Learning Emoji Embeddings using Emoji Co-occurrence Network Graph

Anurag Illendula
Department Of Mathematics
IIT Kharagpur
aianurag09@iitkgp.ac.in

Manish Reddy Yedulla
Department of Engineering Science
IIT Hyderabad
es15btech11012@iith.ac.in

Abstract

Usage of emoji in social media platforms has seen a rapid increase over the last few years. Majority of the social media posts are laden with emoji and users often use more than one emoji in a single social media post to express their emotions and to emphasize certain words in a message. Utilizing the emoji co-occurrence can be helpful to understand how emoji are used in social media posts and their meanings in the context of social media posts. In this paper, we investigate whether emoji co-occurrences can be used as a feature to learn emoji embeddings which can be used in many downstream applications such sentiment analysis and emotion identification in social media text. We utilize 147 million tweets which have emojis in them and build an emoji co-occurrence network. Then, we train a network embedding model to embed emojis into a low dimensional vector space. We evaluate our embeddings using sentiment analysis and emoji similarity experiments, and experimental results show that our embeddings outperform the current state-of-the-art results for sentiment analysis tasks.

1 Introduction

Emojis are the 21st century's successor to the emoticon. They arose from the need to communicate body language and facial expressions during text conversations. They are two-dimensional visual embodiments of everyday aspects of life which were standardized by the Unicode Consortium in 2010 as part of Unicode 6.0. Emoji proliferated throughout the globe and has particularly become a part of the popular culture in the west. It has been adopted by almost all social media platforms and messaging services. Emojis serve many purposes during online communication, among which conveying emotion is one of the primary uses. According to the latest statistics released by Emojipedia in June 2017, the number of emojis has increased to 2,666, posing challenges to applications that list them in small hand-held devices such as mobile phones. To overcome this challenge, emoji keyboards in most of the smartphones contains categorizes emoji into several categories listed in Table 1.

Many recent Natural Language Processing (NLP) systems rely on word representations in finite-dimensional vector space. These NLP systems mainly use pre-trained word embeddings obtained from word2vec [MSC+13] or GloVe [PSM14] or fastText [BGJM16]. Earlier GloVe embeddings were used for training most NLP systems, but fastText trained word embeddings could achieve much higher accuracies of NLP systems involving social media data because the fastText model could learn sub-word information. Emoji embeddings have been of fundamental importance to improve the accuracies of many emoji understanding tasks. Recent research proved that emoji embeddings could enhance the performance of emoji prediction [FMS+17, BBS17], emoji similarity [WBSD17b], and emoji sense disambiguation tasks [WBSD17a, SPWT17]. These emoji representations have also been efficient in understanding the behavior of emojis in different contexts. The need to learn emoji representations for improving the performance of social NLP systems has been recognized by Eisner et al. [ERA+16] and Francesco et al. [BRS16]

among others, where they used traditional approaches which include skip-gram and CBOW model to learn emoji embeddings.

Information networks such as publication networks, World Wide Web are characterized by the interplay between various content and a sophisticated underlying knowledge structure. Graph embedding models are helpful to scale information from large-scale information networks and embed them into a finite-dimensional vector space, and these embeddings have shown great success in various NLP tasks such as node classification [BCM11], link prediction [LNK07] and classification [YRS+14] tasks. These graph embedding models have been of crucial importance and have enhanced the performance of word similarity and word analogical reasoning tasks using language networks [TQW+15]. The analysis of emoji co-occurrence network graphs can help us understand emojis from different perspectives. We hypothesize that emojis which co-occur in a tweet contains the same sentiment as the overall sentiment of the tweet. Consider a tweet, "I got betrayed by 🕵, I want to kill you 🔫", here both the emojis 🕵,🔫 contain negative sentiment, the overall sentiment of the tweet is also negative. Hence we investigate whether emoji co-occurrence could be a better feature to learn emoji representations to improve the accuracy of classification tasks. In this paper, we introduce an approach to learn emoji representations using emoji co-occurrence network graph and large-scale information network embedding model and evaluate our embeddings using the gold-standard dataset for sentiment analysis task.

Table 1: Emoji Categories

Category	Emoji Examples
Smiley and People	😂,😄,😊
Animals and Nature	🐕,🐿,🐑
Food and Drink	🍉,🍉,🍅
Activity	🛀,🚶,🏌
Travel and Places	🚢,🚩,🏔
Objects	⌨,⏳,⌚
Symbols	💙,💕,💔
Flags	🇭🇺,🚩,🇷🇴

This paper is organized as follows. Section 2 discusses the related work done by other researchers in the field of emoji understanding and learning network representations. Section 3 discusses the process of creating an emoji co-occurrence network using our twitter corpus. Section 4 explains our model architecture to
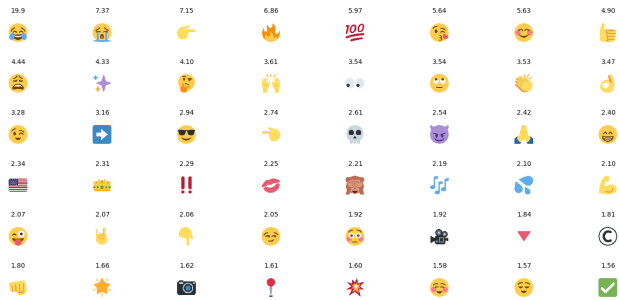

Figure 1: Distribution of Tweets across various emojis in lakhs

.

learn emoji representations from emoji co-occurrence network graph. Section 5 reports the accuracies obtained by our emoji embeddings on the gold-standard dataset for sentiment analysis task, emoji similarity tasks. We discuss the reason behind high accuracies obtained for sentiment analysis task in Section 6 followed by plans for future work in Section 7.

2 Related Work

One of the exciting work by Wijeratne et al. [WBSD16, WBSD17a] in the field of emoji understanding is EmojiNet (http://emojinet.knoesis.org/home.php), the largest machine readable emoji sense inventory, this inventory helps computers understand emojis. In this work Wijeratne et al. tried to connect emojis and their senses to corresponding words in babelnet ([NP12]) using their respective babelnetId. EmojiNet opened doors to many of the emoji understanding tasks like emoji similarity, emoji prediction, emoji sense disambiguation.

The other interesting work done by Wijeratne et al. [WBSD17b] addressed the challenge of measuring emoji similarity using the semantics of emoji. They defined two types of semantics embeddings using the textual senses and the textual descriptions of emojis. Prior work by Francesco et al. ([BRS16]) and Eisner et al. ([ERA+16]) used traditional approaches to learn emoji embeddings. The semantic embeddings have achieved accuracies which outperformed the previous state-of-the-art results in sentiment analysis task; this high accuracy is due to the fact that semantic embeddings can learn syntactic, semantic, sentiment features of emojis.

Seyednezhad et al. ([SM17]) created a network using the emoji co-occurrences in the same tweet; they claim that each edge weight can help us understand the user's context to use multiple emojis. This emoji network also enabled them to justify the use of co-occurred emojis in different perceptions. This also en-
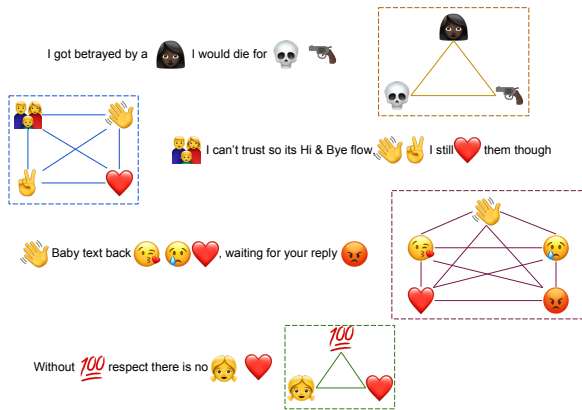
Figure 2: Construction of Emoji polygons
.

abled them to understand emoji usage by understanding possible relations between these special characters in common text. Fede et al. ([FHSM17]) studied different characteristics of this emoji co-occurrence network graph which include studying user's behavior to use a sequence of emojis in different contexts.

Information networks have been of primary use to store large amounts of information. Many researchers have proposed different graph embedding models in machine learning literature which allow us to embed nodes of large information networks into a low dimensional vector space ([PARS14] [GL16] [CLX15]). These embeddings helped address many tasks such as node classification, visualization, and link prediction tasks.

3 Data and Network

The emoji network is constructed using a twitter corpus of 147 million tweets crawled through a period of 2 months (from 6th August 2016 to 8th September 2016) by Wijeratne *et al.* [WBSD17a]. We filter the tweets and only consider the tweets which have multiple emojis embedded in a tweet. This reduces the number of distinct tweets in the dataset to 14.3 million. Figure 1 shows the distribution of the number of tweets of the most frequently occurring emojis. Each tweet generates a polygon of n sides where n is the number of emojis embedded in the tweet. The construction of emoji network is straightforward and Figure 2 explains the construction of emoji polygons with the help of different examples.

The weight of an edge signifies the number of co-occurrences of the emojis sharing the edge considering the complete twitter corpus. For example in the case of tweets shown in Figure 2 the emoji pair (❤️,👋) appeared twice hence the weight of the edge correspond-

Table 2: Most frequently co-occurring emoji pairs

Emoji Pair	No of Co-occurrences
(😂,🙈)	230957
(👉,👈)	196970
(💀,😂)	135595
(😂,😩)	102612
(😒,😂)	102408

Table 3: Least frequently co-occurring emoji pairs

Emoji Pair	No of Co-occurrences
(📰,💾)	1
(😵,📰)	1
(📰,🎰)	1
(😇,📰)	1
(📰,✨)	1

ing to these two emojis is considered as 2. Similarly, the weight of all the edges in the emoji network is calculated.

The emoji co-occurrence network created using the tweets in Figure 2 is represented in Figure 3. We input the emoji co-occurrence network graph to our graph embedding model to learn 300-dimensional emoji embeddings, and we evaluate our embeddings using the gold-standard dataset for sentiment analysis. We use the gold-standard dataset ([NSSM15]) to evaluate our embeddings because the current state-of-the-art results [ERA$^+$16] for sentiment analysis were obtained on this dataset.

4 Model

4.1 Description

Here we discuss two different types of measures which signify the proximity between two nodes of the co-occurrence network graph, and the model developed by Jian et al. [TQW$^+$15] to learn the node representations of a network graph.

First Order Proximity : The first order proximity is defined as the local pairwise proximity which can be related to the weight of the edge formed by joining the two vertices. The first order proximity between an edge (u,v) is the weight W_{uv} of the edge formed by vertices u, v. It can also be inferred from the definition that the first-order proximity between any two non-connected vertices is zero.
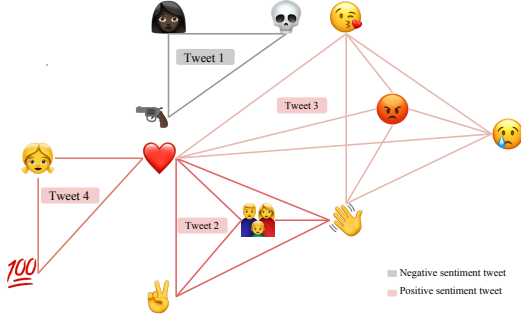
Figure 3: Construction of Emoji Network

.

Second Order Proximity : The second order proximity is defined as the similarity between neighbourhood network structures. For example, consider u to be an emoji node, let $p_u = (w_{(u,1)}, w_{(u,2)}, ..., w_{(u,|V|)})$ denote the first order proximity of the emoji node "u" with all the vertices then the second order proximity is defined as the similarity between p_u and p_v. If there exists no common vertex between u and v, then second-order proximity is zero.

4.1.1 Network embedding using first order proximity:

Let u_i and u_j represent the network embedding in d dimensional vector space, where (i,j) is an undirected edge in the network graph. The joint probability which signifies the proximity between vertices v_i, v_j is defined as

$$p_1(v_i, v_j) = \frac{1}{1 + exp(\vec{u}_i^T \cdot \vec{u}_j)} \tag{1}$$

where $\vec{u_i} \in R^d$ is a low dimensional representation also called as embedding for emoji node v_i, w_{ij} represents the weight of the edge between the nodes v_i and v_j The probability distribution between different pair of vertices is defined as p(.,.) over the vector space V x V and the empirical probability is defined as $\widetilde{p_1}(i,j)$

$$\widetilde{p_1}(i,j) = \frac{w_{ij}}{W} \quad and \quad W = \sum_{(i,j) \in E} w_{ij} \tag{2}$$

To maintain the first Order proximity between the vertices of the network graph, the objective function (O_1) which is the distance between the empirical probability function and the proximity function is to be optimized.

$$O_1(i,j) = d(\widetilde{p_1}(i,j), p_1(i,j)) \tag{3}$$

where $d(\widetilde{p_1}(i,j), p_1(i,j))$ is defined as the distance between the two probability distributions. Replacing $d(\cdot, \cdot)$ by KL-divergence, the objective function reduces to

$$O_1 = - \sum_{(i,j) \in E} O_1(i,j) \tag{4}$$

$$O_1 = - \sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j) \tag{5}$$

4.1.2 Network embedding using second order proximity:

The second order proximity of two nodes (v_i, v_j) measures the similarity of the neighbourhood network structures of nodes (v_i, v_j). This measure is applicable for both directed and undirected graphs. Hence our objective, in this case, is to look at the vertex and the "context" of the vertex which can also be related to the distribution of neighbours of the given vertex. Hence for each edge (v_i, v_j) the probability of "context" is defined by

$$p_2(v_j|v_i) = \frac{exp(\vec{u'}_j^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} exp(\vec{u'}_k^T \cdot \vec{u}_i)} \tag{6}$$

Where $|V|$ is the number of vertices. As mentioned before, the second order proximity assumes that vertices with similar distribution over the contexts as similar vertices. To maintain the second order proximity, the similarity distance between the contexts $p_2(\cdot|v_i)$ represented in the low dimensional vector space and the empirical distribution $\widetilde{p_2}(\cdot|j)$ must be optimized. Hence our objective function (O_2) in this case is

$$O_2 = \sum_{v_i \in V} \lambda_i d(\widetilde{p_2}(\cdot|v_i), p_2(\cdot|v_i)) \tag{7}$$

where $d(\cdot, \cdot)$ is the distance between two probability distributions, here the variable λ_i is used to consider the importance of the vertex v_i during the process of optimization. As defined in the previous case the empirical distribution is defined as

$$\widetilde{p_2}(i,j) = \frac{w_{ij}}{d_i} \quad and \quad d_i = \sum_{k \in N(i)} w_{ik} \tag{8}$$

w_{ij} is the weight of edge (v_i, v_j) and d_i is defined as the out-degree of vertex and N(i) is the set of neighbours of v_i. Considering $\lambda_i = d_i$ for the purpose of simplicity, and replacing $d(\cdot, \cdot)$ with KL-divergence

$$O_2 = - \sum_{(i,j) \in E} w_{ij} \log p_2(v_j|v_i) \tag{9}$$

4.2 Model Optimization

The approach of negative sampling proposed by Mikolov et al. [MSC⁺13] is used to optimize the objective function which helps us to represent every vertex of the network graph in the low dimensional vector space. Hence the objective function simplifies to:

$$\log \sigma(\vec{u'}_j^T \cdot \vec{u}_i)) + \sum_{i=1}^{K} E_{v_n P_n(v)}[\log \sigma(\vec{u'}_j^T \cdot \vec{u}_i)] \quad (10)$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function. We use the stochastic gradient descent algorithm [RRWN11] for optimizing the objective function and we update the model parameters on a batch of edges. Thus after completion of the training process, we get the embeddings corresponding to each vertex. The gradient with respect to an embedding $\vec{u}_i$ of vertex v_i will be:

$$\frac{\partial O_1}{\partial \vec{u}_i} = w_{ij} \cdot \frac{\partial \log(p_1)(v_i, v_j)}{\partial \vec{u}_i} \quad (11)$$

$$\frac{\partial O_2}{\partial \vec{u}_i} = w_{ij} \cdot \frac{\partial \log(p_2)(v_j|v_i)}{\partial \vec{u}_i} \quad (12)$$

We learn the node embeddings ($\vec{u}_i$) by optimizing the objective function in both cases and call the embeddings as first order embeddings and second order embeddings respectively. The model is trained using the Tensorflow ([ABC⁺16]) library on a cuda GPU. Model is trained using RMS Propagation gradient descent algorithm with learning rate as 0.025, and we used a batch size as 128, the number of batches = 300000 and 300-dimensional embeddings. The code is made available on Github[1], 300-dimensional emoji embeddings learned using the emoji co-occurrence network can also be accessed at this link.

5 Experiments

5.1 Sentiment Analysis

In this section, we report our accuracies obtained for the sentiment analysis task on the gold-standard dataset developed by Novak et al. [NSSM15]. Our experiments have achieved accuracies which outperform the current state-of-the-art results for sentiment analysis on the gold-standard dataset. The gold-standard dataset[2] consists of 64599 manually labelled tweets classified into positive, negative, neutral sentiment. The dataset is divided into training set that consists 51679 tweets, 9405 out of which contain emoji and testing set that consists of 12920 tweets, 2295 out of

which contain emoji. In both training and testing sets, 29% are labelled as positive, 25% are labelled as negative, and 46% are labelled as neural. We use the pre-trained FastText word embeddings[3] [MGB⁺18] to embed words into a low dimensional vector space. We calculate the bag of words vector for each tweet and then use this vector as a feature to train a support vector machine and a random forest model on the training set, and evaluate the accuracies obtained for classification task on whole testing dataset consisting of 12920 tweets. The accuracies obtained for classification task using the first order embeddings surpass the current state-of-the-art [WBSD17b] results.

Table 4: Accuracy of Sentiment Analysis task

Word Embeddings	Classification accuracy using RF	Classification accuracy using SVM
State-of-the-art results	60.7	63.6
First Order Embedding	**62.1**	**65.2**
Second Order Embedding	58.7	61.9

5.2 Emoji Similarity

Emoji similarity[4] is one of the important challenges which should be addressed for the development of emoji keyboards since the current emoji keyboard consists of 2666 emojis, and the complete list cannot be accommodated in a small screen. These emoji embeddings learned using the emoji co-occurrence network graph could be helpful to calculate the similarity between emojis using cosine distance as the similarity measure and group emojis which have high similarity values. This grouping of emojis can decrease the number of distinct emojis and helps us accommodate this grouped emojis on a small screen. In this section, we report the emoji similarity values found considering the first order embeddings and second order embeddings.

We consider the cosine distance to be the similarity measure between two embeddings. Let $\vec{a}$ and $\vec{b}$ be two vectors which represent embeddings of emojis e_1 and e_2 respectively, the similarity measure between these two emojis (e_1 and e_2) is calculated as

---

[1]https://bit.ly/2I5hYNd
[2]https://bit.ly/2pLaKVZ

[3]https://bit.ly/2FMTB4N
[4]Our main objective is not to address the emoji similarity task. Our main objective is to demonstrate the usefulness of our emoji embeddings for sentiment analysis task.

$$similarity(e_1, e_2) = \frac{\vec{a} \cdot \vec{b}}{|a| \cdot |b|} \qquad (13)$$

Table 5 and Table 6 reports the most similar emojis found considering the first order embeddings and second order embeddings respectively.The observed results are explained in Section 6.

Table 5: Emoji Similarity Measured using first order embeddings

Emoji Pair	Similarity Measure	Semantic Similarity
(,)	0.921	0.442
(,)	0.916	0.598
(,)	0.911	0.623
(,)	0.909	0.546
(,)	0.856	0.723
(,)	0.889	0.702
(,)	0.881	0.737

5.3 Analogical Reasoning

The analogical reasoning task introduced by Mikolov et al. [MSC+13], defines the syntactic and semantic analogies. For example, consider the semantic analogy such as USA : Washington = India : ? where we fill the gap (represented by "?") by finding a word from the vocabulary whose embedding(represented by vec(x)) is closest to vec(Washington) - vec(USA) + vec(India). Here cosine distance is considered as the similarity measure between the two vectors.

Table 6: Emoji Similarity Measured using second order embeddings

Emoji Pair	Similarity Measure	Semantic Similarity
(,)	0.646	0.662
(,)	0.606	0.598
(,)	0.596	0.623
(,)	0.556	0.622
(,)	0.546	0.916
(,)	0.540	0.945

5.3.1 Emoji to emoji analogy

We extrapolate the semantic analogy task introduced by Mikolov et al. [MSC+13] in the context of emojis, by replacing words with emojis. Consider an emoji

Table 7: Spearman's Rank Correlation Results

Emoji Embeddings	$\rho * 100$
First Order Embeddings	**74**
Second Order Embedding	66

analogy, (:) = (: ?), we fill the gap (represented by "?") by finding an emoji from the complete list of emojis whose embedding(represented by vec(x)) is closest to vec( ) - vec( ) + vec( ). Table 8 reports some of the interesting analogies found using first order and second order embeddings.

Table 8: Emoji to Emoji Analogical Reasoning using Emoji Embeddings

First Emoji Pair	Second Emoji Pair
(:)	(:)
(:)	(:)
(:)	(:)
(:)	(:)
(:)	(:)

6 Discussion

The high accuracy for classification task using the first order embedding model is due to the fact that all co-occurring emojis in a tweet possess the same sentiment feature, hence during classification these embeddings would increase the accuracy of the classification model. Consider the tweet, "Who uses this emoji 😂, I miss the one that had this mouth 😀 and these eyes 😍 ! ... where did he go?! Why did he leave?!" , in this tweet we observe the overall sentiment to be positive, and we also observe that all the emojis embedded in the tweet possess the same sentiment. Hence co-occurring emojis would be better attribute to learn emoji embeddings which can increase the accuracy of sentiment analysis and other related classification tasks.

We use the Spearman's rank correlation coefficient to evaluate the emoji similarity ranks obtained using first order and second order embeddings learned using emoji co-occurrence network with the emoji similarity ranks of gold-standard dataset[5]. Table 7 reports the Spearman's correlation coefficient obtained by our emoji embeddings. According to the correlation coefficients the first emoji embeddings show a strong correlation ($0.6 < \rho < 0.79$).

The top 6 most similar emoji pairs observed considering the first order embeddings are reported in Table

---

[5] https://bit.ly/2GztSR2

5. As we see from Table 5 the most similar emoji pair observed is (👊,🤢) with similarity measure of 0.921. The usage of the first emoji 👊 would be in the context where the user wishes to express his dis-concern over certain issue through an act of hitting, the usage of the other emoji 🤢 would be in the context where the user wishes to express his dis-concern over certain issue through an expression of uneasiness. Hence the high similarity measure has sound even if we consider the context of use of the emojis. The results show that our embeddings give higher similarity measures than the semantic similarity[6] measure.

The top 6 most similar emoji pairs observed considering the second order embeddings are reported in Table 6. As we see from Table 6 the most similar emoji pair observer is (🔔,🎉) with similarity measure of 0.586. The usage of the first emoji 🔔 would be in the context where the user wishes to generate a sound or ring a bell or in the context of celebration, the usage of the second emoji 🎉 would be in the context of celebration. EmojiNet lists "celebration" as a sense form for both the emojis, hence the observed similarity has sound even if we consider the context of use of this emojis.

7 Future Work

Usage of external knowledge has improved the accuracies of various natural language processing tasks and outperformed many state-of-the-art results. Jian et al. [BGL14] have worked on leveraging external knowledge in learning word embeddings which gave better accuracies in word similarity and word analogy tasks. The first set of examples in EmoSim508[7] dataset look more convincing than the results in Table 5 and Table 6; the reason being semantic knowledge helps to us compare the similarity between different emojis efficiently. Using Jian et al.'s work as a reference, we could work on incorporating external knowledge from EmojiNet to our network embedding model which might further improve the accuracies of sentiment analysis and emoji similarity tasks.

Acknowledgement

---

[6] The semantic similarity is the similarity measure obtained using semantic embeddings developed by Wijeratne et al.

[7] https://bit.ly/2GztSR2

References

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[BBS17] Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. Are emojis predictable? *arXiv preprint arXiv:1702.07285*, 2017.

[BCM11] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. In *Social network data analytics*, pages 115–148. Springer, 2011.

[BGJM16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[BGL14] Jiang Bian, Bin Gao, and Tie-Yan Liu. Knowledge-powered deep learning for word embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 132–148. Springer, 2014.

[BRS16] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *LREC*, 2016.

[CLX15] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[ERA+16] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*, 2016.

[FHSM17] Halley Fede, Isaiah Herrera, SM Mahdi Seyednezhad, and Ronaldo Menezes. Representing emoji usage using directed networks: A twitter case study. In *International Workshop on Complex Networks and their Applications*, pages 829–842. Springer, 2017.

[FMS+17] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*, 2017.

[GL16] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[LNK07] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[MGB+18] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[NP12] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[NSSM15] Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. Sentiment of emojis. *PloS one*, 10(12):e0144296, 2015.

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[RRWN11] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

[SM17] SM Mahdi Seyednezhad and Ronaldo Menezes. Understanding subject-based emoji usage using network science. In *Workshop on Complex Networks CompleNet*, pages 151–159. Springer, 2017.

[SPWT17] Amit Sheth, Sujan Perera, Sanjaya Wijeratne, and Krishnaprasad Thirunarayan. Knowledge will propel machine understanding of content: Extrapolating from current examples. In *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*, pages 1–9, 2017.

[TQW+15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[WBSD16] Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran. Emojinet: Building a machine readable sense inventory for emoji. In *International Conference on Social Informatics*, pages 527–541. Springer, 2016.

[WBSD17a] Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran. Emojinet: An open service and api for emoji sense discovery. In *11th International AAAI Conference on Web and Social Media (ICWSM)*, pages 437–446, Montreal, Canada, May 2017.

[WBSD17b] Sanjaya Wijeratne, Lakshika Balasuriya, Amit P. Sheth, and Derek Doran. A

semantics-based measure of emoji similarity. In *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*, pages 646–653, 2017.

[YRS⁺14] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292. ACM, 2014.