Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 03, March -2018

EMOTICON PROGNOSIS USING MACHINE LEARNING

Nirosha.C[1], Dr. S.Harihara Gopalan[2], Priyadharshini.R[3], Vidhya.J[4]

Sri Ramakrishna Engineering College, Coimbatore

Abstract— *Emojis are ideograms which are naturally combined with plain text to visually complement or condense the meaning of a message. It is an essential component in dialogues which has been broadly utilized on almost all social platforms. It could express more delicate feelings beyond plain texts and thus smooth the communications between users, making dialogue systems more anthropomorphic and vivid. In this paper, we investigate the relation between words and emojis, studying the novel task of predicting which emojis are evoked by text-based messages. We frame this investigation as a multi class text-classification problem. Some machine learning algorithms are used to train the model with the help of trained dataset. Experimental results demonstrate that our method achieves the best performances on all evaluation metrics. We conclude the paper by proposing future works in this area.*

Keywords— *Emojis, Multi Class text-classification, Machine Learning Algorithms.*

1. INTRODUCTION

According to the Oxford English Dictionary, the term emoji is Japanese coinage meaning 'pictogram', created by combining e-(picture) with moji (letter or character). Emoji as we know them were first introduced as a set of 176 pictogram available to users to Japanese mobile phones.

The advent of social media has brought along a novel way of communication where meaning is composed by combining short text messages and visual enhancements, the so-called emojis. It expresses our feelings and emotions. The bloom of Emojis has changed conventional communication schemes that only use plain texts, making the conversations between two speakers much more vivid and interesting. This visual language is as of now a standard for online communication, available in online platforms such as Facebook, Whatsapp, or Instagram. Moreover, emojis are informative and flexible that could even express some profound meanings beyond words and sentences.

In this paper, we aim to automatically recommend appropriate emojis to the text messages. It is intuitive to recommend emojis according to the reply sentences directly. More concretely, we frame our investigation as a multi-class text classification problem. The below given block diagram represents the flow of text classification process:

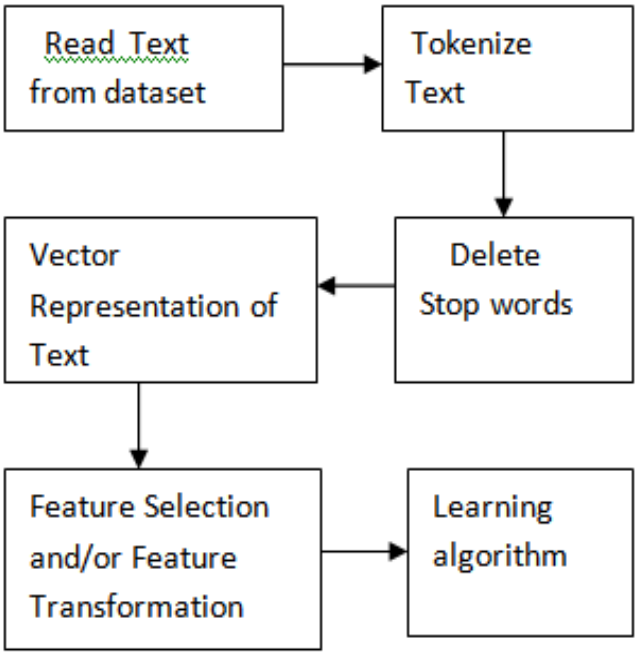

Fig.1. Text Classification Process

Given an input message with sequence of words w1, w2,…, wl, where l is the length of the message, we aim to predict an emoji class. We will evaluate our model on a separate test set and calculate various standard classification metrics such as accuracy and F1 score for evaluation. This work makes the following contributions:

- We present a large scale dataset composed of real-world emoji. This dataset as well as the training splits will be available for future researchers.
- We propose a challenge task for relating emoji to text and present state-of-the-art baseline results on these.
  TASK : Emoji prediction from the text messages.

2. DATASET AND PREPROCESSING

The dataset consists of two fields namely text messages and emojis stored in **MySQL** database . This collection consists of regular text messages and heavily distorted text messages, which were collected from various websites. Over 2000 tuples were collected. The collected dataset to be converted into a pre-processed dataset with techniques like tokenization, Stop Word Removal, Stemming.

Before starting with training we must preprocess the messages. In preprocessing, we implement some data cleaning procedure on the collected dataset. We will remove missing values in "Text messages" column, and add a column, encoding the Emoji class as an integer because categorical variables are often better represented by integers than strings. This makes the dataset to get reduced from its original size and easy to handle. Then we shall make all the character lowercase. This is because 'free' and 'FREE' mean the same and we do not want to treat them as two different words. After preprocessing, the dataset are randomly split into train set and test set. Training set consists of **80%** and testing set consists of **20%**.

3. FEATURE EXTRACTION

Text Representation:
The classifiers and learning algorithms cannot directly process the text documents in their original form, as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. Therefore, during the preprocessing step, the texts are converted to a more manageable representation.

One common approach for extracting features from text is to use the **Bag of Words Model**: a model where for each document, a text messages in our case, the presence (and often the frequency) of words is taken into consideration, but the order in which they occur is ignored. Specifically, for each term in our dataset, we will calculate a measure called Term Frequency, Inverse Document Frequency, abbreviated to TF-IDF.

- **TF-IDF**: TF-IDF stands for Term Frequency-Inverse Document Frequency. In addition to Term Frequency we compute Inverse document frequency. For example, there are two messages in the dataset. 'hello world' and 'hello foo bar'. TF('hello') is 2. IDF('hello') is log(2/2). If a word occurs a lot, it means that the word gives less information. In this model each word has a score, which is TF(w)*IDF(w).

The feature extraction techniques are used when the input data is too large and it is redundant in nature so feature is extracted to improve the Efficiency, Scalability and Accuracy.

4. EXPERIMENTS

In order to study the relation between words and emojis, we performed two different experiments. In the first experiment, we compare our machine learning models, and in the second experiment, we pick the best performing system.

4.1. MACHINE LEARNING MODELS

Machine Learning:
Machine learning field is a subfield from the broad field of artificial intelligence, this aims to make machines able to learn like human. Learning here means understood, observe and represent information about some statistical phenomenon. Machine Learning is automatically learn to make predictions on current data based on past history. It is divided into **Supervised** and **Unsupervised Learning**. Predicting a continuous quantitative Output value is referred as **Regression** Problem. Predicting a non-numerical, Qualitative value or categorical Output value is **Classification**. Observing only Input Variables and No Output variables and grouping those input variables depending on their characteristics called **Clustering.** Input variables are referred as Predictors, Independent or Features. Output variables are referred as Response or Dependent variable .

Machine Learning Algorithms:
After feature selection and transformation the messages can be easily represented in a form that can be used by a ML algorithm. Many text classifiers have been proposed using machine learning techniques. They often differ in the

approach adopted: decision trees, naïve-bayes, rule induction, neural networks, nearest neighbors, and lately, support vector machines. In this paper we have compared three algorithms namely Naïve-bayes, SVM and logistic regression.

Naïve-bayes Classifier:
Naïve-bayes classifier is simple classifier, based on Bayes Theorem of conditional probability and strong independence assumptions. It is one of the most popular and simplest methods for classification. Naive Bayesian Classifiers are highly scalable. Training of the large data simple can be easily done with Naive Bayesian Classifier, which takes a very less time as compared to other classifier. The accuracy of system increases using Naïve Bayesian Classifier. It is easier for implementation, fast to classify and more efficient. It is non sensitive to irrelevant features.

Advantages:
1. Naïve-bayes Classifier algorithm performs well when the input variables are categorical.
2. A Naïve-bayes classifier converges faster, requiring relatively little training data than other discriminative models like logistic regression, when the Naïve-bayes conditional independence assumption holds.
3. With Naïve-bayes Classifier algorithm, it is easier to predict class of the test data set. A good bet for multi class predictions as well.
4. Though it requires conditional independence assumption, Naïve-bayes Classifier has presented good performance in various application domains.

Support Vector Machine:
Support Vector Machine is a supervised machine learning algorithm for classification or regression problems where the dataset teaches SVM about the classes so that SVM can classify any new data. It works by classifying the data into different classes by finding a line (hyperplane) which separates the training data set into classes. As there are many such linear hyperplanes, SVM algorithm tries to maximize the distance between the various classes that are involved and this is referred as margin maximization. If the line that maximizes the distance between the classes is identified, the probability to generalize well to unseen data is increased.
SVM's are classified into two categories:
- Linear SVM's – In linear SVM's the training data i.e. classifiers are separated by a hyperplane.
- Non-Linear SVM's- In non-linear SVM's it is not possible to separate the training data using a hyperplane.

Advantages:
1. SVM offers best classification performance (accuracy) on the training data.
2. SVM renders more efficiency for correct classification of the future data.
3. The best thing about SVM is that it does not make any strong assumptions on data.
4. It does not over-fit the data.

Logistic Regression:
Logistic Regression machine learning algorithm is for classification tasks and not regression problems. The name 'Regression' here implies that a linear model is fit into the feature space. This algorithm applies a logistic function to a linear combination of features to predict the outcome of a categorical dependent variable based on predictor variables.
Logistic regression algorithms helps estimate the probability of falling into a specific level of the categorical dependent variable based on the given predictor variables.

Based on the nature of categorical response, logistic regression is classified into 3 types –

Binary Logistic Regression:
The most commonly used logistic regression when the categorical response has 2 possible outcomes i.e. either yes or not. Example –Predicting whether a student will pass or fail an exam, predicting whether a student will have low or high blood pressure, predicting whether a tumour is cancerous or not.

Multi-nominal Logistic Regression :
Categorical response has 3 or more possible outcomes with no ordering. Example- Predicting what kind of search engine (Yahoo, Bing, Google, and MSN) is used by majority of US citizens.

Ordinal Logistic Regression:
Categorical response has 3 or more possible outcomes with natural ordering. Example- How a customer rates the service and quality of food at a restaurant based on a scale of 1 to 10.

Advantages:

1. Easier to inspect and less complex.
2. Robust algorithm as the independent variables need not have equal variance or normal distribution.
3. These algorithms do not assume a linear relationship between the dependent and independent variables and hence can also handle non-linear effects.
4. Controls confounding and tests interaction.

4.2. PERFORMANCE COMPARISON

Applying the above mentioned Classification Algorithm for a sample Dataset which contains a multi class emoji. We summarize the performance result of the machine learning models in terms of accuracy. In term of accuracy we can find that the **Linear SVM** method is the most accurate while the **Multinomial Naïve-Bayes** and the **Logistic Regression** give us approximately the lower percentage.

Algorithm	Accuracy (%)
MultinomialNB	0.688519
LinearSVM	0.822890
Multi-nominalLR	0.792927

Table-1: Performance Measure

Results:

At last Linear Support Vector Machine has been chosen to train the model and predict the relevant emoji class for the given text messages. This experiment is done using **Scikit-Learn:** Open source library, simple and efficient, Built on NumPy, SciPy, and matplotlib. **Python Language** is used, which contains several modules such as NumPy, Pandas etc.. MySQL database is used to store the dataset which contains more than 2000 tuples consist of top 20 emoji class.

5. CONCLUSION

Emojis are used extensively in social media, however little is known about their use and semantics, especially because emojis are used differently over different communities. In this paper, we have trained the model to predict emoji from text messages. Although our initial work is promising, the models investigated still have significant room to improve. Emoji are everywhere, and are becoming only more pervasive. They already possess a distinct semantic space that can be utilized as a strong information signal as well as a novel means of interaction with data. Furthermore, their semantic richness will only increase as new emoji continue to be introduced .

We will explore the following research directions in future:

1. It is still challenging for our models to predict the emoji due to minor differences in text messages.
2. In future we are going to use more emoji class were in the above work we used only top 20 emoji class.

It is our hope that this work defined within will spur further research and understanding of emoji within the multimedia community.

6. REFERENCES

[1] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, Sung H Myaeng, Some Effective Techniques for Naive Bayes Text Classification, 2006.

[2] Korde V., Mahender C.N., Text classification and classifiers: A survey, International Journal of Artificial Intelligence & Applications 3(2) (2012).

[3] Kim S. B., Rim H. C., Yook D. S. and Lim H. S., "Effective Methods for Improving Naive Bayes Text Classifiers", LNAI 2417, 2002, pp. 414-423.

[4] Leopold, Edda & Kindermann, Jörg, "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?", Machine Learning 46, 2002, pp. 423 – 444.

[5] S.Wijeratne, L.Balasuriya, A.Sheth, and D.Doran. Emojinet: Building a machine readable sense inventory for emoji. In SocInfo,2016.

[6] Zu G., Ohyama W., Wakabayashi T., Kimura F., "Accuracy improvement of automatic text classification based on feature transformation": Proc: the 2003 ACM Symposium on Document Engineering, November 20-22, 2003, pp.118-120.

[7] Shanahan J. and Roma N., Improving SVM Text Classification Performance through Threshold Adjustment, LNAI 2837, 2003, 361- 372 [27]

[8] Soucy P. and Mineau G., "Feature Selection Strategies for Text Categorization", AI 2003, LNAI 2671, 2003, pp. 505-509.

[9] M. Bouazizi and T. Ohtsuki, ''Sentiment analysis: From binary to multiclass classification: A pattern-based approach for multi-class sentiment analysis in Twitter,'' in Proc. IEEE ICC, May 2016, pp. 1–6.