Emoji Prediction based on Twitter Data

Andrew Joseph Magri
B. Sc. I.T. in Artificial Intelligence
University of Malta
andrew.magri.18@um.edu.mt

Daniel Attard
B. Sc. I.T. in Artificial Intelligence
University of Malta
daniel.attard.18@um.edu.mt

1. INTRODUCTION

Social Media is an ever-growing platform, which offers a wide range of utility such as posting digital media, sharing personal content and in particular, communicating with other people on a day-to-day basis. This is a useful tool which helps people to integrate with each other, be it in an informal way or a more formal way. Prior to 1999, textual communication on social media was totally reliant on text-based communication. The notion of an emoticon did exist using various symbols such as ':-)' and ':-(', but unfortunately, it is hard to express certain emotions using only these symbols. Luckily, this type of communication was revolutionised in 1999 by Shigetaka Kurita, a Japanese artist who designed the first emoji set for the Japanese mobile phone operator company NTT DoCoMo [1]. After this, emojis were adopted by all communication companies, from online platforms like Facebook and Instagram, to mobile application such as SMS on smart phones, as a means of expressing emotions within the text which one writes.

It was found that an estimated 19.6% of all tweets on Twitter contain the usage of an emoji. Furthermore, 37.6% of all Twitter users make use of emojis in their tweets [2]. It can be said without a doubt, that the usage of emojis in social media is becoming more and more popular, especially with the younger and growing generations. This increasing usage of emojis, produces the need for new studies about emojis in the field of Natural Language Processing (NLP) to be performed. Despite this, research in this field is relatively limited and only covers

a very small portion of the NLP area. Therefore, through this paper, we aim to contribute further research to this domain.

In this paper, we propose classification techniques, which given a Tweet, determine which is the most appropriate emoji to used based on the content of the Tweet. The classification task being solved in this paper is a multi-class classification task, where a single Tweet can only be assigned a single label, unlike a multi-label classification task where each Tweet can be assigned an array of labels [3]. In order to see which classification methodology is best, we implement a Random Forest (RF) classifier, Support Vector Machine (SVM), and a Neural Network (NN) model, all using different techniques, to predict which emoji is fit for a specific Tweet. We then compare which of the classifiers performed best when compared to each other in order to determine which is the best framework for the task at hand. As a baseline for the results, we consider the results of a Random classifier and a Naïve Bayes classifier.

In the following sections, we will go through the procedure of implementing such a system. In Section 2, we will have a look at previous work done in the area of natural language processing paired up with the usage of emojis. In Sections 3 and 4, we describe the dataset which will be used, the pre-processing which is performed on the data, and the different models which were implemented in this paper. Finally, in Section 5 we will discuss and evaluate the results achieved by each individual model implemented.

2. LITERATURE REVIEW

In this section we will have a look at other related work which also aim to use emojis in NLP. The task at hand is a classification problem and therefore, the following research regards related work focusing on classification.

Feature Extraction

Feature extraction is a key part of each classification task. Classifiers do not allow for direct natural language input, and thus feature extraction is used to change the natural language data into numerical data which the classifiers can handle.

A technique commonly used as a feature extractor is the bag of words (BOW) [4-7]. The BOW makes use of n-grams in order to store the count of a sequence of terms. N-grams consist of a sequence of n terms, which can take different forms such as a sequence of words or a sequence of characters. This BOW is in turn used to build a TF-IDF feature matrix which assigns a higher weighting to words which are more important to a particular tweet. Words which are very common, like the word “the” is assigned a smaller weighting than a word which is used to identify a subject, such as “smile”. Using this technique, it can be concluded that the feature lies with the word “smile” and not with the word “the”.

Another feature extraction method which is often used in NLP is word embeddings [4], [8-10]. These word embedding are used to map the sequence of words onto a vector with a numerical representation which can be used as an input to the classifier. A common tool used by multiple papers [9] [11] is word2vec [12]. Other research [10] [13], suggests using pre-trained word embeddings like GloVe [14], which are word embeddings directly trained on twitter data. Eisner et al. [15], find a lack of embeddings which take emojis into consideration, and to this end, they created emoji2vec. Emoji2vec are word embeddings which also account for emojis within a given sentence.

Classifiers

There are various classifiers which can be used to classify a given input into a specific label. One of the most basic type of classifier is the Naïve Bayes Classifier. The Naïve Bayes is a mathematical model which applies the Bayes Theorem to calculate the probabilities of an

object belonging to a particular label. It assumes that different features are independent of each other, and due to this, a Naïve Bayes classifier tends to achieve worse results than other algorithms and so it is used as a baseline. Kopev et al. [13] apply the Multinomial variation of the Naïve Bayes, whilst Owusu and Beaulieu [5] apply the Bernoulli variation. Owusu and Beaulieu state that they used the Bernoulli variation as it performs better with shorter texts, in this case tweets.

Another classifier used by various papers was the Random Forest (RF) Classifier. This classifier uses an ensemble of decision trees, where each tree classifies an input to a specific label and the label which is most common is chosen to be the classification label. This model achieves better results than the Naïve Bayes classifier however, the results are still unsatisfactory. Çöltekin and Rama [7] disregard this model due to its poor results. Kopev et al. [13] achieve relatively good results (macro F1 = 0.16167) when compared with their Mulinomial Naïve Bayes classifier (macro F1 = 0.01763).

Support Vector Machines (SVMs) are another model which offer strong performance when applied to classification problems. An SVM model attempts to split the data using a hyperplane which partitions the data into its respective labels. Multiple papers achieve their best results using this model. The BOW paired with a SVM [5] [7] is an implementation of the SVM model which achieves very good results. Using this model together with a TF-IDF Vectorizer, Çöltekin and Rama [7] placed first in the SemEval-2018, achieving an F1-Score of 0.3599. A slightly different approach was taken by Kopev et al. [13], who paired the SVM with a TF-IDF Vectorizer without the use of the BOW. They achieved an F1-Score of 0.233, which is significantly worse than that of Çöltekin and Rama [7]. Inan [3] proposes an improvement on the SVM, the Sequential Minimal Optimization (SMO) model. This model splits the complex quadratic programming problem of the SVM into smaller tasks which are solved iteratively.

The final classifier explored was the Neural Network model. A lot of variations are proposed to solve such classification tasks, however, the most common was found to be the Long Short-Term Memory Network (LSTM), a variation of Recurrent Neural Networks (RNN). When building the embeddings of the input, the most common approach was that of word embeddings, however, promising results were also achieved by character-based embeddings [4], with an F1-Score of 0.34. By far, the most common approach is the LSTM and its variation the Bidirectional LSTM (Bi-LSTM) [10,11,16]. Using a Bi-LSTM allows for the model to also consider data which is not yet fed into a unidirectional LSTM and thus would have a better understanding of the overall context. A further implementation is that of including a convolutional layer to the LSTM, making it a CNN-LSTM [8]. Baziotis et al. [9] add a context-aware self-attention layer to their LSTM model. This layer is added to the end of the model to give important words a larger weighting based on the context vector of that particular word. This technique achieved great results, obtaining an F1-Score of 0.3531, and taking second overall in the SemEval-2018.

3. DATASET

The data used to train and test our systems was provided by the organisers of SemEval-2018. The English data consisted of approximately 500,000 training tweets and 50,000 testing tweets. The data collected by the provided crawler was around 110,000 tweets less for the training tweets, as since 2018 some tweets have been deleted by the users. Similarly, the final Spanish dataset consisted of 84,000 training tweets and 10,000 testing tweets. It was noted that the training set and the testing set had a slight overlap, and therefore, these duplicates were removed from the testing set.

Together with the tweets as text, another file contained the labels which correctly classify each tweet. In total, there were 20 unique labels for the English dataset, and 19 unique

labels for the Spanish dataset, each representing a different emoji. Each tweet only contained one emoji and thus a one-to-one mapping between the tweet text and the tweet label was to be ensured. Such a restriction limits the results of the implemented systems as one tweet can be classified into multiple labels. An instance of this was that out of the 20 labels, 4 of them represented a heart.

4. METHODOLOGY

In this section, the pre-processing applied to the dataset for each of the classifiers, will be discussed and the section will continue by describing the chosen classifiers. For implementation, various NLTK [17], sci-kit learn [18], and Tensorflow [19] libraries are used.

Pre-processing

Processing data before modelling the classifier to it, is essential for any machine learning algorithm and is especially so in the domain of NLP. Unlike images which are fundamentally formed by matrices of real numbers, natural language is composed of words which are combined together in different lengths. The non-standard length and non-numerical property of language need to be addressed and catered for in order to work with standard classifiers. Further to these properties, language, especially text written on social media, is cluttered with extra and non-standard words which need to be rectified.

Firstly, each tweet is tokenised using the standard NLTK TweetTokenizer function which is adapted to tokenise tweets such as eliminating user handles (@user), cropping repeated characters to a maximum of 3, where for example “looooool” is converted into “loool”, while also lower casing each word. Secondly, each word is lemmatised using the WordNetLemmatizer. Lemmatisation replaces different forms of a word into a single form and unlike stemming, takes into consideration the context the word is in. Stopwords, urls and numbers are also removed. With respect to punctuation, all punctuation was removed except

for the question mark and exclamation mark symbols. These punctuation symbols were allowed in tweets since they give an indication of the emotion conveyed by the tweet. On visual inspection of the processed tweets, groups of underscores could be noted and so these were eliminated. Hashtags had the leading '#' symbol removed and the rest of the hashtag was included as a normal word. The majority of tweets also contained location information. Both the inclusion and exclusion of this information was tested, and the results obtained are discussed in Section 7. These pre-processing techniques were a combination of techniques utilised in [9] and [13].

Baselines

Apart from comparing our results to those obtained in the SemEval-2018 competition [20], baseline metrics are also established on our set of tweets to be able to compare and contrast our state-of-the-art methods. An initial baseline was set by using a random classifier. The Naïve Bayes is a probabilistic classifier which is used as a baseline in numerous research [5], [13], [21]. As with previous research, our Naïve Bayes classifier is trained on the TF-IDF, term-document matrix.

Apart from comparing our results against each other, these baselines also serve as a minimum standard which the below classifiers should outperform.

Random Forest

Several research has used random forests for dealing with emoji prediction. Our implementation uses a TF-IDF sparse matrix as input with no further processing apart from the above stated pre-processing. Guibon, Ochs and Bellot [6] implement a RF using similar processing, whilst the use of RF was also proposed in [7], [22].

Support Vector Machine

We base our implementation of the SVM classifier on the best performing submission to the SemEval-2018 task 2 competition, Çöltekin and Rama [7]. Apart from this, an SVM classifier was chosen for its simplicity which may result in better overall performance due to less likelihood of overfitting, as may be the case with NN models. A one-vs-rest linear SVM which uses both word and character n-grams, is implemented. Adding all the n-grams together, creating a BOW of n-grams, these features are weighted by applying TF-IDF to build the term-document matrix. Different number of n-grams are used, where the character n-grams vary from 1 to 9 while word n-grams vary from 1 to 4. The C parameter for the Linear SVM was set to 0.1 as stated in [7]. No limit on vocabulary size or word frequency was included.

Neural Networks

As can be seen in Section 2, plenty of research has been done in the field of emoji classification using RNNs. Furthermore, the second-best performing submission, used a Bi-Directional LSTM neural network [9]. We based our implementation of an LSTM combined with convolutional layers NN, on the work of Owusu and Beaulieu [5]. Each tweet is converted into a sequence of integers and padded so that all tweets have an identical length of 20 tokens. An embedding matrix is populated using pre-trained GloVe embeddings [14]. The NN composes of an embedding layer, which is followed by a dropout layer to reduce overfitting. A 1D convolutional layer with an input size of 64 and a 5 by 5 kernel are used. A LSTM layer with a size of 64 units is applied before propagating the input through a dense layer containing 20 neurons, identical to the number of classes. 20 epochs, with a batch size of 32, are used since the model is observed to converge at around 15 epochs. ReLU and sigmoid activation functions are used for the convolutional layer and LSTM layer respectively. A maximum vocabulary length of 20000 words was imposed on the input.

5. EXPERIMENTS AND EVALUATION

When evaluating each classifier, the macro, precision, recall and F1 Score were calculated. The macro F1 score is the same metric used in SemEval-2018 competition which allow us to compare our results to the competition's. Table 1 and Table 2 illustrate the results obtained by the English and Spanish tweets respectively. As discussed in Section 4, both the inclusion and exclusion of location information was tested. Since the random classifier does not take into consideration the tweet itself, only 1 result is shown.

As expected, the random classifier performs poorly particularly due to the label imbalance present in the datasets which is not coherent with the equal distribution of labels from the random classifier. The Naïve Bayes classifier is also used as a baseline algorithm and obtains slightly superior results to the random classifier in terms of F1 Score. Interestingly, on the English dataset, the Naïve Bayes classifier produces the highest macro precision value, when compared to all other classifiers. This is due to a low number of false positive on the '💔' emoji. All other classifiers perform significantly better than these baselines with regards to their F1 score.

When comparing the macro F1 score of the same classifier, with and without location information, no significant difference can be observed in the English dataset for all classifiers. Despite this, when comparing the same values of the Spanish dataset, a 12% increase in macro F1 score can be observed for the NN and SVM models using location information. The increased number of tokens provided to the Spanish classifiers, having limited amount of training data may be the cause of this significant increase. This increase is not apparent in the English classifiers due to the greater amount of training data.

English	Random Classifier	Naïve Bayes (location)	Naïve Bayes (no location)	Random Forest (location)	Random Forest (no location)	LSTM + CNN (location)	LSTM + CNN (no location)	SVM (location)	SVM (no location)
Precision (Macro)	5.0	46.3	37.9	27.2	25.9	25.6	30.6	23.8	24.0
Recall (Macro)	5.0	6.9	7.9	19.3	20.1	20.5	20.8	28.7	28.8
F1 Score (Macro)	4.4	5.0	6.3	19.2	20.1	19.4	19.5	24.7	25.0

Table 1: Macro Precision, Recall and F1 Score results for various classifiers on English tweets.

Spanish	Random Classifier	Naïve Bayes (location)	Naïve Bayes (no location)	Random Forest (location)	Random Forest (no location)	LSTM + CNN (location)	LSTM + CNN (no location)	SVM (location)	SVM (no location)
Precision (Macro)	4.4	13.3	13.1	18.5	15.2	15.4	13.8	16.5	15.3
Recall (Macro)	4.9	6.6	7.1	10.8	11.7	14.7	14.1	20.8	18.8
F1 Score (Macro)	4.2	4.3	5.3	10.4	10.1	13.6	12.1	16.4	14.7

Table 2: Macro Precision, Recall and F1 Score results for various classifiers on Spanish tweets.

For both the English and Spanish datasets, the SVM classifier produced superior results. For the English dataset set, the RF and NN classifiers obtained similar results while the SVM outperformed the NN classifier by 28%. This is different than the results obtained on the Spanish dataset, where a significant increase is observed both between the RF and NN and the NN and SVM classifiers. The large increase in F1 score obtained by the English SVM when compared to the NN model, and the similar performance between the RF and NN models, can be attributed to the NN overfitting.

The SVM results shown in Table 1 and Table 2, make use of both character and word n-grams. Further testing was made to conclude whether the removal of the character n-grams heavily affected the results. A 4% and 10% increase in macro F1 score was obtained with the English and Spanish dataset respectively, when including character n-grams. Again, the greater increase in the Spanish dataset is due to the limited training data available. Similar improvements on the English dataset were obtained in [7], when including and excluding character n-grams.

When comparing our results to the results obtained by the research our implementations are based on, a significant decrease can be observed. Çöltekin and Rama [7] obtained a macro F1 score of 35.99, compared to our 25.0, while Owusu and Beaulieu [5] obtained a macro F1 score of 31.83, compared to our 19.5, both on the English dataset. A similar decrease was also observed on the Spanish dataset. Pre-processing the tweets with less alterations, resulted is inferior results. The difference in performance can therefore be attributed to the lack of training data we were able to obtain. Another possible cause may be the pre-processing steps applied to the data, but the processing performed is frequently used as mentioned in Section 4.

Figure 1 illustrates a confusion matrix of our SVM classifier on the English dataset with location information. A good number of true positives can be observed for the emojis '❤️', '😍', '💕', '🔥', '😘', '💥', '😉', '🎄'. As expected, the '❤️' emoji which constitutes to around 20% of the training data, obtains the greatest number of true positives. Similarly, the '😍' emoji makes up for 10% of the training data and also obtains a high number of true positives. On the other hand, the '😂' emoji which also makes up for 10% of the training data, only obtains a third of the true positives obtained by '😍'. This can be attributed to the use of the '😂' emoji in various contexts, where some may use it to signify joy while other use it to convey remorse. Figure 2 illustrates an example where the classifier detected the satire in the

tweet and correctly detected the emoji while Figure 3 shows an emoji conveying remorse which the classifier incorrectly predicted as the '💥' emoji.

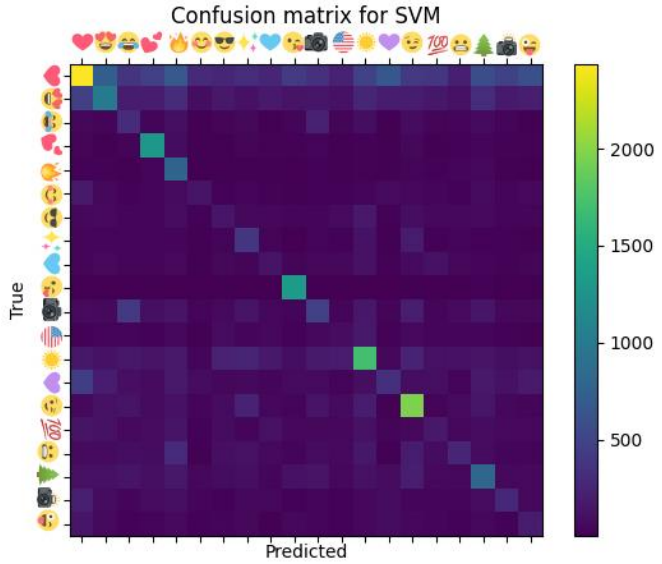

Figure 2: Confusion matrix for our best performing SVM model on the English Dataset, with location information

Original Tweet: Someone in the crowd was hilarious. 5 more shows at The Vegas Laugh Factory!! #manarms @user

Pre-processed Tweet: someone crowd wa hilarious show vega laugh factory ! ! manarms

Actual Emoji: '😂'

Predicted Emoji: '😂'

Figure 1: SVM classifier with correct detection.

Original Tweet: Can someone please call a coroner.....1st degree mayne #Repost…

Pre-processed Tweet: someone please call coroner st degree mayne repost

Actual Emoji: '😂' **Predicted Emoji:** '💥'

Figure 3: SVM classifier with incorrect detection

6. CONCLUSION

In this research, different classification techniques including a RF, a NN with LSTM and convolutional layers, and an SVM, were used to predict an emoji for a given tweet. The inclusion or exclusion of location information was also tested which was observed to produce no significant difference for the English dataset while producing a significant increase in the Spanish dataset. The SVM classifier produced superior results, obtaining a 16.4 and 25 macro F1 score on the English and Spanish datasets respectively. More training data is imperative to increase the performance of these classifiers, especially NN models which tend to overfit the training data.

7. REFERENCES

[1] K. Steinmetz, “Oxford's 2015 Word of the Year Is This Emoji,” Time, 09-Oct-2017. [Online]. Available: https://time.com/4114886/oxford-word-of-the-year-2015-emoji/. [Accessed: 28-Feb-2021].

[2] N. Ljubešić and D. Fišer, “A Global Analysis of Emoji Usage,” in Proceedings of the 10th Web as Corpus Workshop, 2016.

[3] E. Inan, “An Active Learning Based Emoji Prediction Method in Turkish”, in International Journal of Intelligent Systems and Applications in Engineering, vol. 8, no. 1, pp. 1–5, Feb. 2020

[4] F. Barbieri, M. Ballesteros, and H. Saggion, “Are emojis predictable?”, in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 105–111, Valencia Spain, 2017.

[5] J. Beaulieu and D. A. Owusu, “UMDuluth-CS8761 at SemEval-2018 Task 2: Emojis: Too many Choices?”, in Proceedings of The 12th International Workshop on Semantic Evaluation, pp. 400–404, June 2018.

[6] G. Guibon, M. Ochs, and P. Bellot, “Emoji Recommendation in Private Instant Messages”, in Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 1821–1823 New York, USA, 2018.

[7] C. Çöltekin and T. Rama, “Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs in Emoji Prediction”, in Proceedings of The 12th International Workshop on Semantic Evaluation pp. 34-38, June 2018.

[8] C. Wu, F. Wu, S. Wu, Z. Yuan, J. Liu, and Y. Huang, “THU_NGN at SemEval-2018 task 2: Residual CNN-LSTM Network with Attention for English Emoji Prediction,” in Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 410-414, New Orleans, Louisiana, June 2018.

[9] C. Baziotis, A. Nikolaos, A. Kolovou, G. Paraskevopoulos, N. Ellinas, and A. Potamianos, “NTUA-SLP at SemEval-2018 Task 2: Predicting Emojis using RNNs with Context-aware Attention”, in Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 438-444, New Orleans, Louisiana, June 2018.

[10] M. Liu. “EmoNLP at SemEval-2018 Task 2: English Emoji Prediction with Gradient Boosting Regression Tree Method and Bidirectional LSTM”, in Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 390–394, New Orleans, Louisiana, June 2018.

[11] 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”, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space”, 2013.

[13] D. Kopev, A. Atanasov, D. Zlatkova, M. Hardalov and I. Koychev, “Tweety at SemEval-2018 Task 2: Predicting Emojis using Hierarchical Attention Neural Networks and Support Vector Machine”, in Proceedings of The 12th International Workshop on Semantic Evaluation, pp. 497-501, New Orleans, Louisiana, June 2018.

[14] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global Vectors for Word Representation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), vol. 14, pp. 1532–1543, 2014.

[15] B. Eisner, T. Rocktaschel, I. Augenstein, M. Bosnjak, and S. Riedel, “emoji2vec: Learning Emoji Representations from their Description,” in Proceedings of the 4th International Workshop on Natural Language Processing for Social Media, pp. 48–54, Nov. 2016.

[16] F. Barbieri, L. E. Anke, J. Camacho-Collados, S. Schockaert, and H. Saggion, “Interpretable Emoji Prediction via Label-Wise Attention LSTMs”, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4766–4771, 2018.

[17] “Natural Language Toolkit,” Natural Language Toolkit - NLTK 3.5 documentation. [Online]. Available: http://www.nltk.org/index.html. [Accessed: 25-Feb-2021].

[18] “Machine Learning in Python,” scikit. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 25-Feb-2021].

[19] TensorFlow. [Online]. Available: https://www.tensorflow.org/. [Accessed: 25-Feb-2021].

[20] 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, pp. 24-33, New Orleans, Louisiana, June 2018.

[21] A. Basile and K. W. Lino, “TAJJEB at SemEval-2018 Task 2: Traditional Approaches Just Do the Job with Emoji Prediction”, in Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 467–473, New Orleans, Louisiana, June 2018.

[22] S. Jin and T. Pedersen, “Duluth UROP at SemEval-2018 Task 2: Multilingual Emoji Rrediction with Ensemble Learning and Oversampling”, in Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 482–485, New Orleans, Louisiana, June 2018.