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 dataset4 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

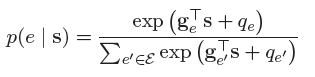
context.

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 componentwise 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) embedding5 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.

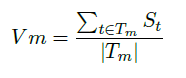
**Word Representations:** Word embeddings are generated which are learned together with the updates to the model. Stochastically (with p = 0.5) each word that occurs only once in the training data is replaced with a fixed represenation (out-of-vocabulary words vector). When pretrained word embeddings 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).