The validity of word vectors over the time for the EVALITA 2018 Emoji prediction task (ITAmoji)

Mauro Bennici
You Are My GUide
mauro@youaremyguide.com

Xileny Seijas Portocarrero
You Are My GUide
xileny@youaremyguide.com

Abstract

English. This document describes the results of our system in the evaluation campaign on the prediction of Emoji in Italian, organized in the context of EVALITA 2018[1] (Ronzano et al., 2018). Given the text of a tweet in Italian, the task is to predict the emoji most likely associated with that tweet among the 25 emojis selected by the organizers. In this report, we describe the three proposed systems for evaluation. The approach described starts from the possibility of creating two different models, one for the part of categorization, and the other for the part of polarity. And to combine the two models to get a better understanding of the dataset.

Italiano. Questo documento descrive i nostri risultati del nostro sistema nella campagna di valutazione sulla predizione delle Emoji in italiano, organizzata nel contesto di EVALITA 2018.

Dato il testo di un tweet in italiano, il task consiste nel predire l'emoji più probabilmente associata a quel tweet tra le 25 emojis selezionate dagli organizzatori. In questo report descriviamo i tre sistemi proposti per la valutazione.

L'approccio descritto parte dalla possibilità di creare due modelli diversi, uno per la parte di categorizzazione, e l'altro per la parte di polarità. E di unire i due modelli per ottenere una maggiore comprensione del dataset.

1 Introduction

In the field of communication, the importance of addressing your audience with a common language in which the customer can recognize and identify with each other is fundamental. In social interactions, an increasing amount of communication occurs in a non-verbal way, as with emoji. Being able to predict the best emoji to use in a message can increase the perception of the same and give strength to the message itself.

In the context of the Italian Emoji Prediction task called ITAmoji, we have tried to predict one of 25 possible emojis from different tweets.

Despite the knowledge of how a system of SVM could be the best solution for the problem, as per the previous context SemEval 2018 (Rama & Çöltekin, 2018), a different approach was chosen to focus on the effectiveness of a Neural Network based model

2 Description of the system

We first started by cleaning the given data from all the noise information. All the punctuation marks were removed from the text of tweets, and we focused on cleaning the text and removing ambiguities such as shortened words and abbreviations. We substituted all the hyperlinks with a more generic word "LINK" and we did the same with the usernames preceded by '@' (users' tags), after seeing that it was not relevant in the prediction of the most likely emoji for the tweet.

We tried removing the stop words from the tweets' text to leave only the words with relevant meaning in it, but the results were poor.

[1] https://sites.google.com/view/itamoji/

Then we converted every word of the tweet's text into its lemma, and while doing the lemmatization, we saw that sometimes the username was misleading in the text, so we chose to remove it and substitute it with a more generic word 'USERNAME'.

We used two different fastText[2] vectors created in the 2016 and the other created in 2017, all with Italian tweets containing at least one emojis. The idea is to analyze if different fastText vectors created with tweets published in different periods could discover the use of the emojis and its evolution over the time.

The system created is an Ensemble of two different models to replicate the result obtained in the emotion classification (Akhtar at al., 2018).
The first model is a bi-directional Long Short-Term Memory (BI-LSTM) implemented in Keras[3].

```
Layer (type)          Output Shape          Param #
===================================================
e (Embedding)      (None, 25, 200)         34978200

b (Bidirectional)  (None, 512)               935936

d (Dense)          (None, 25)                 12825
===================================================
```

A dropout and a recurrent_dropout of 0.9.
The optimizer is the RMSProp. The embedding is trainable.

The second is a LightGBM[4], where the following properties are extracted from the tweet text:

- length of the tweet
- percentage of special characters
- the number of exclamation points
- the number of question marks
- the number of words
- the number of characters
- the number of spaces
- the number of stop words
- the ratio between words and stop words
- the ratio between words and spaces
- the ratio between words and hashtags

and are joined to the vector created by the bigram and the trigram of the tweet itself at word and character level.
The number of leaves is 250, the learner set as 'Feature', and the learning rate at 0.04.

The ensemble is done in the weighted average when the BI_LSTM decide the 60% of the vote and the LightGBM the 40%.

It was also tried to add a linear classifier but the attempt did not provide any advantage. The cross-validation task to find a good weight was ineffectual and the provision was insignificant.

3 Results

The results of the Bi-LSTM were:

BI-LSTM with 2016 fastText		
precision	recall	F1 score
0.3595	0.2519	0.2715

Table 1: precision, recall, and F1 score with 2016 fastText vector.

BI-LSTM with 2017 fastText		
precision	recall	F1 score
0.3520	0.2577	0.2772

Table 2: precision, recall, and F1 score with 2017 fastText vector.

The model trained with the data published during the 2017 is quite similar to the model trained with the data published on the 2016.

The results of the LightGBM were:

[2] https://fasttext.cc
[3] https://keras.io
[4] https://github.com/Microsoft/LightGBM

LightGBM only text		
precision	recall	F1 score
0.2399	0.3094	0.2460

Table 3: precision, recall, and F1 score

The LightGBM model was also tested by adding to the already mentioned properties additional information such as the user ID and information extracted from the tweet date such as day, month, the day of the week and time.

The results obtained also indicate here that there is a correspondence between the use of emojis, the user, the time and the day. For example the Christmas tree in December or the heart emoji in the evening hours.

LightGBM with user and date		
precision	recall	F1 score
0.5044	0.2331	0.2702

Table 4: precision, recall, and F1 score

The level of Precision obtained in this way was very high even if the F1 score is still lower than the BI-LSTM model.

To avoid the unbalancing of the emojis present in the training dataset various undersampling and oversampling operations were performed without any appreciable results.
Turning to the result of the ensemble of the two models we had a marked increase in the F1 score thanks to the substantial growth of the Recall in both cases.

In the tables 5 and 6 there are the results from the minimum and the maximum F1 score obtained during the process of the ensemble.

BI-LSTM with 2016 fastText + LightGBM only text		
precision	recall	F1 score
0.4121	0.2715	0.2955

Table 5: precision, recall, and F1 score

BI-LSTM with 2017 fastText + LightGBM with user and date		
precision	recall	F1 score
0.3650	0.2917	0.3048

Table 6: precision, recall, and F1 score

The result of the validation was however very far from that obtained during the training phase. It will be necessary to evaluate if, as in the research Exploring Emoji Usage and Prediction Through a Temporal Variation Lens (Barbieri et al., 2018), it was the time of the publication of the tweets is to be distant from the date of the tweets analyzed.

If the tweets analyzed were too different from those of the training dataset, if the users in the test dataset have different behaviors, or if the system suffered from some kind of overfitting (visible in the third submission, gw2017_pe).

	gw2017_e	gw2017_p	gw2017_pe
Macro F1	0.222082	**0.232940**	0.037520
Micro F1	**0.421920**	0.400920	0.119480
Weighted F1	0.368996	**0.378105**	0.109664
Coverage error	**4.601440**	5.661600	13.489400
Accuracy at 5	**0.713000**	0.671840	0.279280
Accuracy at 10	**0.859040**	0.814880	0.430360
Accuracy at 15	**0.943080**	0.894160	0.560000
Accuracy at 20	**0.982520**	0.929920	0.662720

Table 7: macro F1, micro F1, weighted F1, coverage error, accuracy at 5, 10, 15 and 20 for the three runs submitted.

In table 8 we can observe the result of the three submissions split for each emoji.

Runs	**gw2017_e**			**gw2017_p**			**gw2017_pe**			
Label	**precision**	**recall**	**f1-score**	**precision**	**recall**	**f1-score**	**precision**	**recall**	**f1-score**	**quantity**

😁	0.2150	0.0224	0.0405	0.1395	0.0642	0.0879	0.0242	0.0107	0.0148	1028
💙	0.4429	0.1917	0.2676	0.3608	0.2075	0.2635	0.0215	0.0178	0.0195	506
😘	0.3142	0.3417	0.3274	0.2726	0.3681	0.3133	0.0343	0.0468	0.0396	834
😋	0.3624	0.3540	0.3582	0.3204	0.3850	0.3498	0.0107	0.0155	0.0127	387
😱	0.3137	0.0360	0.0646	0.1608	0.0518	0.0784	0.0077	0.0023	0.0035	444
😂	0.3533	**0.8357**	0.4967	0.4185	0.6104	0.4965	0.2024	**0.2648**	**0.2294**	4966
💪	0.3902	0.1535	0.2203	0.3257	0.2038	0.2507	0.0263	0.0264	0.0263	417
😀	0.2917	0.0554	0.0931	0.2190	0.0678	0.1035	0.0328	0.0102	0.0155	885
😅	0.0800	0.0053	0.0099	0.0581	0.0132	0.0215	0.0380	0.0079	0.0131	379
💋	0.5143	0.2581	0.3437	0.4464	0.2688	0.3356	0.0044	0.0036	0.0039	279
😭	0.3144	0.1635	0.2152	0.1895	0.2520	0.2163	0.0135	0.0134	0.0135	373
❤️	**0.7567**	0.7497	**0.7531**	**0.7803**	**0.7358**	**0.7574**	**0.2101**	0.2016	0.2058	5069
🤣	0.1714	0.0110	0.0207	0.1053	0.0183	0.0312	0.0137	0.0018	0.0032	546
🌹	0.3769	0.1849	0.2481	0.3439	0.2038	0.2559	0.0142	0.0113	0.0126	265
😍	0.3137	0.4109	0.3558	0.2952	0.4824	0.3663	0.0904	0.1583	0.1151	2363
😊	0.2384	0.1607	0.1920	0.2068	0.1747	0.1894	0.0526	0.0546	0.0536	1282
😎	0.3174	0.1043	0.1570	0.2432	0.1157	0.1568	0.0317	0.0243	0.0275	700
✨	0.4667	0.1579	0.2360	0.3239	0.1729	0.2255	0.0096	0.0075	0.0084	266
☀️	0.6735	0.3103	0.4249	0.6221	0.3354	0.4358	0.0106	0.0063	0.0079	319
🤔	0.3204	0.1220	0.1767	0.2101	0.2680	0.2356	0.0193	0.0185	0.0189	541
👍	0.4278	0.1199	0.1873	0.3043	0.1526	0.2033	0.0249	0.0171	0.0203	642
🔝	0.3220	0.0548	0.0936	0.2368	0.0778	0.1171	0.0187	0.0086	0.0118	347
💕	0.3590	0.0411	0.0737	0.2537	0.0499	0.0833	0.0161	0.0059	0.0086	341
😉	0.2082	0.1181	0.1507	0.1584	0.2451	0.1924	0.0369	0.0419	0.0392	1338
😜	0.2609	0.0248	0.0454	0.1860	0.0331	0.0562	0.0336	0.0083	0.0133	483
avg / total	**0.4071**	**0.4219**	0.3690	0.3870	0.4009	**0.3781**	0.1051	0.1195	0.1097	25000

Table 8: Precision, Recall, F1 Score, and quantity in the test set of the 25 most frequent emojis.

It is important to note that despite the significant presence of the dataset the 😁 has a meager final F1 score. On the other hand, the ☀️ has a high F1 score even if only present in 319 items.

4 Discussion

In the study of the dataset, three critical issues emerged.

- The first is that the use of similar emojis seems more dictated by a personal choice of the user.

  There are not many pieces of evidence because the use of one emoji is preferred.

  In particular for the following emoji: 😀 🤣 😂 😀
- The second is that, especially in cases where a tweet begins by indicating a USERNAME, or in a mention or a direct response, the use of emoji takes on a sub-language value. That is, the use of a specific word or emoji has a meaning that only the tweet recipients know. Use of emoji 😅 and 🤣 could be irony or just references to previous pasted experiences in common.
- Thirdly, the strong imbalance of the training dataset is not the only reason for the unbalanced prediction of some emojis, as in the case of 😀 and ☀️.

5 Conclusion

The result of the ensemble was pretty good and demonstrate the validity of this kind of approach. The use of emoji is personal and also depends on the context and the people in the discussion. A system with the emojis with the same meaning merged could be more proficient and ready for the production.

In the near future, we will evaluate the speed and effectiveness of a CNN model in which the operation of the BI-LSTM and the features extrapolation used in the LightGBM model can be merged during the same training session.

We will also focus on the creation of fastText vectors of different size containing tweets for specific contexts and published in different periods to identify the periodicity and variation in the use of particular emoji. The intent is to discover other hidden patterns, more than the obvious that has emerged for the holiday periods.

Reference

Francesco Ronzano, Francesco Barbieri, Endang Wahyu Pamungkas, Viviana Patti, and Francesca Chiusaroli (2018) ITAmoji: Overview of the Italian emoji prediction task @ Evalita 2018. In Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018), CEUR.org, Turin, Italy.

Taraka Rama and Çagri Çöltekin. (2018, June). Tübingen-Oslo at SemEval-2018 Task 2: SVMs perform better than RNNs in Emoji Prediction. Retrieved from https://aclanthology.coli.uni-saarland.de/papers/S18-1004/s18-1004

Francesco Barbieri, José Camacho-Collados, Francesco Ronzano, Luis Espinosa Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, Horacio. (2018) Saggion: SemEval 2018 Task 2: Multilingual Emoji Prediction. SemEval@NAACL-HLT 2018: 24-33. ACL.

Md Shad Akhtar, Deepanway Ghosal, Asif Ekbal, Pushpak Bhattacharyya, Sadao Kurohashi. (2018, October 15). A Multi-task Ensemble Framework for Emotion, Sentiment and Intensity Prediction. Retrieved from https://arxiv.org/abs/1808.01216

Francesco Barbieri, Luis Marujo, Pradeep Karuturi, William Brendel, Horacio Saggion. (2018, May 02). Exploring Emoji Usage and Prediction Through a Temporal Variation Lens. Retrieved from https://arxiv.org/abs/1805.00731