Emotion Detection in Tweets

Sarthak Gupte

Abstract **- Twitter is a well established social media platform and the users post tweets to express and share their emotions with the followers. This paper focuses on identification of emotions in the tweets. The problem which is tried to solve is multi-label classification, i. e., a tweet can belong to more than one emotion and in order to correctly identify the emotion of a tweet all of the emotions should be identified correctly. Along with classical multi-label classifiers such as Binary Relevance and Classification chain, BERT deep learning model was implemented to evaluate the performance on the SemEval 2018 Task 1 dataset. The BERT model performed the best out of all the three models.**

1. INTRODUCTION

Emotion Detection is the process of recognition and identification of human emotions. In the field of computational linguistics emotion detection is identification and extraction of human emotions from the given text. Emotion detection falls under the category of sentimental analysis, which is a part of Natural Language Processing(NLP). Sentimental analysis uses natural language processing to extract and classify emotion in positive, negative or neutral emotions. The problem statement discussed in the paper is identification of human emotions from the given tweets. Posts on social media platform Twitter are known as tweets, there is a character limitation on length of words in tweets, i.e., 280 words long. It seems to be a simple task for humans, but is equally challenging for the machines. It is easier to detect emotions in speech than in written format, as in speech the way of utterance changes depending upon the emotion. Machine can understand the emotion behind a text if it has the proper knowledge required to understand the language. Words play an important part to understand which sentence carries which emotions. Although, it is difficult for machines to identify or detect emotions in text because of the use of various figures of the speeches, such as irony or sarcasm.

Emotion detection is one of the fields in NLP, in which a lot of advancement can be observed. Emotion detection is a part of sentiment analysis. In sentimental analysis, a given text is to be categorized between either positive or negative or neutral. Emotion detection is a type of sentimental analysis as specific emotion needs to be identified in the text like happy, sad, surprise, regret anger etc. It is known that in text, many times there can be more than one emotion represented, for example "Barcelona won UCL!! even though I support Liverpool, I am a huge Messi fan!". In the previous sentence it can be observed that a statement can have multiple emotions, sadness and appreciation or happiness. The dataset selected is a SemEval 2018 competition dataset which focuses on emotion detection for multiple emotions. This problem in machine learning is known as multi-label classification problem[8].

Emotion detection is considered to be one of the fields in which big companies are making progressive advancement. One of the best examples is Google. Google has used emotion detection in features like Google keyboard where if a user types a word which is an emotion, automatically the emojis related to the word is being suggested. Small businesses also use emotion detection to evaluate the product rating. Customers often use the review section to write and criticize the product. Using emotion detection the reviews can be converted to emotion and will be easy to get an overall understanding of how the product is received by the customers. Apart from emotion detection in text, emotion detection using face recognition is also in demand. Apple has made a subsequent advancement in this field and the product Animoji or Memoji. It uses facial recognition to read facial emotions and replicate into emoji.

The section 2 talks about the related work and researches which were used for this project. Section 3 talks about the dataset used in the project. Section 4 discusses about the traditional ML models and it's implementation on the dataset. Section 5 discusses about implementation of deep learning model on the SemEval dataset. Section 6 talks about comparative analysis of the results from the traditional ML models and the deep learning model. Section 7 includes the conclusion or the observation made from the result obtain. Finally, section 8 includes references used in the paper.

2. RELATED WORK

Canales et. al.[1] performed a survey on the work done in the field of emotion detection to compare and constant some of the previous work done in this area of NLP. The paper discussed lexical and machine learning approaches that has been carried out since the development of the field. In total

15 different methods and the emotions that were detected were discussed in the paper.

Gupta el. al.[3] used emotion detection to generate feedback and improve contact center responses from the emails received in the customer care of a news organization. The paper main focus was to extract some additional features known as the salient features from the dataset created after processing the emails received in customer care. Based on these salient features, the Boostexter model was implemented which improved the baseline F-1 score by 20

3. METHODOLOGY

To perform emotion detection in text certain steps are needed to be followed. Data or text that is used for predicting emotions should be cleaned. Cleaning and preparation of the data help remove stopwords and other noise in the data, such as punctuation marks etc. Tweets are converted into vectors for input to classical machine learning models.

3.1 Data Description

The dataset used for the project is from SemEval Task 2018 task 1 - Ec. The tweets are in English language and are scrapped from Twitter. The dataset consists of around 6,837 tweets. With each tweet there is an ID associated. The ID is an encrypted unique key to identify each tweet. There are 11 emotions as columns in the dataset 'anger', 'anticipation', 'disgust', 'fear', 'joy', 'love', 'optimism', 'pessimism', 'sadness', 'surprise', 'trust'. Each column consists of a binary value either 0 or 1. A row consists of a tweet and corresponding emotions. Every emotion present in the tweet contains 1 as a value in that particular column otherwise 0. The emotion describes -

3.2 Data Preprocessing and Preparation

To process the data it was stored in a pandas dataframe of python library. Pandas helps python users to store data in a tabular manner. Where the first row is converted to column name and provides several other features to process data easily. It was observed that tweets contained extra full stops and some of the words were not spelled correctly. Basically while extraction of tweets they were kept as it is and were not cleaned. To clean the dataset we used NLTK[4] library supported by python language. This library helps us clean the data so that it can be used by the models as input. Following operations were performed on the tweets present in the dataset -

- Removing stopwords - Stopwords are the words which are very frequently used in English language such as it, the, as, a etc. These words does not help in prediction of emotion. Hence we do not use these words for training of the models. Best way to handle stop word is to remove them from the tweets.

- Removing punctuations - Removal of punctuation is the process used to remove symbols from the tweets. Usually it is observed that in tweets users often use multiple symbols at once. For example multiple exclamation marks, multiple full stops etc. These symbols also do no contribute in prediction of emotions, hence the symbols are removed before training of models.

- Lemmatization - Lemmatization is the process of conversion of words to there root origin. The root word is also known as lemma. For example went will be converted to the lemma go and the process is called as lemmatization. It helps reduce the processing time of the models as lemma are more easily related to the emotions.

- Converting to lower case - All the words and character in the tweets are converted to lower case. These steps does not seems so significant but is actually is. When we train the data without converting the data to lower case, it treats similar words differently if first letter of one of the word is in upper case but other is in lower case. Hence to make sure all words which are same should be considered as single, all words are converted to lower case

3.3 Baseline Model

A baseline model is a very basic model, whose aim is to create a benchmark for other models. Baseline model helps for comparison and understanding of how well other machine learning and deep learning models are performing on the given dataset. For the given dataset the baseline model chosen is Zero rule classifier. Zero rule classifier is a very simple classifier which finds the class with the maximum occurrence and predicts the maximum class as output for any input. The problem faced in the given dataset is that it is a multi-label classification problem. The baseline accuracy would have been really low if multiple emotions were matched. Hence for output only the maximum occurred class was matched. This will set the baseline model higher and there might be a chance where other models perform worse than baseline.

3.4 Multi-Label Models

Multi-label classification is a type of classification where there are more than one class or label and a data point can belong to more than one class or label. Multi-label classification is very common in text classification. A method to implement multi-label classification is using classical classification models and using it in such a way to train on all the classes. Two of the traditional ways it is implemented is Binary Relevance and Classifier Chain.

3.4.1 Binary Relevance

Binary Relevance[5] acts as an ensemble method. First it takes a classification model, in this experimentation it is provided with Logistic Regression. Binary relevance created different models for different classes. For example if there are x classes, x different logistic regression[9] models are created and each model is trained on a different class or label. Once each model is trained, union of all models is taken for a single output. To implement Binary relevance scikit multilearn[6] library is used for python language.

3.4.2 Classifier Chain

Classifier chain[7] model is very similar to binary relevance. The only difference between the two models is that binary relevance is parallel since every model works parallely in different classes, while the classifier chain model is sequential. It also creates a model equal to the number of the classes. The first model is trained on the first class and the output provided by the first model is used as the input to the second model. This is how sequential classification is

achieved. The second model is trained on the second class or label, this is continued till all the classes are trained.

3.5 BERT model

BERT (Bidirectional Encoder Representations from Transformers)[2] model was developed by Google for mainly prediction of the next sentence to improve their search results. BERT is a pre-trained model on 112 languages. The BERT base model is used for experimentation which is a 12-layer, 768-hidden, 12-heads, 110M parameter neural network architecture. To implement a BERT simpletransformer library is used which is supported by python. This library is built on another library called transformer which is created by the famous Hugging face library in python. The simpletransformer library used bert for sequential classification and is specifically created for multi-label classification.

4. EXPLORATORY ANALYSIS

Exploratory analysis is the process of understanding the dataset and the features by plotting graphs and diagrams. It helps to understand the hidden features of the data. Sometimes analysis of data makes the processing of the data easier. Word cloud is an important concept of data mining while performing on text. Word cloud contains words which are different in size and orientation. Bigger the size of the word more frequent the word in the dataset. Smaller the size of the word less frequent the word in the dataset. To create a word cloud we used WordCloud library in python.

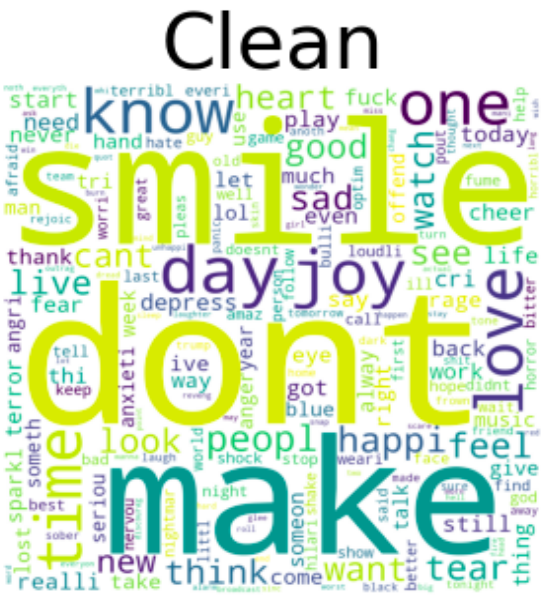

Figure 1: Word Cloud of the Dataset

In the word cloud it can be observed that the words like smile, joy and love are very frequency this could imply that many tweet have emotion corresponding to these words. Some words like make, dont, know and one are also very frequent, although these words doesn't provide too much of context when used individually.

Another analysis performed on the dataset was to identify which emotion was present the most in the dataset. This shoes that disgust, anger and joy all these are approximately present in around 2,500 tweets. Out of which disgust is present in most of the tweets.

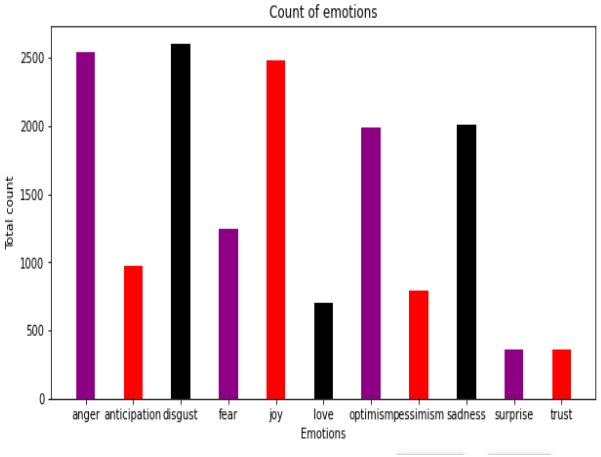

Figure 2: Emotion distribution in Dataset

5. EVALUATION AND RESULTS

Evaluation is used to measure the performance of the models on the given datasets. Since the baseline model is very basic and does not create a confusion matrix, accuracy was used to measure the performance of the baseline model. Other models were evaluated on the basis of precision, recall and F-1 score.

- Precision - Precision is known as positively predicted values. It is used to answer the question of what part of the true class were actually predicted as true.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- Recall - Recall is also known as sensitivity. It is used to answer the question what part of the class was predicted correctly and true out of all correct predictions made.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- F-1 Score - F score is a type of F measure. F score is harmonic mean of precision and recall for a given beta value. There are multiple types of F scores, when the weight of precision and recall is similar and it creates a harmonic mean that F score is known as F-1 score.

$$F - 1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

On performing experimentation on all the models discussed in methodology section, it was observed that the baseline accuracy obtained from the zero rule classifier is 38 percent.

Classifier	Accuracy
Zero classifier	38%

The metric used to evaluate rest of the models, that is, binary relevance, classifier chain and BERT is precision recall and F-1 score. The confusion matrix is being shared in the pdf attached in the folder.

Model	Precision	Recall	F-1 Score
Binary Relevance	0.61	0.35	0.42
Classifier Chain	0.57	0.38	0.44
BERT	0.77	0.50	0.61

It was observed that BERT model performed the best in all the evaluation parameters. Between classifier chain and binary relevance, binary relevance performed better in precision while in recall and F-1 score classifier chain performed better.

6. CONCLUSION AND FUTURE WORK

The problem of classification of multiple labels or classes for the dataset used is solved best by deep learning model BERT with the score of 0.61 F-1 score. It can be said that deep learning model are able to correlate text better than classical ML models, as models like BERT emphasizes on the semantics and the context in which the word is being used. For future work it would be interesting to see if any other classification model (other than linear regression) would be able to help improve the results for classical multi-label classification model.

7. REFERENCES

[1] L. Canales and P. Martínez-Barco. Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, 2014.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] N. Gupta, M. Gilbert, and G. D. Fabbrizio. Emotion detection in email customer care. *Computational Intelligence*, 29(3):489–505, 2013.

[4] E. Loper and S. Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[5] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, and A. Bahamonde. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, 1(4):303–313, 2012.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[7] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.

[8] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[9] R. E. Wright. Logistic regression. 1995.