SemEval-2018: Task 1, 2 3: Emotion Recognition, Emoji Prediction and Irony Detection through Bag of Words and Supervised Machine Learning models

[*]Note: Project Final Github

Julio Ernesto Pérez Lara - 2003838
School of Computer Science and Electronic Engineering
University of Essex

***Abstract*—In this paper, a methodology to recognize emotions, predict emojis and detect irony in tweets will be described. The approach of this project is based on the classic bag-of-words model together with 2 supervised Machine Learning techniques. For the classification process, the proposed algorithms were (i) Multinomial Naïve Bayes Classifier, and (ii) Logistic Regression Classifier.**

In any event, this system will be used and evaluated in the context of the SemEval 2018 project, especially in task 1, task 2 and task 3.

I. Introduction

In the last 10 years, technology has clearly changed the way humans communicate, specifically due to social media like Twitter[1], Facebook[2], WhatsApp[3], among others. Such applications provide users with the ability to express their feelings and beliefs not only with text, but also with digital images: emojis (❤️🔥).

At present, Twitter usage is expanding speedily [1]. For example, Twitter had 330 million monthly active users in 2019 that shared positive and negative tweets that included emojis regarding a wide range of topics [2]. Consequently, as explained in [3], these comments included judgements with emoticons about companies, services, or products that could be examined to gain a general understanding of data.

Similarly, as explained by [4], emojis were first used to highlight discussions before becoming representations of particular ideas or feelings. As of today, emojis in conjunction with slang words and hashtags are used in almost every conversation in Twitter. [5].

This complex situation is emerging a new challenge for the NLP community [6] and is captivating more and more research interest. [7]. For example, the authors in [6] and [7] found that automatic emoji recognition may reveal precious information in customer service. In such cases, a customer is chatting with an automatic chatbot; empowering the algorithm with the ability to detect user's emotion is a big step to the construction of an emotionally intelligence agent capable of generate an empathetic response.

Therefore, a methodology to identify emotions, predict emojis and detect irony has been developed for the SemEval 2018 task 1, 2 and 3 [8], in which the goal is to achieve better results than the TweetEval leader-board on test set [8].

This methodology is based on the classic bag-of-words model and applying classification algorithms, such as (i) Multinomial Naïve Bayes Classifier and (ii) Logistic Regression Classifier (LRC)

The SemEval 2018-tasks consist in:

- **Task 1**: Emotion Recognition.
  This activity consists of recognizing the emotion evoked by a tweet. The dataset is called ”Affects in Tweets” [9]
- **Task 2**: Emoji Prediction.
  This activity consists in, given a tweet, predicting its most likely emoji, which is based on the emoji prediction dataset [10]
- **Task 3**: Irony Detection
  This activity consists of recognizing whether a tweet includes ironic intents or not, based on the dataset [11].

This work is organized as follows: (II) Literature review, which explains some related works. (III) Methodology, which addresses the methodology applied in the 3 tasks [8]. (IV) Results, which presents the outcome of the project. (V) Discussion, which presents the final remarks as well as future works, and (VI) Conclusion, which describes the main insights from the created models.

II. Literature review

A. Emoji Prediction

Emoji meanings are very difficult to examine in detail not only due to the subjective nature of emoji semantics, but also because it is a relative novel research problem [10]. Similarly, the process of analyzing emojis could be very arduous if we

[1] https://twitter.com/.
[2] https://www.facebook.com/.
[3] https://www.whatsapp.com//.

consider the coexistence of many possible meanings for one digital symbol[4].

For example, one emoji may be displayed differently across distinct devices or platforms [10], and perhaps, the same digital icon is used in different cultures and communities over the world who interpret the emoji individually [12]. Under these circumstances, it is quite complicated to apply traditional Natural Language Processing (NLP) techniques to analyze the small images.

Nevertheless, there certain research works that have described an emoji, either with a set of semantically close emojis, or by emoji pair co-occurrence in the same tweet [5]. For example, as described in [13], a collection of emoji definitions can enable a semantics-based measure of similarity though vector word embeddings. Similarly, more evidence of this approach can be also explained in [10], in which concludes that modeling emoji semantics via vector representations is a well-defined avenue of work.

In fact, the field of emoji vector evaluation has also experienced a notable growth as of recent. For instance, this contribution [10] mentions the introduction of: EmoTwi50 [14] and EmoSim508 [13], two new datasets with pairwise emoji similarity with human annotations.

Having considered the vector representation approach, it is also reasonable to look at two different proposals in regard to the analyze of emoji semantics. The first perspective includes models trained on Twitter data [14] [15], while the other collaborations have worked in the context of sentiment analysis; for example, in creating sentiment lexicons for emojis. [16] [17]

B. *Word Embeddings*

Originally, pictures were used instead of text as the source of emoji prediction. [18]. However, as stated by [4], several research papers focus on emoji prediction, and most of them use word embeddings in order to do multi-class emoji prediction. [6] [4] [7].

Briefly, word embeddings is nothing but the process of converting text data to numerical vectors, and it is used to capture not only the semantic of the emojis, but also their emotional content. Below are the popular and simple ”vectorization” techniques:

- Bag of words
- TF-IDF
- Word2vec
- GloVe embedding
- ELMO (Embeddings for Language models)

For more information about these methods, go to this post: Word embeddings in 2020 (Towards Data Science)

There appears to be an acceleration in the use of text vectorization techniques. For example, this paper [15] used simple embeddings techniques based on emoji description in the Unicode3 list and obtained 85% of accuracy, while the authors of this paper [19] trained a neural network model on Weibo4 to predict emojis and obtained more than 50% of accuracy in their algorithm. Similarly, this research [20] predicted 20 emojis in million of tweets using Long Short-Term Memory (LSTM) and obtained 65% of F1-Score.

C. *Twitter Sentiment Analysis*

While opinions are important because they influence people’s behavior, sentiment analysis is the process of studying people’s opinions and emotions towards institutions, events, or topics [21]. As demonstrated in [22], a Twitter Sentiment Analysis determines whether a tweet is positive, negative, or neutral, by combining Natural Language Processing (NLP) tools and Machine Learning (ML) techniques to assign a sentiment score to the entities or themes.

However, Twitter Sentiment Analysis (TSA) can be a difficult task, because each tweet, with a maximum of 280 characters, is defined by a combination of emojis, abbreviations, URLs, acronyms, hashtags, and slangs [23].

Nevertheless, several investigations during the last decade propose two big approaches to perform Twitter Sentiment Analysis: (i) Supervised Learning approach, and (ii) Unsupervised Learning approach

III. Methodology

The methodology applied in task 1, task 2 and task 3 has been organized logically according to the 5 stages of Machine Learning models proposed by Yufeng in [24] (Figure 1)

Therefore, this work has been divided into 5 big stages, as shown in 1, which consists of two big stages: (i) importing the mappings for each task, (ii) importing the train and test subsets for each task, (iii) preprocessing the train and test subset for each task, (iv) performing word embeddings, and (v) building 2 supervised Machine Learning models. (shown in Figure 1)

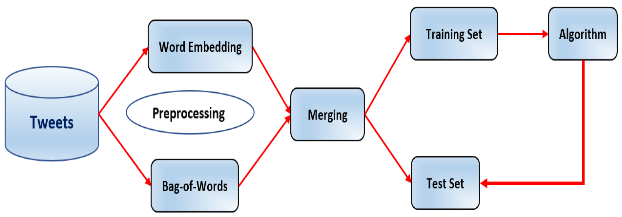

Fig. 1. Model used to solve task 1, 2 and 3

A. *Dataset*

The data for the 3 tasks consists of 46,119 tweets in English for training, 42,205 for testing and 11,329 for validating.

[4] As explained by [10], the image that represents the same emoji can vary. For example, the emoji U+1F40F, which has different rendering by platform in Unicode v11 (up to February 2021): Apple 🐏, Google 🐏, Twitter 🐏, Facebook 🐏, Samsung 🐏, Windows 🐏.

As described by [8], the tweets were retrieved with the Twitter APIs from October 2015 to February 2017 and ”geolocalized” in United States.

TABLE I
SemEval-2018 Dataset for Task 1, Task 2 and Task 3

	Dataset	*Labels*	*Training*	*Testing*	*Validating*
Task 1	Emotion Recognition	4	3257	1421	374
Task 2	Emoji Prediction	20	40,000	40,000	10,000
Task 3	Irony Detection	2	2,862	784	955

- Emotion Recognition dataset [9]
- Emoji Prediction dataset [10]
- Irony Detection dataset [11]

B. *Data preprocessing*

The second stage is about ”preprocessing the data”, which according to [25], it refers to the methods of cleaning, organizing and making ready the raw text, so that a Machine Learning model can perform better.

For this, the most common preprocessing techniques have been applied to the train and test subset:

- **Tokenization**:
  This is the process of dividing the text into single words or characters [25]
- **Lower-casing**:
  This is the process of converting a word to lower case (e.g. Texans = texans). [25]
- **User objects removal**:
  This is the process of deleting the Twitter User reference[5] (e.g. @user or @paulina_100)
- **Hashtag removal**:
  This is the process of deleting the Twitter hashtags (e.g. #SundayFunday #SanDiego)
- **Stop-words removal**:
  This is the process of dropping the very commonly used words, such as “a”, ”an”, ”the”, or ”from” [25]
- **Punctuation marks removal**:
  This is the process of dropping the very commonly used symbols, such as !?.,”() [25]
- **New line characters removal**:
  This is the process of dropping new line characters (\n)
- **Two or more spaces removal**:
  This is the process of dropping two or more spaces, such as (’   ”   ”   ’)
- **Non-ASCII characters removal**:
  This is the process of dropping symbols from other languages, such the Chinese or Russian symbols.
- **Stemming**:
  This the process of transforming a word to its root form. (e.g. excited = excit) [25]

C. *Bag of Words (BoW)*

This step consists of applying the Bag of Words (BoW) technique, which according to [27], is a representation of text that describes the occurrence of words within a document collection. In other words, the BoW technique uses word occurrence frequencies to measure the content of a tweet and see how often each words appeared.

In general, Bag of Words is a method to extract features (X) from tweets. These attributes can be used for training the Machine Learning algorithms proposed: (i) Multinomial Naïve Bayes Classifier and (ii) Logistic Regression Classifier

D. *Naïve Bayes Multinomial classifier*

The baseline system for task 2 is a Machine Learning algorithm based on the Naïve Bayes Theorem. The authors in [28] mentioned that a Naïve Bayes Classifier is one of the Machine Learning supervised classification methods that are very useful with text, since it categorizes the words that belongs to a particular class. [29]

In general speaking, this method is based on the Bayes Theorem [30] and assumes that attributes are independent, meaning that the presence of one particular feature doesn’t affect the other:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

$$P(B|A) = \frac{P(B|A) * P(A)}{P(B|A) * P(A) + P(B|A) * P(B)}$$

- P(A|B) means the Probability of A occurring given evidence B has already occurred
- P(A|B) means the Probability of B occurring given evidence A has already occurred
- P(A) means the probability of A occurring.
- P(B) means the probability of B occurring.
- Using Bayes’ Theorem, we can find the probability of A happening, given that B has occurred
- In Bayes’ Theorem B is the evidence and A is the hypothesis.
- Bayes’ Theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the even

With this technique, it is possible to calculate “the probability of a emoji occurring, given a tweet by looking through a series of examples of labels and counting how often it occurs in each class” [31]

Finally, the class given the predictors can be obtained by using the Naïve Bayes Multinomial, a variation of the Naive Bayes Classifier that considers the frequency of words [28]:

$$c(d) = \underset{c \, \in \, C}{argmax} \left[\log P(c) + \sum_{i=1}^{m} f_i \log P(W_i|c) \right]$$

- d means a tweet (text document)
- c means the class
- c(d) means the class of d (tweet)
- C means the set of all possible class labels.

[5] As defined in [26], User Objects describe the Twitter user reference. This happens when a user quotes other users tweets, replies to tweets, follows users, or is mentioned in tweets

- m means the number of different words in the collection of documents
- w_i means the ith word appearing in the document (i = 1, 2, ..., m)
- f_i means the frequency of w_i in a Tweet (document) (i = 1, 2, ..., m)
- P(c) means the prior probability of class
- P(w_i | c) means the conditional probability of each word of the document collection

Similarly, the Prior probability P(c) and the conditional probability P(w_i | c) can be calculated as [28]:

$$P(c) = \frac{\sum_{j=1}^{n} \delta(c_j, c) + 1}{n + l}$$

$$P(w_i \mid c) = \frac{\sum_{j=1}^{n} f_{ji}\ \delta(c_j, c) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji}\ \delta(c_j, c) + m}$$

- n is the number of training documents
- l means the number of classes
- c(d) means the class of d (tweet)
- f_{ji} means the frequency of w_i in the jth training document
- c_j is the class of the jth training document.
- The binary function δ (c_j , c) can be defined using the following equation:

$$\delta(c_j, c) = \begin{cases} 1, & \text{if } c_j = c \\ 0, & \text{otherwise} \end{cases}$$

E. *Logistic regression classifier*

In machine learning, logistic regression is a widely recognized classification method that is used to predict the probability of a multi-categorical dependent variable [32]. In other words, as [33] states, a logistic regression model will identify relationships between our target feature (irony/not irony), and our remaining features to apply probabilistic calculations for determining which class the twitter should belong to (Figure 2

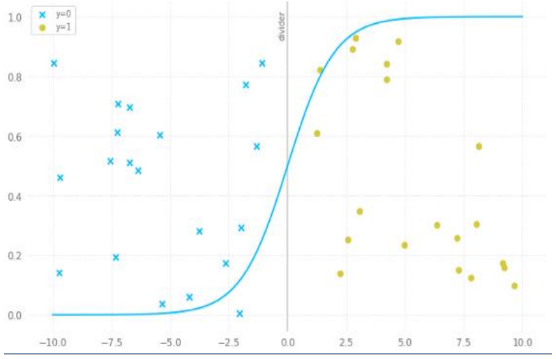

Fig. 2. Illustration of the Sigmoid function in some random data (logistic regression for binary classification). Reprinted from [34]

IV. Results and Evaluation

After training and testing the ML model, the essential model evaluation techniques for predictive multinomial and binary classification[6] have been applied to find the effectiveness of the algorithms built in task 1, 2 and 3

Confusion Matrix is a good way to demonstrate the performance of classification algorithms, specially where the number of records in each class is not equal. As defined in [30], Confusion matrix is composed of four main parts: (i) True Positives, (ii) True Negatives, (iii) False Positive and False Negative (iv).

TABLE II
Confusion Matrix

	Actual Positive	***Actual Negative***
Predicted Positive	TP	FP
Predicted Negative	FN	TN

- **TP** is the number of true positives. It means: “observations that are part of the positive class and that we predicted correctly” [30]
- **FN** is the number of false negatives (Type II error). It means “observations predicted to be part of the negative class that are actually part of the positive class” [11]. For example, predicting the emoji ❤️ when indeed is 🔥
- **FP** is the number of false positives (Type I error). It means ”observations predicted to be part of the positive class that are actually negative class”. For example, predicting positive sentiment when indeed the tweet contains a negative thought [30]
- **TN** is the number of true negatives. It means: “observations that are part of the negative class and that we predicted correctly” [30]

Precision is defined as the number of true positives over the number of true positives plus the number of false positives. It is a value between zero and one, which the higher the number, the better the accuracy. [30]

For example, precision score tell us what proportion of tweets that we predicted as negative, actually contain negative opinions.

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

Recall is defined as the number of true positives over the number of true positives plus the number of false negatives. It is a value between zero and one, which the higher the number, the better the accuracy [30]

For example, Recall tells us what proportion of tweets that actually contained negative emotions were predicted by the algorithm before.

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

For individual classes, F1 score is computed as [35]:

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

The overall classification effectiveness of the model is measured by macro-averaged F1 score: [35]

$$\mathbf{F_{macro}} = \frac{1}{k}\sum_{i=1}^{k} f_i$$

[6] As described in [30], the essential metrics and methods used for assessing the performance of predictive binary and multinomial classification models include: (i) Average Classification Accuracy, (ii) Confusion Matrix, (iii) Precision, (iv) Recall, (v) ROC Curve, and Specificity

Consequently, Table III shows the scores of the classic Precision, Recall, F1-Score, and Accuracy metrics obtained after applied the machine learning models to the test subset (unseen data).

TABLE III
RESULTS OF SEMEVAL 2018: TASK 1, 2 AND 3.

	Model	*F1 Score*	*Precision*	*Recall*	*Accuracy*
Task 1	BoW+MNB	30%	29%	33%	33%
	BoW+logr	28%	28%	32%	32%
Task 2	BoW+MNB	8%	9%	8%	8%
	BoW+logr	10%	13%	9%	13%
Task 3	BoW+MNB	47%	50%	46%	46%
	BoW+logr	56%	55%	56%	56%

This table show model's performances

As observed in Table III, the accuracy of the Multinomial Naïves Bayes classifier for task 1 is nearly 32% whereas for task 2 is 8% and for task 3 46%.
In other words, if this model analyzes 100 twitters from the whole database, it will be misclassifying 68 twitters as twitters with different emotions (e.g. predicting joy sentiment when indeed the tweet contains a sad feelings)

This is really bad if we consider that anything over 50% means the model is better than random. Therefore, we should consider either applying different ML algorithms, preprocesssing techniques, word embeddings methods, features selection or hyperparameter tuning to improve the accuracy metric.

Finally, this same analysis applies to every the rest metrics obtained in Naïves Bayes classifier and to all the scores derived from the Logistic Regression model.

V. DISCUSSION

Initially, the model presented in this project has used text vectorization to represent each word as features by performing Bag of Words technique via Scikit-learn python library[7]

Additionally, the classification process has been performed using the Multinomial Naïves Bayes classifier to attempt to match the objective of macro F1-Score proposed in the TweetEval leader-board [8]

In general, the preprocessing step has been considered as the most crucial process for the quality of this NLP model. For this reason, Future works that focus only on this essential task could be very useful to improve the quality of the next NLP models.

In addition of this, there are some limitations in the proposed methodology and there are always expected work to be done in the future, such as the following recommendations:

- Analyze the imbalanced classes and determine if is required to apply oversampling or undersampling techniques to tackle this problem. Tomek Links or SMOTE approaches can help with this.
- Build other supervised learning algorithms for multi-classification such as Support Vector Machine (SVM) or K-Nearest Neighbors.
- Apply more preprocessing techniques to the train and test subsets. For example lemmatization, POS Tagging, or regular expressions not only to detect user objects with spaces and 2 or more words.(e.g. @ Peak District), but also to detect duplicated letters at the end of the word (e.g. ahhhhhhhh, zzzzzz)
- Tune the machine learning model
- Apply advanced vectorization techniques, such as Word2vec, GloVe embedding or ELMO (Embeddings for Language models)

VI. CONCLUSION

This report presented 3 NLP-Machine Learning models to identify emotions, predict emojis and detect irony for the SemEval task 1, 2, and 3 [8] given certain text-features.

These models are based on 2 supervised learning algorithms: (i) Multinomial Naïves Bayes classifier and (ii) Logistic Regression Classifier.

The comparative experiments were carried out and provided the following results:

- A different approach to clean text data based on 12 preprocessing techniques and using mostly Pandas dataframe
- The effectiveness of the 2 ML techniques, in which the Logisitc Regression performed better in task 2 and 3 in terms of Recall and Precision Metrics.
  In fact, the best average accuracy value obtained by applying this model in task 1 33%, whereas in task 2 is 13% and 56% in task 3.
- Multinomial Naïves Bayes classifier performed almost equal to Logistic Regression in task 1
- A basic approach to people who is starting Data Science

As future works, it could be considered the idea to learn at least 2 more robust embedding methods such as word2vec or doc2vec [36], 2 more robust preprocessing techniques and Deep Neural Networks or BERT models

[7] https://scikit-learn.org.

REFERENCES

[1] J. Sampson, F. Morstatter, R. Maciejewski, and H. Liu, "Surpassing the limit: Keyword clustering to improve twitter sample coverage," in *Proceedings of the 26th ACM conference on hypertext & social media*, 2015, pp. 237–245.

[2] (2019) Q1 2019 Earnings Report" twitter. [Online]. Available: https://s22.q4cdn.com/826641620/files/doc_financials/2019/q1/Q1-2019-Earnings-Release.pdf

[3] M. Hao, C. Rohrdantz, H. Janetzko, U. Dayal, D. A. Keim, L.-E. Haug, and M.-C. Hsu, "Visual sentiment analysis on twitter data streams," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2011, pp. 277–278.

[4] G. Guibon, M. Ochs, and P. Bellot, "Lis at semeval-2018 task 2: Mixing word embeddings and bag of features for multilingual emoji prediction," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 502–506.

[5] F. Barbieri, L. Marujo, P. Karuturi, W. Brendel, and H. Saggion, “Exploring emoji usage and prediction through a temporal variation lens,” *arXiv preprint arXiv:1805.00731*, 2018.

[6] C. Huang, A. Trabelsi, and O. R. Zaïane, “Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert,” *arXiv preprint arXiv:1904.00132*, 2019.

[7] P. Du and J.-Y. Nie, “Mutux at semeval-2018 task 1: exploring impacts of context information on emotion detection,” in *Proceedings of the 12th international workshop on semantic evaluation*, 2018, pp. 345–349.

[8] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, “Tweeteval: Unified benchmark and comparative evaluation for tweet classification,” *arXiv preprint arXiv:2010.12421*, 2020.

[9] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, “Semeval-2018 task 1: Affect in tweets,” in *Proceedings of the 12th international workshop on semantic evaluation*, 2018, pp. 1–17.

[10] F. Barbieri, J. Camacho-Collados, F. Ronzano, L. E. Anke, M. Ballesteros, V. Basile, V. Patti, and H. Saggion, “Semeval 2018 task 2: Multilingual emoji prediction,” in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 24–33.

[11] C. Van Hee, E. Lefever, and V. Hoste, “Semeval-2018 task 3: Irony detection in english tweets,” in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 39–50.

[12] J. Ge and S. C. Herring, “Communicative functions of emoji sequences on sina weibo,” *First Monday*, 2018.

[13] S. Wijeratne, L. Balasuriya, A. Sheth, and D. Doran, “A semantics-based measure of emoji similarity,” in *Proceedings of the International Conference on Web Intelligence*, 2017, pp. 646–653.

[14] F. Barbieri, F. Ronzano, and H. Saggion, “What does this emoji mean? a vector space skip-gram model for twitter emojis,” in *Calzolari N, Choukri K, Declerck T, et al, editors. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016); 2016 May 23-28; Portorož, Slovenia. Paris: European Language Resources Association (ELRA); 2016. p. 3967-72.* ELRA (European Language Resources Association), 2016.

[15] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bošnjak, and S. Riedel, “emoji2vec: Learning emoji representations from their description,” *arXiv preprint arXiv:1609.08359*, 2016.

[16] D. Rodrigues, M. Prada, R. Gaspar, M. V. Garrido, and D. Lopes, “Lisbon emoji and emoticon database (leed): Norms for emoji and emoticons in seven evaluative dimensions,” *Behavior research methods*, vol. 50, no. 1, pp. 392–405, 2018.

[17] M. Kimura and M. Katsurai, “Automatic construction of an emoji sentiment lexicon,” in *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, 2017, pp. 1033–1036.

[18] S. Cappallo, T. Mensink, and C. G. Snoek, “Image2emoji: Zero-shot emoji prediction for visual media,” in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1311–1314.

[19] R. Xie, Z. Liu, R. Yan, and M. Sun, “Neural emoji recommendation in dialogue systems,” *arXiv preprint arXiv:1612.04609*, 2016.

[20] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,” *arXiv preprint arXiv:1708.00524*, 2017.

[21] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[22] E. Kouloumpis, T. Wilson, and J. Moore, “Twitter sentiment analysis: The good the bad and the omg!” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, 2011.

[23] H. M. Ismail, B. Belkhouche, and N. Zaki, “Semantic twitter sentiment analysis based on a fuzzy thesaurus,” *Soft Computing*, vol. 22, no. 18, pp. 6011–6024, 2018.

[24] G. Yufeng. (2017) The 7 Steps of Machine Learning.” towards data science. [Online]. Available: https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e

[25] J. Weng. (2019) NLP Text Preprocessing: A Practical Guide and Template.” towards data science. [Online]. Available: https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79

[26] (2021) Data dictionary: Standard v1.1” twitter developer website. [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/user#:~:text=The%20User%20object%20contains%20Twitter,can%20be%20grouped%20into%20lists.

[27] M. Nadeem, “Identifying depression on twitter,” *arXiv preprint arXiv:1607.07384*, 2016.

[28] J. Song, K. T. Kim, B. Lee, S. Kim, and H. Y. Youn, “A novel classification approach based on naïve bayes for twitter sentiment analysis,” *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 6, pp. 2996–3011, 2017.

[29] K. Suppala and N. Rao, “Sentiment analysis using naïve bayes classifier,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, pp. 264–269, 2019.

[30] C. Albon, *Machine learning with python cookbook: Practical solutions from preprocessing to deep learning*. ” O’Reilly Media, Inc.”, 2018.

[31] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, “Sentiment analysis of facebook statuses using naive bayes classifier for language learning,” in *IISA 2013*. IEEE, 2013, pp. 1–6.

[32] J. Pesantez-Narvaez, M. Guillen, and M. Alcañiz, “Predicting motor insurance claims using telematics data—xgboost versus logistic regression,” *Risks*, vol. 7, no. 2, p. 70, 2019.

[33] (2019) ”Building a Logistic Regression in Python, step by step” towards data science. [Online]. Available: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

[34] (2019) ”Binary Classification with Logistic Regression” towards data science. [Online]. Available: https://towardsdatascience.com/binary-classification-with-logistic-regression-31b5a25693c4

[35] S. Jin and T. Pedersen, “Duluth urop at semeval-2018 task 2: Multilingual emoji prediction with ensemble learning and oversampling,” *arXiv preprint arXiv:1805.10267*, 2018.

[36] M. Abdullah and S. Shaikh, “Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning,” in *Proceedings of the 12th international workshop on semantic evaluation*, 2018, pp. 350–357.