EMOTION RECOGNITION IN TWEETS

Behnam Nikbakhtbideh

University of Calgary
Electrical and Software Engineering

ABSTRACT

The ever-increasing user-generated content on social media has made opinion mining an arduous job. As a microblogging platform, Twitter is being used to collect views about products, trends, and politics. Emotion recognition is a technique used to analyze the attitude, emotions and opinions of different people towards anything, and it can be carried out on tweets to analyze public opinion on news, policies, social movements, and personalities.

By implementing techniques in machine learning, emotion recognition aims to classify input to an emotional state where the input might be of textual or media type. With respect to tweets, the main aspect of the work is within NLP[1] to analyze textual data. In this work, a Bayesian based model plus multiple enhancements is investigated and results are interpreted and compared with one of the open source solutions.

***Index Terms*—** tweet, text-analysis, text-mining, nlp, classification, bayesian, emotion-recognition

1. INTRODUCTION

Emotion recognition is an important domain in AI[2] because of its impact on governments and market to apply the correct decisions. By moving the global trend towards Big Data, Nowadays social media platforms like Twitter are producing numerous amounts of data that can reflect the general attitude of society.

Emotion Recognition and Sentiment Analysis are two kinds of classification problems to detect what emotion/sentiment the human actor has in media or text. The focus of sentiment analysis is to derive information from human language for interpreting views and feelings to assign a label like positive, negative, or neutral. However, emotion detection aims at finding out more specific sentiment tones.

According to [1], global emotions are limited to six basic categories: **happiness**, **surprise**, **sadness**, **anger**, **fear**, and **disgust**. In fact, other complex emotional states can be derived via these universal categories with respect to situations like culture and sex.

Emotion Recognition in Tweets is a kind of emotion recognition that focuses on tweets as input data. Tweets have some characteristics that make them distinctive from other resources. One feature is the multimedia nature of tweets that might contain image, audio and video. But the majority of works in this domain concentrate on text. The other feature related to social media is that when it comes to analyse data, some special inputs like emoji could be considered in NLP. Also gaining benefit from the nature of social networks might be helpful in detection of emotion based on the connections and interaction of people towards contents that their emotion are previously detected.

This study aims to develop a specific model that can reach an acceptable level of accuracy compared with existing solutions. The key contribution of this study is to examine multiple pre-processing approaches to configure a reasonable model for Bayesian classifier in emotion recognition.

The rest of the paper is organized as follows. Section 2 discusses literature related to the current research work. Section 3 presents the proposed methodology. Results are presented in Section 4 and also future work is suggested. Section 5 finally concludes the research work.

2. RELATED WORK

In terms of input type, techniques in this area could be categorized in three main sections:

2.1. Multi Modal

Most of the work in emotion recognition focus on text because tweets are mostly in text. But in some papers like [2], it tries to fuse both visual and textual models to get a more comprehensive result.

Emotion Recognition can be applied to text, speech, and facial expressions. So there is a solution to combine the result for each of these problems. In this research, a test for interpreting the emotion from media (audio and image) is done with the following results.

Audio: By using tools like PANNs[3] that is based on Py-

[1] Natural Language Processing
[2] Artificial Intelligence
[3] https://github.com/qiuqiangkong/audioset_tagging_cnn

torch[4] and Pandas[5], we can analyze a voice (not necessarily speech) and get some classes that could be tagged as emotion like the figure 1.

Image: By using the dlib[6] library for emotion recognition in images (or video frames), a sample result is extracted as figure 2 that conveys some emotions like *happiness*.

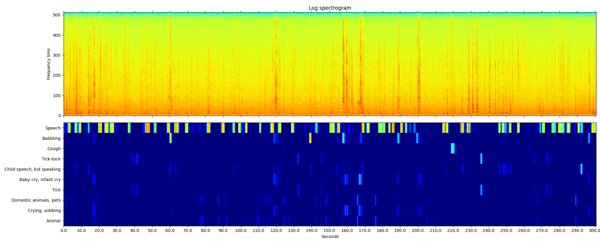

Fig. 1. emotion recognition on audio

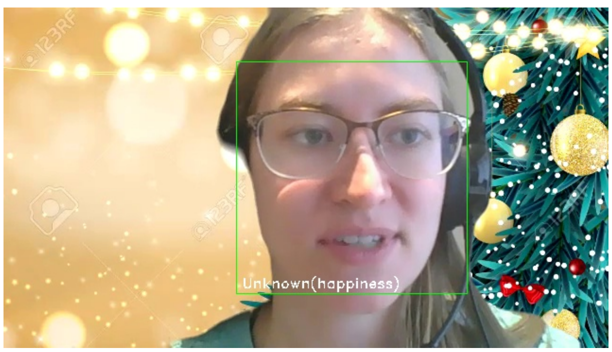

Fig. 2. emotion recognition on image

The main approach here is to connect the output of prior classification to CNN[7], and then the problem is to build a model like SVM[8] that minimizes CCC[9]. Although the most problematic part of this model is to combine (or fuse) these results that need specific dataset.

2.2. Social Network Features

Some papers try to utilize information that can be gained by social relationships and behaviour. For example in [3], the focus is on social media features like user's opinions, amount of user activities (number of tweets for example), and user's behaviour (for example, periods of being active or inactive).

[4]https://pytorch.org/
[5]https://pandas.pydata.org/
[6]https://github.com/davisking/dlib
[7]Convolutional Neural Networks
[8]Support Vector Machine
[9]Concordance Correlation Coefficient

2.3. Text Based

Works in this domain concentrate on NLP models to analyze tweets as textual data. In [4] it uses a non-supervised learning model that requires no manual annotation. This technique starts with a couple of hashtags like *#angry*, *#happy*, or *emoticons*, and assign a label to each tweet. Then it extends its training set automatically. The limitation of this technique is that such initial data is not present in all kinds of tweets, and it will cause a bias in the training model.

Also in [5], it uses another kind of non-supervised learning model that is based on linguistic and acoustic features of language to automatically assign labels to each tweet. It seems that this technique doesn't remove the need for learning, and only transfers it to another level which might decrease accuracy.

On the other hand, supervised models try to construct a model based on training data. Multiple techniques like SVM, decision-trees and Bayesian models, and also deep-learning models for text classification are provided in this.

Supervised solutions could be divided in two general parts: **statistical models** and **NLP models**.

Statistical models concentrate on entropy features of the language, but in **NLP solutions** the very basic infrastructure of work is individual words or sentences, and not single characters. The most important difference in feature extraction that aims to map text into vectors.

Bag of Words is a straightforward technique for feature extraction that the vector is a matrix of words in dictionary with corresponding frequency in the input text. Although this is a simple solution, it is not efficient because of the sparse matrix with the size equals to the dictionary.

To categorize text (tweet), measures like TFIDF[10] can be used to normalize vectors and gain more meaningful data instead of raw frequencies. Meaning that it normalizes frequency against how much the frequency of that word is in document, and how much is in the dictionary. For example, a word like *world* is quite frequent in English literature and might not convey a special meaning or feeling, but a word like *disgust* can be determinant to find out about semantic state of the text. After that, some kind of text similarity i.e. Levenshtein Distance can be applied to find the most similar set of texts to the input, and assign a label (emotion) based on it.

However, one deficiency of these vectors is that they don't preserve the order of the words, and n-gram solutions can resolve this issue [6]. For example, in a 2-gram model, *do not worry at all* will be mapped to a set of bigrams like {"*do not*", "*not worry*", "*worry at*", "*at all*"}. Here a token like *not worry* can be very helpful for detection of emotion, but in ordinary tokenization, even token *not* might be dropped by stop-words removal.

On the other hand, statistical approaches might use n-gram based on the characters [7]. In this approach, there is

[10]Term Frequency - Inverse Document Frequency

no linguistic dependency between the model and a specific language.

In addition to these pre-processing steps, other tasks to remove stop-words, doing Stemming or Lemmatization can be helpful in some cases.

3. MATERIALS AND METHODS

Building a comprehensive model is not possible without trying to connect multiple models and components, and applying pre-processing. in [8], it combines various machine learning-based classifiers including SVM, DTC[11], NB[12], RF[13], GBM[14] and LR[15] and then compare the results.

The proposed model is based on a Bayesian classifier with multiple enhancements in pre-processing and is comparable with the results in [8]. Also an open-source tool [9] is selected to compare the accuracy against a dataset in [10] with 16000 classified text entries.

Due to some differences between these two models, 14696 entries are analyzed and among them, only 5425 items are classified correctly which makes about 37% of accuracy, and the overall time was about 1ms for each item. The proposed model reaches average accuracy of 72% with this dataset.

3.1. Dataset

Regarding the inter-dependence connection between emotional classes, the level of accuracy is also dependent to the dataset. With another dataset in [11] with categories [*sadness*, *enthusiasm*, *neutral*, *worry*, *surprise*, *love*, *fun*, *hate*, *happiness*, *boredom*, *relief*, *anger*] it only gives half the accuracy in comparison with [10]. For example, a category like *worry* might be inter-related to both categories *fear* or *sadness*, and this will result in lower accuracy. But the dataset in [10] has only six global categories without any correlation including [*sadness*, *anger*, *love*, *surprise*, *fear*, *joy*].

3.2. Preparation

Prior to enter to the classifier, various tests are done to increase the accuracy of the model. In this section, a number of them with positive impact will be discussed.

Tokenization is the first preparation step to convert the text into separated tokens. A function named *word_tokenize* in *NLTK* is used here that split words by space characters and normalize them to lowercase.

Two enhancements on *tokenization* is done. The first is to apply an optional *n-grams* combination on tokens. For example, tokens *[I, am, not, happy]* will be converted to a new bigram set of tokens *[I am, am not, not happy]*. Due to lack of enough data for these tokens, the accuracy of this enhancement is fallen by more than 40% and we skip it for now.

The next enhancement is to concatenate token *not* to its next token. So the input *[I, am, not, happy]* will be turned into *[I, am, not happy]*. This enhancement will reduce accuracy by 5% but results in more accurate prediction for sentences like *I am not happy* by 50%.

Stop-Words Removal is the next step that increases average accuracy by about 10% and is used to remove some frequent tokens without any specific meaning. For example, for tokens *[I, am, happy]*, the result of this step is *[happy]*.

Stemming is the next optional state that reduces the words to their structural roots. For example, both two words *book*, *books* will turn into *book*. Most of the well-known facilities in this part are based on statistical analysis without considering the meaning of words. For example, considering the *Porter-Stemmer* in *NLTK*, *Homes* is transferred to *home*, *Winning* is transferred to *win*, but *Alone* will be converted to *alon*. Having this step will increase accuracy by 2% on the dataset that is used.

Lemmatization is another replacement for *Stemming* that keeps the semantic of the tokens. In this step, a utility from *NLTK* named *WordNetLemmatizer* is used that is based on *Wordnet*. By applying this model, the token *good* will be converted to *good*. Using this model has not a significant improvement more than *Stemming* on the dataset that is used, and this is resulted from the diversity of samples in the dataset. Although for some categories *Lemmatization* will increase the performance.

3.3. Model

The model itself is a Bayesian classifier with multiple preprocessing in the preparation phase. The dataset is split into 90% as trainset and 10% for test. The accuracy and performance of the model differs between categories and in some cases is higher than [8], but in average is about 6% lower than it. Also the average accuracy is two times more than the open-source tool in [9].

3.4. Implementation

The python libraries that is used in this project is *NLTK* for general methods needed in *NLP* and also for managing and fetching the required datasets needed for stop-words removal, wordnet and punkt (tokenization).

A python class named *TextItem* is responsible to cover all operations applied on a single text(Tweet) including *Tokenization*, *Stemming* and *Lemmatization*. Another class named *NltkClassifier* is to maintain and handle *train*, *test*, and *prediction*. The other part of the work is to receive user's command from CLI, parse the arguments and apply the required operations. There is also a possibility to store

[11] Decision Tree Classifier
[12] Naive Bayes
[13] Random Forest
[14] Gradient Boosting Machine
[15] Logistic Regression

the trained model in that the prediction for each single input (Tweet) will be possible without the need to re-train and re-build the model.

4. RESULTS AND DISCUSSIONS

Table 1 shows a comparison between performance of multiple categories. It shows a wide diversity between these classes that reduces the total accuracy by about 73%. The results achieved in [8] are shown in Table 2 and this shows a acceptable result in our work for categories *Sadness*, *Anger*, *Fear*, and *Joy*. Although the model in [8] used the dataset with categories [*Positive*, *Neutral*, *Negative*] that are more distributed, and less inter-correlated. So we believe that our approach is comparable with it although has less average performance.

Category	Precision	Recall	F1-Score
Sadness	0.90	0.69	0.78
Anger	0.81	0.74	0.77
Love	0.59	0.76	0.66
Surprise	0.20	0.85	0.32
Fear	0.77	0.75	0.76
Joy	0.91	0.71	0.80

Table 1. performance results of the proposed model

Model	Precision	Recall	F1-Score
RF	0.74	0.79	0.77
SVM	0.76	0.80	0.78
NB	0.75	0.78	0.75
DT	0.74	0.77	0.76
GBM	0.72	0.79	0.76
LR	0.79	0.82	0.80
Proposed model	0.78	0.84	0.81

Table 2. performance results of the model [8]

Another comparison is on the effect of pre-processing steps on the results as illustrated in 3 that shows about 10% of improvement in category *sadness*.

Pre-processing	Precision	Recall	F1-Score
No	0.94	0.57	0.71
Yes	0.90	0.70	0.78

Table 3. performance results of the proposed model for category *sadness*

Further Work: As given in table 1, there is a diversity between classes that is resulted from lack of enough data and inter-connection between two categories *Love* and *Surprise*. The results show high rate of FP[16] for these two classes rather than the others, also the number of samples for these two categories are lower than the others.

These problems show anomaly in dataset that needs various examination in feature-selection, feature-extraction and validation. Chi-square test is a technique to examine independence of features, and k-fold cross-validation is useful to distribute the space of test/validation over training set.

[16]False Positive

5. CONCLUSIONS

Emotion Recognition is a general task in classification that assigns a label as emotion to the input as text or media. Regarding the text as the main media for tweets, most of the work in this domain focus on NLP, and a variety of models based on classification algorithms like Bayesian model, decision-tree and SVM are proposed. In this work, a Bayesian model is investigated with a configuration in pre-processing phase to achieve acceptable results in comparison with an open-source tool and a recent paper in this field.

6. REFERENCES

[1] Du, S., Tao, Y., Martinez, A. M. (2014). Compound facial expressions of emotion. Proceedings of the National Academy of Sciences, 111(15), E1454–E1462. https://doi.org/10.1073/PNAS.1322355111

[2] Lin, C., Hu, P., Su, H., Li, S., Mei, J., Zhou, J., Leung, H. (2020). SenseMood: Depression detection on social media. ICMR 2020 - Proceedings of the 2020 International Conference on Multimedia Retrieval, 407–411. https://doi.org/10.1145/3372278.3391932

[3] Shen, G., Jia, J., Nie, L., Feng, F., Zhang, C., Hu, T., Chua, T. S., Zhu, W. (2017). Depression detection via harvesting social media: A multimodal dictionary learning solution. IJCAI International Joint Conference on Artificial Intelligence, 0, 3838–3844. https://doi.org/10.24963/IJCAI.2017/536

[4] SintsovaValentina, PuPearl. (2016). Dystemo. ACM Transactions on Intelligent Systems and Technology (TIST), 8(1). https://doi.org/10.1145/2912147

[5] Hines, C., Sethu, V., Epps, J. (2015). Twitter: A new online source of automatically tagged data for conversational speech emotion recognition. ASM 2015 - Proceedings of the 1st International Workshop on Affect and Sentiment in Multimedia, Co-Located with ACM MM 2015, 9–14. https://doi.org/10.1145/2813524.2813529

[6] Abdaoui, Amine, et al. "Feel: a french expanded emotion lexicon." Language Resources and Evaluation 51.3 (2017): 833-855.

[7] Kruczek, J., Kruczek, P., Kuta, M. (2020). Are n-gram Categories Helpful in Text Classification? Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12138 LNCS, 524–537. https://doi.org/10.1007/978-3-030-50417-5_39

[8] Yousaf, A., Umer, M., Sadiq, S., Ullah, S., Mirjalili, S., Rupapara, V., Nappi, M. (2021). Emotion Recognition by Textual Tweets Classification Using Voting Classifier (LR-SGD). IEEE Access, 9, 6286–6295. https://doi.org/10.1109/ACCESS.2020.3047831

[9] aman2656/text2emotion-library. (n.d.). Retrieved November 26, 2021, from https://github.com/aman2656/text2emotion-library

[10] Emotions dataset for NLP — Kaggle. (n.d.). Retrieved November 26, 2021, from https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp?select=train.txt

[11] text_emotion — Kaggle. (n.d.). Retrieved November 26, 2021, from https://www.kaggle.com/maysaasalama/text-emotion/version/1