

Multilingual Emoji Prediction

LIN3011

Sean Diacono

0408600L



Faculty of ICT

University of Malta

Contents

1	Introduction	1
2	Literature Review	2
3	Data & Methodology	3
3.1	Data	3
3.2	Text Pre-Processing and Vectorization	3
3.3	LSTM Implementation	4
3.3.1	Pre-Trained Word Embeddings	5
4	Results	5
4.1	Quantitative Analysis	5
4.2	Qualitative Analysis	7
4.2.1	English	7
4.2.2	Spanish	9
5	Conclusion	9

1 Introduction

Emojis are ideograms and smileys which are used in electronic text, they are an improvement over the character-based emoticons. Various types of Emojis exist, including faces (😄), objects (🚗), animals (🐱) and other symbols (🇫🇷). These emojis are usually used to convey moods, foods, actions, places, etc. therefore, they may be useful when performing sentiment analysis on some text. Figure 1 shows an example of a Tweet with an emoji which matches the meaning of the text.

Fun in the sun 🌞 @ Brownstone Park , Portland , CT
--

Figure 1: Example Tweet with an Emoji

The shared SemEval 2018 Task 2, proposed by Barbieri et al. [1], is to predict appropriate emojis for Tweets in English and Spanish, the different emojis taken into consideration may be seen in Figure 2. By being able to predict an emoji from a Tweet it would then be possible to extract the sentiment of that Tweet [2]. For example if the predicted emoji for a certain piece of text is a red heart (❤️) then it may be said that the sentiment of that text is more likely to be positive than negative.

English																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
❤️	😄	😄	😄	💖	🔥	😄	😄	🌟	💙	😄	📷	🇺🇸	🌞	💖	😄	100	😄	🎄	📷

Spanish																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
❤️	😄	😄	💖	😄	😄	💪	😄	👉	🇪🇸	😄	💙	💖	😄	💖	🌟	🎵	💖	😄	.

Figure 2: The Emojis considered for this task

In this artifact an approach for predicting emojis from Tweets will be implemented and discussed. This approach will take the form of a Bidirectional Long Short Term Memory (LSTM) network. The result of this model will then be compared to the baseline set by Barbieri et al. and the other results of the competition.

2 Literature Review

Barbieri et al. summarised the significant solutions submitted to the competition in [1]. The baseline set for the task was a classifier based on *FastText*, with F1 scores of 30.98 and 16.72 being set for English and Spanish respectively.

The top ranking submission for both the English and Spanish scores was *Tübingen-Oslo* by Çöltekin et al. [3] in which they propose two solutions to the problem and compare the results. One solution made use of a classical SVM approach with bag of n-gram features while the other solution was a Recurrent Neural Network (RNN) with word and character sequences as input. Their best solution was the SVM approach which gave F1 scores of 35.99 and 22.36 for English and Spanish respectively.

The second best solution for the English dataset was *NTUA-SLP* by Baziotis et al. [4] which made use of a Bidirectional LSTM (Bi-LSTM) with pre-trained word2vec vectors. Another notable system that was submitted for the English dataset was *EmoNLP* by Liu [5] which was based on two machine learning algorithms, a Gradient Boosting Regression Tree Method (GBM) with a Bi-LSTM using word and character n-grams.

Hatching Chick by Coster et al. [6] ranked second best for the Spanish dataset, the system made use of an SVM with word and character n-grams. However, this solution did not perform better than the baseline for the English dataset. An interesting system proposed for the Spanish dataset was *Duluth UROP* [7] which made use of several machine learning algorithms such as Naive Bayes, Logistic Regression, Random Forests and some others and then combined the results of all the algorithms using a soft voting ensemble approach.

Some elements of the aforementioned systems, such as using pre-trained word vectors, were taken into consideration when developing the system for this artifact.

3 Data & Methodology

Text pre-processing and implementation of the models was done using Python and Jupyter Notebooks. Some important Python libraries used were Scikit-Learn, Keras and NLTK.

3.1 Data

The dataset made available by Barbieri et al. [1] was used for this task. The dataset readily included the Trial (Validation) and Test data for both languages. Due to Twitter restrictions the Training data had to be downloaded using a Java executable provided by Barbieri et al. which crawls Twitter to download the tweets. The Trial and Test data included 50k tweets each for English and 10k tweets each for Spanish. The Train data was supposed to have 500k tweets in English and 100k tweets in Spanish, however, some tweets have been deleted from Twitter since the beginning of the task, so the actual number of tweets downloaded differs from the official dataset. Table 1 shows the number of tweets for Trial, Training, and Test for each language.

	Trial	Train	Test
English	50,000	405,272	50,000
Spanish	50,000	83,975	50,000

Table 1: Number of Tweets in the Dataset

3.2 Text Pre-Processing and Vectorization

Before training any models the tweets were pre-processed and vectorized. Several pre-processing techniques were tested to find the best combination of methods. First the characters in the tweets were all set to lowercase and any leading and trailing spaces were removed. Then 3 boolean variables were created, these variables are used to indicate which text pre-processing techniques to perform on the tweets. The first variable provides the

option to perform word lemmatization, which groups together different forms of a word into a single word, this was added to reduce the size of the vocabulary and to reduce the variation between tweets. The second variable provides the option to remove stop words from the tweets, stop words are commonly used words, such as ‘the’, ‘this’, and ‘that’, which usually do not have any use when trying to infer meaning from text, therefore, they can sometimes be ignored. Finally the third variable provides the choice of removing punctuation marks, this was created to try and reduce noise from the tweets such as hashtags and at signs. The combination of techniques which gave the best results for the models, was found to be not doing any pre-processing besides making all characters in the tweets lowercase.

After the tweets are processed they are then vectorized to be able to be used in the model. A tokenizer was created using Keras’ functions, which converts text into a sequence of integers. The sequences were also padded so that all the vectors were of the same length.

3.3 LSTM Implementation

A Bidirectional LSTM model was implemented using Keras, the model architecture may be seen in Figure 3.

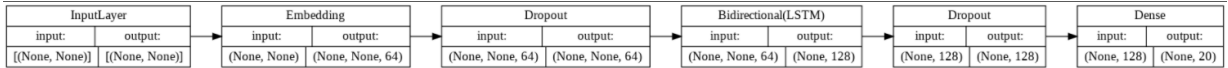


Figure 3: LSTM Model Architecture

First the embedding layer learns embeddings for the text fed to it, it is then followed by a dropout with a rate of 0.2. The second layer is a Bidirectional LSTM layer which learns from the data from beginning to the end and vice versa. A Unidirectional LSTM was tested too however, the bidirectional variant was found to be more accurate. A dropout with a rate of 0.2 follows the LSTM layer before the final layer which is a dense softmax layer with a neuron for every emoji class.

The Adam algorithm was used as an optimizer and Sparse Categorical Cross Entropy as a loss function. The model was then trained on the train data and the trial data was used as validation. The ideal number of training epochs was found to be 3, as this provided the highest accuracy without overfitting on the training data. The results of the LSTM will be discussed in the next section

3.3.1 Pre-Trained Word Embeddings

Pre-trained Global Vectors for Word Representation (GloVe) [8] for English were downloaded to investigate their affect on the results of the LSTM model. An embedding matrix was created and used in the Embedding layer. The embedding layer’s *trainable* attribute was set to False so as not to change the embeddings further during training. However, these embeddings did not improve the results of the LSTM model.

4 Results

In this section the results the LSTM model will be discussed. To evaluate the result the official evaluation script of the competition was used, this script calculates a Macro F-score. This Macro F-score is calculated as an average of the F-scores of all the individual emoji classes, the purpose of this is to score systems which overfit on the most common emojis lower and to encourage the development of systems with high F-scores for all the emojis.





4.1 Quantitative Analysis

The bidirectional LSTM model with no pre-processing applied to it besides making all characters lowercase achieved a Macro F-Score in English of **31.285** which exceeds the baseline and places the system in 7th place whilst in Spanish it achieved a score of **17.02** which also beats the baseline and places it in 5th place. These scores are respectable

especially when considering that the amount of training data available was significantly less. These scores can be seen in Table 2.

Name of System	F1	Name of System	F1
Tübingen-Oslo	35.99	Tübingen-Oslo	22.36
NTUA-SLP	35.36	Hatching Chick	18.73
hgsgnlp	34.02	UMDuluth-CS8761	18.18
EmonLP	33.67	TAJJEB	17.08
ECNU	33.35	This System	17.02
UMDuluth-CS8761	31.83	Duluth UROP	16.75
This System	31.29	Baseline	16.72
Baseline	30.98		

Table 2: F-scores for English (left) and Spanish (right)

A confusion matrix was created to clearly see the performance of all the different emojis. Figure 4a shows the confusion matrix for English, some interesting insights which may be gathered from it are the correlation between similar emojis. For example classes 1 () and 3 () which are two similar emojis see some correlation between them indicating the system's difficulty with differentiating between them. This is also clear between classes 10 () and 18 () which again are two emojis with very similar meanings.

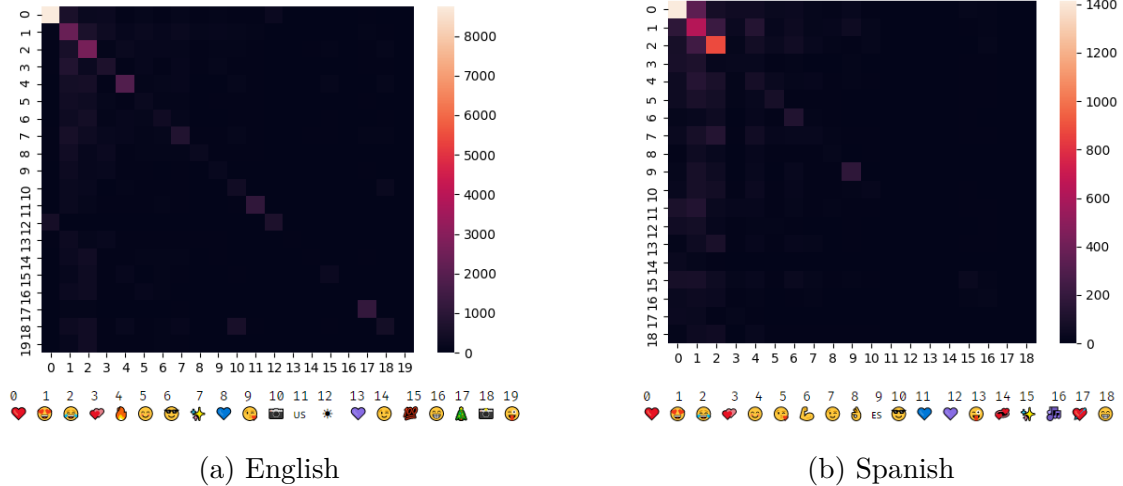


Figure 4: Confusion Matrices

Figure 4b shows the confusion matrix for Spanish and the lower accuracy of the system is clearly shown from the number of false positives shown on the plot. For example the plot shows a high number of false positives between the classes 2 (😂) and 7 (😊) which usually do not have the same meaning.

4.2 Qualitative Analysis

The model was also evaluated qualitatively by looking at the output labels predicted by the model for several tweets and comparing them to the actual labels. An interesting finding which was made was that the model rarely predicts emojis which are completely out of context and when it does make incorrect predictions the predicted emojis are usually very close to the meaning of the actual outputs.

4.2.1 English

Figure 5 shows a correct prediction by the model, interestingly although there are no direct mentions to USA in the tweet the predicted emoji is still the US flag possibly indicating that words such as *freedom* have been marked as words which are associated to the US flag emoji. This also shows the potential of such an emoji prediction system, it was able to infer that the text in the tweet was related to the USA.

The Tweet is: There is only one thing greater than freedom and that's our God..
The Predicted Emoji is: u s
The Actual Emoji is: u s

Figure 5: Correct Prediction

Similarly to the previous tweet, the tweet in Figure 6 does not explicitly mention Christmas however, the model was still able to predict the correct emoji. This possibly shows that the system associates words such as *December* with the Christmas Tree emoji.

The Tweet is: Happy December! 'Tis the most wonderful time of the year! @ The Grove
The Predicted Emoji is: 🎄
The Actual Emoji is: 🎄

Figure 6: Correct Prediction

In Figure 7 the system was able to make the correct prediction. The predicted emoji allows for conclusions to be made about the tweet. In this case the sentiment of the tweet may be said to be positive and one of love.

The Tweet is: Our Niece and Nephew wishing everyone Happy Holidays from PamLuxuryTravel #familytime ...
The Predicted Emoji is: ❤️
The Actual Emoji is: ❤️

Figure 7: Correct Prediction

Figure 8 shows an instance where the model made the incorrect prediction however, when looking at the content of the tweet it is clear that the predicted emoji is also an appropriate emoji for the tweet and not far off from the actual emoji.

The Tweet is: Photos of photos of food. Hand model: @user Join us tomorrow @user for..
The Predicted Emoji is: 🍷
The Actual Emoji is: 🍷

Figure 8: Incorrect Prediction

The prediction in Figure 9 is incorrect, nevertheless the meanings of the predicted emoji and actual emoji are very similar, so the predicted emoji would still be appropriate for that tweet.

The Tweet is: Scarlett and Summer Time And I love this Snapchat filter! I love Scarlett's Company she's...
The Predicted Emoji is: 😊
The Actual Emoji is: ❤️

Figure 9: Incorrect Prediction

4.2.2 Spanish

Figure 10 shows the system making a correct prediction for a tweet in Spanish. This prediction potentially indicates the model correctly associated the word *España* with the Spanish flag emoji.

```
The Tweet is: ESPAÑA UNA, GRANDE Y LIBRE en España  
The Predicted Emoji is: 🇪🇸  
The Actual Emoji is: 🇪🇸
```

Figure 10: Correct Prediction

In Figure 11 the model makes an incorrect prediction however similarly, to the cases in English the meaning of the predicted emoji and actual emoji are very similar.

```
The Tweet is: Que bonito eres @user  
The Predicted Emoji is: 🍷  
The Actual Emoji is: ❤️
```

Figure 11: Incorrect Prediction

5 Conclusion

In this artifact a Bidirectional LSTM approach for the task of predicting emojis from tweets in English and Spanish was presented. Various methods of text pre-processing, such as Word Lemmatization, and the use of pre-trained word embeddings were investigated. The results were satisfactory with both the English and Spanish models exceeding the baseline scores of the SemEval 2018 competition.

Future work could include expanding the number of pre-processing techniques available in the system and training the system on more languages.

References

- [1] F. Barbieri, J. Camacho-Collados, F. Ronzano, L. Espinosa-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*, (New Orleans, Louisiana), pp. 24–33, Association for Computational Linguistics, June 2018.
- [2] P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič, “Sentiment of Emojis,” *PLOS ONE*, vol. 10, p. e0144296, Dec. 2015. Publisher: Public Library of Science.
- [3] 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*, (New Orleans, Louisiana), pp. 34–38, Association for Computational Linguistics, June 2018.
- [4] C. Baziotis, N. Athanasiou, G. Paraskevopoulos, N. Ellinas, A. Kolovou, and A. Potamianos, “NTUA-SLP at SemEval-2018 Task 2: Predicting Emojis using RNNs with Context-aware Attention,” *arXiv:1804.06657 [cs]*, Apr. 2018. arXiv: 1804.06657.
- [5] 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*, (New Orleans, Louisiana), pp. 390–394, Association for Computational Linguistics, June 2018.
- [6] J. Coster, R. G. van Dalen, and N. A. J. Stierman, “Hatching Chick at SemEval-2018 Task 2: Multilingual Emoji Prediction,” in *Proceedings of The 12th International Workshop on Semantic Evaluation*, (New Orleans, Louisiana), pp. 445–448, Association for Computational Linguistics, June 2018.

- [7] S. Jin and T. Pedersen, “Duluth UROP at SemEval-2018 Task 2: Multilingual Emoji Prediction with Ensemble Learning and Oversampling,” *arXiv:1805.10267 [cs]*, May 2018. arXiv: 1805.10267.
- [8] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, 2014.