by Banijay to access the performance on a larger scale and address the limitation if needed.

5 Discussion & Conclusion

Despite our best efforts, which involved experimenting with different datasets, features as input, models, and hyperparameters, we were only able to achieve incremental improvements. Ultimately, our model's performance did not meet our initial expectations. However, through iterative processes and experimentation, we gained valuable insights into potential areas for improvement.

During our analysis, one key limitation became apparent: the quality and consistency of the labelled training data. The effectiveness of any machine learning model heavily relies on the quality of the training data. In our case, due to time constraints, we were unable to ensure that the training set was adequately labelled, and representative of the emotional diversity present in the test set. This mismatch between the training and test data likely contributed to the suboptimal performance of our model.

Furthermore, due to time and capacity constraints, we could not do as many experiments as we aimed to. Given the limited resources, we were unable to thoroughly explore various model architectures, hyperparameters, and feature representations to optimize performance.

Additionally, we observed a significant disparity in language use between the training and test datasets, which hindered the model's ability to generalize effectively. The language used in the training set differed substantially from that in the test set, posing a challenge for the model to accurately capture emotional nuances in unseen data.

Moving forward and based on the identified limitations, we recommend several strategies to enhance the performance of our emotion classification pipeline:

1. Improved Data Labelling: It is crucial to invest more time and effort into meticulously labelling the training data, ensuring it accurately reflects the emotional nuances present in the test set. This may involve employing domain experts or utilizing advanced annotation techniques to achieve higher-quality labelled data.

2. Model Tuning and Optimization: Continuously fine-tuning and optimizing the model architecture, hyperparameters, and feature representations can help enhance the model's ability to capture emotions present in the text data.

3. Domain-Specific Adaptation: Fine-tune the model on domain-specific data, such as transcripts from similar TV shows or Dutch language corpora. This can help the model better understand the specific language and emotional expressions relevant to the target domain.

In conclusion, while our journey through NLP and emotion classification has been both enlightening and challenging, there is still much room for improvement. By addressing the identified limitations and implementing the recommended strategies, we remain optimistic about the potential of NLP in unravelling the complexities of human emotions and enhancing various applications across domains.

References

Chen, Y., Perozzi, B., Al-Rfou, R., and Skiena, S. (2013). The expressive power of word embeddings. *arXiv preprint arXiv:1301.3226*.

Ekman, P. and Friesen, W. V. (1971). Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of Personality and Social Psychology*, 17(2):124–129.

Hoque, M. (2023). A comprehensive overview of transformer-based models: encoders, decoders, and more.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., and ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Mittal, A., Murthy, R., Kumar, V., and Bhat, R. (2024). Towards understanding and mitigating the hallucinations in nlp and speech. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, pages 489–492.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., and ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wambsganss, T., Engel, C., and Fromm, H. (2021). Improving explainability and accuracy through feature engineering: A taxonomy of features in nlp-based machine learning. In *ICIS*.