COMP90049 Project 2 Report

Which emoji is missing?

1. Introduction

The aim of this report is to express the knowledge gained after critically analyzing the effectiveness of few supervised Machine Learning methods on the problem of determining which emoji was used in a tweet, while reinforcing concepts in data mining. The Supervised Machine Learning Algorithms used here are Naïve Bayes and Random Forest.

2. Dataset

The Data (Jeremy Nicholson and Rao Kotagiri, 2018) is collected by sending rate-limited queries through the Twitter API2, and further filtered by removing tweets containing two or more of the 10 queried emoji's and also tweets not containing the emoji in its visible text along with some other special characters, giving a dataset which consists of only one of the 10 selected frequent emoji's. This Dataset is further shuffled, and randomly assigned to training, development and test sets.

2.1 Training Dataset:

37143 Tweets are used as training data, out of which the distribution of the tweets over the emoji's (classes) are formulated as a table as shown in Table 2.1,

Emoji	No. of Instances in this class for Training Dataset	No. of Instances in this class for Development Dataset
Upside	5024	1624
Cry	4820	1587
Neutral	4580	1496
Think	4400	1420
Explode	4118	1334
Clap	3786	1265
Shrug	3756	1222
Hands	3744	1322
FacePalm	1517	433
Disappoint	1398	461

Table 2.1: No. of Instances in a selected class

2.2 Development Dataset:

12164 Tweets are used in the development data, out of which the distribution of the tweets over the emoji's (classes) are formulated as a table as shown in Table 2.1.

2.3 Data Analysis and Feature Representations:

Tweets are in the form of text represented as a single line sentence in the datasets, we can feature engineer this text of tweets as the below representations which will be later processed over supervised classification algorithms.

- **Most100:**
  A list of 100 tokens from the tweets which appear with the greatest frequency in the training collection are generated and extracted and each tweet is represented as 100 features with values 1 or 0 i.e. If that attribute(word) is in the tweet its given a value of 1 else a value of 0.

- **Top10:**
  A list of tokens from the tweets, whose presence was automatically determined to be predictive of one of more emoji classes, using two statistical methods i.e. The Mutual Information and Chi-Square is generated and each tweet is represented as features with values 1 or 0 i.e. If that attribute(word) is in the tweet its given a value of 1 else a value of 0.

- **Vector Space Model with TF-IDF (VSM):**
  Tweets are actually series of words. One way of feature representation is to use the Bag of Words model, Robertson, S.E. and Sparck Jones (1976). Here, we segment each tweet into words and count the number of times each word occurs in each document and finally assign each word an integer id. Each unique word in our dictionary will correspond to a feature (descriptive feature). Just counting the number of words in each tweet has 1 issue, it will give more weightage to longer tweets than shorter tweets. To avoid this, we can use term frequency (TF - Term Frequencies) i.e.
  TF = (word count) / (Total words in the tweet)

  Later, we can even reduce the weightage of more common words like the, is, an etc. which occurs in all document. This is called the Inverse Document Frequency. This collective measure of weight using the Term Frequency and Inverse Document Frequency is generally called as TF-IDF.

3. Evaluation Metrics

The current Dataset used is a class-imbalanced dataset. Hence, Accuracy alone cannot evaluate a model, here we use Precision, Recall and F1-score along with accuracy to evaluate the models. Throughout this paper, the following terms will be used to evaluate each classification model:

3.1 Accuracy:

Accuracy is the fraction of correct classification of tweets predicted by the model into one of the 10 respective emoji's (classes).

$$\text{Accuracy} = \frac{\text{No. of Correct Predictions}}{\text{Total No. of Predictions}}$$

3.2 Precision:

Precision deals with "How many of the predicted emoji are correct?". It is the ratio of the total number of True Positives made and the total number of predictions made for that emoji i.e. True Positive's + False Positives.

$$\text{Precision} = TP / (TP+FP)$$

3.3 Recall:

Recall deals with "How many relevant correct predictions of emoji are made?". Recall is the ratio of total number of True Positives made for this emoji class and True Positives + False Negatives.

$$\text{Recall} = TP / (TP+FN)$$

3.4 F1-Score:

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$\text{F1-Score} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall} + \text{Precision}}$$

4. Methodology and Evaluation

4.1 Naive Bayes Classifier

Naive Bayes classifiers as stated by George John and Pat Langley (1995) is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Hence, here Clap, Hands etc., are considered independent to each other.

Bayes Theorem:
Posterior = (prior * likelihood) / evidence

- **Results:**

While using Naive Bayes classifier as the classification algorithm for classifying which emoji is missing for a tweet, the below metrics were recorded for the provided development dataset as mentioned in section2.2, the following table can be formulated for three different feature representations. It is found that the accuracy of the Bag of Words model is the highest compared to the other feature representations.

Most100	30.3765
Top10	26.2414
Vector Space Model	55.8533

Table 4.1.1: Results for Emoji Prediction using Naive Bayes Classifier.

Precision, Recall and F1-Measure for all the 10 Emoji classes predicted using Naive Bayes Classifier can be tabulated with respect to the 3 different Feature(Attribute) inputs as shown below,

Emoji Class	Precision	Recall	F1Score
Clap	0.245	0.534	0.336
Cry	0.361	0.294	0.324
Disappoint	0.191	0.141	0.162
Explode	0.459	0.261	0.333
FacePalm	0.232	0.148	0.181
Hands	0.629	0.378	0.472
Neutral	0.250	0.235	0.242
Shrug	0.208	0.184	0.195
Think	0.298	0.385	0.336
Upside	0.279	0.279	0.279

Table 4.1.2: Precision, Recall and F1Score for Emoji Prediction using Most100.

Emoji Class	Precision	Recall	F1Score
Clap	0.206	0.623	0.309
Cry	0.326	0.142	0.198
Disappoint	0.441	0.178	0.253
Explode	0.751	0.289	0.418
FacePalm	0.410	0.111	0.175
Hands	0.696	0.249	0.367
Neutral	0.165	0.444	0.240
Shrug	0.285	0.106	0.154
Think	0.893	0.053	0.100
Upside	0.261	0.286	0.273

Table 4.1.3: Precision, Recall and F1Score for Emoji Prediction using Top10.

Emoji Class	Precision	Recall	F1Score
Clap	0.85	0.63	0.72

Cry	0.60	0.65	0.63
Disappoint	0.95	0.18	0.30
Explode	0.82	0.58	0.68
FacePalm	0.97	0.25	0.39
Hands	0.87	0.71	0.79
Neutral	0.48	0.41	0.44
Shrug	0.73	0.30	0.43
Think	0.69	0.53	0.60
Upside	0.30	0.81	0.44

Table 4.1.4: Precision, Recall and F1Score for Emoji Prediction using V.S.M. (Bag of Words).

4.2 Random Forest (RD):

Random forests or random decision forests as stated by Breiman L (2001) is an ensemble learning method for classifying tweets based on emoji's by constructing a multitude of decision trees at training time and outputting the emoji that is the mode of the classes of the individual trees. Random decision forests correct for decision trees habit of overfitting to their training set.

- **Results:**

While using Random Forest classifier as the classification algorithm for classifying which emoji is missing for a tweet, the below metrics were recorded for the provided datasets as mentioned in section2.2, the following table can be formulated.

Most100	47.1884
Top10	27.047
Vector Space Model	59.8487

Table 4.2.1: Accuracy for Emoji Prediction.

Precision, Recall and F1-Measure for all the 10 Emoji classes predicted using Naive Bayes Classifier can be tabulated with respect to the 3 different Feature(Attribute) inputs as shown below,

Emoji Class	Precision	Recall	F1Score
Clap	0.622	0.511	0.561
Cry	0.576	0.486	0.527
Disappoint	0.346	0.217	0.267
Explode	0.428	0.512	0.467
FacePalm	0.383	0.316	0.346
Hands	0.607	0.734	0.665
Neutral	0.410	0.374	0.391
Shrug	0.408	0.334	0.367
Think	0.530	0.471	0.499
Upside	0.343	0.490	0.403

Table 4.2.2: Precision, Recall and F1Score for Emoji Prediction using Most100.

Emoji Class	Precision	Recall	F1Score
Clap	0.685	0.158	0.257
Cry	0.374	0.177	0.240
Disappoint	0.356	0.193	0.250
Explode	0.200	0.546	0.293
FacePalm	0.282	0.132	0.180
Hands	0.291	0.663	0.404
Neutral	0.267	0.168	0.206
Shrug	0.246	0.156	0.191
Think	0.244	0.129	0.169
Upside	0.280	0.267	0.273

Table 4.2.3: Precision, Recall and F1Score for Emoji Prediction using Top10.

Emoji Class	Precision	Recall	F1Score
Clap	0.80	0.70	0.75
Cry	0.64	0.64	0.64
Disappoint	0.67	0.32	0.43
Explode	0.83	0.64	0.72
FacePalm	0.88	0.39	0.54
Hands	0.89	0.76	0.82
Neutral	0.47	0.48	0.47
Shrug	0.57	0.40	0.47
Think	0.68	0.58	0.63
Upside	0.36	0.72	0.48

Table 4.2.4: Precision, Recall and F1Score for Emoji Prediction using VSM.

5. Evaluation:

The following table can be formulated, comparing the accuracy after predicting the correct emoji.

	Most100	Top10	VSM
Naive Bayes	30.3765	26.241	55.8533
Random Forest	47.1884	27.047	59.8487

Table 5: Comparing the accuracy of classification algorithms with different feature representations.

It is clearly shown that the accuracy of Random Forest Classification with input feature representation as Vector Space Model is the

highest followed by Naïve Bayes with the same feature representation and then Random Forest with the Most100 Feature representation as input and also it is observed that the time taken to predict emoji for tweets is the highest for Random Forest with Vector Space Model as the feature representation and the least for Naïve Bayes with Most100 as the feature representation.

A bar graph can be plotted mapping the Accuracy as shown in Figure 5.1

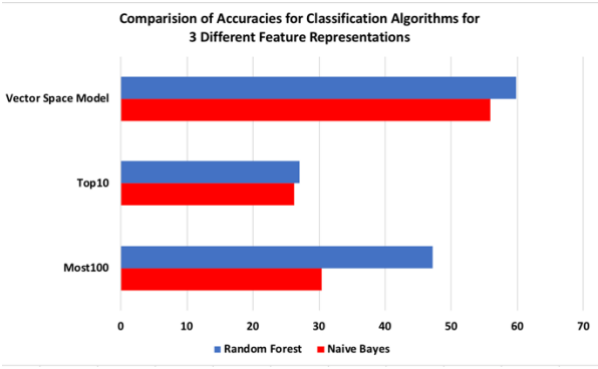

Figure 5.1: Comparing the Accuracy for classifying emoji using Random Forest and Naïve Bayes over Most100, Top10 and VSM feature representations.

6. Extension:

6.1 Removal of Stop Words:

The Removal of stop words like a, is, the etc., can be removed from the tweets and then they could be feature engineered using the Bag of Words model and fed as input to the Classifier. This removal of stop words from the inputs has increased the accuracy of Naïve Bayes from 55.85 to 56.74 and Random Forest from 59.84 to 62.538

6.2 Stemming:

Stemming as stated by Lovins(1968) is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form i.e. by reducing the words "clap", "applaud" and "clapping" to the root word, "clap". Hence, The features can further be stemmed and sent as input to the model stemming increased the accuracy of Naïve Bayes from 56.74 to 57.

7. Conclusions

This report established and presented emoji prediction for a tweet using classification methods. It is observed that since this is a multiclass classification problem, precision and recall are necessary to be considered along with accuracy and also that features in the data play a major role in predicting emoji's over the predictive models, as improvement in features was proportional to the increase in accuracy. The quality and quantity of the features will have great influence on whether the model is good or not. It is observed that the Vector Space Model (Bag of Words) with TF-IDF approach has the highest accuracy as shown in Table5.

8. References

1) Jeremy Nicholson and Rao Kotagiri (2018) COMP90049 2018S1 Project 2: Which emoji is missing? Unpublished technical report, the University of Melbourne.
2) G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. Commun. ACM 18, 11 (November 1975), 613-620.
3) Robertson, S.E. and Sparck Jones, K. 'Relevance weighting of search terms', Journal of the American Society for Information Science, 27, 1976, 129-146.
4) George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.
5) Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.
6) Breiman L (2001). "Random Forests". Machine Learning. 45 (1): 5–32.
7) Lovins, J. B. (1968); Development of a Stemming Algorithm, Mechanical Translation and Computational Linguistics, 11, 22—31