emoji2vec: Learning Emoji Representations from their Description

Abstract

Many current natural language processing applications for social media rely on representation learning and utilize pre-trained word embeddings. There currently exist several publicly-available, pre-trained sets of word embeddings, but they contain few or no emoji representations even as emoji usage in social media has increased. emoji2vec are pre-trained embeddings for all Unicode emojis which are learned from their description in the Unicode emoji standard. The resulting emoji embeddings can be readily used in downstream social natural language processing applications alongside word2vec. For the downstream task of sentiment analysis, emoji embeddings learned from short descriptions outperforms a skip-gram model trained on a large collection of tweets, while avoiding the need for contexts in which emojis need to appear frequently in order to estimate a representation.

1 Introduction

First introduced in 1997, emojis, a standardized set of small pictorial glyphs depicting everything from smiling faces to international flags, have seen a drastic increase in usage in social media over the last decade. The Oxford Dictionary named 2015 the year of the emoji, citing an increase in usage of over 800% during the course of the year, and elected the 'Face with Tears of Joy' emoji () as the Word of the Year. As of this writing, over 10% of Twitter posts and over 50% of text on Instagram contain one or more emojis. Due to their popularity and broad usage, they have been the subject of much formal and informal research in language and social communication, as well as in natural language processing (NLP).

In the context of social sciences, research has focused on emoji usage as a means of expressing emotions on mobile platforms. Interestingly, although essentially thought of as means of expressing emotions, emojis have been adopted as tools to express relationally useful roles in conversation. Emojis are culturally and contextually bound, and are open to reinterpretation and misinterpretation. These findings have paved the way for many formal analyses of semantic characteristics of emojis.

Concurrently we observe an increased interest in natural language processing on social media data. Many current NLP systems applied to social media rely on representation learning and word embeddings. Such systems often rely on pre-trained word embeddings that can for instance be obtained from word2vec or GloVe. Yet, neither resource contain a complete set of Unicode emoji representations, which suggests that many social NLP applications could be improved by the addition of robust emoji representations.

Embeddings for emoji Unicode symbols are learned from their description in the Unicode emoji standard. The usefulness of emoji representations trained in this way is demonstrated by evaluating on a Twitter sentiment analysis task. Furthermore, a qualitative analysis by investigating emoji analogy examples and visualizing the emoji embedding space is provided.

2 Related Work

There has been little work in distributional embeddings of emojis. The first research done in this direction was an informal blog post by the Instagram Data Team in 2015. They generated vector embeddings for emojis similar to skip-gram-based vectors by training on the entire corpus of Instagram posts. Their research gave valuable insight into the usage of emojis on Instagram, and showed that distributed representations can help understanding emoji semantics in everyday usage. The second contribution, closest to ours, trained emoji embeddings from a large Twitter dataset of over 100 million English tweets using the skip-gram method. These pre-trained emoji representations led to increased accuracy on a similarity task, and a meaningful clustering of the emoji embedding space. While this method is able to learn robust representations for frequently-used emojis, representations of less frequent emojis are estimated rather poorly or not available at all. In fact, only around 700 emojis can be found in this corpus, while there is support of over 1600 emojis in the Unicode standard.

The approach differs in two important aspects. First, since the representation of emojis are estimated directly from their description, robust representations are obtained for all supported emoji symbols — even the long tail of infrequently used ones. Secondly, the method works with much less data. Instead of training on millions of tweets, the representations are trained on only a few thousand descriptions. Still, higher accuracy results are obtained on a Twitter sentiment analysis task.

In addition, the work relates to building word representations for words and concepts based on their description in a dictionary. Similarly to their approach, representations are build for emojis based on their descriptions and keyword phrases.

Some of the limitations are evident in the work who showed that different cultural phenomena and languages may co-opt conventional emoji sentiment. Since training is only on English-language definitions and ignore temporal definitions of emojis, the training method might not capture the full semantic characteristics of an emoji.

3 Methodology

The method maps emoji symbols into the same space as the 300-dimensional Google News word2vec embeddings. Thus, the resulting emoji2vec embeddings can be used in addition to 300-dimensional word2vec embeddings in any application. To this end emojis, their name and their keyword phrases are crawled from the Unicode emoji list, resulting in 6088 descriptions of 1661 emoji symbols.

3.1 Model

Emoji embeddings are trained using a simple method. For every training example consisting of an emoji and a sequence of words $w_1, ..., w_N$ describing that emoji, we take the sum of the individual word vectors in the descriptive phrase as found in the Google News word2vec embeddings

$$v = \sum_{k=1}^{N} w_k, \tag{1}$$

where w_k is the word2vec vector for word w_k if that vector exists (otherwise we drop the summand) and v_j is the vector representation of the description. A trainable vector x_i for every emoji in our training set is defined, and the probability of a match between the emoji representation x_i and its description representation v_j is modeled using the sigmoid of the dot product of the two representations $\sigma(x_i^T v_j)$. For training we use the logistic loss

$$L(i, j, y_{ij}) = -\log(\sigma(y_{ij} x_i^T v_j - (1 - y_{ij}) x_i^T v_j)) \tag{2}$$

where y_{ij} is 1 if description j is valid for emoji i and 0 otherwise.

3.2 Optimization

The model is implemented in TensorFlow and optimized using stochastic gradient descent with Adam as optimizer. As we do not observe any negative training examples (invalid descriptions of emojis do not appear in the original training set), to increase generalization performance we randomly sample descriptions for emojis as negative instances (i.e. induce a mismatched description). One of the parameters of our model is the ratio of negative samples to positive samples; we found that having one positive example per negative example produced the best results. We perform early-stopping on a held-out development set and found 80 epochs of training to give the best results. As we are only training on emoji descriptions and our method is simple and cheap, training takes less than 3 minutes on a 2013 MacBook Pro.

4 Experiments

The approach is quantitatively evaluated on an intrinsic (emoji-description classification) and extrinsic (Twitter sentiment analysis) task. Furthermore, a qualitative analysis is given by visualizing the learned emoji embedding space and investigating emoji analogy examples.

4.1 Emoji-Description Classification

To analyze how well the method models the distribution of correct emoji descriptions, a manually-labeled test set containing pairs of emojis and phrases, as well as a correspondence label was created. For instance, the test set includes the example: , "crying", True, as well as the example , "fish", False. $\sigma(x_i^T v_i)$ is calculated for each example in the test set, measuring the similarity between the emoji vector and the sum of word vectors in the phrase.

When a classifier thresholds the above prediction at 0.5 to determine a positive or negative correlation, an accuracy of 85.5% is obtained for classifying whether an emoji-description pair is valid or not. By varying the threshold used for this classifier, a receiver operating characteristic curve with an area-under-the-curve of 0.933 is obtained, which demonstrates that high quality of the learned emoji representations.

4.2 Sentiment Analysis on Tweets

As downstream task the accuracy of sentiment classification of tweets for various classifiers with three different sets of pre-trained word embeddings are compared: (1) the original Google News word2vec embeddings, (2) word2vec augmented with emoji embeddings trained by skip-gram model, and (3) word2vec augmented with emoji2vec trained from Unicode descriptions. A dataset is used which consists of over 67k English tweets labelled manually for positive, neutral, or negative sentiment. In both the training set and the test set, 46% of tweets are labeled neutral, 29% are labeled positive, and 25% are labeled negative. To compute the feature vectors for training, we summed the vectors corresponding to each word or emoji in the text of the Tweet. The goal of this simple sentiment analysis model is not to produce state-of-the-art results in sentiment analysis; it is simply to show that including emojis adds discriminating information to a model, which could potentially be exploited in more advanced social NLP systems.

Because the labels are rather evenly distributed, accuracy is an effective metric in determining performance on this classification task. Results are reported in Table 1. Augmenting word2vec with emoji embeddings improves overall classification accuracy on the full corpus, and substantially improves classification performance for tweets that contain emojis. It suggests that emoji embeddings could improve performance for other social NLP tasks as well. Furthermore, emoji2vec generally outperforms the emoji embeddings trained by the skip-gram model, despite being trained on much less data using a simple model.

4.3 Analogy Task

A well-known property of word2vec is that embeddings trained with this method to some extent capture meaningful linear relationships between words directly in the vector space. For instance, it holds that the vector representation of 'king' minus 'man' plus 'woman' is closest to 'queen'. Word embeddings have commonly been evaluated on such word analogy tasks. Unfortunately, it is difficult to build such an analogy task for emojis due to the small number and semantically distinct categories of emojis. Nevertheless, a few intuitive examples were collected. For every query the closest five

Table 1: Three-way classification accuracy on the Twitter sentiment analysis corpus using Random Forrests and Linear SVM classifier with different word embeddings.

	Classification accuracy on entire dataset, N = 12920	
Word Embeddings	Random Forest	Linear SVM
Google News	57.5	58.5
Google News + (skip-gram model)	58.2*	60.0*
Google News + emoji2vec	59.5*	60.5*

	Classification accuracy on tweets containing emoji, N = 2295	
Word Embeddings	Random Forest	Linear SVM
Google News	46.0	47.1
Google News + (skip-gram model)	52.4*	57.4*
Google News + emoji2vec	54.4*	59.2*

	Classification accuracy on 90% most frequent emoji, N = 2186	
Word Embeddings	Random Forest	Linear SVM
Google News	47.3	45.1
Google News + (skip-gram model)	52.8*	56.9*
Google News + emoji2vec	55.0*	59.5*

	Classification accuracy on 10% least frequent emoji, N = 308	
Word Embeddings	Random Forest	Linear SVM
Google News	44.7	43.2
Google News + (skip-gram model)	53.9*	52.9*
Google News + emoji2vec	54.5*	55.2*

emojis were retrieved. Though the correct answer is sometimes not the top one, it is often contained in the top three.

5 Conclusion

Since existing pre-trained word embeddings such as Google News word2vec embeddings or GloVe fail to provide emoji embeddings, emoji2vec — embeddings of 1661 emoji symbols were released. Instead of running word2vec's skip-gram model on a large collection of emojis and their contexts appearing in tweets, emoji2vec is directly trained on Unicode descriptions of emojis. The resulting emoji embeddings can be used to augment any downstream task that currently uses word2vec embeddings, and might prove especially useful in social NLP tasks where emojis are used frequently (e.g. Twitter, Instagram, etc.). Despite the fact that the model is simpler and trained on much less data, it outperforms the skip-gram model on the task of Twitter sentiment analysis.

As the approach directly works on Unicode descriptions, it is not restricted to emoji symbols. In the future the usefulness of the method for other Unicode symbol embeddings will be investigated. Furthermore, plans are made to improve emoji2vec in the future by also reading full text emoji descriptions and using a recurrent neural network instead of a bag-of-word-vectors approach for enocoding descriptions. In addition, since the approach does not capture the context-dependent definitions of emojis (such as sarcasm, or appropriation via other cultural phenomena), mechanisms will be explored to efficiently capturing these nuanced meanings.

6 Data Release and Reproducibility

Pre-trained emoji2vec embeddings as well as the training data and code are released at https://github.com/uclmr/emoji2vec. Note that the emoji2vec format is compatible with word2vec and can be loaded into gensim or similar libraries.