wordmoji2vec: Embeddings of Words and Emojis

Diego Bravo
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON M5S 3G4
d.bravovelasquez@mail.utoronto.ca

Abstract

The use of emojis in social media has increased drastically since their inception. Researchers are looking to leverage social media's vast amounts of data to fuel their research. Understanding emojis are their meanings is important to know how to apply and interpret new methods and results. This project created embeddings for emojis and words using the Word2Vec model and evaluated them using the analogical reasoning task. The difficulty of measuring the quality of emoji embeddings is discussed, and it is suggested to create a task for this purpose similar to the analogical reasoning task. The results of the analogical task were compared to similar works and a significant disparity in accuracy was discovered. It is hypothesized that the datasets used are the source of the discrepancy.

1 Introduction

With the rise of smartphones and social media, we can share pictures of our recent trip or let our friends know what we had for breakfast in real time. Our methods of communication are rapidly evolving alongside new technology, adopting whimsical little images that help us convey our ideas in 140 characters or less. Emojis are not just a fad, they are here to stay.

Researchers are turning to social media to use the vast amounts of data generated as fuel for their research. Recent natural language processing techniques rely on word embeddings to boost performance in certain tasks such as sentiment analysis, which can be used to determine if a tweet is positive, negative or neutral. Existing models that produce word embeddings do not account for the unicode representation of emojis, an integral part of the language used in social media. A recent example of using emoji to improve word embeddings is the work by Eisner et al [3]. They showed improved performance in sentiment analysis of tweet by using existing word embeddings of emoji descriptions in the Unicode standard to provide additional information.

This project seeks to uncover some semantic meaning of emojis based on their use in social media by employing the word2vec model on words alongside emojis. Lebduska showed that emojis are culturally and contextually bound, and are open to interpretation [4]. Even so, Barbieri et all showed that a subset of emojis share meaning between languages[**?**]. As such, datasets from two regions will be used to highlight any differences in the meaning of emojis that might exist.

2 Previous Work

Recent natural language processing techniques depend on the distributional hypothesis, which states that words that occur in similar contexts share semantic meaning [8]. These techniques represent words as vectors in a continuous space based on the context in which they appear. Semantic and syntactic similarities are captured in the geometric properties of the vectors. For example, the result of a vector calculation vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vector.

There are two main categories for generating word embeddings each with its benefits and drawbacks. Count-based models, such as Latent Semantic Analysis, leverage global statistics of co-occurring words. Predictive based techniques, such as word2vec, look at co-occurrence in a local window resulting in better performance at analogical tasks. Baroni et al showed that predictive-based methods perform better in various tasks through a systematic comparison [2]. Even so, Pennington et al argues otherwise and proposes GloVe, leveraging global statistic and taking advantage of the local window, resulting in state-of-the-art performance in various tasks including analogical reasoning[7].

In 2013, Mikolov et al proposed two predictive-based methods for generating word embeddings, continuous bag-of-words (CBOW), and continuous skip-gram model. CBOW is computationally faster, but the skip-gram model produces higher-quality vectors [5]. That same year, Mikolov et al added extensions to the skip-gram model to improve the quality of the vectors through phrase learning, as well as optimize it by approximating the softmax using negative sampling[6].

The work most similar to this project was contributed by Barbieri et al [1]. The authors used the skip-gram model on a twitter dataset consisting of 10 million tweets. The dataset was filtered to create multiple variants to determine how the embeddings were affected. The authors filtered hashtags, usernames, punctuation, and even removed all words so that emoji embeddings only depended on other emojis.

Eisner et al took a different approach and produced emoji embeddings by using pre-trained word embeddings on their descriptions in the Unicode standard. The authors showed that for downstream tasks such as sentiment analysis, these embeddings produced better results than embeddings produced by a skip-gram model.

3 Method

Similar to the approach by Barbieri et al[1], this project implements the word2vec model a Twitter data set. Emojis are treated simply as words in order to be able to use word2vec. The main difference is in the evaluation. Barbieri et al created a dataset to evaluate the similarity and relatedness of emojis, while this project uses the analogical reasoning task proposed by Mikolov et al in the original word2vec paper. For qualitative evaluation, analogical reasoning with emojis will be used.

4 Dataset and Preprocessing

The Twitter API was used to gather tweets with geolocation data since they come from real users, filtering out bots and spam[1]. Tweets were gathered from the United States and Great Britain, to compare the usage of emojis between the two regions. Data collection occured during October and November of 2016. From the United States about 11.7 million tweets were collected, and about 10.2 million from Great Britain. Of the 21.9 million tweets, about 5.1 million or 23% included emojis.

The tweets were split into tokens with a modified python port of the CMU ARK Twitter Part-of-Speech Tagger[1]. Modifications were required to properly parse emojis due to updates to the Unicode standard. The dataset was cleaned up by identifying and removing Twitter-specific tokens and punctuation. The tokens in question are at-mentions, URLS, and hashtags. An example can be seen in Figure 1.

Similar to the original word2vec model, the input is a stream of words, thus all of the tokens were concatenated with spaces resulting in a single-line file. A consequence of this choice sentence structure and boundaries between tweets are lost. Beginning or ending words are effectively non-existent.

5 Word2Vec Model

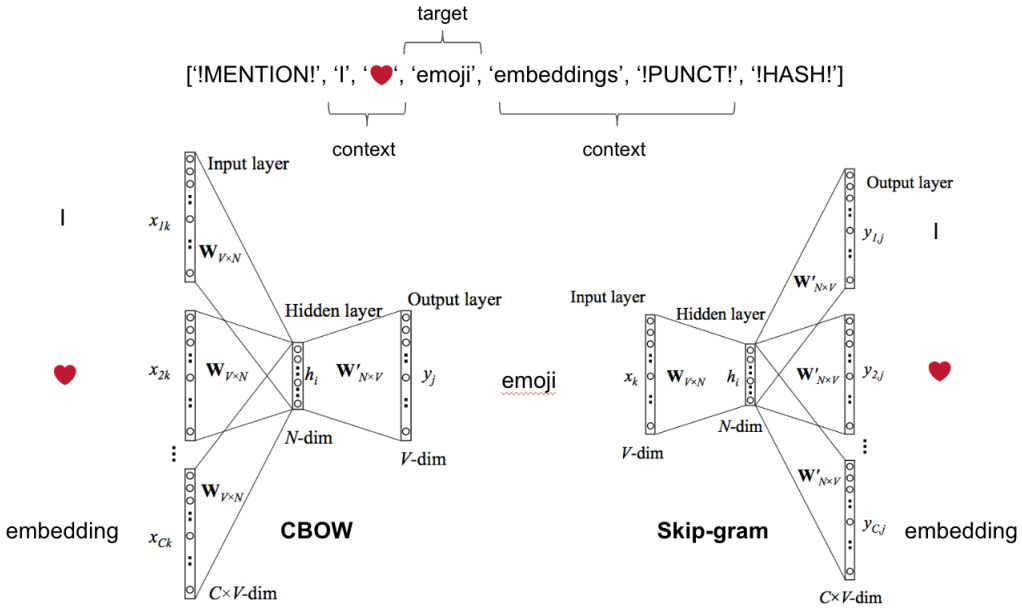
5.1 Skip-gram and Continuous Bag-of-Words

The Word2Vec model uses a single hidden layer, inputs and outputs are one-hot-encoded vectors of the vocabulary. There are two architectures for Word2Vec, which are somewhat opposites. The

[1] https://github.com/myleott/ark-twokenize-py

Figure 1: A tweet is split into tokens as part of the preprocessing step. Some of these tokens are replaced by placeholders, or even removed.

Figure 2: Diagram of the Word2Vec model. A single hidden layer is implemented with inputs of outputs of one-hot-encoding vectors representing words in the vocabulary. In CBOW, the context words are used as input to predict the target word. In Skip-gram, the target word is used to predict its context. Diagrams taken from [**?**] and modified



Skip-gram architecture predicts the context words around a target word, while the Continuous Bag-of-Words (CBOW) architecture predicts a target word given the context words. A context word is a word that is near the target word within a threshold called the window size. The data is a set of word pairs based on the input and output words. In the case of skip-gram, the input word is the target and the output words are the context. For CBOW, the input words are the context, and the output word is the target. The input layer of CBOW is a weighted sum of the context words. The output layer of Skip-gram are multinomial distribution for each context word. A diagram of the two architectures can be seen in Figure 2.

The W and W' different matrices and not tranposes of each other. They are weights between their respective layers. Due to the one-hot-encoding inputi. For a word w_i the operation $W^\top x_i$ is essentially selecting its vector representation.

As such, the objective is to maximize the log-probability of a word given its context, or the context given the word:

Skip-Gram

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c, j\neq 0} \log p(w_{t+j}|w_t) \quad (1)$$

CBOW

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c, j\neq 0} \log p(w_t|w_{t+j}) \quad (2)$$

Where w_t is the input word, c is the size of the context window, and T is size of corpus. The conditional probability is defined using a softmax functions:

$$p(w_O|w_I) = \frac{exp(v'^{\top}_{w_O} v_{w_I})}{\sum_{w=1}^{W} exp(v'^{\top}_{w} v_{wI})} \quad (3)$$

Where v_w and v'_w represent the "input" and "output" vector of w, and W is the size of the vocabulary. Calculating $\nabla \log p(w_O|w_I)$ is computationally expensive as it scales with W. Updating every conditional probability would require the normalizing constant to be calculated each time. Mikolov et al proposed negative sampling to avoid this issue.

5.2 Negative Sampling

Negative Sampling is based on Noise Contrastive Estimation (NCE), which transforms the problem from a predictive model of language to a problem of probabilistic binary classification[**?**]. NCE states that a good model can distinguish between noise and real samples through logistic regression. In a given context, a word is sampled from the true distribution that models the data. Random words are sampled from the noise distribution. The goal is to distinguish the sample that came from the true distribution. NCE is used to learn parameters of a model that estimates the true distribution. The objective is maximized when high probabilities are assigned to the real words and low probabilities to the noise words.

NCE has been shown to approximately maximize the log probability of the softmax, but Mikolov et al proposed a simplification as they are only concerned with learning high-quality vectors[6] and word2vec does not require a full probabilistic model[8]. The objective of Negative Sampling is then:

$$E = \log \sigma(v'^{\top}_{w_O} h) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} [\log \sigma(-v'^{\top}_{w_i} h)] \quad (4)$$

Where the value of h depends on the architectures: for skip-gram $h = v_{w_I}$, for CBOW $h = \frac{1}{C}\sum_{c=1}^{C} v_{w_c}$. The noise words are drawn from a distribution that is a free parameter. Mikolov et al propose a unigram distrbution raised to the 3/4rd power.

6 Quantitative Evaluation

The first evaluation was done on the clean dataset that consists of all tweets collected. Both Skip-gram and CBOW architectures were used and evaluated on the analogical reasoning tasks. The hyper parameters were varied as follows:

- Vector size = $[200, 300, 400]$
- Window size = $[3, 5, 7]$
- Negative samples = $[3, 5, 7]$

The highest accuracy achieved by Skip-gram was 63.6% with hyperparameters VS=400, WS=5, NS=7, and CBOW achieved 58.8% with hyperparameters VS=400, WS=5, NS=[5, 7]. Skip-gram had better accuracy than CBOW on all combinations of hyperparameters, but the run-time was about twice as long. See Tables A1 and A2 in the appendix for the complete results. These results agree with Mikolov et al's observation that Skip-gram produces high-quality embeddings but is more computationally expensive. It was expected for these vectors to not produced state-of-the-art results, as the dataset is not a good source of proper language usage. Mikolov et al used a Google News dataset to showcase the word2vec model.

Table 1: The top 10 emojis in the United States and the top 5 associated words.

Emoji	% of total	Top 5 similar words
😂	17.9	lmfao, keese, lmao, lmaoooo, Lmao
😭	6.2	CDFUUUUU, cryyy, cryingg, cryinnnnn, cantttt
😍	3.9	aviii, lawddd, BADDDDD, excitedddd, Cuteeeee
❤️	3.7	love, I, and, this, for
🙄	2.5	smh, lol, Smh, lmao, jamia
💀	2.2	lmfao, lmao, likaho, LMFAO, chilllllllll
😩	2.2	KollieRedd, mannnnnnnnn, luckyyyy, himm, lordddd
🤔	1.9	hmmm, Hmm, Hmmmm, hmmmm, hmm
🔥	1.6	CuPreme, L.A.X., Fie, fireeeeee, courtlinjabrae
😊	1.5	Kawal, Yumiko, Tereza

Table 2: The top 10 emojis in Great Britain and the top 5 associated words.

Emoji	% of total	Top 5 similar words
😂	16.1	Bwaaaaaaaaahaha, Deaddd, dkm, deadd, ffsssssss
❤️	3.9	you, it, xxx, nicoedesign, this
😭	3.5	ffs, DEADDDDD, chwest, DYINGGGG, goneeeee
😍	2.9	omgosh, Eba, Rosanda, fooood, favvvvv
😊	1.8	Kags, Pic/, Pics/, Bibiana, Spans
👍🏻	1.7	Garin, HAGWE, Njoy, Toony, Chooooooooooooooooooooooooon
😩	1.7	fml, ffs, goneeeee, Whyyyyyyy, omdsss
😘	1.4	xxx, xxxx, Loe&Hugs, Love&hugs, Tylo
😳	1.4	ffs, smh, urghhhhhh, mxim, sozzzz
🎄	1.3	MerryChristmas, Christmasssss, Christmasssssss, Eeeeep, Christmassss

Barbieri et al made their embeddings available on their website[2]. The name of the files suggest hyperparameters used were VS=300, WS=6, NS=3. The embeddings for both the clean and raw datasets were evaluated on the same analogical reasoning task. The clean embeddings had an accuracy of 19.1% and the raw embeddings had an accuracy of 22.4%.

7 Analysis

One of the goals of this project was to compare the usage and meaning of emojis between the United States and Great Britain. A list of the top 10 emojis and their top 5 associated words are presented in Table 1 for the US, and Table 2 for GB. The emojis 😂, 😭, and 😍 are some of the most popular in both regions. This observation is consistent with Barbieri et al's paper[**?**]. Note that the meaning of 😭 consistent between the two regions and contradicts the meaning given to it by the Unicode standard. 😭 represents sobbing, crying, and sadness, whereas in the US and GB it has a similar meaning to 😂. The associated words for these emojis are also quite different between the two regions. Another interesting usage is that of 💀. It seems to be used as dying of laughter in the US, but its usage is not popular in GB.

Analogical reasoning was done with emojis and the results can be seen in Table 3. Since emojis were simply treated as words in this model, relationships between emojis was captured. Though usage of emojis is not well defined and may lead to incorrect relationships captured by the model, there are a few instances that are accurate. For example, query 1 shows that the relationship of gender is accurate if using 👦, but no tso accurate for 👨. Similarly, Queries 6 and 7 show that sadness can be subtracted and another emotion can be added.

Additionally, a t-SNE visualization was created for both regions showing similar clusters. This observation suggests that the meaning of these emojis are consistent between regions. The visualization can be found in Figures A1 and A2 in the appendix.

[2]http://sempub.taln.upf.edu/tw/emojis/

Table 3: Analogical reasoning of handpicked emoji showing variance in quality.

#	Query	United States	Great Britain
1	👱 + ♀ - ♂	👱, 👳, 👲, 👱◻, 🕌	👱, 👱, 👱, 👱, 🛁
2	👱 + ♀ - ♂	👲, 👱, 👱, 👱◻, ⬛	👱, 👱, 🥦, 👲, 👱
3	👯 + 👱 - 👱	👯, 👯, 👯, 👯, 👯	👯, 👯, 👯, 👯, 👯
4	👯 + 👱 - 👱	👯, 🛋, ⇶, 💏, 💏	👯, 💏, 👯, 👯, 👯
5	😿 + 😍 - 😢	😻, aviii, 👿, prettyyyyy, lawddd	😻, omgosh, 👌, faveee, cat_cafe_manchester
6	😿 + 😁 - 😢	😁, 😸, 😀, 😊, purrrrrfect	😆, 😁, 😃, Truestory, 😻
7	👸 + ♂ - ♀	Kween, UUU, Malika, JOC, Princesas	👑, 👛, birthdaaaaaay, bestiee, 👸

8 Discussion

The difference in accuracy between the results of this project compared to the results from Barbieri et al is difficult to explain. On their website, you can download their vectors in a format readable by the gensim library, suggesting they used that library for their analysis. This project also utilized the gensim library. Let's assume they used the default values for parameters they do not mention, such as the subsampling rate, as this project does. Then the main disparity between this project and Barbieri et al's work is the dataset used. The accuracy report above was for the complete dataset (i.e. 21.9 million tweets), whereas Barbieri et al used 10 million tweets.

Do the 11.9 million tweets really make that big of a difference? Not really. In this project, the US dataset contains 11.7 million tweets and achieved an accuracy of 59.3%, and the GB dataset contains 10.2 million tweets and achieved an accuracy of 55.4%. My hypothesis is that the difference in accuracy is inherent to the dataset. Twitter aggregates a tremendous amount of data and their API yields grabs a fraction of that data. The content in the dataset also depends on the state of affairs at the time of collection. For example, the christmas tree is one of the top 10 emojis used in Great Britain. One would expect to see less usage of this emoji during the summer.

Keep in mind that the analogical reasoning task is used to determine the quality of word embeddings and not emoji embeddings. Even if the results of this project achieved greater accuracy, it does not mean that the emoji embeddings are better than Barbieri et al's. The qualitative evaluation showed that usage of emojis between the US and GB are similar in quantities but their associated words are quite different. Considering Lebduska's claim that emojis are open to interpretation [4] and Barbieri et al's observation that emojis have different usage and meaning across languages and regions[**?**], it begs the question of how emoji embeddings should be evaluated.

In the original word2vec paper, Mikolov et al proposed the analogical reasoning task to evaluate the word embeddings. This task consists of "questions" that measure if the model can capture relationships between words such as capitals, currencies, and lexical categories. For example, comparatives (bad worse, big bigger), superlatives (bad worst, big biggest), present-participles (code coding, dance dancing), the list goes on. These are relationship between words that arise from linguistics.

As far as I know, there are no guidelines or structures to emojis are there are for language. Emojis don't fit similar categories nicely, and thus a new list of categories that may be used to evaluate emojis is needed. One of the categories that is easy to see is the concept of gender. Based on how the emoji is constructed according to the Unicode standard, the concept of gender can be easily tested. There are emojis constructed by combining the emoji of man or woman and another emoji such as the medical symbol to create doctor, or a school to create a teacher. Another template is to use a base emoji such as a construction worker or cop, that is genderless, and add the male or female sign emoji.

Instead of relying on the analogical reasoning task, Barbieri et al tackled the challenge of quantitatively evaluating emojis by creating a similarity and relatedness task which they called the EmoTwi50 dataset. This task consists of 50 emoji pairs, half of which were manually chosen and the other half were randomly chosen. Each pair was assessed by 8 participants who gave a score on equivalence of the emojis (similarity) and if the emojis could be used together (relatedness).

The EmoTwi50 dataset and Barbieri et al's method of evaluation is very similar to the work done by Levy et al[**?**]. Levy et al used the WordSim353 dataset which consists of 350 word pairs and 29 participants. This type of evaluation for emojis is flawed based on the fact that emojis are open to interpretation. The participants may score the pair of emojis in the EmoTwi50 dataset differently than how they are used in the wild, but that does not mean the the usage in the wild is less correct. Interpretation of emojis may also change over time, and the datasets like EmoTwi50 would have to be updated. In essence, this type of evaluation is measuring whether or not the similarities and relatedness in the EmoTwi50 dataset exist in the wild.

For future reasearch in emoji embeddings I think it would be helpful to have a consistent dataset in order to better determine if a model is better than another. Using different datasets may result in inconsisteint results for the same model and evaluation as shown in the quantitative analysis. Due to changes in Twitter's Terms of Service[3], it is difficult to share a dataset of tweets. The new changes prevent sharing Twitter data unless it is the Tweet ID. Thus every project would require to download the tweets based on the list of Tweet IDs.

9 Conclusions

In this project, embeddings of emojis alongside words were created using the Word2Vec model. The analogical reasoning task was used to evaluate the results of the SKip-gram and CBOW architecture, showing that Skip-gram produces high-quality embeddings but is also slow. These results are consistent with literature. The results of this project were also compared with Barbieri et al's work, showing some inconsistency in the accuracy measured. The dataset used to create the embeddings is believed to be the cause of the disparity. The difficulty in evaluating the quality of emoji embeddings was discussed, suggesting a task similar to the analogical reasoning task may be created, but it may not capture the ever-changing nature of the meaning of emojis.

[3] http://inkdroid.org/2014/08/31/on-archiving-tweets/

References

[1] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *Language Resources and Evaluation conference, LREC*, Portoroz, Slovenia, May 2016.

[2] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[3] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. emoji2vec: Learning emoji representations from their description. *CoRR*, abs/1609.08359, 2016.

[4] Lisa Lebduska. Emoji, emoji, what for art thou? `http://harlotofthearts.org/index.php/harlot/article/view/186/157`, Nov 2014. Online. Accessed 2016-11-09.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[7] 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, Doha, Qatar, October 2014. Association for Computational Linguistics.

[8] Tensorflow Team. Vector representations of words. `https://www.tensorflow.org/versions/r0.11/tutorials/word2vec/index.html`. Online. Accessed 2016-11-09.

10 Appendix

Table A1: Accuracy results of the analogical reasoning task for the clean dataset

Vector Size	Window Size	Negative Samples		
		3	5	7
200	3	59.4	61.1	62.8
	5	59.6	61.2	61.4
	7	57.0	59.4	60.8
300	3	61.1	62.2	63.3
	5	61.8	61.6	63.2
	7	60.9	61.4	62.3
400	3	60.7	61.6	62.9
	5	62.0	62.7	63.6
	7	60.1	61.4	61.2

Table A2: Accuracy results of the analogical reasoning task for the placeholder dataset

Vector Size	Window Size	Negative Samples		
		3	5	7
200	3	52.4	53.5	52.5
	5	52.5	53.7	53.8
	7	51.7	52.8	54.6
300	3	53.7	55.8	55.9
	5	55.1	56.6	57.2
	7	55.6	56.9	57.4
400	3	56.2	56.3	56.9
	5	56.4	57.7	58.7
	7	56.8	58.8	58.8

Figure A1: t-SNE visualization of emoji embeddings from Great Britain. Notice the cluster of flags near the top left and the center, sad smileys near the top, and surprised smileys near the bottom. The cluster of male, female, and family emojis are located just to the right of the center. Not all emojis renderings are supported so they show up as a question mark or a group of 2 or more emojis.

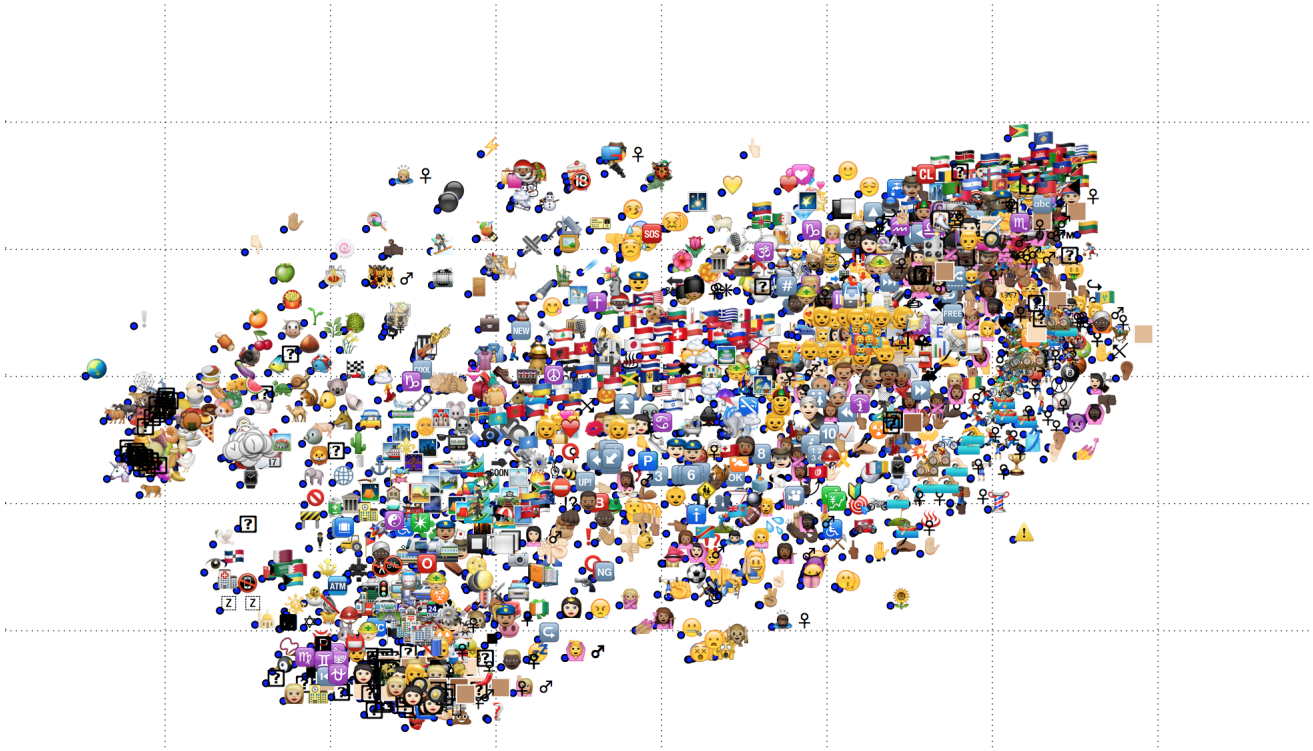


Figure A2: t-SNE visualization of emoji embeddings from United States. Notice the cluster of flags near the top, happy smileys near the bottom, and suprised smileys near the left. The cluster of male, female and family emojis seemt to be located in the upper-left corner, but are covered by other emojis. Not all emojis renderings are supported so they show up as a question mark or a group of 2 or more emojis