The Dabblers at SemEval-2018 Task 2: Multilingual Emoji Prediction

Larisa Alexa, **Alina Lorenț**,
Daniela Gîfu, **Diana Trandabăț**

Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi
Institute of Computer Science, Romanian Academy - Iasi Branch
Cognos Business Consulting S.R.L., 32 Bd. Regina Maria, Bucharest, Romania
{larisa.alexa04, alina.lorent}@gmail.com, {daniela.gifu, dtrandabat}@info.uaic.ro

Abstract

The “Multilingual Emoji Prediction” task focuses on the ability of predicting the correspondent emoji for a certain tweet. In this paper, we investigate the relation between words and emojis. In order to do that, we used supervised machine learning (Naive Bayes) and deep learning (Recursive Neural Network).

Keywords
Virtual, emojis, human sentiment, understanding of emojis.

1. Introduction

In the last few years, Social Media has evolved very fast, becoming at the moment a very important part of our daily life. There are several social networking platforms such as Twitter, Facebook, Instagram, WhatsApp, which were created in order to allow us to communicate with each other, to share our feelings or opinions related to different topics. Despite their differences or their purposes, each of these platforms shares one aspect: the use of emojis. From facial expressions to animals, objects or places, they are all used to communicate simple things or to enhance feelings and emotions. Due to the fact that their meaning is not always the same, processing emojis remains a challenge for the NLP researchers. From a language to another, for different cultures or depending on the user’s sentiments, emojis meaning can vary a lot. Understanding their meaning depending on the context of use has a huge relevance in multiple fields, like: human computer interaction, multimedia retrieval, etc.

Twitter Emojis are pictures usually combined with text in order to emphasize the meaning of that text. Although these pictures are the same all over the world, they can be interpreted and used in different ways, depending on culture differences. Despite their widely usage in social media, their underlying semantics have received little attention from a Natural Language Processing standpoint.

2. Related work

Over the past few years, there has been an increased public and enterprise interest in social media. Therefore, analyzing emojis has become an important aspect for NLP researchers, because their meaning has remained for the time unexplored.

Go et al. [9] and Castellucci *et al.* [6] used in their papers distant supervision over emotion-labeled textual contents in order to train a sentiment classier and to build a polarity lexicon. Aoki *et al.* [1] described in his research a methodology to represent each emoticon as a vector of emotions, while Jiang [10] proposed a sentiment and emotion classier based on semantic spaces of emojis in the Chinese Website Sina Weibo. In his research, Cappallo *et al.* [5] proposed a multimodal approach for generating emoji labels for images (Image2Emoji). Boia *et al.* (2013) [4] analyzed sentiment lexicons generated by considering emoticons, showing that in many cases they do

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 405–409
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

not outperform lexicons created only with textual features.

Barbieri et al. [2] tried to predict the most likely emoji a Twitter message evokes. They used a model based on Bidirectional Long Short-term Memory Networks (BLSTMs) with standard lookup, word representations and character-based representation of tokens. For the word representations they replaced each word that occurred only once in the training data with a fixed representation (out- of-vocabulary words vector) (similar to the treatment of word embeddings by Dyer et al. (2015)). For the character-based representations, they computed character- based continuous-space vector embeddings (Ling et al., 2015b; Ballesteros et al., 2015) of the tokens in each tweet, using bidirectional LSTMs.

3. Data Set and Methods

In this Section, we present the data set format and the architecture we used to predict emojis. We implemented two main modules: first one is based on a Recurrent Neural Network (3.3.1) and the second one implements Naïve Bayes algorithm (**Error! Reference source not found.**).

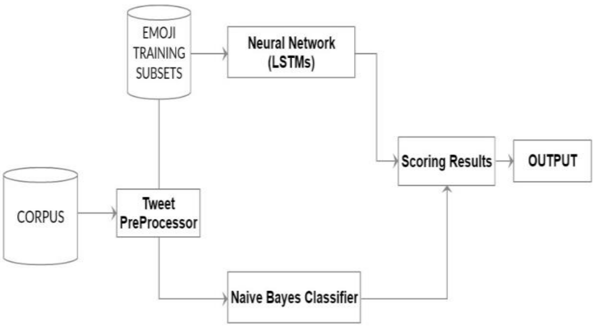


Figure 1. Architectural Diagram

3.1 Data Set

The corpus is formed from 500k tweets in English and 100K tweets in Spanish. The tweets were retrieved with the Twitter APIs, from October 2015 to February 2017, from United States and Spain. The dataset includes tweets that contain one and only one emoji from the 20 most frequent emojis. Data was split into Training Data (80%), Trial Data (10%) and Test Data (10%).

Data set is related to the 20 most frequent emoji of each language.

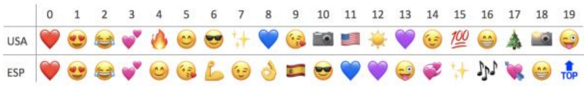

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
USA
ESP

Figure 2. Emoji labels distribution

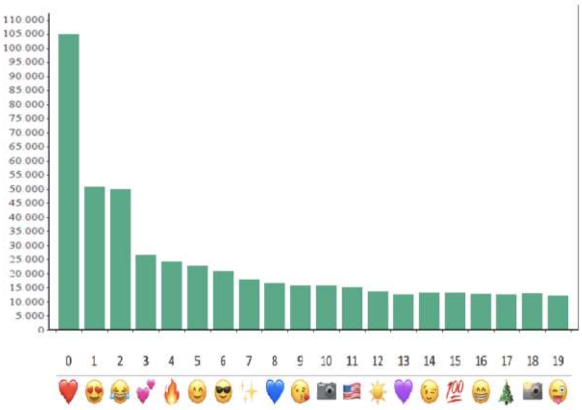
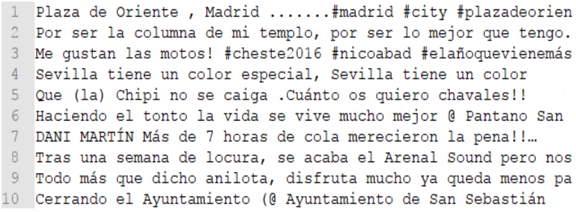


Figure 3. Label frequency in training dataset

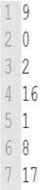
In order to generate the training data, we used the tools given by organizers: a crawler for extracting the tweets and an emoji extractor.

For each language, data is represented through two files: one file containing one tweet per line and the other file containing the corresponded emoji label.



```
Plaza de Oriente , Madrid .......#madrid #city #plazadeorien
Por ser la columna de mi templo, por ser lo mejor que tengo.
Me gustan las motos! #cheste2016 #nicoabad #elañoquevienemás
Sevilla tiene un color especial, Sevilla tiene un color
Que (la) Chipi no se caiga .Cuánto os quiero chavales!!
Haciendo el tonto la vida se vive mucho mejor @ Pantano San
DANI MARTÍN Más de 7 horas de cola merecieron la pena!!…
Tras una semana de locura, se acaba el Arenal Sound pero nos
Todo más que dicho anilota, disfruta mucho ya queda menos pa
Cerrando el Ayuntamiento (@ Ayuntamiento de San Sebastián
```

Figure 4. Corpus-tweets file



```
9
0
2
16
1
8
17
```

Figure 5. Corpus-labels file

3.2 Tweet Pre-Processor

The first step from the preprocessor module consists in cleaning up the data set (punctuation, stop words) in order to avoid noise in the implemented algorithms.

This step consists in removing punctuation marks and links. We identify them by using the regular expression:

(([-\"'/`_%$&*+<>^•()=|¡ ▪ ;:.,!?@#~]+)|([0-9]+))

We removed stop words and user mentions, but we decided not to eliminate the hashtag word because many tweets were made only by this kind of words. We removed instead the Hashtag sign and passed the words to the next step of preprocessing.

For the last step, we used Stanford Tokenizer in order to obtain the list of tokens for each tweet. Then we replaced each word with the correspondent lemma using WordNet Dictionary. The words that didn't have a lemma were considered noise and we choose to eliminate them.

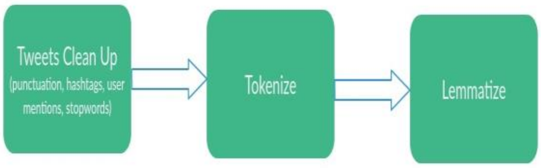
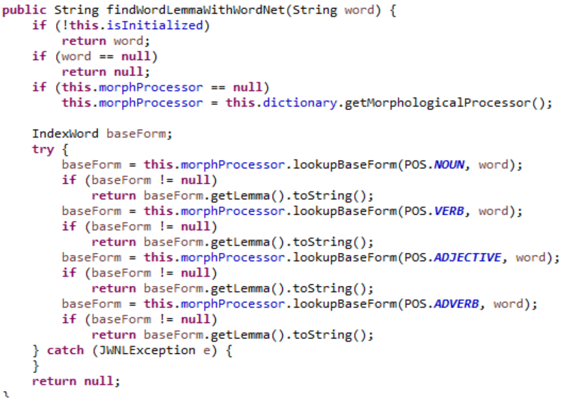


Figure 6. Preprocessing architecture



```
public String findWordLemmaWithWordNet(String word) {
    if (!this.isInitialized)
        return word;
    if (word == null)
        return null;
    if (this.morphProcessor == null)
        this.morphProcessor = this.dictionary.getMorphologicalProcessor();

    IndexWord baseForm;
    try {
        baseForm = this.morphProcessor.lookupBaseForm(POS.NOUN, word);
        if (baseForm != null)
            return baseForm.getLemma().toString();
        baseForm = this.morphProcessor.lookupBaseForm(POS.VERB, word);
        if (baseForm != null)
            return baseForm.getLemma().toString();
        baseForm = this.morphProcessor.lookupBaseForm(POS.ADJECTIVE, word);
        if (baseForm != null)
            return baseForm.getLemma().toString();
        baseForm = this.morphProcessor.lookupBaseForm(POS.ADVERB, word);
        if (baseForm != null)
            return baseForm.getLemma().toString();
    } catch (JWNLException e) {
    }
    return null;
}
```

Figure 7. Code sample for finding word lemma using WordNet Dictionary

3.3 Deep Learning Models

3.3.1 Recursive Neural Networks

Recursive neural network (RvNN) is a kind of deep neural network created by applying the same set of weights recursively over a structure, in order to produce a structured prediction over variable-size input structures, or a scalar prediction on it, by traversing a given structure in topological order**.**

Given the proven effectiveness and the impact of recurrent neural networks in different topics (sentiment analysis, etc.), we intend to build an emoji prediction model based on a Long Short-Term Memory Network (LSTM).

For each subtask, we divided the train data set into twenty smaller train sets, one for each emoji label. Each train subset contains tweets with only two labels (classes). For instance, the train set for label "0" contains tweets with classified with 0 or !0. We then created a model for every emoji label a trained it with the correspondent train dataset.

In order to unify the models output, we run the test dataset on each one of these model. Then, based on the probabilities of each classified label, we chose the emoji with the highest score.

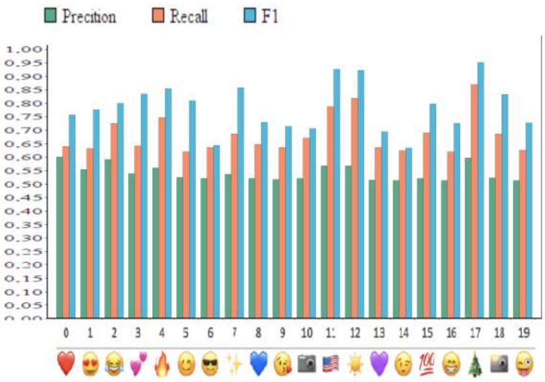


Figure 8. Evaluation for English models

3.3.2 Naïve Bayes Classifier

Naïve Bayes it's a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Likelihood

Class Prior Probability

$$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$$

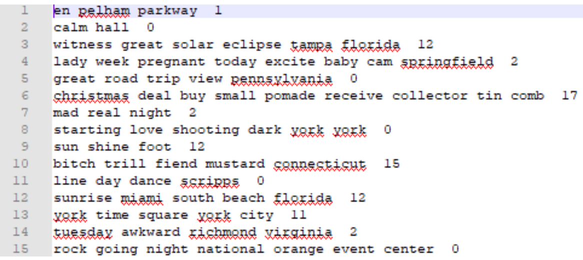
Posterior Probability

Predictor Prior Probability

$$P(c|\mathrm{X}) = P(x_1|c)\times P(x_2|c)\times\cdots\times P(x_n|c)\times P(c)$$

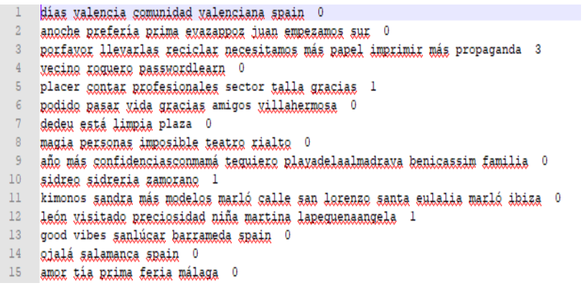
- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Naïve Bayes Classifier is written in JavaScript. We used several libraries, applications already made to make the best and smarter module. Train data receives as input ex: “en_train.txt” and “en_train.labels”, after the files are read, it reads files line by line, makes a tweet array, then it assigns to each tweet from the array a number from the label file, then it calls the Naïve Bayes algorithm implementation. Finally, the test data is entered and the values are generated for them. It represents an interface that easily generates a label according to the introduced text. Moreover, it does not change values if the same tweet is entered multiple times. The output format is txt and a label is generated for each line from test data.



```
en pelham parkway  1
calm hall  0
witness great solar eclipse tampa florida  12
lady week pregnant today excite baby cam springfield  2
great road trip view pennsylvania  0
christmas deal buy small pomade receive collector tin comb  17
mad real night  2
starting love shooting dark york york  0
sun shine foot  12
bitch trill fiend mustard connecticut  15
line day dance scripps  0
sunrise miami south beach florida  12
york time square york city  11
tuesday awkward richmond virginia  2
rock going night national orange event center  0
```

Figure 9. Short example of US result



```
días valencia comunidad valenciana spain  0
anoche prefería prima evazappoz juan empezamos sur  0
porfavor llevarlas reciclar necesitamos más papel imprimir más propaganda  3
vecino roquero passwordlearn  0
placer contar profesionales sector talla gracias  1
podido pasar vida gracias amigos villahermosa  0
dedeu está limpia plaza  0
magia personas imposible teatro rialto  0
año más confidenciasconmamá tequiero playadelaalmadrava benicassim familia  0
sidreo sidreria zamorano  1
kimonos sandra más modelos marló calle san lorenzo santa eulalia marló ibiza  0
león visitado preciosidad niña martina lapequenaangela  1
good vibes sanlúcar barrameda spain  0
ojalá salamanca spain  0
amor tía prima feria málaga  0
```

Figure 10. Short example of ES result

4. Discussions

Based on the things observed during the project implementation, we think that a possible improvement consists in trying to minimize the tweets noise. For instance, many words from tweets have duplicated letters (e.g. “aaaaand”). Eliminating those duplicated letters till the word has a correspondent lemma could significantly reduce the noise.

5. Conclusions

Emojis are very used on social sites, but not much is known about their use and semantics. However, it has been noticed that emojis are used in different communities. In this paper, we tried to predict the correspondent emoji for a given tweet using a deep learning module based on a Recurrent Neural Network and a Naïve Bayes module. The results for the Naïve Bayes implementation were better than those from the network module.

Acknowledgments

This survey was published with the support by two grants of the Romanian National Authority for Scientific Research and Innovation, UEFISCDI, project number PN-III-P2-2.1-BG-2016-0390, contract 126BG/2016 and project number PN-III-P1-1.2-PCCDI-2017-0818, contract 73PCCDI/2018 within PNCDI III, and partially by the README project "Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users' writing styles", contract no. 114/15.09.2017, MySMIS 2014 code 119286.

References

[1] Aoki, S. and Uchida, O. 2011. *A method for automatically generating the emotional vectors of emoticons using weblog articles.* In Proceedings of 10th WSEAS Int. Conf. On Applied Computer and Applied Computational Science, Stevens Point, Wisconsin, USA, pages 132{136, 2011.

[2] Barbieri, F., Ballesteros, M., Saggion, H., 2017. *Are Emojis Predictable?* in Large Scale Text Understanding Systems Lab, TALN Group Universitat Pompeu Fabra, Barcelona, Spain IBM T.J Watson Research Center, U.S

[3] Barbieri, Francesco and Camacho-Collados, Jose and Ronzano, Francesco and Espinosa-Anke, Luis and Ballesteros, Miguel and Basile, Valerio and Patti, Viviana and Saggion, Horacio, 2018. *SemEval-2018 Task 2: Multilingual Emoji Prediction.* In Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), New Orleans, LA, United States. Association for Computational Linguistics.

[4] Boia, M., Faltings, B., Musat, C.-C., and Pu, P., 2013. *A:) is worth a thousand words: How people attach sentiment to emoticons and words in tweets.* In Social Computing (SocialCom), 2013

International Conference on, pages 345–350. IEEE.

[5] Cappallo, S., Mensink, T. and Snoek, C. G. M., 2015. *Image2emoji: Zero-shot emoji predictionfor visual media.* In Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, pp.1311–1314. ACM, (2015).

[6] Castellucci, G. , Croce, D. and Basili R. 2015. *Acquiring a large scale polarity lexicon through unsupervised distributional methods.* In Natural Language Processing and Information Systems, pages 73{86. Springer, 2015.

[7] Dzmitry, B., Kyunghyun, C. and Yoshua, B., 2014. *Neural machine translation by jointly learning to align and translate.* In Proceeding of the third International Conference on Learning Representations, Toulon, France, May.

[8] Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M. and Riedel, S., 2016. *Emoji2vec: Learning emoji representations from their description.* In Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media, pages 48–54, Austin, TX, USA, November. Association for Computational Linguistics.

[9] Go, A. , Bhayani, R. and Huang, L. 2009. *Twitter sentiment classication using distant upervision.* CS224N Project Report, Stanford, 1:12, 2009

[10] Jiang, F., Liu, Y.-Q., Luan, H.-B., Sun, J.-S., Zhu, X., Zhang, M. and Ma, S.-P., 2015. *Microblog sentiment analysis with emoticon space model.* Journal of Computer Science and Technology, 30(5):1120{1129, 2015.