Zero Shot Emoji Prediction using Multimodal Emoji Embeddings

Stanford CS224N Custom Project

Isabelle Lim
MS&E Stanford University
limxri@stanford.edu

Janani Balasubramanian
MS&E Stanford University
janani98@stanford.edu

Hailong Chen
MS&E Stanford University
hlgchen@stanford.edu

Abstract

With the advent of social media, emojis have become a key feature of online textual communication, yet they are not well understood from an NLP point of view. Existing literature focuses on a small range of emotional emoji, with less focus on exploring the long tail of emoji. Zero-shot prediction, a practically useful capability that can cater to new emojis, is often overlooked. We construct a multimodal model that can leverage emoji image and description information to generate meaningful embeddings for all available emojis and combine these with a fine-tuned Sentence-BERT to make emoji predictions which can handle unseen classes. We outperform previous models and can predict for unseen emojis. Further, we highlight difficulties in evaluating zero-shot performance and suggest some ways to overcome those along with an analysis of different ways emojis are used today.

1 Key Informations

- Project Mentor: Gabriel Poesia (poesia@stanford.edu)
- TA Mentor: Vincent Li
- External Collaborators (if you have any): NA
- Sharing project: No
- The project code: https://github.com/hlgchen/emoji_prediction

2 Introduction

The rise of social media has introduced a new way of communication where meaning can be composed by combining textual messages with emojis. Analyzing conversations without considering emojis would be a loss of valuable information as emojis are often used to either convey the sentiments and emotions of the user, or clarify ambiguous text phrases. Despite their prevalence as a language form, emojis and their underlying semantics have not been widely studied from a Natural Language Processing (NLP) standpoint, and the interplay between text-based messages and emojis has not been explored in-depth. Current approaches tend to focus on emoji usage relating to sentiment expression, or restrict the emoji they consider to the most commonly used emoji which also tend to be emotional emoji, thus neglecting the long tail of largely non-emotional emoji. Multimodal approaches to emoji prediction combining information from text and image on social media posts were found to enhance emoji prediction [1], [2]. However, to the best of our knowledge, emoji embeddings have been constructed with emoji name and description data and have neglected to use emoji image information. Finally, zero-shot emoji prediction based on textual input, an important capability for practical emoji recommendation systems, has been explored by very few papers [2].

We construct a multimodal model that leverages emoji image and description information obtained from emojipedia.org and hotemoji.com, and uses a finetuned Sentence-BERT model to make

emoji predictions. We outperform [2]'s model on general emoji prediction, show that emojis are used in 2 ways (sentiment expression and word emphasis), and propose a practical system of predicting emojis that yield respectable performances across different settings.

3 Related Work

3.1 Emoji Prediction

Interest in emoji prediction has grown over the years due to the usage of emoji exploding on social media. Existing literature achieved high accuracies with various state-of-the-art models from bi-directional LSTM [3] to BERT [4], showing that there is a non-spurious relation between words and emojis that can be learned by automated systems.

Most existing papers on emoji prediction tend to use either a handcrafted emoji set [5] or the most frequent emojis in their respective datasets (e.g. [3] uses the top 20, [6] uses the top 100 and [4] uses the top 300). This means that emoji sets considered in existing literature are limited in size and topics, as the most commonly used emojis are typically emotional emoji such as facial expressions. [2] is the only exception that considered a wide range of over 1000 emoji including the often overlooked long tail emojis. Our work chooses to adopt [2]'s dataset as we argue that limiting our emoji set means neglecting a large amount of information that can be conveyed through non-emotional emoji.

[3], [2] consider a multimodal approach to predict emojis. The former combines image embeddings generated by Resnets with text embeddings generated by FastText to predict emojis based on the picture and text caption associated with an Instagram post, while the latter combines the confidence scores generated from a text-based bi-directional LSTM and image-based convolutional network to predict emojis based on text and image inputs associated with a tweet. Both papers show that a multimodal approach outperform their unimodal counterparts based only on textual contents, suggesting that textual and visual content embed different but complementary features of emoji usage. However, to the best of our knowledge, there is no existing work that currently leverages the rich visual information embedded in an emojis visual image for emoji prediction. Certain emoji usage correlates heavily with the emojis visual image rather than the emojis name. For instance, 🍆 is often used to represent a penis due to the similarity of the image's shape with the penis shape. Thus, we distinguish our work by supplementing emoji embeddings with emoji image information and not just emoji name and description.

3.2 Zero-shot learning

Zero-shot learning is when you train a model that can predict outcomes for unseen classes. This has become increasingly popular when the is focus on transferable learning to recognize unseen classes. There are two popular approaches used for zero-shot models: (1) Embedding models [7],[8]where data is projected in embedding space and a non-parametric approach is used to predict the class; (2) Generative models [9] are used to create fake samples to predict for unknown classes. The embedding approach is seen to follow a class-inductive approach while the generative models follow a class-transductive approach.[10]

A popular approach for zero-shot learning is embedding the data (text/image) into a vector space. A model is trained to embed the data with the objective of minimizing distance between the ground truth class vector and the embedding of the data. The embedder be used for unseen classes as the it is generally trained on train classes. When embedding the data there are multiple approaches which include creating embeddings manually[11], generating automatic word vectors like Bag-of-Words [12] or using context aware embedding approaches such as GloVe[13] and Word2Vec [14]. Further, Augmenting the information while training the embedding and classifier proves to be better at zero-shot learning [15]. This inspired us to consider augmenting the information of the emojis by scraping the data from `emojipedia.org` and `hotemoji.com`.

Image2Emoji [16] tries to zero-shot predict emojis for visual data. They combine textual and image information to predict the emojis. They used pretrained models like Word2Vec to embed the text along with generating probabilities for visual concepts and find the similarity with the output vector by measuring the distance from the emoji names vector. Our approach varies from this approach by

using BERT model to generate embeddings leveraging transformers along with a CNN to extract information from the image.

4 Model Architecture: SEMBERT (Sentence-Emoji BERT)

Our zero-shot emoji prediction model consists of three main components as seen in figure 1:

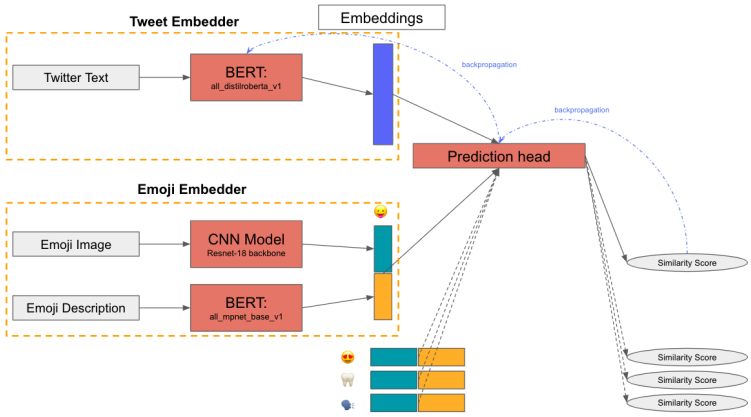

Figure 1: Overview of the Proposed Model

4.1 Emoji Embedder

Here, we create embeddings for all the emojis in our dataset. The idea is to create multimodal embeddings where the embeddings incorporate the image of the emoji along with its name and description. Thus, the emoji embedder has two components:

Emoji Image Embedder: Here, the image is mapped to a vector using a convolutional neural network.The backbone for our image embedder is a resnet-18 that is pretrained on image-net. The output dimension of the last linear layer giving us a 200 dimensional vector v_i. The resnet-18 is trained by comparing the vector from the CNN with the word vector obtained from passing the description and emoji name through a pretrained Glove Model t_i. We create positive and negative samples where we pair the emoji vector with the right emoji description vector for positive samples and incorrect pairing for negative samples. We maintain a ratio of 1:9 for positive to negative samples ($y_i = 1$ if pairing is positive). The model uses cosine distance to measure the similarity between the CNN vector and the text vector.

A contrastive loss function is used to train the CNN. We minimize the following function:

$$L = \sum_i (\frac{1}{2} y_i d(v_i, t_i)^2 + \frac{1}{2}(1 - y_i) \max(1 - d(v_i, t_i), 0)^2)$$

Emoji Name and Description Embedder: Here, we use a pretrained *all-mpnet-base-v1* model to convert the emoji name, its aliases and its corresponding description (concatenated with "\u25A1" and passed through the model) to vectors of size 768. The data we scraped ourselves from `emojipedia.org` and `hotemoji.com`).

To get the final emoji embedding we concatenate image embeddings with textual embeddings. This is done for all emojis in our dataset. Note that none of these models are finetuned on the emoji prediction task.

4.2 Tweet Embedder

Each tweet is embedded using a SentenceBert [17] (*all-distilroberta-v1*) and a vector of 768 is created. In contrast to the emoji embedding models, we finetune this model on our emoji prediction task.

4.2.1 Prediction Head

The prediction head projects the 768+200=968 dimensional emoji embeddings and the 768 dimensional tweet embedding into a 500 dimensional vector space (by applying a linear transformation to each). In our final model versions we use a dropout layer with 0.2 dropout probability before feeding the embeddings into the linear layers. In the 500 dimensional project space the dot product between emojis and tweets is taken. A softmax layer normalizes all dot products (for better ensembling) and the emoji embedding(s) with the highest similarity score(s) are returned as the predictions of the model. An illustration of the model architecture can be found in Figure 1.

4.3 Baselines

We develop two baseline models that make predictions based on the *similarity* of tweets and the descriptions.

Sentence-Baseline: we use the emoji name, alias and description embeddings as described earlier and use the same model (*all-mpnet-base-v1*) to encode tweets. We normalize the euclidian length of all embeddings to 1 and measure the cosine similarity between the two. The emoji with the highest score is the predicted emoji.

Word-Baseline: we try to check whether emoji names appear in the given tweet (if the tweet contains the word "crocodiles" we would expect the emoji with the name "crocodile" to have a high similarity score. As Glove word vectors are sensitive to misspellings, we decide to use a small SentenceBERT for encoding each word (*all-MiniLM-L6-v2*). Taking the average of individual embeddings of each word in emoji names we get the query. We compare this to the embedding of each individual word in a tweet. The maximum similarity over all words in the tweet with the emoji is the final similarity score between an emoji and a tweet.

5 Experiments

5.1 Data

Dataset	# of Observations	# of Emojis
Train Set	11.3M	1122
Validation Set	0.9M	1068
Full Test Set	1M	1064
Balanced Test Set	10K	1063
Zero-shot Test Set	1.1M	99

Table 1: Dataset constructed from [2]. Note that we also remove emojis representing numbers as we do not have emojipedia+hotemoji data for those.

The main dataset is based on the Twemoji dataset obtained from [2], which contained 13M tweets in the train set, 1M tweets in the validation set, and 1M tweets in their full test set. They constructed a class-balanced test set that consists of a subset of 10k tweets where no single emoji appears more than 10 times. For our purposes of zero-shot prediction, we randomly select 30 emojis from the 10-60 most frequently used emojis and 69 emojis from the rest, for a total of 99 emojis to be removed from the original train and validation set, but not from the test set, and used these 1.1M observations of the 99 zero-shot emojis to form the Twemoji zero-shot test set (any tweet where one of the 99 emojis appear at least once). Thus, we obtain a modified Twemoji dataset with statistics as can be seen in Table 1.

Secondly, to increase our range of emoji available, we constructed our own zero-shot test set by scraping 50K tweets from twitter that contain one of 279 emoji types that were not in the training set.

Finally, in order to create our emoji embeddings, we scraped data from `emojipedia.org` and `hotemoji.com` to get emoji description, images and metadata for 1.7K emojis.

5.1.1 Data preprocessing

When preprocessing tweets, researchers tend to play around with 2 areas: (1) removing links and mentions, and (2) limiting the dataset to tweets with a minimum number of words. [4] find links within tweets contain useful information. However, many researchers including [2] chose to remove links and mentions, neglecting potentially useful information that may be conveyed by links which motivated us to conduct experiments on this. Additionally, [1] limited their dataset to 4 or more words per tweet as they argue that tweets with low word count tend to be noisier. Hence, we ran an experiment with our Sembert models to verify if using a limited dataset of tweets with 3 or more words would lead to better model performance. This has not been the case, which is why we decide to leave the tweets generally unprocessed. See Table 6 in Appendix A for experimental results.

5.2 Evaluation method

For evaluation, we use top-k accuracy with k=1, 5, 10 and 100. As emojis can be used interchangeably with no definite correct answer, top-1 accuracy might not be the best metric for evaluating the task and top-5 or top-10 accuracy may be better. We defined a true match the way Cappallo et al. (2018)[2] defined their true positive. If there is even one emoji amongst the top k prediction that was correctly predicted among all the emojis used by a user in a tweet, we consider it to be a correct prediction.

5.3 Training details

Loss function: We train all of our models with a loss function that was inspired by triplet loss.

Let $\mathcal{E}$ be the set of all emojis. $|\mathcal{E}| = N$
Let $\hat{h} \in \mathrm{R}^N$ be the predicted logits for each emoji given a tweet (dot product of projected embeddings). $\hat{h}_i$ (the ith entry in $\hat{h}$) is the logit for the emoji with emoji-id i. Feeding this through a softmax we get probabilities $\hat{p}$ for each emoji. Let y be the non-empty set of emoji ids that appear in a tweet.The loss for a single tweet is:

$$\mathcal{L} = -(\sum_{i \in y} \log(\hat{p}_i) + \sum_{j \in \mathcal{E} \setminus y} \log(1 - \hat{p}_j))$$

The batch loss is the average loss over all individual tweet losses. The model tries to predict higher probabilities for emojis in y and lower probabilities for $\mathcal{E} \setminus y$. This loss is propagated till the Tweet embedder for each batch.

Optimization: We use the Adam optimizer and trained our models over chunks consisting of 128k datapoints each. We stopped the training when there was no improvements on the validation set after 3-5 consecutive steps. We use a learning rate of 5e-5 along with a batchsize of 64 and exponential learning rate decay of 0.95 after every chunk.

5.4 Sub-model architectures

Given the base architecture as described in Section 4, we experiment with the architecture and training methodology. In the subsequent Results section we will report the performance on the following models:

Sembert: has the same architecture as described in Section 4.

Sembert dropout: we add a dropout layer before the linear projections of embeddings in Sembert with dropout probability of 0.2.

Sembert balanced: we take the trained Sembert model and finetune it using the balanced training data.

Sembert balanced dropout: taking the trained Sembert balanced model, a dropout layer with probability of 0.2 was added before the linear projections.

Emsemble dropout: each of our models outputs a probability distribution after the softmax layer. We take the average over the predictions of Sembert dropout and Sembert balanced dropout.

6 Results and Discussion

6.1 Results on Full Test Set

Table 2 provides a comparison of the performance of our models with our baseline as well as Cappallo et al. (2018) [2] model for predicting emojis on the full Twemoji test set.

Our Sembert models feature significantly higher scores than [2] and our own baseline. [2] only reports the Top-1 accuracy for their model trained on unbalanced training set. The significant improvement can be attributed to the use of State-of-the-art transformer models instead of Bi-directional LSTM as the language model and secondly, the augmentation of emoji information with data from `emojipedia.org` and `hotemoji.com` has led to information rich vectors.

Model	Top-1	Top-5	Top-10	Top-100
Cappallo et al. (2018) [2]	21.4	-	-	-
Cappallo et al. (2018) balanced	13.0	30.0	41.0	84.0
Sentence Baseline	1.3	4.0	6.0	21.7
Sembert	26.2	49.1	60.7	92.8
Sembert dropout	26.5	49.5	61.0	91.6
Sembert balanced	20.3	37.9	47.6	83.7
Sembert balanced dropout	20.8	38.0	47.5	83.0
Ensemble dropout	26.5	48.8	59.8	90.9

Table 2: Accuracy in percent on the test set: Our models significantly performs better than Cappallo et al. (2018) [2] model. 21.4 is the accuracy reported by the authors on a non-balanced dataset.

We observe that when we experimented with a dropout of 0.2 for Sembert trained on balanced/un-balanced data, we see that the models with dropout performed slightly better or comparably to the models without dropout. This holds true when evaluating models on the remaining datasets, hence we only consider models with dropout.

6.2 Results on Balanced Test Set

Table 7 provides a comparison of the performance of our models against the baseline and [2]'s model on the balanced Twemoji test set. Here, our models outperformed our baseline but yielded worse results than Cappallo et al.(2018)[2].

Model	Top-1	Top-5	Top-10	Top-100
Cappallo et al. (2018)	19.9	-	-	-
Cappallo et al. (2018) balanced	35.1	48.3	54.7	87.7
Sentence Baseline	5.3	13.3	17.8	42.8
Sembert dropout	21.3	37.3	45.0	70.2
Sembert balanced dropout	30.8	46.5	53.5	76.9
Ensemble dropout	30.5	47.5	54.9	78.4

Table 3: Accuracy in percent on the balanced testset: our models underperform the author's model in this task as we optimize for zero shotting.

The difference in performance may be attributed to the removal of observations containing the 99 emoji types while training our models, resulting in loss of more than 1M training data points compared to Cappallo's model and an increase in unseen emojis for our model. As the balanced test set contain only 10K observations (no more than 10 for each emoji), the presence of 99 unseen emojis are likely to have a significant impact on the performance of our model.

6.3 Results on Zero-Shot Set

Table 4 provides a comparison of the performance of our models against our baseline on the Twemoji zero-shot test set, unrestricted and with output restricted to the zero-shot emoji set only. We only

consider tweets for prediction where there is exactly one emoji (which is a zero shot emoji). Note that our accuracy measure is equivalent to the standard accuracy measure if there is only one "true" emoji for a tweet. Our Sembert and Ensemble models outperforms our baseline on the zero-shot Twemoji test set. It yields a decent accuracy on the restricted prediction. The model is able to generalize for these 99 emojis.

Since emojis can be used interchangably, there are many substitutes for different emojis. Thus, it is expected that a model will rather predict emojis that it has seen during training. Looking at top-5 or top-10 accuracy or restricting the emojis the model should take into consideration (by removing the top-10 most frequent emojis in the dataset or even all emojis seen during training as done in Table 4) gives a more realistic and fair picture of a model's zero shot capabilities.

	unrestricted				restricted			
Model	Top-1	Top-5	Top-10	Top-100	Top-1	Top-5	Top-10	Top-100
Sentence Baseline	0.9	4.8	7.6	23.4	7.9	20.9	29.3	1
Sembert dropout	2.8	9.9	19.2	69.0	43.0	71.4	82.1	1
Sembert balanced dropout	2.0	7.6	14.4	60.0	34.7	67.0	76.8	1
Ensemble dropout	2.6	9.5	17.9	66.4	41.7	71.7	81.1	1

Table 4: Accuracy in percent on the 99 zero-shot emojis

6.4 Ensembling

Our ensemble model Ensemble dropout features comparable accuracies to the best model available on every dataset. This means that the model is able to predict even rare emojis reasonably well, if it has seen these during training. Given that Sembert is not able to make good predictions on the balanced dataset and Sembert balanced is not able to make good predictions on the full test set with frequency information about emojis, it comes as a suprise that it is possible to do well on both dataset at the same time.

6.5 Results on Custom-Scraped Zero-Shot Set

Table 5 provides a comparison of the performance of our models with our baseline on the our own scraped zero-shot test set, with emojis considered for prediction unrestricted and restricted to only those that have never been seen before. Surprisingly here only our baseline model performs well. None of our other models are able to make predictions with high accuracy. We analyze these results and propose a solution in the subsequent Section.

	unrestricted				restricted			
Model	Top-1	Top-5	Top-10	Top-100	Top-1	Top-5	Top-10	Top-100
Sentence Baseline	3.9	8.5	11.3	29.7	6.3	12.9	16.9	41.9
Sembert dropout	0.0	0.1	0.1	33.6	0.1	11.1	28.1	83.1
Sembert balanced dropout	0.0	0.3	0.7	19.5	2.4	10.9	20.3	73.1
Ensemble dropout	0.0	0.2	0.03	23.2	1.4	11.5	25.4	81.4

Table 5: Accuracy in percent on custom scraped zero shot set

7 Analysis - Towards Practical Emoji Prediction

7.1 Different ways of using emojis

Analysing the results on our custom scraped zero-shot set, we observe that emojis can be used in two ways:

- **Sentiment expression:** In this case, the emojis capture the sentiment of the tweet. For example, "I won the championship 🍾". Here, understanding the emotions and sentiments of the tweet is critical in predicting the correct emoji.
- **Word or idea emphasis (literal usage of emojis):** In this case, the emoji is used as a way to put emphasis on certain words or ideas used in the tweet. For example, "Seals are like the dogs of the sea 🐕". This would need a model more focused on particular words' meaning present in the sentence and finding the emoji that best represents the word or idea.

As the Twemoji dataset was highly skewed towards emotional emoji and largely consists of tweets that used emoji for sentiment expression, the Sembert and Ensemble models which were trained on Twemoji data could perform better than baseline on the Twemoji zero-shot test set as the observations were skewed towards sentiment expression. However, in our scraped zero-shot test set, we considered emoji outside of the Twemoji dataset, the observations consist of largely non-emotional emoji and emoji usage was skewed towards word or idea emphasis. This would explain why the baseline, which is based on sentence/word matching, could significantly outperform our trained Sembert and Ensemble models.

To validate this hypothesis we conduct a small experiment and make predictions for 3 artificially created tweets for each emoji:

- I like *emoji-name*
- I am angered by *emoji-name*
- *emoji-name* is very lit, I wanna see it

We define the correct label as the emoji-name for each emoji. Indeed, we find that our trained Sembert models feature much lower accuracy scores, showing that they put too much focus on the sentiment part of the sentence, not being very good in predicting word or idea emphasis usage of emojis. The results can be seen in Appendix B Table 7.

7.1.1 EREC: Emoji Recommendation

Based on the above understanding, we combine baseline model predictions with trained model predictions to create a practical Emoji Recomendation system (EREC). Because there is no correct way of using emojis, we believe that in a practical system the model performance for top5 and top10 emoji prediction should be more crucial. EREC recommends a user specified number of prediction from our two baseline models (Sentence Baseline, Word Baseline) if the predicted cosine similarities are above a certain threshold. These predictions are the literal emoji predictions. The rest of emojis needed for top k prediction are taken from the Ensemble dropout model, which are more likely sentiment expressing emojis.

This model not only accounts for the use of emojis to capture the emotional sentiments associated with the piece of text (captured by ensemble SEMBERT) but also the word to emoji mapping (literal mapping) done by the baseline both at the sentence and word level. Examples of its predictions can be found in Appendix C.2 Its performance on our 4 testsets can be found in Appendix C.1 Table 8.

8 Conclusion and Outlook

In this project, we use state-of-the art transformers along with CNNs to build a model capable of doing seen-class and zero-shot emoji prediction. Our models perform comparably or better than [2]'s models on the dataset they used.

We find that ensembling can create models that perform well on low-frequency emojis without performance loss on high-frequency emojis. We note that zero-shot emoji prediction is difficult due to the trained models' bias towards predicting seen emoji, and propose evaluating zero-shot prediction while ignoring top-k most frequent emojis for prediction (extreme case is k= number emojis in the training set). Importantly, we find that emojis are used in two different way - sentiment expression, and word emphasis - and models trained on empirical data perform stronger on the former while word-matching models tend to perform stronger on the latter. This motivated us to develop EREC which is an ensemble of SEMBERT which does well with sentiment expression and the baseline that handles word emphasis, making the predictions more practical and versatile. The introduction of emoji image information potentially led to enhanced embeddings leading to better performance.

With the understanding that emojis can be used in different ways, future work can explore creating datasets for each emoji use case and leveraging the corresponding dataset for the desired use case (e.g. emotional emoji usage for sentiment analysis). There is also room to explore the extent to which emoji image information improves emoji prediction, especially for most recently released emoji with sparse emoji description.

References

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

[2] Spencer Cappallo, Stacey Svetlichnaya, Pierre Garrigues, Thomas Mensink, and Cees GM Snoek. New modality: Emoji challenges in prediction, anticipation, and retrieval. *IEEE Transactions on Multimedia*, 21(2):402–415, 2018.

[3] Francesco Barbieri, Miguel Ballesteros, Francesco Ronzano, and Horacio Saggion. Multimodal emoji prediction. *arXiv preprint arXiv:1803.02392*, 2018.

[4] Weicheng Ma, Ruibo Liu, Lili Wang, and Soroush Vosoughi. Multi-resolution annotations for emoji prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6684–6694, Online, November 2020. Association for Computational Linguistics.

[5] 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. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[6] Weijian Li, Yuxiao Chen, Tianran Hu, and Jiebo Luo. Mining the relationship between emoji usage patterns and personality. In *Proceedings of the international AAAI conference on web and social media*, volume 12, 2018.

[7] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[8] Hongguang Zhang and Piotr Koniusz. Zero-shot kernel learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7670–7679, 2018.

[9] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289, 2018.

[10] Jiayi Shen, Haochen Wang, Anran Zhang, Qiang Qiu, Xiantong Zhen, and Xianbin Cao. Model-agnostic metric for zero-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 786–795, 2020.

[11] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009.

[12] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1):43–52, 2010.

[13] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural and Information Processing System (NIPS)*, 2013.

[15] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. *arXiv preprint arXiv:1903.12626*, 2019.

[16] Spencer Cappallo, Thomas Mensink, and Cees Snoek. Image2emoji: Zero-shot emoji prediction for visual media. 10 2015.

[17] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

A Supplementary Experimental Results

Model	Top-1	Top-5	Top-10	Top-100
Sembert	26.2	49.1	60.7	92.8
Sembert-min3	25.6	48.1	60.0	92.4
Sembert-m3-cleaned	21.7	44.6	56.6	91.3

Table 6: Results of model trained with different types of processing evaluated on a testset with the same processing. Sembert-min3 is a model trained with tweets that have at least 3 words. Sembert-m3-cleaned is a model trained on a dataset in which each tweet has at least 3 words and link, hashtags and mentions are removed.

B Different ways of using emojis - Experiment

We conduct a small experiment and make predictions for 3 artificially created tweets for each emoji, we define the correct label as the emoji-name for each emoji:

- I like *emoji-name* (table column like)
- I am angered by *emoji-name* (table column anger)
- *emoji-name* is very lit, I wanna see it (table column lit)

Model	like	anger	lit
Sentence Baseline	95.14	93.98	92.76
Word Baseline	79.89	79.5	76.19
Sembert dropout	7.57	2.38	1.27
Sembert balanced dropout	36.46	26.69	17.96

Table 7: Accuracy on emoji usage artificial tweets.

C EREC supplementary information

C.1 EREC performance

Dataset	Top-1	Top-5	Top-10	Top-100
Twemoji full test set	26.5	45.0	57.4	90.8
Twemoji balanced test set	30.5	46.3	54.9	79.0
Twemoji zero-shot test set	2.6	8.6	15.7	66.9
Scraped zero-shot test set	0.0	8.4	9.1	28.5

Table 8: Results of EREC. The top-5/top-10 accuracies are similar to the best models we have for each dataset.

C.2 Example Sentence

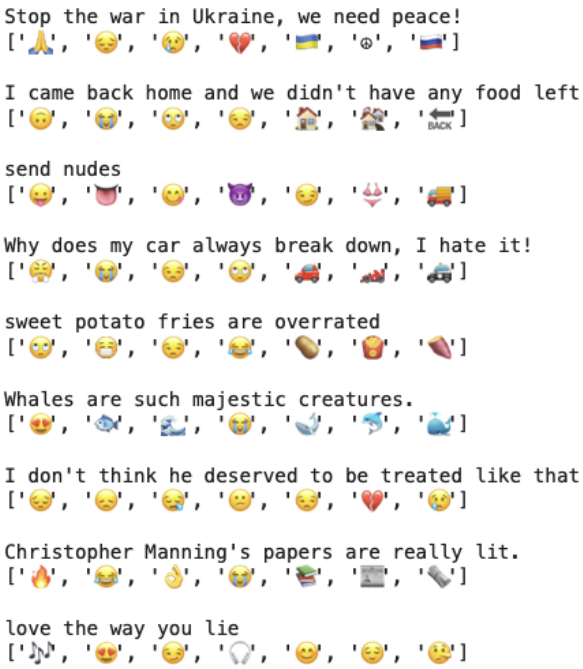

Figure 2: Example predictions of EREC