How do you feel, my dear

Fabio Brambilla[978983]

Università Degli Studi di Milano

Abstract. The project aims to study the emotional profiles of fictional scripts in movies and TV series, examining how they change over time and how they are affected by character relationships. We will use a categorical representation of emotions.

Emotion detection in text, also known as sentiment analysis or opinion mining, uses natural language processing (NLP) to identify and extract subjective information from text. The goal is to classify the emotional content of a text or speech, such as determining the emotion contained in the text.

There are several approaches to emotion detection in text, including rule-based, lexicon-based, and machine learning-based methods. Rule-based approaches use manually defined rules or patterns to identify specific emotions, while lexicon-based approaches use a pre-defined list of words or phrases associated with specific emotions. Machine learning-based approaches use algorithms to classify text based on a training dataset of annotated text but require a larger amount of data for training.

Emotion detection in text has made significant progress in recent years, but it is still a challenging task due to the subjectivity of emotions and the complexity of natural language. There are also ethical considerations, such as the potential for bias in the training data or the use of emotion detection for nefarious purposes.

Keywords: Emotion · Outcome prediction · Information retrieval

1 Research question and methodology

Given a dataset of text with labeled emotions, the goal of this project is to build a model that can predict emotions in new, unseen text. The model should be able to classify emotions in categorical classes and analyze the emotional profile of main characters in a movie. The model should also be able to understand how the emotional profile changes over time and how it is affected by relationships between characters.

To achieve this goal, natural language processing techniques will be used to identify and extract subjective information from the text data. There are several approaches to emotion detection in text, including rule-based, lexicon-based, and machine learning-based methods. The project will use machine learning-based approaches, which can handle more complex or nuanced emotions but require a larger amount of annotated data for training.

The performance of the emotion detection model will be evaluated using standard metrics such as precision, recall, and F1 score, and the model will be tested on a separate test set to assess its generalization ability.

Specifically, the proposed approach is to use a Bi-LSTM model, as well as other machine learning models from the sklearn library including Logistic Regression, MLP, Decision Tree, Random Forest, and K-Nearest Neighbor. These models will be evaluated and compared to determine their accuracy in emotion prediction. The text data will be vectorized using the GloVe word embedding technique, which represents words in a high-dimensional vector space. A pre-trained embedding model will be used because the text in this project expresses general human emotions and feelings, rather than being specific to a particular corpus or task.

Once the emotion prediction model is trained and evaluated, it will be used to study the emotional profile of main characters in movies from the Cornell Movie-Dialogs Corpus. The model will classify the emotional content of the character's dialogues and the resulting emotion distributions will be analyzed to understand how the emotional profile changes over the course of the movie and how it is affected by relationships between characters. In addition, the exchanges of dialogues between characters will be analyzed to observe how the emotion changes depending on the character with whom the protagonist is interacting.

2 Data Sources

To obtain the necessary data for the project, two datasets were used: one containing written texts with labeled emotions, and the other containing movie scripts with related information.

For the emotion dataset, the 'Emotion Detection from Text' dataset on kaggle.com was utilized. This dataset includes a collection of 40,000 English-language tweets annotated with the corresponding emotion. The emotions are classified into 13 categories [Fig. 1]:

- Anger
- Empty
- Boredom
- Enthusiasm
- Fun
- Happiness
- Hate
- Love
- Relief
- Sadness
- Surprise
- Neutral
- Worry

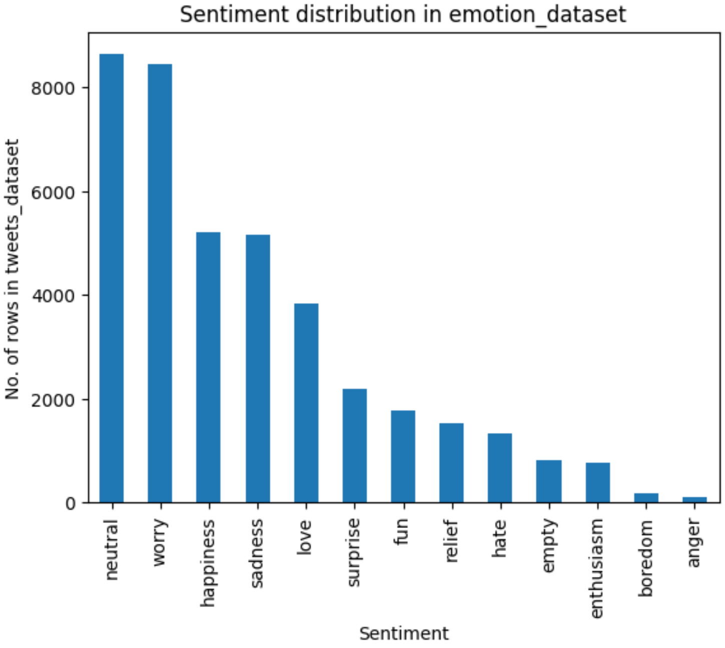


Fig. 1. Emotion distribution among different classes in the dataset

The script dataset used was the ’Cornell Movie-Dialogs Corpus,’ which is a comprehensive collection of fictional conversations extracted from raw film scripts, including rich metadata.

3 Experimental methodology

The experimental methodology for this project involves the following steps:

- Preprocessing
- Word embedding
- Emotion prediction model training
- Model evaluation
- Emotional profile analysis

3.1 Preprocessing

To train the model, the emotion dataset must be split into three different dataframes: one for training the model, one for evaluating the model during the training phase, and one for testing the trained model.

A pre-processing library called text_hammer was used to clean the text by removing:

- punctuation
- stopwords
- emails, HTML tags, website and unnecessary links
- contraction of words
- normalizations of words

This process was important to avoid training the model on irrelevant data and potentially degrading its performance. The dataset contains tweets, so special characters such as ’@’ and ’’ were also removed.

To consider only the most relevant and frequent words in the dataset, a Tokenizer was used to count word frequency and keep only the 10,000 most frequent words in the corpus. To input the sequences into the model, they must all be the same length, so the pad_sequences function was used to add padding (zeros) to the beginning or end of each sequence until it has the same length as the longest sequence. If the sequence length is greater than the specified maxlen, it will also be trimmed from the end.

3.2 Word embedding

To create a mathematical representation of the textual data that can be understood by machine learning algorithms, a text embedding was created using the pre-trained ’glove-wiki-gigaword-100’ model contained in the gensim library. This model, called GloVe, is an unsupervised learning algorithm that produces vector representations for words by training on global word-to-word statistics from a large text corpora. The resulting vectors show interesting linear relations in the vector space. The final embedding matrix had the shape [10000, 100] and contained GloVe vectors for each cleaned word in the dataset.

3.3 Emotion prediction model training

The final step in the process was to create a Bi-LSTM based sentence model. A Bi-LSTM is a sequence processing model that consists of two LSTMs: one that processes the input in a forward direction, and the other that processes it in a backward direction. By using a Bi-LSTM, we can increase the amount of information available to the network and improve the context available to the algorithm, allowing it to better understand the relationships between words in a sentence.

The model was trained using 25 epochs with a batch size of 120 and an early stopping patience of 5. The results of the training showed that the model began to overfit around the eighth epoch, stopping at the fourteenth epoch with a training accuracy of 43% and a validation accuracy of 37% [Fig. 2].

After training the Bi-LSTM model, the other five models in the sklearn library were also trained for comparison.

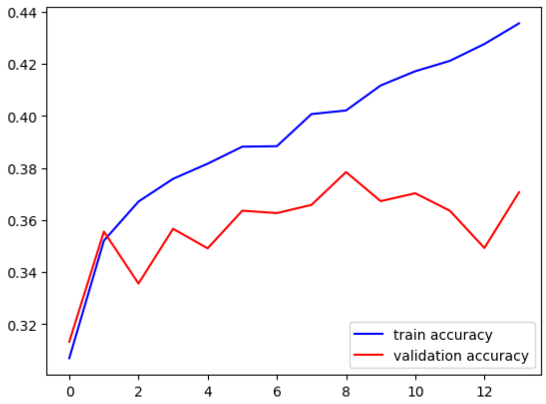
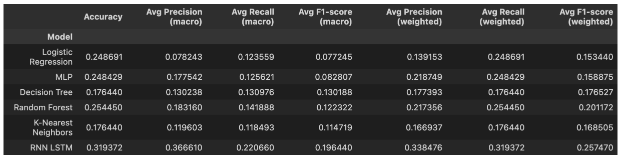


Fig. 2. Difference between training and validation accuracy throughout the epochs

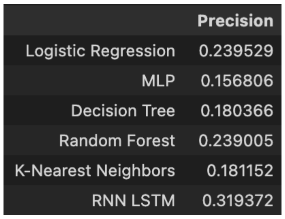
3.4 Model evaluation

To further assess the performance of the models, additional evaluation metrics were also calculated, including precision, recall, and F1 score [Fig. 3]. In addition, cross validation was also used to further evaluate the model's performance and validate its results [Fig. 4]. The precision of a model refers to the proportion of true positive predictions made by the model, while the recall measures the proportion of actual positive cases that were correctly predicted. The F1 score is a combination of precision and recall, and is a useful metric for comparing the performance of different models.



	Accuracy	Avg Precision (macro)	Avg Recall (macro)	Avg F1-score (macro)	Avg Precision (weighted)	Avg Recall (weighted)	Avg F1-score (weighted)
Model							
Logistic Regression	0.248691	0.078243	0.123559	0.077245	0.139153	0.248691	0.153440
MLP	0.248429	0.177542	0.125621	0.082807	0.218749	0.248429	0.158875
Decision Tree	0.176440	0.130238	0.130976	0.130188	0.177393	0.176440	0.176527
Random Forest	0.254450	0.183160	0.141888	0.122322	0.217356	0.254450	0.201172
K-Nearest Neighbors	0.176440	0.119603	0.118493	0.114719	0.166937	0.176440	0.168505
RNN LSTM	0.319372	0.366610	0.220660	0.196440	0.338476	0.319372	0.257470

Fig. 3. Accuracy of models on test data



	Precision
Logistic Regression	0.239529
MLP	0.156806
Decision Tree	0.180366
Random Forest	0.239005
K-Nearest Neighbors	0.181152
RNN LSTM	0.319372

Fig. 4. Accuracy of models on test data using cross validation

The results of these additional evaluation metrics showed that the Bi-LSTM model performed the best, with a precision of 33%, a recall of 31%, and an F1 score of 25%. However, it is important to note that the overall performance of all the models was relatively low, indicating that there is still room for improvement in the emotion detection task.

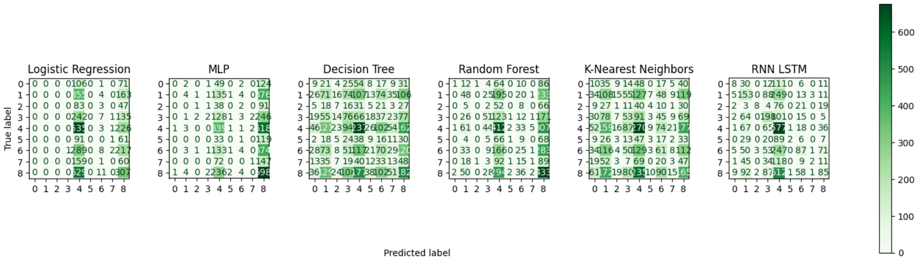


Fig. 5. Confusion matrix of the different models

In conclusion, the Bi-LSTM model was found to be the most accurate in predicting emotions in text, although its performance was not particularly strong. Future work could focus on improving the model's performance by incorporating additional features or using more advanced machine learning techniques.

3.5 Emotional profile analysis

The analysis of the protagonist's emotional state is carried out using different types of graphs. For a better comparison, the graphs taken by all the trained models are displayed.

First, an analysis in percentage of the emotions experienced throughout the film by the character is carried out. This is done by using a pie chart [Fig. 6].

Then, an analysis of the emotions is carried out, for this purpose a graph representing colored points indicating the emotion at a specific moment in the film. Each point corresponds to a line of the protagonist and the points are arranged in chronological order to have an idea of the evolution of the emotional state throughout the film [Fig. 7].

After this, using a column graph, the emotions experienced with the various characters with which the protagonist in question interacts during the film are displayed. Through this graph it is possible to observe which emotions predominate when interacting with other characters [Fig. 8].

Finally, another column graph is displayed indicating the emotions experienced during interaction with characters of different gender [Fig. 9].

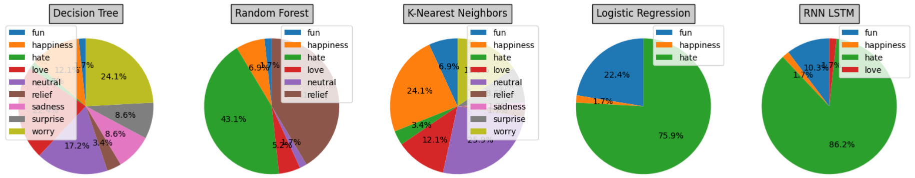


Fig. 6. Percentage of the emotions experienced throughout the film by the character

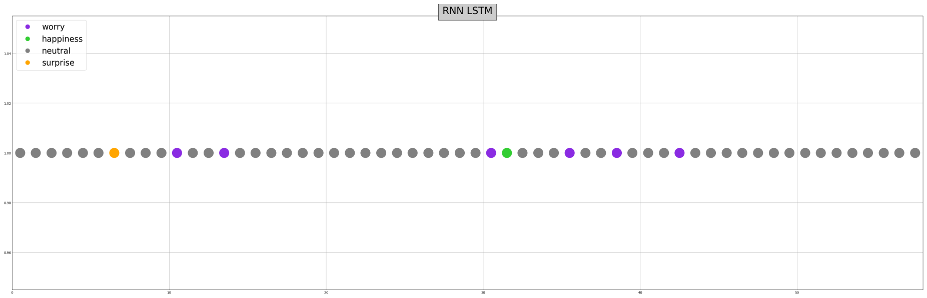


Fig. 7. Chronological order of the evolution of the emotional state throughout the film

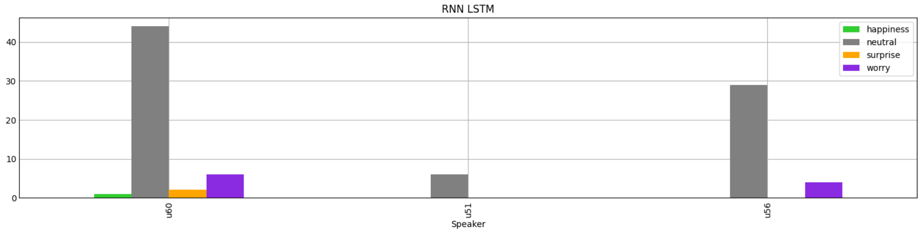


Fig. 8. Emotions experienced with the various characters with which the protagonist in question interacts during the film

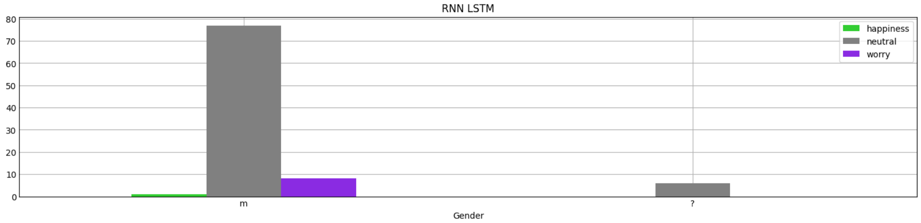


Fig. 9. Emotions experienced during interaction with characters of different gender

4 Conclusion

It is clear from the results that none of the models were able to achieve a high level of accuracy in predicting emotions in text. One possible reason for this could be the limited size of the dataset, as 40,000 tweets may not be sufficient to capture the complexity and nuance of human emotions. Additionally, the use of pre-trained word embeddings and the basic text pre-processing steps may not have adequately captured the contextual and semantic information necessary for accurate emotion detection.

In order to improve the performance of the model, it may be necessary to use a larger and more diverse dataset, as well as more advanced pre-processing and word embedding techniques. For example, utilizing more advanced techniques such as word sense disambiguation or sentiment analysis could potentially provide the model with a deeper understanding of the text and lead to more accurate emotion predictions.

Overall, while the results of this experiment are not particularly promising, there is still much potential for improving the accuracy of emotion detection in text through the use of more advanced techniques and a larger and more diverse dataset.