How do you feel, my dear?

Alessio Bonatesta

Department of Computer Science, Università degli Studi di Milano, via Celoria 18, Milano, 20133, Italy .

Author's email: alessio.bonatesta@studenti.unimi.it;

Abstract

Recently, emotion detection in text has received attention in the literature on sentiment analysis. Detecting emotions is important for studying human communication in different domains, including fictional scripts for TV series and movies. The project aims to study and analyze different machine learning algorithms in combination with different methods of natural language representation in order to implement an emotion detection model and exploit it to study the emotional profile of the main characters in one of the movies included in the Cornell Movie–Dialogs Corpus.

Keywords: NLP, Emotion Detection, Machine Learning, Word Embedding, GloVe, TF-IDF

1 Introduction

Recently, the evolution of technology and the great advances made in the field of natural language analysis have opened new doors for the different applications of LLM in everyday life: Virtual Assistants, Machine Translation,Customer Support, Content Generation, and so on. One of the situations in which these language models are exploited concerns sentiment analysis (just think of the analysis of the feelings of the customers of the own company in the reviews and the incredible gain of time deriving from the use of an automatic model of recognition), technique that aims to classify sentences in classes of feeling like "negative", "neutral", "positive".
A similar but more specific task is emotion detection one: it goes beyond the simple identification of positive or negative feelings and is able to identify specific emotions, such as happiness, sadness, anger, fear, etc. This type of analysis seeks to answer the question "What specific emotions were expressed in the input?". Emotion detection is therefore more complex, as it requires classification into a larger set of emotions, and emotions can be nuanced and complex. This class of problems finds applications in areas such as psychology, mental health, the analysis of interactions with chatbots and others, up to the world of analyzing scripts of movies and TV series. What, then, is an emotion? According to WordNet Search 3.0, an emotion is "any intense feeling." Wikipedia defines an emotion as "a mental and physiological state associated with a wide range of sensations, thoughts, and behaviors." According to Paul Ekman, a well-known American psychologist and pioneer in the study of emotions, "Emotions are a process, a particular kind of automatic appraisal influenced by our evolutionary and

personal past, in which we sense that something important to our welfare is occurring, and a set of psychological changes and emotional behaviors begins to deal with the situation." [1]. According to him, emotions are classified into 6 different types: joy, sadness, fear, surprise, anger and disgust [2].
Several researchers have based their work on the emotion model defined by Ortony, Clore, and Collins (the OCC model) [3], a widely used emotion model that states that the strength of a given emotion depends primarily on the events, agents, or objects in the environment of the agent exhibiting the emotion. The model specifies about 22 categories of emotions, a list that also contains the 6 emotions identified by Ekman.

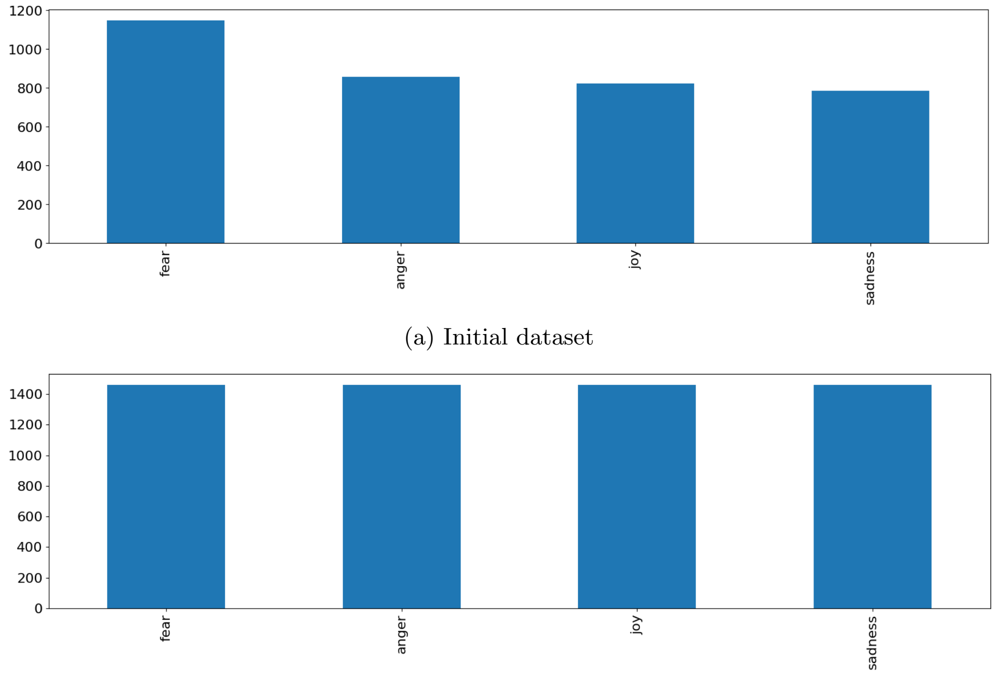
Currently, the identification of emotions is mainly addressed on two different levels: the textual and the visual. There are several modern systems, as well as several ongoing research in the field of computer vision, that aim to recognize the emotions expressed by a subject's face. The second way, on the other hand, concerns the classification of text written in natural language. Unfortunately, classifying a text in its expressed emotions presents some significant challenges due to the complexity of human emotions and the nuanced nature of natural language, such as: problems of semantic ambiguity, emotional nuances, cultural variability, irony and sarcasm, etc.
The text-based emotion identification experiments carried out in [4] lead to the conclusion that in-depth semantic analysis of text is necessary to obtain more accurate results. As noted in [5], there are three approaches to the emotion detection task: keyword-based, learning-based and hybrid based.
The learning-based approach uses a trained classifier to classify input text in emotion classes using keywords. Such a model is easier and faster to adapt to the change in domain.

In this project we will try to analyze different machine learning algorithms by studying and comparing the performance of each one in combination with two different ways of representing natural language: Bag of Words and Vector representation by Glove. Then, after comparing the accuracy of the models by cross-validation, we will apply the selected model for the analysis of a movie script, in particular "Will Hunting", trying to understand if the results obtained are in agreement with the knowledge we have about the film.

2 Dataset

Initially the work was done working on the dataset "Emotion Detection from Text", present in Kaggle [6]. Obtaining poor results in the tests carried out (which will be explained in the next chapter) it was decided to change the data source, moving the analysis to the data presented in the WASSA-2017 Shared Task on Emotion Intensity [7]. It is a labeled dataset dedicated to the supervised development of a machine learning model for emotion detection, composed of numerous tweets associated to 4 classes of emotions: fear, anger, joy, sadness.

At the beginning, extracting the training data offered on the competition web page, the situation that occurred was that shown in Figure 1a. Given the small number of records per class and the significant overrepresentation of instances within the "fear" category, it was decided to augment the training set by incorporating the data found in what the authors referred to as the "development set" and subsequently performing the resampling operation (more specifically, random downsampling to the number of records contained in the lower cardinality class). After this operation, the dataset was formed by four classes each of 1400 samples, as shown in Figure 1b.



(a) Initial dataset

(b) Dataset after union and resampling

Fig. 1: Dataset balance

2.1 Data preprocessing

Data preprocessing is a very important part of developing an effective emotion detection model, which aims to identify emotions or feelings expressed in a text or sentence. There are several necessary steps to prepare the textual data in order to make them suitable for the analysis of emotions; for this project, the steps taken are as follows:

1. Contractions in sentences have been removed using the library contractions. This step was necessary to separate the negative contracted verbs (for example, ”can’t”) and keep the negative part while removing the stop words (see point 6 in this list);

2. Given the presence of mentions and URLs inside the tweets, this type of words have been removed;

3. Text tokenization was performed with the word tokenizer of the library nltk;

4. Uppercase letters have been lowercased;

5. Punctuation has been removed;

6. Thanks to the nltk stop words corpus these kind of words were filtered out. Since the negative terms could be fundamental to distinguish an emotion from another, it was decided to remove from the stop list the negative terms ”not”, ”nor”, and ”no”, thus managing to keep them.

7. Finally, thanks to WordNetLemmatizer, the lemmatization operation has been performed.

Since in the dataset there were some duplicates and some with different associated labels, the last operation of the preprocessing phase was to remove these duplicates, in order to avoid that these duplicates adversely affect the models.

3 Experiments

3.1 Bag of words

The first experiments carried out concerned the use of the Bag of Words model for the representation of the text involved. As explained in [REF], following this view, a sentence is represented by retaining only information about the number of occurrences of each term, and the exact ordering of the terms in a text is ignored. Despite this, the most widely used approach for a BOW representation is the TF-IDF representation: it also captures information about the frequencies of words in documents by calculating the Term Frequency, but the Inverse Document Frequency value weights words according to their importance in the entire collection of documents. The TF-IDF value for a word in a document is obtained by multiplying its TF in the document by its IDF. So, the higher the TF-IDF value of a word in a document, the greater its importance in that specific context.

The experiments in question were carried out following the same procedure:

1. Initially, the dataset was split into training sets and test sets using sklearn, ensuring that the data split maintained the same distribution as the target classes or labels present in the original dataset. The test set, despite its name, was actually used as a validation set during the tuning of the hyper-parameters.

2. The parameters were then tuned. A list of n-gram values and a number of other parameters dependent on the specific model were then defined, and different models were then trained by applying the different combinations of hyper-parameters. Each model has been defined within a pipeline containing sklearn's TfidfVectorizer followed by that model.

3. The accuracy of the various models was taken into account for the selection of the representative model of the machine learning algorithm under consideration.

In the .ipynb file, at the end of each section dedicated to the model under consideration, there are the results of the model selected in classifying the test set. We report these results below.

3.1.1 Multinomial Logistic Regression

The first algorithm analyzed was multinomial logistic regression. Here are the hyper-parameters tested:

- n-grams range: [(1,1),(1,2),(2,2),(1,3),(2,3),(3,3)]
- solvers: ['lbfgs', 'liblinear']

The model with the highest accuracy was found to be the one using both 1-grams and 2-grams, i.e. with *range=(1,2)*, in combination with the *lbfgs* solver (and therefore l2-type penalties). In Figure 2a you can see the accuracy and the classification report obtained by the selected model, while in Figure 2b you can see the relative confusion matrix.

3.1.2 Random Forest

Here are the hyper-parameters tested for the devolopment of a Random Forest:

- n-grams range: [(1,1),(1,2),(2,2),(1,3),(2,3),(3,3)]
- number of estimators: ['lbfgs', 'liblinear']
- losses: ['gini', 'entropy', 'log loss']

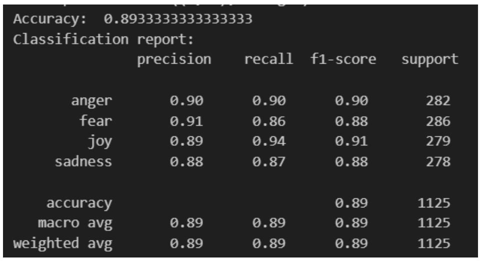
The model with the highest accuracy was found to be the one using 1-grams, 2-grams and 3-grams, i.e. with *range=(1,3)*, in combination with 100 estimators and the *entropy* loss function. In Figure 3a you can see the accuracy and the classification report obtained by the selected model, while in Figure 3b you can see the relative confusion matrix.

3.1.3 SVM

Here are the hyper-parameters tested for the devolopment of a SVM classifier:

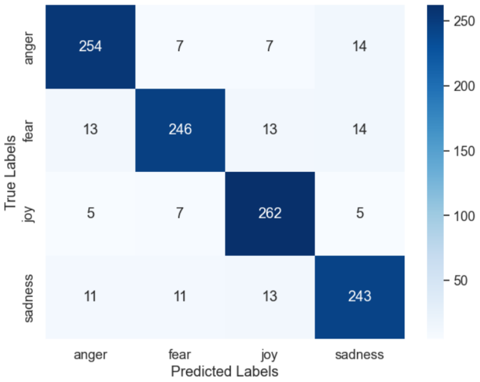
- n-grams range: [(1,1),(1,2),(2,2),(1,3),(2,3),(3,3)]
- kernels: ['linear', 'poly', 'rbf', 'sigmoid']

The model with the highest accuracy was found to be the one using 1-grams and 2-grams, i.e. with *range=(1,2)*, in combination with the *sigmoid* kernel. In Figure 7a you can see the accuracy and the classification report obtained by the selected model, while in Figure 7b you can see the relative confusion matrix.



```
Accuracy:  0.8933333333333333
Classification report:
               precision    recall  f1-score   support

       anger       0.90      0.90      0.90       282
        fear       0.91      0.86      0.88       286
         joy       0.89      0.94      0.91       279
     sadness       0.88      0.87      0.88       278

    accuracy                           0.89      1125
   macro avg       0.89      0.89      0.89      1125
weighted avg       0.89      0.89      0.89      1125
```

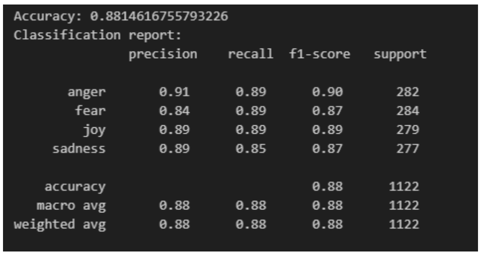
(a) Classification report.



True Labels \ Predicted Labels	anger	fear	joy	sadness
anger	254	7	7	14
fear	13	246	13	14
joy	5	7	262	5
sadness	11	11	13	243

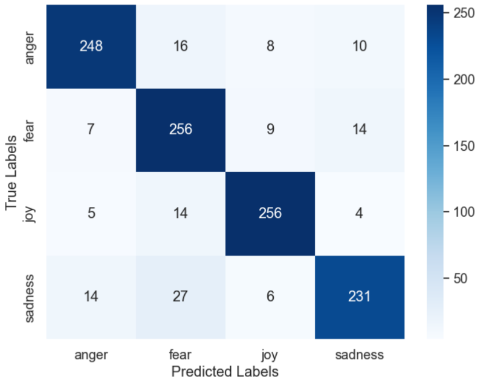
(b) Matrix profile.

Fig. 2: Multinomial Logistic Regression.



```
Accuracy: 0.8814616755793226
Classification report:
              precision    recall  f1-score   support

       anger       0.91      0.89      0.90       282
        fear       0.84      0.89      0.87       284
         joy       0.89      0.89      0.89       279
     sadness       0.89      0.85      0.87       277

    accuracy                           0.88      1122
   macro avg       0.88      0.88      0.88      1122
weighted avg       0.88      0.88      0.88      1122
```

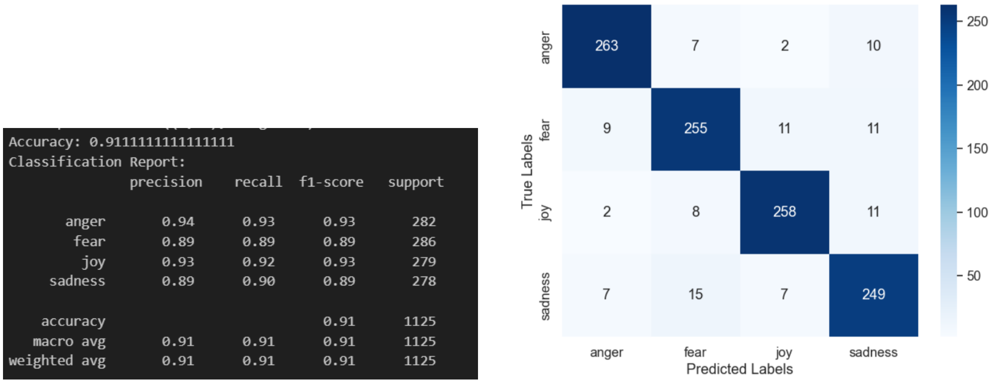
(a) Classification report.



True Labels \ Predicted Labels	anger	fear	joy	sadness
anger	248	16	8	10
fear	7	256	9	14
joy	5	14	256	4
sadness	14	27	6	231

(b) Matrix profile.

Fig. 3: Random Forest.



```
Accuracy: 0.9111111111111111
Classification Report:
              precision    recall  f1-score   support

       anger       0.94      0.93      0.93       282
        fear       0.89      0.89      0.89       286
         joy       0.93      0.92      0.93       279
     sadness       0.89      0.90      0.89       278

    accuracy                           0.91      1125
   macro avg       0.91      0.91      0.91      1125
weighted avg       0.91      0.91      0.91      1125
```

(a) Classification report.

(b) Matrix profile.

Fig. 4: Support Vector Machine.

3.2 GloVe

The second type of word representation on which the analysis focused on is a semantic vector space model, in which each word is represented as a real-valued vector. These vectors capture semantic relationships, allowing the model to understand the meaning of words based on the context in which they appear. The model selected for this analysis is GloVe, because as reported in [8] it performs significantly better than the other baselines, often with smaller vector sizes and smaller corpora. The pre-trained GloVe has been imported thanks to the gensim library. Every trained model was tuned as in previous section. One of the hyper parameter tested for every model is the number of GloVe's vectors' dimensions: 100, 200 or 300.

3.2.1 Multinomial Logistic Regression

As in the previous chapter, the first algorithm tested in combination with Glove was multinomial logistic regression. The model selected after the hyper parameter tuning is a regressor that works on vectors of 300 values and with a *liblinear* solver. As for the other experiments, in Figure 5 there are the results.

3.2.2 Random Forest

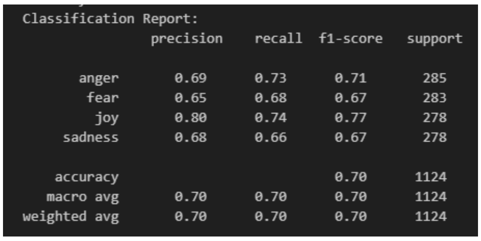
Here the results for the Random Forest combined with GloVe. The model selected after the hyper parameter tuning is a forest with 200 trees that work on vectors of 300 values and with a *log loss* criterion. In Figure 6 there are the results.

3.2.3 SVM

Clearly, also here we evaluate among the different algorithms the use of a SVM classifier. The model selected after the hyper parameter tuning is a classifier that works on vectors of 300 values and with a *linear* kernel. The results are shown in Figure 7.

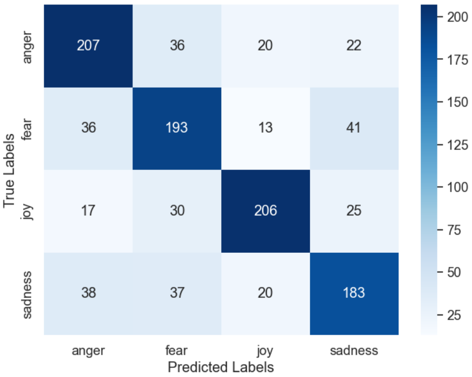
3.2.4 Neural Network

Regarding the test on Neural Network, the process has been the same. We chose to train some simple networks, all of them composed by two Dense layers, with the SGD optimizer and catogorical crossentropy loss function. The resulted model was a NN with two layers of size, in order, 512 and 256, based on a vector representation of 300 dimensions. The accuracy on the test set, in this case, was 0.7613.



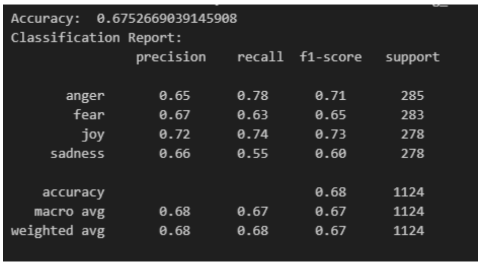
```
Classification Report:
              precision    recall  f1-score   support

       anger       0.69      0.73      0.71       285
        fear       0.65      0.68      0.67       283
         joy       0.80      0.74      0.77       278
     sadness       0.68      0.66      0.67       278

    accuracy                           0.70      1124
   macro avg       0.70      0.70      0.70      1124
weighted avg       0.70      0.70      0.70      1124
```

(a) Classification report.



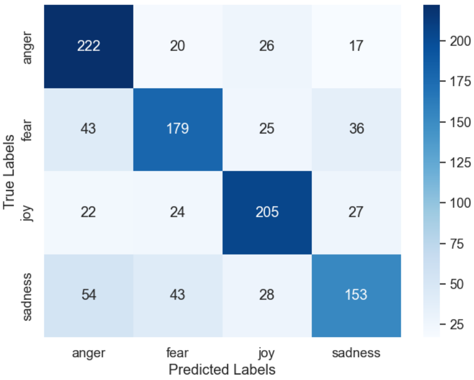
(b) Matrix profile.

Fig. 5: Logistic Regression with GloVe representation.



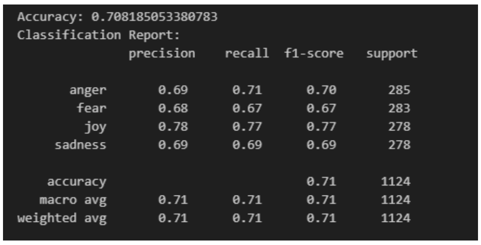
```
Accuracy:  0.6752669039145908
Classification Report:
              precision    recall  f1-score   support

       anger       0.65      0.78      0.71       285
        fear       0.67      0.63      0.65       283
         joy       0.72      0.74      0.73       278
     sadness       0.66      0.55      0.60       278

    accuracy                           0.68      1124
   macro avg       0.68      0.67      0.67      1124
weighted avg       0.68      0.68      0.67      1124
```

(a) Classification report.



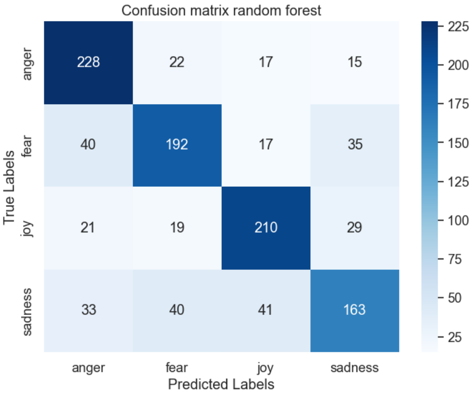
(b) Matrix profile.

Fig. 6: Random Forest with GloVe representation.



```
Accuracy: 0.708185053380783
Classification Report:
              precision    recall  f1-score   support

       anger       0.69      0.71      0.70       285
        fear       0.68      0.67      0.67       283
         joy       0.78      0.77      0.77       278
     sadness       0.69      0.69      0.69       278

    accuracy                           0.71      1124
   macro avg       0.71      0.71      0.71      1124
weighted avg       0.71      0.71      0.71      1124
```

(a) Classification report.



(b) Matrix profile.

Fig. 7: Support Vector Machine with GloVe representation.

4 Validation

All models selected as representatives of the algorithms taken into analysis were compared through *k-fold cross-validation*. In order to obtain balanced folds the method used was the StratifiedKFold, present in the *sklearn.model_selection* library. We chose to use the mean accuracy as the metric to compare the models and select the final one. In Table 1 the mean accuracy value for each model combined with the two representations.

Table 1: Results of cross validation

	Logistic Regression		Random Forest		SVM		NN[1]
Scoring	TF-IDF	GloVe	TF-IDF	GloVe	TF-IDF	GloVe	
Accuracy	0.8827	0.7039	0.8630	0.6895	0.8947	0.6938	0.7606
Precision	0.8830	0.7042	0.8654	0.6895	0.8952	0.6954	0.7501
Recall	0.8829	0.7041	0.8631	0.6893	0.8949	0.6939	0.7726
F1 score	0.8826	0.7039	0.8633	0.6865	0.8948	0.6942	0.7613

Note: Precision, recall and F1 present the macro-averaged results.

[1] Neural Network tested only with GloVe

4.1 Results

The experiments conducted lead us to highlight mainly three observations.
The first is that the models involved in the analysis can achieve better results when combined with a TF-IDF representation, at least in this specific case where we used the dataset shown. One possible explanation might be the small size of the dataset, combined with the fact that the emotions within it are strongly related to keywords or specific terms, overshadowing the semantic context.
Secondly, GloVe-based models perform better when they work with vectors of higher dimensions. This makes sense when you think that the use of larger vectors allows the model to more accurately represent the meaning of words and the relationships between them. This is because larger dimensions create a larger semantic space, in which words can be more precisely positioned according to their meaning. This allows the model to capture more subtle nuances in the meaning of words.
Finally, the model best suited to this task in this context is, in accordance with results in [5], an SVM (in our case based on a representation of the words of type TF-IDF) .

5 Film analysis

As was mentioned in the introduction, the purpose of this project is to train a model from a tweet dataset and use it for emotional analysis of a character in a movie, giving input to the trained model the conversations within the script. The dataset from which the conversations of the chosen film were extracted is the Cornell Movie-Dialogs Corpus [9]. The main character of ”Good Will Hunting” has been chosen for the analysis, since he is known for his complexity. Will is a genius with a difficult past, characterized by traumatic experiences. This complexity provides a fertile ground for conducting an analysis of his emotional profile.

Analyzing the conversations involving the protagonist, giving input to the model the sentences spoken by him and associating the output to the person to whom he turns in the conversation, What emerges is that the emotions expressed by the main character are predominantly negative. In particular, we can see in Figure 8 how the

predominant emotion is fear. Fear is one of the key emotions that define him because of his difficult past and traumatic experiences. Fear of abandonment and lack of self-confidence lead him to build emotional barriers and reject people who try to approach him. This fear of being vulnerable and injured is an important component of his emotional profile and deeply affects his behavior.
The other two prevailing emotions turn out to be anger and fear. These are in fact key feelings that define the character of Will Hunting, and we can therefore say that the results of the model reflect with an acceptable degree of confidence the complexity of the emotional profile of the protagonist of the film. In support of this statement we can also note that the character of Skylar (the girl Will falls in love with) is the one with whom he shows most joy in conversations. Skylar's presence in Will's life drives him to consider his future and his chances for happiness. Skylar's presence in Will's life is a source of happiness and joy for him. His romantic relationship with her offers him a sense of belonging and hope for the future, which are positive emotions in contrast to the more negative ones we have already talked about.

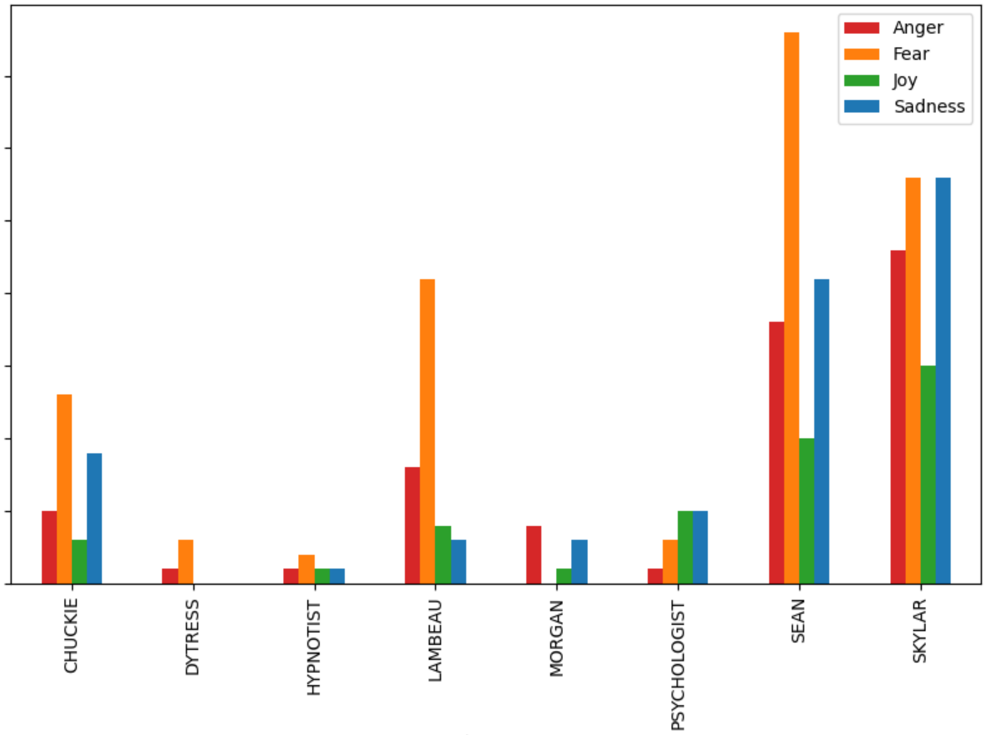


Fig. 8: Emotions related to interlocutors.

In order to analyze the character's emotional profile during the film, it was decided to divide the entire duration of the film into 7 sub-periods and analyze the conversations in relation to the period instead of the interlocutors. Figure 9 shows the results.

6 Conclusion

To summarize, we have demonstrated how machine learning can be utilized to detect emotions in cinematic scripts and how it could lead to a deeper understanding of the emotional dynamics that underlie the narratives. Through the accurate analysis of

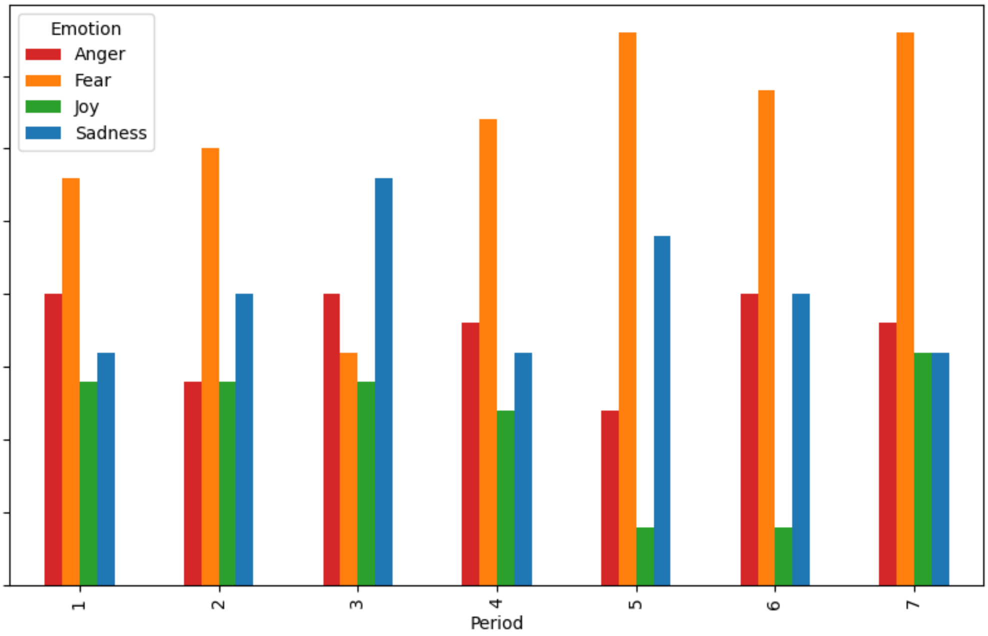


Fig. 9: Emotions related to interlocutors.

the words and linguistic nuances present in the scripts, machine learning models can identify and interpret the emotions of the characters in a more or less sophisticated and detailed way.

The effectiveness of such systems is influenced by the quality of training data and the complexity of human emotions, which can be difficult to categorize. Therefore, the continuous development of more advanced algorithms and the deepening of human understanding of emotions are essential to ensure significant progress in this fascinating field of research.

References

[1] Paul Ekman emotions. https://www.paulekman.com/universal-emotions/

[2] Ekman, P., *et al.*: Basic emotions. Handbook of cognition and emotion **98**(45-60), 16 (1999)

[3] Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1988)

[4] Aman, S.: Recognizing emotions in text. PhD thesis, University of Ottawa (Canada) (2007)

[5] Binali, H., Wu, C., Potdar, V.: Computational approaches for emotion detection in text. In: 4th IEEE International Conference on Digital Ecosystems and Technologies, pp. 172–177 (2010). IEEE

[6] Emotion Detection from Text. Kaggle. https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text

[7] Mohammad, S., Bravo-Marquez, F.: WASSA-2017 shared task on emotion intensity. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 34–49. Association for

Computational Linguistics, Copenhagen, Denmark (2017). https://doi.org/10.18653/v1/W17-5205 . https://aclanthology.org/W17-5205

[8] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

[9] Danescu-Niculescu-Mizil, C., Lee, L.: Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In: Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011 (2011)