

# Emojis Dataset

Yann MENTHA,  
DLab, EPFL, Switzerland  
January 15, 2021

**Abstract**—Emojis have become a central part of democratized digital communication over the last few decades, and being able to accurately represent them in an appropriate semantic space has therefore become a crucial aspect of Natural language processing. When it comes to extracting the emotion and meaning out of a text, such a representation needs to reflect the commonly accepted, potentially multiple, meanings of an emoji - ideally we should be able to embed the emojis in the same space as the words themselves.

In this report, we present an overview of the creation of the Emojis dataset, which consists of human labeled single-word descriptions of the 1325 most common emojis (30 annotations per emoji on average). We then proceed to an in-depth presentation of the various tools and methods utilised to collect the data.

## I. INTRODUCTION

The use of emojis is a great way to add valuable insights on the emotion and context of a text, in conversations or social network posts for example. However, in many cases the emoji chosen by the writer does not share the significance of the rest of the sentence, but rather completes it. For example, in the sentence "No way, did you go to the park? 😄" no word relates to a euphoric feeling, and the same sentence "No way, did you go to the park? 😡" can carry a totally different message. Therefore, using the words surrounding an arbitrary emoji, as in a skip-gram model [1] does not present the most promising method to model them. In the Emoji2vec paper [2], researchers tackled this issue by using a few expert-labelled sources for emoji descriptions. However, this approach might miss some commonly-accepted meanings, as the same emoji may be used in many contexts with a different meaning each time. In addition to this, the utilisation of certain emojis and their respective meanings might vary over time.

Hence, a dataset containing human-labeled descriptions of the most common emojis could be used to allow researchers to train NLP models on data that better fit the meaning attributed to each emoji by the general population. As no such dataset existed in the literature at the time of writing, we aimed to fill this gap.

## II. MODELS AND METHODS

The approach we followed consists of the following parts:

- (a) **Selection of emojis:** data-driven manner of selecting which emojis to keep in the dataset
- (b) **Quantity needs:** data-driven manner of choosing the number of answers per emoji
- (c) **Data Collection:** techniques used to create the forms used to gather data and monitor workers in real time
- (d) **Post Processing:** data cleaning and refactoring
- (e) **Validation:** t-sne visualization of a word2vec/bert embedding created on-top of the Emojis dataset

### A. Selection of Emojis

Although more than 3859 UTF-8 emojis existed at the time of writing (emoji pypi package, v. 0.6.0), only a small proportion of them represent the majority of emoji use on social networks (Fig. 1) (in the same way that we tend to use only a small portion of the english vocabulary on an everyday basis). Therefore, we decided to focus more of our resources on the analysis of more frequently used emojis, rather than producing a balanced dataset which may miss data for the most critical emojis and give too much weight to marginal ones. The covered ratio of emoji use, sorting emojis by their frequency, is shown on Fig. 2 .

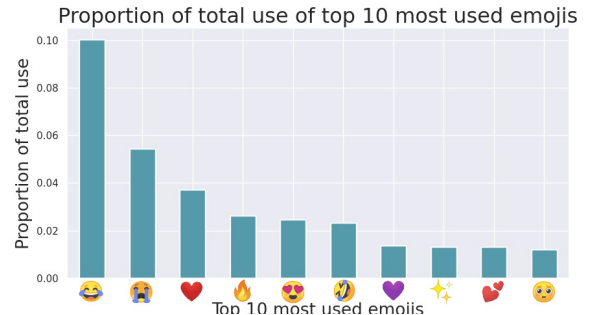


Figure 1: **Top 10 emojis** : ratio of total use for the top 10 most used emojis on 1Gb of twitter data. These 10 emojis account for 31.72% of the total use. Data extracted from 1Gb of random english (u.s) twits in the year 2018

As Twitter data does not represent the only context in which our dataset should be useful, we performed the selection of emojis by taking into account frequency and practicality. (Removal of those emojis representing hours, flags, letters, and numbers) Furthermore, we aimed at keeping clear of gendered and skin-toned emojis by utilising only their corresponding neutral versions (yellow for skin) and mapping these emojis back to their neutral version.

Our final emojis selection covers 94.7% of the total emojis use of our twitter data, and represents 34.34% of all emojis.

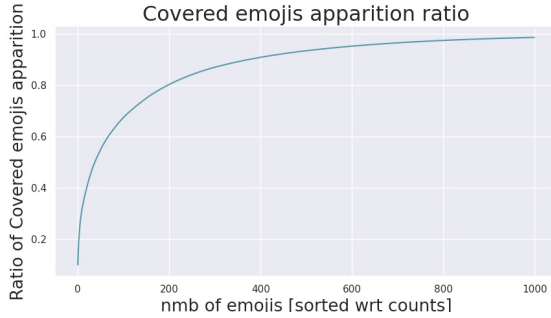


Figure 2: **Covered emojis apparition ratio** : the vast majority of emoji use is represented by only a small subset of emojis. The top 600 most used emojis covers > 95% of total emoji use.

### B. Quantity needs

Depending on their nature, emojis can require more or less words to describe them accurately. For instance, an emoji as simple as 🔥 could intuitively be described by "fire" or "burn" whereas 😞 would present a broader lexical field, including words such as "confused" "skeptical" or "thinking".

As such, we had to define a data-driven limit on the number of annotations to collect per emoji: gathering a large amount of annotations for 🔥 could turn out to be pointless just as gathering too little data for the 😞 could not represent its' true distribution. ( 😞 was described by 13 distinct words by 30 annotations, whereas the vocabulary used for 🔥 consisted of only two words - "fire" and "hot")

### Distribution shift

The point is to keep collecting data as long as new information is added to the existing distribution, or, in other words, as long as we observe a shift of the distribution in question. We therefore selected the convergence of word distribution for each emoji as a metric.

First, a pilot dataset of 12 emojis and 160 annotations per emoji was gathered. Each such annotation was assumed to be statistically independent given that each worker's ID is ensured to be distinct. The fact that each such annotation was made at the same moment in time was not taken into account in this assumption. Once this was done, we computed the trajectories for every emoji: such a trajectory consists of a random ordering of the words used to describe a given emoji in the pilot dataset. We computed the Jensen-shannon (JS) divergence between  $f_n(w)$  and  $f_{(n+1)}(w)$  with  $f_n(w)$  the distribution of words including all words up to  $n \in \mathbb{N}$  in the trajectory. The experience was repeated several times in order to reduce the variance due to the stochastic nature of such trajectories. Figure Fig. 3 shows the Jensen-Shannon divergence for 20 of these trajectories per emoji for a set of emojis selected to represent several levels of complexity.

The distribution shift induced by the addition of a single new word becomes significantly smaller as the number of samples  $n$  present in that distribution increases. Indeed, the JS convergence of a trajectory presenting a new word at every step still converges, as can be seen on the purple line of figure Fig. 3, where such a trajectory was artificially computed. The divergence evolution of every emoji was therefore normalized with respect to this artificial trajectory in order to obtain relevant insights concerning the convergence of distribution shifts as  $t_{norm}(n) = \frac{t(n) - r(n)}{r(n)}$  with  $t(n)$  a trajectory,  $t_{norm}(n)$  its normalized version and  $r(n)$  the artificial reference. Fig. 4 highlights this behaviour.

Despite the significant noise level in the signal due to the limited size of the pilot dataset, it seems to be that several emojis converge after the threshold of 120 annotations per emoji. Due to practical constraints, it was necessary to reduce this number to 30 annotations per emoji in the final dataset, which might not accurately represent the true distribution of certain complex emojis. 🍕 was described with 2 words, whereas 📌 had a vocabulary size of 13, in line with the obtained results.

### Number of words per annotation

A crucial point concerned the number of words a worker could give for one emoji: many annotations can be necessary to describe complex emojis, but useless for simpler ones. We therefore tested two frameworks: the first allowing 3 words per emojis, the second allowing 1 word only. As similar quality of results was obtained in both cases, the second framework being preferred for its simplicity.

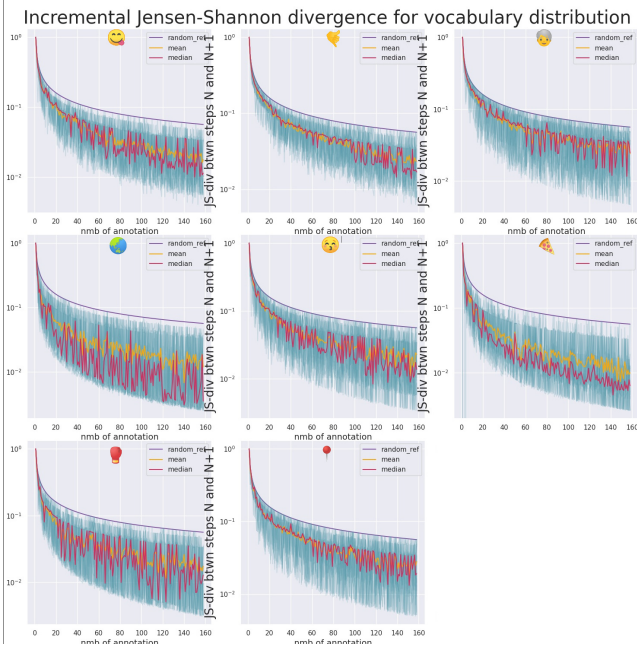


Figure 3: **Incremental JS distance**: the Jensen-Shannon (JS) divergence of the distribution before and after the addition of an annotation to a specific emoji vocabulary tends to decrease for two reasons. First, as the number of annotations increases, the newly added sample represents a fraction decreasing as  $\frac{1}{n}$ . As such, its effect on the distribution shift will be less and less impactful. The second reason is the salient one : we observe a stabilization of the distribution as the most used words are covered by the gathered vocabulary. An annotation using a word already seen many times will therefore have less impact than a brand new word. The purple curve represents a synthetic word distribution that adds a new word at each step and serves as a reference.

### C. Data Collection

In order to collect human-labeled data, we chose the Amazon MTurk crowdsourcing marketplace (Mturk) for its convenience and the presence of a python library and api (boto3), allowing automation of the task’s creation and control. The annotations had to be collected through forms presenting the following properties:

- On-the-fly answer format checking
- Confirmation code checking
- Cross-browser compatible emojis formats.
- Automatic form creation from emojis list

As Mturk did not present such native features, we combined it with Google Forms (Gform) using Appscript

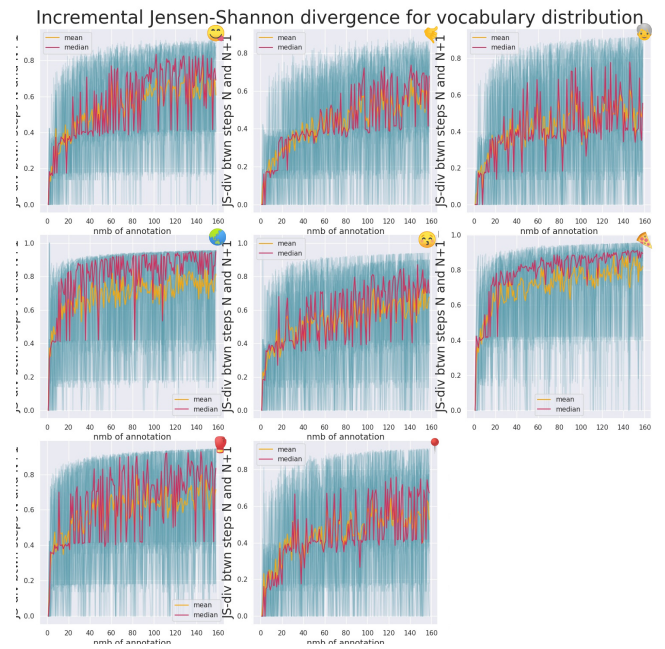


Figure 4: **Normalized Incremental JS distance**: similar plot as in Fig. 3, where each curve was normalized with respect to the artificial curve in order to get rid of the converging behavior due to the scaling of the distribution.

to automate the form creation. Every MTurk task was then linked to a corresponding Gform.

### Mturk2Gform

In order to ensure a minimum diversity in the source of the data, it was necessary to limit the number of forms one worker could answer. Fig. 5 Neither Mturk nor Gform provided such control: we therefore implemented it along with all features cited above in a new python package: Mturk2gform. Notably, the library allows for synchronization of the Mturk framework with Gforms, automation of Mturk HITs creation/handling, and control of the number of forms answered per worker. (cf Fig. 6). Though data quality control could have been carried out on-the-fly using Mturk2gform, all workers inputs were accepted fpostprocessingor the sake of Mturk requester reputation: bad feedback from workers makes it undoubtedly less likely for future forms to be answered.

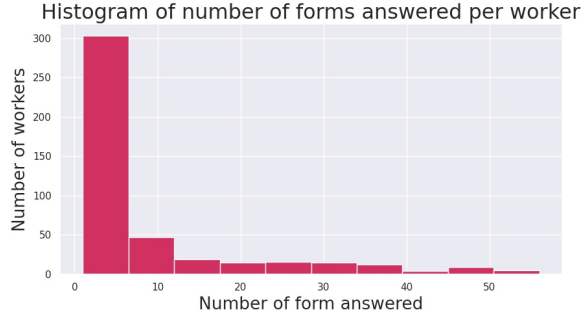


Figure 5: **Number of forms answered per worker:** The high number of workers answering a small amount of forms ensures some minimal diversity in the data. Mturk2gform was used to set a maximum limit of 50.

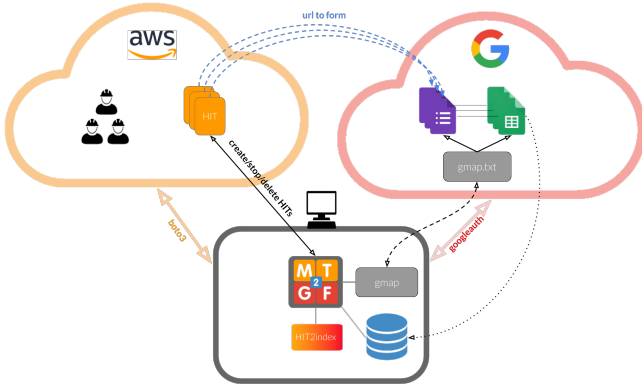


Figure 6: **Mturk2gform:** Mturk2gform (mt2gf) allows one to synchronize the mturk framework using google forms. A one-to-one mapping between Mturk HITs and google forms indexes is maintained (hit2indx). Once the forms are created, mt2gf creates Mturk hits allowing workers to access the gforms via a link. During the time that workers use to fill out the forms, mt2gf regularly downloads local versions of the results for quality checking/tagging of fraudulent workers.

Information regarding age, gender and mother tongue were additionally collected from the workers.

The overall characteristics of the dataset can be found in Table I and Fig. 7

Table I: Emojis Dataset characteristics.

Total nmb of emojis	1325
Nmb of forms	118
Nmb of emojis per form	10
Max nmb of forms per worker	56

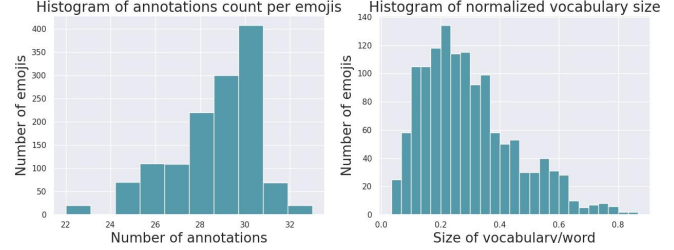


Figure 7: **Annotations and vocabulary size per emoji:** (left) Due to removal of invalid inputs (cf Sec. II-D) some emojis ended up with less than the expected 30 annotations. (right) Plot of the normalized vocabulary: emojis having received less annotations are more likely to have a smaller vocabulary size. We therefore display the vocabulary size per annotation, showing a mean of 0.3 words/annotation, corresponding to a vocabulary size of approx. 10 per 30 annotations.

#### Display format

The pilot datasets revealed that the visualisation of UTF-8 emojis could be problematic depending on the browser type and version: some browsers would display a default rectangle instead of the actual emoji, making it impossible for workers to describe it in a meaningful manner. We therefore decided to display emojis as images rather than using the plain text utf-8 format. As no dataset mapping emojis to their corresponding image existed, we created one by automatically capturing a screenshot of the sequentially displayed emojis in a jupyter notebook.

#### D. Postprocessing

A post-processing pipeline was implemented to detect malformed data. Each of the 3 steps presented in this subsection can be optionally turned on or off when generating the usable version of the dataset.

##### Repeated answers

It was observed that a slight minority of workers used the same answer for all of the emojis in a single form, regardless of their aspect. We therefore discarded workers whose utilised vocabulary size was inferior to a given ratio for the maximum number of answers in the form.

##### Honeypots

One commonly used emoji sampled from the honeypots emoji list was placed in every form: this way, answers falling outside of a Levenshtein distance of 3 for any word in the expected Honeypot vocabulary were discarded.



### Spelling

As spelling mistakes within the answers could not be checked server-side in the Gforms, it was necessary to carry this out as a post-processing step. A misspelled word would therefore go through the following steps:

- 1) If the word is used by multiple workers to describe the emoji, it is considered valid.
- 2) If the word is a valid english word (enchant python library used as reference) it is kept as is.
- 3) We compute suggestions for the word in question, that is, valid words that are within a maximum Levenshtein distance of 2. If some of these suggestions have been used by other workers, we keep the suggestion that has been used the most frequently.
- 4) If the word can be split into several valid words, we keep the splitted version separated by a ' ' (ex: "uparrow" becomes "up arrow")
- 5) If the word is still considered as invalid, we reuse the suggestions of step 3 and select the one which obtains the highest score in the logits computed by an uncased BERT for masked ML model, using as context all of the valid words gathered for the emoji in question
- 6) If no suggestion was available at step 5, we keep the word as is.

```
Modified: riceball --> rice ball (disassembled2)
Modified: ricebowl --> rice bowl (disassembled2)
Modified: ufo --> ufos (corrected)
Modified: omlette --> omelette (corrected)
Modified: omlett --> molest (corrected)
Modified: frenchfries --> french fries (disassembled1)
Modified: vegetable --> vegetable (cross_suggested)
Modified: raddish --> radish (cross_suggested)
Modified: sweetpotato --> sweet potato (disassembled1)
Modified: cakepops --> cake pops (disassembled2)
Modified: kebob --> kabob (cross_suggested)
Modified: popcakes --> pop cakes (disassembled2)
Modified: smores --> s mores (disassembled2)
Modified: kabbobs --> kabobs (corrected)
Modified: smores --> s mores (disassembled2)
Modified: smore --> s more (disassembled2)
Modified: bento --> bent o (disassembled2)
Modified: stricker --> tricker (corrected)
Modified: bottlecap --> bottle cap (disassembled1)
Modified: softserve --> soft serve (disassembled1)
Modified: creamee --> creamer (corrected)
Not found: icecrem
```

Figure 8: **Word correction:** sample extracted from the log of the word correction algorithm

The results shown in the Sec. III Result section were obtained by discarding the workers whose vocabulary size was inferior to 0.8 of the maximum size. Honeypots and spelling corrections were used as well.

### E. Validation

As the purpose of this dataset consists of training NLP models, we used one of the multiple applications that are able to run on top of it as a quality check. For comparison purposes, we decided to reproduce the t-sne visualization of embeddings obtained in the emoji2vec paper [2] with both the original data and ours. This dimension reduction is performed on embeddings created using a word2vec model. Ideally, the word classification test (determining whether a given word is associated with an emoji on a dataset with negative sampling) present in this paper should have been run as well: unfortunately, the experiment turned out to be challenging to reproduce. We therefore kept ourselves to the t-sne embeddings. The same experiment was performed using the last attention layer of an english-trained BERT model for embedding in order to check if the quality difference between our dataset and the emoji2vec baseline would be reproducible.

### Parameters

The parameters for the word2vec pipeline were kept identical as in the word2vec paper. (40 epochs with a 1-1 negative-positive sampling, 300 input dimension and 300 output dimension). All t-sne plots were performed with 5000 iterations and with a perplexity of 30 (sklearn manifold library).

## III. RESULTS

A sample of the final dataset is displayed in Table II. Table III and Table IV show subset of results gathered for emojis presenting the most and least diverse vocabulary respectively. Table V shows a subset of the demographic information gathered along with the dataset.

The T-sne embeddings reproduced using the original emoji2vec data can be seen on Fig. 11 and Fig. 11 (word2vec and bert methods respectively): the same is done for our own dataset on Fig. 10 and Fig. 12.

Table II: **Random sample**: example random sample of the emoji dataset















WorkerID	FormId	Duration	emoji <sub>i</sub> <i>index</i>	emoji	word
A2CK0OXMPOR9LE	41	52.0	725		graph
A3774HPOUKYTX7	52	458.0	257		rose
A2D2JX8R0QU9G4	51	454.0	642		worker
A3VL1CBZ3BGQK7	28	153.0	83	?	medical
A39MKVROUZ1UWR	95	460.0	208		morning
AB66CTVQ90RCV	92	181.0	1023		walking
A2KX8SX5M47GPX	123	98.0	1255		vessel
A3OP9SCMH0KPJL	20	455.0	223		moon
AOMFEAWQHU3D8	92	793.0	957		upset
A1S8KMGKJACGVO	36	318.0	638		woman
AF0M3066S5UF1	44	89.0	755		mail
A34NDXO56MBC3D	87	554.0	1210		minuscule
AOMFEAWQHU3D8	45	179.0	1235		knee
A33DVD9ZMS0XXN	28	90.0	64		aries
AMA18W8F60Y2J	65	616.0	1320		chair

Table III: **Varied emoji**: Example of words sampled from an emoji presenting a rich vocabulary











WorkerID	FormId	Duration	emoji <sub>i</sub> <i>index</i>	emoji	word
A337Y4X67PY4QI	83	117.0	186		key
ATP2BJPDK4K2B	83	233.0	186		box
AJRY9ALX8069Y	83	84.0	186		rewind
A2T007HZK66WM	83	168.0	186		cc
A2MOKIEQZ0OF2M	83	NaN	186		clip

Table IV: **Constant emoji**: Example of words sampled from an emoji presenting a poor vocabulary

WorkerID	FormId	Duration	emoji <sub>i</sub> <i>index</i>	emoji	word
A1N1ULK71RHVMM	42	432.0	492		tiger
A2T1LNI80EPOQR	42	466.0	492		tiger
A1J1ZM062PH3FN	42	277.0	492		tiger
A1DD23J1WBGQUU	42	363.0	492		tiger
AOMFEAWQHU3D8	42	279.0	492		tiger

TSNE embedding of e2v using w2v

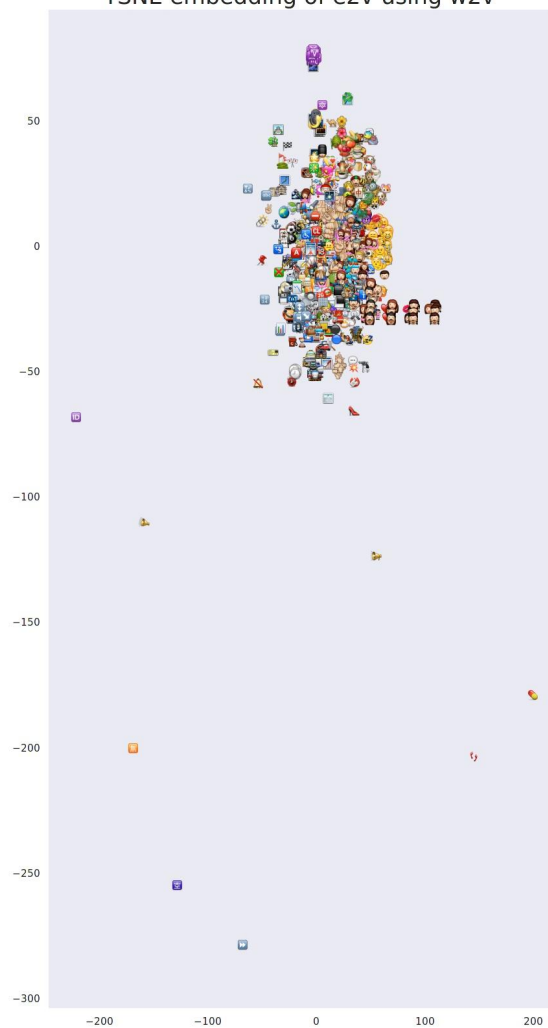


Figure 9

TSNE embedding of em\_dataset using w2v

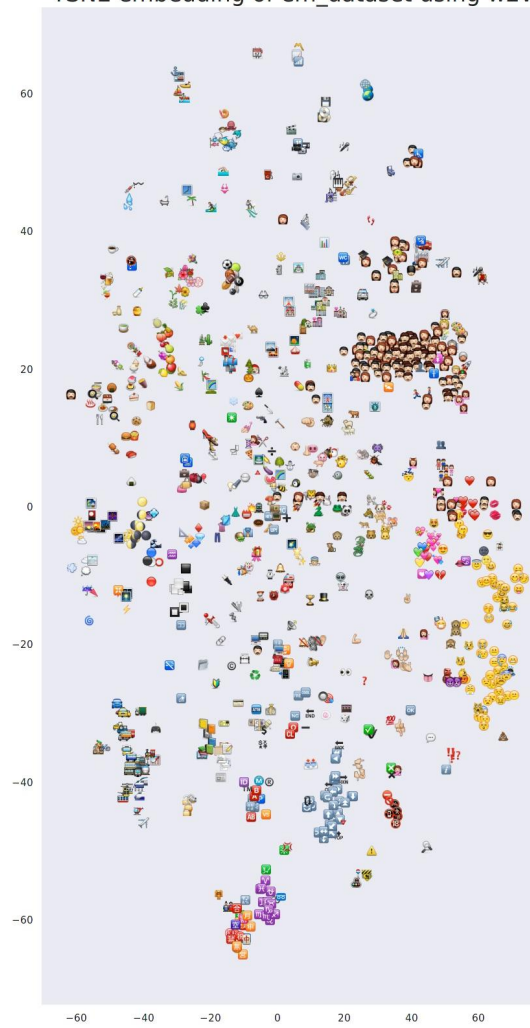


Figure 10

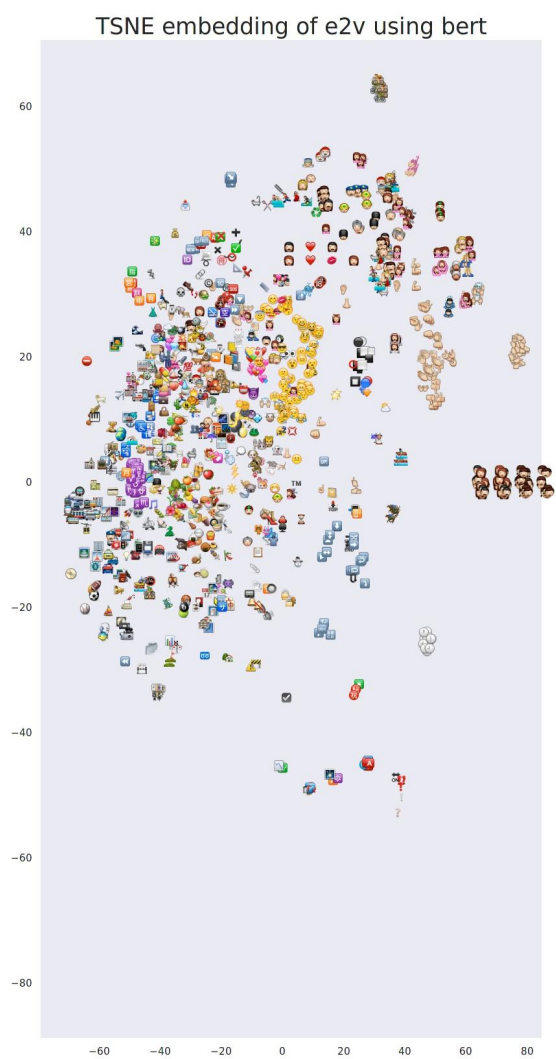


Figure 11

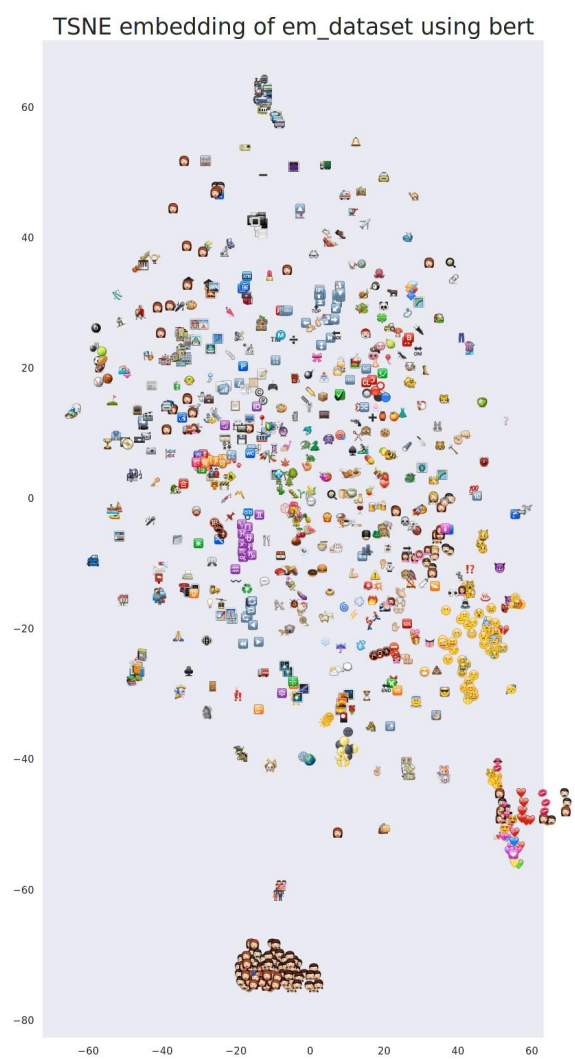


Figure 12



Table V: **Workers Info**: Subset sampled from the demographic workers information table generated with the dataset

WorkerID	Age	Gender	Mothertongue
ALML8V38FDV0	55.0	Female	english
A31UXXZVI3U4E2	31.0	Male	english
AQ53YJDPDDLZ	40.0	Male	english
A143XRCI1YXAFE	28.0	Male	english
A1KR49KPV5J9BM	36.0	Male	english

- [2] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. emoji2vec: Learning emoji representations from their description, 2016.

















emoji	word	emoji	word
	{twirling, loop, repeat, coils, spiral, wire, loops, curlicue, wave, curl, sign, libra, loopy, glasses, swirly, curls, pig nose, swirl, forever, symbol, cord, zodiac, astrology}		{ten}
	{shock, kid, think, worried, praise, surprised, pray, child, worried boy, awake, boy, nervous, confused, sad, confusion, tense, astounded, angry, alert, thinking, reading, stunned, pain}		{lipstick}
	{work, analyst, business, corporate, profession, teacher, woman, leader, present or, co, meeting, businesswoman, worker, statistics, scientist, women, chart, presentation, executive, accountant, stocks, reporter, graph}		{spider}
	{vacation, seasons, night, grass, painting, moon, corn, meditate, daytime, flower, sun, pond, flower vase, sunny day, ferns, frame, spa, cornfield, evening, web, picture, calm, nature}		{butterfly}
	{cacti, island, grow, grass, emerald city, hills, church, unknown, building, cactus, factory, bamboo, tower, an, crowd, ogre, alien, moldy, city, paw, fingers, ship a, plant}		{key}
	{pop, excitement, star, pow, conversation, flash, statement, warning, anger, shout, saying, quotation, crack, spark, comic, boom, yelling, wham, thought, chat box, speech, thinking, explosion}		{family}
			{bee}
			{carrot}
			{family}
			{family}
			{rainbow}
			{spoon}
			{apple}

Figure 13: Vocabulary of 6 emojis with the richest vocabulary (> 23 words) and 18 emojis with the smallest one.

#### IV. DISCUSSION

Although some emojis are not properly rendered due to the font used (Apple Color Emoji.ttf, many emojis mapped to only "man" or "woman"), it seems like our dataset led to slightly better embeddings than the emoji2vec baseline, which converged to a compressed cluster. However some inconsistencies persist: zodiac signs are spread across the image, food cluster is not very centered, etc. Bert embeddings lead to slightly more defined clusters in this regard.

Overall the produced dataset looks usable for many NLP applications: if more data is required in the future, the mturk2gform framework would facilitate the gathering of additional annotations.

#### REFERENCES

- [1] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.