

emoji2vec: Learning Emoji Representations from their Description

Ben Eisner

Princeton University
beisner@princeton.edu

Tim Rocktäschel

University College London
t.rocktaschel@cs.ucl.ac.uk

Isabelle Augenstein

University College London
i.augenstein@cs.ucl.ac.uk

Matko Bošnjak

University College London
m.bosnjak@cs.ucl.ac.uk

Sebastian Riedel

University College London
s.riedel@cs.ucl.ac.uk

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. In this paper we release emoji2vec, **pre-trained embeddings for all Unicode emojis which are learned from their description in the Unicode emoji standard.**¹ The resulting emoji embeddings can be readily used in downstream social natural language processing applications alongside word2vec. We demonstrate, for the downstream task of sentiment analysis, that 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 (Cruse, 2015).² 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, Kelly and Watts (2015) found that although essentially thought of as means of expressing emotions, emojis have been adopted as tools to express relationally useful roles in conversation. (Lebduska, 2014) showed that emojis are culturally and contextually bound, and are open to reinterpretation and misinterpretation, a result confirmed by (Miller et al., 2016). 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 (Ritter et al., 2011; Gattani et al., 2013; Rosenthal et al., 2015). Many current NLP systems applied to social media rely on representation learning and word embeddings (Tang et al., 2014; Dong et al., 2014; Dhingra et al., 2016; Augenstein et al.,

¹<http://www.unicode.org/emoji/charts/full-emoji-list.html>

²See https://twitter.com/Kyle_MacLachlan/status/765390472604971009 for an extreme example.

2016). Such systems often rely on pre-trained word embeddings that can for instance be obtained from `word2vec` (Mikolov et al., 2013a) or `GloVe` (Pennington et al., 2014). 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.

In this paper we release `emoji2vec`, embeddings for emoji Unicode symbols learned from their description in the Unicode emoji standard. We demonstrate the usefulness of emoji representations trained in this way by evaluating on a Twitter sentiment analysis task. Furthermore, we provide a qualitative analysis by investigating emoji analogy examples and visualizing the emoji embedding space.

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 (Dimson, 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, was introduced by (Barbieri et al., 2016). They trained emoji embeddings from a large Twitter dataset of over 100 million English tweets using the skip-gram method (Mikolov et al., 2013a). 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 Barbieri et al. (2016)’s corpus, while there is support of over 1600 emojis in the Unicode standard.

Our approach differs in two important aspects. First, since we are estimating the representation of emojis directly from their description, we obtain robust representations for all supported emoji symbols — even the long tail of infrequently used ones. Sec-



Man in Business Suit Levitating

Figure 1: Example description of U+1F574. We also use *business*, *man* and *suit* keywords for training.

ondly, our method works with much less data. Instead of training on millions of tweets, our representations are trained on only a few thousand descriptions. Still, we obtain higher accuracy results on a Twitter sentiment analysis task.

In addition, our work relates to the work of Hill et al. (2016) who built word representations for words and concepts based on their description in a dictionary. Similarly to their approach, we build representations for emojis based on their descriptions and keyword phrases.

Some of the limitations of our work are evident in the work of Park et al. (2013) who showed that different cultural phenomena and languages may co-opt conventional emoji sentiment. Since we train only on English-language definitions and ignore temporal definitions of emojis, our training method might not capture the full semantic characteristics of an emoji.

3 Method

Our 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 we crawl emojis, their name and their keyword phrases from the Unicode emoji list, resulting in 6088 descriptions of 1661 emoji symbols. Figure 1 shows an example for an uncommon emoji.

3.1 Model

We train emoji embeddings using a simple method. For every training example consisting of an emoji and a sequence of words w_1, \dots, 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

$$v_j = \sum_{k=1}^N w_k$$

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. We define a trainable vector x_i for every emoji in our training set, and model the probability of a match between the emoji representation x_i and its description representation v_j using the sigmoid of the dot product of the two representations $\sigma(x_i^T v_j)$. For training we use the logistic loss

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

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

3.2 Optimization

Our model is implemented in TensorFlow (Abadi et al., 2015) and optimized using stochastic gradient descent with Adam (Kingma and Ba, 2015) 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 Evaluation

We quantitatively evaluate our approach on an intrinsic (emoji-description classification) and extrinsic (Twitter sentiment analysis) task. Furthermore, we give a qualitative analysis by visualizing the learned emoji embedding space and investigating emoji analogy examples.

4.1 Emoji-Description Classification

To analyze how well our method models the distribution of correct emoji descriptions, we created a manually-labeled test set containing pairs of emojis and phrases, as well as a correspondence label. For instance, our test set includes the example: {😭,

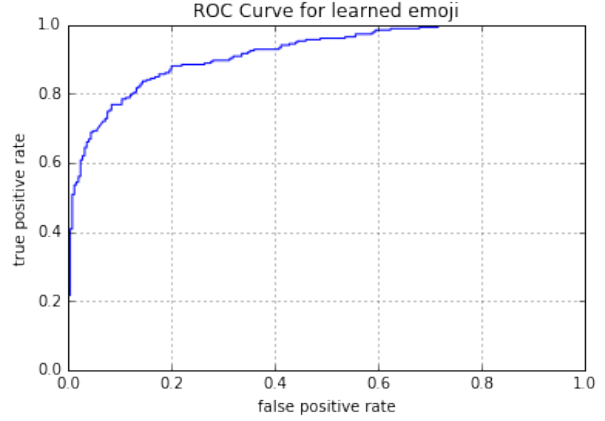


Figure 2: Receiver operating characteristic curve for learned emoji vectors evaluated against the test set.

”crying”, True}, as well as the example {😭, ”fish”, False}. We calculate $\sigma(x_i^T v_i)$ 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, we obtain an accuracy of 85.5% for classifying whether an emoji-description pair is valid or not. By varying the threshold used for this classifier, we obtain a receiver operating characteristic curve (Figure 4.1) with an area-under-the-curve of 0.933, which demonstrates that high quality of the learned emoji representations.

4.2 Sentiment Analysis on Tweets

As downstream task we compare the accuracy of sentiment classification of tweets for various classifiers with three different sets of pre-trained word embeddings: (1) the original Google News `word2vec` embeddings, (2) `word2vec` augmented with emoji embeddings trained by Barbieri et al. (2016), and (3) `word2vec` augmented with emoji2vec trained from Unicode descriptions. We use the recent dataset by Kralj Novak et al. (2015), 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. We find that 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, we find that `emoji2vec` generally outperforms the emoji embeddings trained by Barbieri et al. (2016), despite being trained on much less data using a simple model.

4.3 t-SNE Visualization

To gain further insights, we project the learned emoji embeddings into two-dimensional space using t-SNE (Maaten and Hinton, 2008). This method projects high-dimensional embeddings into a lower-dimensional space while attempting to preserve relative distances. We perform this projection of emoji representation into two-dimensional space.

From Figure 4.3 we see a number of notable semantic clusters, indicating that the vectors we trained have accurately captured some of the semantic properties of the emojis. For instance, all flag symbols are clustered in the bottom, and many smiley faces in the center. Other prominent emoji clusters include fruits, astrological signs, animals, vehicles, or families. On the other hand, symbolic representations of numbers are not properly disentangled in the embedding space, indicating limitations of our simple model. A two-dimensional projection is convenient from a visualization perspective, and certainly shows that some intuitively similar emojis are close to each other in vector space.

4.4 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' (Mikolov et al., 2013b). Word embeddings have commonly been evaluated on such word analogy tasks (Levy and Goldberg, 2014). Unfortunately, it is difficult to build such an analogy task for emojis due to the small number and semantically distinct categories of emojis. Nevertheless, we collected a few intuitive examples in Figure 4. For every query we have retrieved the closest five emojis. Though the correct answer is sometimes not the top one, it is often contained in the top three.



Figure 4: Emoji analogy examples. Notice that the seemingly "correct" emoji often appears in the top three closest vectors, but not always in the top spot (furthest to the left).

5 Conclusion

Since existing pre-trained word embeddings such as Google News `word2vec` embeddings or GloVe fail to provide emoji embeddings, we have released `emoji2vec` — embeddings of 1661 emoji symbols. 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 our model is simpler and trained on much less data, we outperform (Barbieri et al., 2016) on the task of Twitter sentiment analysis.

As our approach directly works on Unicode descriptions, it is not restricted to emoji symbols. In the future we want to investigate the usefulness of our method for other Unicode symbol embeddings. Furthermore, we plan to improve `emoji2vec` in the future by also reading full text emoji description

Classification accuracy on entire dataset, $N = 12920$		
Word Embeddings	Random Forrest	Linear SVM
Google News	57.5	58.5
Google News + (Barbieri et al., 2016)	58.2*	60.0*
Google News + emoji2vec	59.5*	60.5*
Classification accuracy on tweets containing emoji, $N = 2295$		
Word Embeddings	Random Forrest	Linear SVM
Google News	46.0	47.1
Google News + (Barbieri et al., 2016)	52.4*	57.4*
Google News + emoji2vec	54.4*	59.2*
Classification accuracy on 90% most frequent emoji, $N = 2186$		
Word Embeddings	Random Forrest	Linear SVM
Google News	47.3	45.1
Google News + (Barbieri et al., 2016)	52.8*	56.9*
Google News + emoji2vec	55.0*	59.5*
Classification accuracy on 10% least frequent emoji, $N = 308$		
Word Embeddings	Random Forrest	Linear SVM
Google News	44.7	43.2
Google News + (Barbieri et al., 2016)	53.9*	52.9*
Google News + emoji2vec	54.5*	55.2*

Table 1: Three-way classification accuracy on the Twitter sentiment analysis corpus using Random Forrests (Ho, 1995) and Linear SVM (Fan et al., 2008) classifier with different word embeddings. "*" denotes results with significance of $p < 0.05$ as calculated by McNemar's test, with the respect to classification with Google News embeddings per each classifier, and dataset



Figure 3: Emoji vector embeddings, projected down into a 2-dimensional space using the t-SNE technique. Note the clusters of similar emojis like flags (bottom), family emoji (top left), zodiac symbols (top left), animals (left), smileys (middle), etc.

from Emojipedia³ and using a recurrent neural network instead of a bag-of-word-vectors approach for encoding descriptions. In addition, since our approach does not capture the context-dependent definitions of emojis (such as sarcasm, or appropriation via other cultural phenomena), we would like to explore mechanisms of efficiently capturing these nuanced meanings.

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 gensim⁴ or similar libraries.

Acknowledgements

The authors would like to thank Michael Large, Peter Gabriel and Suran Goonatillake for the inspiration of this work, and the anonymous reviewers for their insightful comments. This research was supported by an Allen Distinguished Investigator award, a Marie Curie Career Integration Award, by Microsoft Research through its PhD Scholarship Programme, and by Elsevier.

References

- Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *Software available from tensorflow.org*, 1.
- Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance Detection with Bidirectional Conditional Encoding. In *Proceedings of EMLNP*.
- Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis. In *Proceedings of LREC*, May.
- Joe Cruse. 2015. Emoji usage in TV conversation.
- Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016.

³emojipedia.org

⁴<https://radimrehurek.com/gensim/models/word2vec.html>

- Tweet2Vec: Character-Based Distributed Representations for Social Media. In *Proceedings of ACL*, pages 269–274.
- Thomas Dimson. 2015. Machine Learning for Emoji Trends. <http://instagram-engineering.tumblr.com/post/117889701472/emojineering-part-1-machine-learning-for-emoji>. Accessed: 2016-09-05.
- Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In *Proceedings of ACL*, pages 49–54.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.
- Abhishek Gattani, Digvijay S Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. 2013. Entity Extraction, Linking, Classification, and Tagging for Social Media: A Wikipedia-Based Approach. In *Proceedings of the VLDB Endowment*, 6(11):1126–1137.
- Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to Understand Phrases by Embedding the Dictionary. *TACL*.
- Tin Kam Ho. 1995. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE.
- Ryan Kelly and Leon Watts. 2015. Characterising the inventive appropriation of emoji as relationally meaningful in mediated close personal relationships. *Experiences of Technology Appropriation: Unanticipated Users, Usage, Circumstances, and Design*.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*.
- Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of Emojis. *PLoS ONE*, 10(12):1–22, 12.
- Lisa Lebduska. 2014. Emoji, Emoji, What for Art Thou? *Harlot: A Revealing Look at the Arts of Persuasion*, 1(12).
- Omer Levy and Yoav Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of ConLL*, pages 171–180.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed Representa-

- tions of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, pages 3111–3119.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of NAACL-HLT*, pages 746–751.
- Hannah Miller, Jacob Thebault-Spieker, Shuo Chang, Isaac Johnson, Loren Terveen, and Brent Hecht. 2016. “Blissfully happy” or “ready to fight”: Varying Interpretations of Emoji. In *Proceedings of ICWSM*.
- Jaram Park, Vladimir Barash, Clay Fink, and Meeyoung Cha. 2013. Emoticon style: Interpreting differences in emoticons across cultures. In *ICWSM*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of EMNLP*, pages 1532–1543, October.
- Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of EMNLP*, pages 1524–1534.
- Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the SemEval*, pages 451–463.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of ACL*, pages 1555–1565.