**Emoji Prediction**

**How can emoji be predicted?**

Emojis are ideograms which are naturally combined with plain text to visually complement or condense the meaning of a message. Despite being widely used in social media, their underlying semantics have received little attention from a Natural Language Processing standpoint.

An interesting article covering this subject, written by Francesco Barbieri et al. aims to investigate the relation between words and emojis, studying the problem of predicting which emojis are evoked by textbased tweet messages.

Miller et al. (2016) performed an evaluation asking human annotators the meaning of emojis and the sentiment they evoke. People do not always have the same understanding of emojis, indeed, there seems to exist multiple interpretations of their meaning beyond their designer’s intent or the physical object they evoke1. Their main conclusion was that emojis can lead to misunderstandings. The ambiguity of emojis raises an interesting question in human-computer interaction: how can we teach an artificial agent to correctly interpret and recognise emojis’ use in spontaneous conversation? The main motivation of this research is that an artificial intelligence system that is able to predict emojis could contribute to better natural language understanding (Novak et al., 2015) and thus to different natural language processing tasks such as generating emoji-enriched social media content, enhance emotion/sentiment analysis systems, and improve retrieval of social network material.

In this work, a state of the art classification framework is employed to automatically predict the most likely emoji a Twitter message evokes.

40 million tweets were retrieved with the Twitter APIs3. Tweets were posted between October 2015 and May 2016 geo-localized in the United States of America. All hyperlinks from each tweet were removed and lowercased all textual content in order to reduce noise and sparsity. From the dataset, tweets which include one and only one of the 20 most frequent emojis were selected, resulting in a final dataset composed of 584,600 tweets. In the experiments the subsets of the 10 (502,700 tweets) and 5 most frequent emojis (341,500 tweets) were considered. The emoji from the sequence of tokens is removed and used as a label both for training and testing. The task for the machine learning models described in this paper is to predict the single emoji that appears in the input tweet.

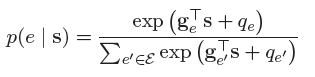
In this Section, the models that are used to predict an emoji given a tweet are as follows: the first model is an architecture based on Recurrent Neural Networks and the second and third are the two baselines. The two major differences between the RNNs and the baselines, is that the RNNs take into account sequences of words and thus, the entire context.

Given the proven effectiveness and the impact of recurrent neural networks in different tasks, the first emoji prediction model is based on bi-directional Long Short-term Memory Networks (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005). The B-LSTM can be formalized as follows:



W is a learned parameter matrix, fw is the forward LSTM encoding of the message, bw is the backward LSTM encoding of the message, and

d is a bias term, then passed through a component wise ReLU. The vector s is then used to compute the probability distribution of the emojis given the message as:



ge' is a column vector representing the (output) embedding of the emoji e, and qe is a bias term for the emoji e. The set E represents the list of emojis. The loss/objective function the network aims to minimize is the following:



m is a tweet of the training set T, s is the encoded vector representation of the tweet and em is the emoji contained in the tweet m. The inputs of the LSTMs are word embeddings. Following, two alternatives explored in the experiments presented in this paper.

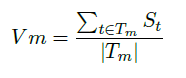
**Word Representations:** Word embeddings are generated which are learned together with the updates to the model. Stochastically (with p = 0.5) each word that occurs only once in the training data is replaced with a fixed represenation (out-of-vocabulary words vector). When pretrained word embeddings are used, these are concatenated with the learned vector representations obtaining a final representation for each word type. This is similar to the treatment of word embeddings by Dyer et al. (2015).

**Character-based Representations:** Character-based continuous-space vector embeddings are computed (Ling et al., 2015b; Ballesteros et al., 2015) of the tokens in each tweet using, again, bidirectional LSTMs. The character-based approach learns representations for words that are orthographically similar, thus, they should be able to handle different alternatives of the same word type occurring in social media.

Going forward, the two baselines mentioned earlier are discussed.

**Bag of words:** A bag of words classifier as baseline was applied, since it has been successfully employed in several classification tasks, like sentiment analysis and topic modeling (Wallach, 2006; Blei, 2012; Titov and McDonald, 2008; Maas et al., 2011; Davidov et al., 2010). Each message is represented with a vector of the most informative tokens (punctuation marks included) selected using term frequency−inverse document frequency (TFIDF). A L2-regularized logistic regression classifier is employed to make the predictions.

**Skip-Gram Vetor Average:** A Skip-gram model is trained (Mikolov et al., 2013), learned from 65M Tweets (where testing instances have been removed) to learn Twitter semantic vectors. Then, a model (henceforth, AVG) is built which represents each message as the average of the vectors corresponding to each token of the tweet. Formally, each message m is represented with the vector Vm:



Tm are the set of tokens included in the message m, St is the vector of token t in the Skipgram model, and |Tm| is the number of tokens in m. After obtaining a representation of each message, a L2-regularized logistic regression is trained (with epsilon equal to 0.001).

**Who did this already to a certain extent?**

Android and iOS did it already, Google does it as well, and there are many programs that were developed to predict those lovely emojis.

**Dango** - is a floating assistant that runs on your phone and predicts emoji, stickers and GIFs based on what you and your friends are writing in any app. This lets you have the same rich conversations everywhere: Messenger, Kik, Whatsapp, Snapchat.

<https://getdango.com/emoji-and-deep-learning/>

**EmojiNet** – “We investigate these questions by turning to the latest craze in computer vision: deep learning. Deep learning lets you train an entire model with low-, mid-, and high-level representations all at once. This comes at the expense of requiring a large amount of training data. For this investigation, we wish to train a model that suggests emojis for a given image. The deep learning technique typically used for computer vision is convolutional neural networks, and thus we call our model EmojiNet.”

<http://engineering.curalate.com/2016/01/20/emojinet.html>

**@Google** - Google has a new trick up its sleeve on Twitter — it can now understand emoji you tweet at it. Simply tweet an emoji  [@Google](https://twitter.com/google) and the company's Twitter account will respond with a fun GIF and a handy link to for local search results.

<https://www.theverge.com/2016/12/6/13856818/google-emoji-twitter-response-local-search>

**Swiftkey** - Keyboard app maker Swiftkey has officially launched its first product since that acquisition — and it’s an emoji-predicting keyboard app, called Swiftmoji.

<https://techcrunch.com/2016/07/20/swiftkey-officially-unwraps-its-emoji-prediction-app/>

**Resources for a better understanding of the project and how to approach it**

[1] **Barbieri F., Ballesteros M., Saggion H., -** [**Are Emojis Predictable?**](https://arxiv.org/abs/1702.07285), European Chapter of the Association for Computational Linguistics Valencia, 3-7 April 2017.

[2] **Marengo, Davide; Giannotta, Fabrizia; Settanni,** Michele - Assessing personality using emoji: An exploratory study – 2017

[3] **Marcel Danesi** - THE SEMIOTICS OF EMOJI – 2017

[4] **Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran** - EmojiNet: Building a Machine Readable Sense Inventory for Emoji

**Technical articles and resources**

[1] Efficient Estimation of Word Representations in Vector Space - <https://arxiv.org/pdf/1301.3781v1.pdf>

[2] : Deep Learning Lectures from Oxford University

<https://www.youtube.com/playlist?list=PLE6Wd9FR--EfW8dtjAuPoTuPcqmOV53Fu>

[3] Reading resources

<http://deeplearning.net/reading-list/>

[4] Getting starting with word2vec and gensim.

<http://rare-technologies.com/deep-learning-with-word2vec-and-gensim/>

[5] Intuition on word embedding methods

<http://www.offconvex.org/2015/12/12/word-embeddings-1/>

[6] RNN –

<http://www.neutronest.moe/2015-11-15-LSTM-survey.html>

[7] RNN- explaining LSTMs

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

List will go on…

**Risks**

The prediction cannot be the one expected. There are multiple approaches when it comes to choosing one single emoji. Sarcasm and irony can break the prediction chain if the sentences before weren’t processed thoroughly.

Ambiguity can strike in and without doubt, there will be a few times when the prediction will go wrong. The “tone” in written communication can be easily confusing, thus leading to a wrong emoji to be used.