Tweet - Eval - Based Multi-Class Tweet Classification

The tweet eval based system is used for image and emoji processing.That is totally bases on NLP. NLP stands for natural language processing.TweetEval introduces an evaluation framework consisting of seven heterogeneous Twitter-specific classification tasks.The main abstract of this how to recognize the emoji, how to predict the emoji. The natural language processing for social media is too fragmented.It means all of the emojis are used in every social media applications. every new application are released with new features and this is the main purpose of that. ***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Waterfall consists of seven heterogeneous tasks in Twitter, all framed as multi-class tweet classification. All tasks have been unified into the same benchmark, with each data set presented in the same format and with fixed training, validation and test splits.

II. 2 TweetEval: The Benchmark

In this section, we describe the compilation, Creation procedure behind the creating of TweetEval and its corresponding tasks, as well as relevant statistics and evaluation metrics.The task should be explain in below sub section.

A. Tasks

Emotion Recognition: Emotion recognition is the process of identifying human emotion. People vary widely in their accuracy at recognizing the emotions of others. Use of technology to help people with emotion recognition is a relatively nascent research area. Generally, the technology works best if it uses multiple modalities in context. To date, the most work has been conducted on automating the recognition of facial expressions from video, spoken expressions from audio, written expressions from text, and physiology as measured by wearable. for instance,Humans show a great deal of variability in their abilities to recognize emotion. A key point to keep in mind when learning about automated emotion recognition is that there are several sources of "ground truth," or truth about what the real emotion is. Suppose we are trying to recognize the emotions of Alex. One source is "what would most people say that Alex is feeling?" In this case, the 'truth' may not correspond to what Alex feels, but may correspond to what most people would say it looks like Alex feels. **Emoji Prediction** Recently, Unicode has been standardized with the penetration of social networking services, the use of emojis has become common. Emojis, as they are also known, are most effective in expressing emotions in sentences. Sentiment analysis in natural language processing manually labels emotions for sentences. The authors can predict sentiment using emoji of text posted on social media without labeling manually. The purpose of this paper is to propose a new model that learns from sentences using emojis as labels, collecting English and Japanese tweets from Twitter as the corpus. The authors verify and compare multiple models based on attention long short-term memory (LSTM) and convolutions neural networks (CNN) and Bidirectional Encoder Representations from Transformers (BERT). Emoji usage has become a new form of social communication, which is important because it can help to improve communication systems such as chat applications. This paper investigates the usage and semantics of emojis over time to analyze seasonal variation of emoji usage. In addition, the authors develop an emoji prediction model based on time information..

Emojis can be considered somehow an evolution of character-based emotions. there are 20 label of emoji prediction such as read heart, smiling face with 2 hearts,face with tear of joys,two hearts, fire and so many..

Emoji Detection. The sentiment analysis of the emojis allows us to draw several interesting conclusions. It turns out that most of the emojis are positive, especially the most popular ones. The sentiment distribution of the tweets with and without emojis is significantly different.there are several types of emojis in this, such as angry,happy,love,sad. Any system targeting the task of modeling social media communication is expected to tackle the usage of emojis. In fact, their semantic load is sufficiently rich that oversimplifying them to sentiment carriers or boosters would be to neglect the semantic richness of these ideograms, which in addition to mood ( ) include in their vocabulary references to food ( ), sports ( ), scenery ( ), etc2. In general, however, effectively predicting the emoji associated with a piece of content may help to improve different NLP tasks

Hate Speech Detection. As online content continues to grow, so does the spread of hate speech. We identify and examine challenges faced by online automatic approaches for hate speech detection in text. Among these difficulties are subtleties in language, differing definitions on what constitutes hate speech, and limitations of data availability for training and testing of these systems. Furthermore, many recent approaches suffer from an interpret ability problem—that is, it can be difficult to understand why the systems make the decisions that they do. We propose a multi-view SVM approach that achieves near state-of-the-art performance, while being simpler and producing more easily interpretative decisions than neural methods. We also discuss both technical and practical challenges that remain for this task. This task consists in predicting whether a tweet is hateful or not against any of two target communities: immigrants and women. there are two labels for this such as hateful or non- hateful.

Offensive Language Identification. Offensive language identification is classification task in natural language processing (NLP) where the aim is to moderate and minimise offensive content in social media. It has been an active area of research in both academia and industry for the past two decades. There is an increasing demand for offensive language identification on social media texts which are largely code-mixed. Code-mixing is a prevalent phenomenon in a multilingual community and the code-mixed texts are sometimes written in non-native scripts. Systems trained on monolingual data fail on code-mixed data due to the complexity of code-switching at different linguistic levels in the text. This shared task presents a new gold standard corpus for offensive language identification of code-mixed text in Dravidian languages (Tamil-English, Malayalam-English, and Kannada-English).

The goal of this task is to identify offensive language content of the code-mixed data-set of comments/posts in Dravidian Languages ((Tamil-English, Malayalam-English, and Kannada-English)) collected from social media. The comment/post may contain more than one sentence but the average sentence length of the corpora is 1. Each comment/post is annotated at the comment/post level. This data-set also has class imbalance problems depicting real-world scenarios. This task consists in identifying whether some form of offensive language is present in a tweet. there are two labels for this such as offensive and non -offensive.

III. Methodology

A. *DATA-SET*

To automatically classify tweets from Twitter of various types based on predefined categories. We took various categories into consideration for classifying twitter data. These categories are business, motorcycles, space, medicines, religion, politics, sports and technology. The data was collected from various sources as shown below: **Input Data**: The real-time data consisting of the emojis **Training data-set:** Fetched from another social media applications sets. **Final Deliverable**: It will return a list of all categories to which the input tweet belongs.

B. *MODELS AND TOOLS*

- **Neural language model:**A language model is a function, or an algorithm for learning such a function, that captures the salient statistical characteristics of the distribution of sequences of words in a natural language, typically allowing one to make probabilistic predictions of the next word given preceding ones.A neural network language model is a language model based on Neural Networks , exploiting their ability to learn distributed representations to reduce the impact of the curse of dimensional.In the context of learning algorithms, the curse of dimensional refers to the need for huge numbers of training examples when learning highly complex functions. When the number of input variables increases, the number of required examples can grow exponentially. The curse of dimensional arises when a huge number of different combinations of values of the input variables must be discriminated from each other, and the learning algorithm needs at least one example per relevant combination of values. In the context of language models, the problem comes from the huge number of possible sequences of words, e.g., with a sequence of 10 words taken from a vocabulary of 100,000 there are 1050 possible sequences...
- In this,we use three types of models which is very useful in tweet eval based system.
- Many types of deep neural networks exist, from perceptions to recurrent, convolutions and long short-term memory networks. GPT is a " transformer " model, which uses " attention " in place of previous recurrence- and convolution-based architectures.GPT — We read from left to right and so we do not know the context after 'machine', 'learning':
- BERT is a method of creating neural networks — algorithms that use layered nodes, or "neurons," to learn to perform a task through training on numerous examples. BERT is trained by repeatedly attempting to fill in words left out of a passage of writing, and its power lies in the gargantuan size of this initial training data set.
- First of all, XL Net is a BERT-like model instead of a totally different one. But it is a very promising and potential one. In one word, XL Net is a generalized auto regressive retraining method.

C. *APPROACHES*

- There are some concepts used in this proposed method to get the desired result. These are: **Outliers removal**: It is used to remove low frequent and high frequent words using Bag of words approach. **Stop words removal**: It is used to remove most common words such as "the", "is", "at", "which", and "on". **Spelling Correction**: It is used to correct spellings using EDIT DISTANCE method. **Named Entity Recognition**: It is used for ranking result category and finding the most appropriate result. **Synonym form**: It is used when a feature of test query is not found as one of dimension in feature space then replaces that word

D. *DATA-SET OF TWEET-EVAL*

We have used different data set of twitter messages for my research,which is focused on three tweet classifications. Three tweet classification is explained below.

- Hate speech detection: The definition of hate speech is neither universally accepted nor are individual facets of the definition fully agreed upon. Ross, et al. believe that a clear definition of hate speech can help the study of detecting hate speech by making annotating hate speech an easier task, and thus, making the annotations more reliable . However, the line between hate speech and appropriate free expression is blurry, making some wary to give hate speech a precise definition. For instance, the American Bar Association does not give an official definition, but instead asserts that speech that contributes to a criminal act can be punished as part of a hate crime . Similarly, we opt not to propose a specific definition, but instead examine existing definitions to gain insights into what typically constitutes hate speech and what technical challenges the definitions might bring. We summarize leading definitions of hate speech from varying sources, as well as some aspects of the definitions that make the detection of hate speech difficult.there is 2 labels of that is 1)hateful or 2) not hateful.

- Irony Detection:This paper proposes the first multilingual (French, English and Arabic) and multicultural (Indo-European languages vs. less culturally close languages) irony detection system. We employ both feature-based models and neural architectures using monolingual word representation. We compare the performance of these systems with state-of-the-art systems to identify their capabilities. We show that these monolingual models trained separately on different languages using multilingual word representation or text-based features can open the door to irony detection in languages that lack of annotated data for irony. there are 2 labels for that 1) irony and 2) not-irony
- Offensive Language Identification:Offensive language identification is classification task in natural language processing (NLP) where the aim is to moderate and minimise offensive content in social media. It has been an active area of research in both academia and industry for the past two decades. there are 2 labels for that 1) offesnsive and 2) not offensive.

here I show the all the analysis which I have done from my side.

1) **Hate Speech Detection:**

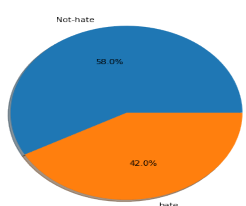


figure1

- In the figure 1, i can analyze which data is hate or not-hate.
- In this, i use pie chart for describing my data .in the figure 1 ,we can show that, i got 42.0
- In this , I used top uni gram,and top biagram,In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. ... Using Latin numerical prefixes, an n-gram of size 1 is referred to as a ”uni gram.size 2 is a ”biagram” (or, less commonly, a ”diagram”); size 3 is a ”trigram”.

  - Distribution of ratings :The ratings are in align with the polarity, that is, most of the ratings are at 4 or 5 range.
  - Reviewers age distribution: Most reviewers are in their 30s to 40s.

- Top unigrams before removing stop words : I’d very much like to have one solution that works for both unigrams and n-grams, although it would be fine to have two versions, one with a ”fixed” flag and one with a ”regex” flag. I’m putting the two aspects of the question together since someone may have a solution that tries a different approach that addresses both fixed and regular expression stop-word patterns.tokens are a list of character vectors, which may be unigrams, or n-grams concatenated by a $_(underscore)character.Stop-$ $wordsareacharactervector.RightnowIamcontenttoletthisbeafixedst$ $wordstoo.TopicModelingwithLSA$: $itmeansLSAstandsforlatentsemanticsanalysis.inthefigureshowsh$

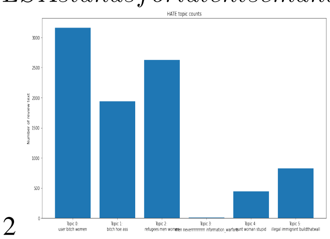


figure 2

- in the figure2,we can show that number of review text on y Axis and and topic are on the x axis. it is simply shows the how many numbers of haters are occurred
  1) Topic 0 : User Bitch Women
  2) Topic 1: Bitch hoe ass
  3) Topic 2: Refugees men women
  4) Topic 3: Never Information
  5) Topic 4: Woman Stupid
  6) Topic 5: Illegal Immigrant build that well

.

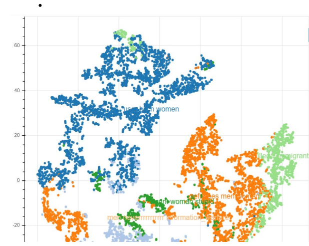


figure 3

2 . **Irony detection:**

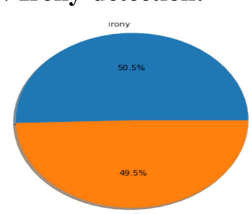


figure 4

- In the figure 1, i can analyze which data is irony or not-irony.
- In this, i use pie chart for describing my data .in the figure 4 ,we can show that, i got 50.5

- Dist plot of review polarity score:Vast majority of the polarity are greater than 0, means most of them are positive.
- Distribution of ratings :The ratings are in align with the polarity, that is, most of the ratings are at 4 or 5 range.
- Reviewers age distribution: Most reviewers are in their 30s to 40s.
- In this , i used top uni gram,and top biagram,In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. ... Using Latin numerical prefixes, an n-gram of size 1 is referred to as a ”uni gram.size 2 is a ”bi gram” (or, less commonly, a ”diagram”); size 3 is a ”triagram”.

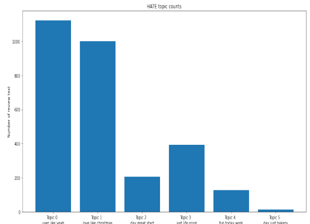


figure 5

- In this figure 5, we can show that Top 20 part-of-speech tagging of review corpus.
- In this , i use bar chart to show the how number of irony detected.
- In the figure2,we can show that number of review text on y Axis and and topic are on the x axis. it is simply shows the how many numbers of haters are occurred.
  1) Topic 0: User like yeah
  2) Topic 1: Love like Christmas
  3) Topic 2:Day great start
  4) Topic 3: Just life good
  5) Topic 4: Fun today work
  6) Topic 5: Day just bakery

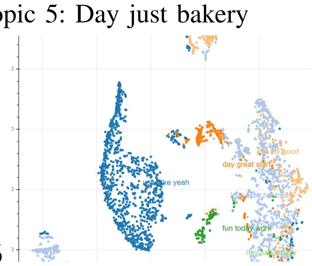
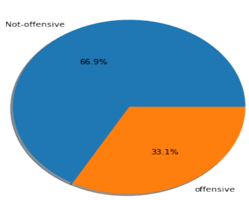


figure 6

3. Offensive Language detection:

- figure 7



-
- In the figure 1, i can analyze which data is irony or not-irony.
- In this, i use pie chart for describing my data .in the figure 4 ,we can show that, i got 33.1

- Dist plot of review polarity score:Vast majority of the polarity are greater than 0, means most of them are positive.
- Distribution of ratings :The ratings are in align with the polarity, that is, most of the ratings are at 4 or 5 range.
- Reviewers age distribution: Most reviewers are in their 30s to 40s.
- In this , i used top uni gram,and top biagram,In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. ... Using Latin numerical prefixes, an n-gram of size 1 is referred to as a ”uni gram.size 2 is a ”bi gram” (or, less commonly, a ”diagram”); size 3 is a ”trigram”.

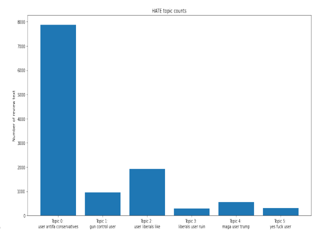
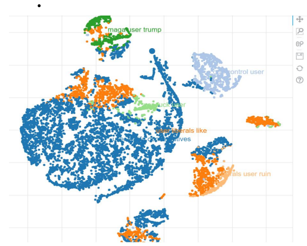


figure 8

- In this figure 5, we can show that Top 20 part-of-speech tagging of review corpus.
- In this , i use bar chart to show the how number of irony detected.
- In the figure2,we can show that number of review text on y Axis and and topic are on the x axis. it is simply shows the how many numbers of haters are occurred.
  1) Topic 0: User Antifa Conservation
  2) Topic 1: Gun control user
  3) Topic 2: User Liberals like
  4) Topic 3: Liberals user ruin
  5) Topic 4: Maga user trump
  6) Topic 5: Yes fuck User

.



In this figure , we can show that i create graph by suing clustering. in this map, you can see different different topics are take a place. Here are the terms that differentiate the review text from a general English corpus. Here are the terms in review text that are most associated with Tops department:

E. Evolution

- Preprocessing: The data we have in the our data-set ,which is very high dimensional,inconsistent, and also noisy. which they will needs to very clean and and processing of that data.we extracted text from tweets by removing laters, URLs, numbers , special character, and alphabet. and also including emojis and punctuation.besides the all data, we can change all the data. and also applied identical pre-processing to all the data set

IV. Practical Work

- CODING: First of all, I have imported all the required resources and methods along with that I assigned paths of the data-sets in google drive.

```python
import os
import numpy as np
import pandas as pd
import torch
import transformers
import json

from tqdm import tqdm
from pathlib import Path
from torch.utils.data import DataLoader,Dataset
from torch import cuda
device = 'cuda' if cuda.is_available() else 'cpu'
print(device)
```

- Following this, I have created function called create data which will be holding data of the data-sets.
- Then after,I have listed directory of data-sets along with this variables of data-sets are also created and we have implemented robetra-base model with assigned values of batches.
- After couple of progressive steps I gave respective label numbers to the particular data-sets through data model.

```python
MAX_LEN = 256
TRAIN_BATCH_SIZE = 1
EPOCHS = 1
LEARNING_RATE = 1e-05
VAL_BATCH_SIZE = 4
TEST_BATCH_SIZE = 8
```

- Respectively , I created function to measure accuracy.
- Moreover , I defined and applied training data-sets, models to measure respective accuracy of each data-sets.
- Output file will be generated as BIN file.

```python
acc = valid(emoji_model, val_emoji_loader)
print("Accuracy on test data = %0.2f%%" % acc)


acc = valid(emotion_model, val_emotion_loader)
print("Accuracy on test data = %0.2f%%" % acc)


acc = valid(hate_model, val_hate_loader)
print("Accuracy on val data = %0.2f%%" % acc)
```

- To validate data-sets I defined valid function and passed values from the data-sets.
- I created predict function and gave all the data-sets values.

```python
emoji_predictions = list()
for i in range(len(emoji_pred)):
  for j in range(TEST_BATCH_SIZE):
    out_arr = emoji_pred[i][j]
    out_arr = out_arr.cpu().data.numpy()
    max_id = np.argmax(out_arr)
    emoji_predictions.append(max_id)
```

- At-last, I created text file with respect to all data-sets and all the values generated by predict function will be stored in that file.

```
[ ] len(emoji_predictions)
```

```
[ ] file = open('emoji_pred.txt','w')
    for i in emoji_predictions:
      file.write(str(i))
      print("\n")
    file.close()
```

ACKNOWLEDGMENT

– A project or a thesis requires guidance and help from a lot of people, and showing gratitude and appreciating their help in form of a letter or an article is known to be an Acknowledgment. Many might still be wondering about what does acknowledgment means. It is nothing but an act of acceptance of something or someone.

REFERENCES

1) Sara Rosenthal, Noura Farra, and Preslav Nakov. 2019. Semeval-2017 task 4: Sentiment ana
2) David Vilares, Miguel A Alonso, and Carlos Gomez- ´ Rodr´ıguez. 2016. En-es-cs: An english-spanish code-switching twitter corpus for multilingual sentiment analysis. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 4149–4153.
3) Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In Proceedings of the conference on empirical methods in natural language processing, pages 1524–1534. Association for Computational Linguistics

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.