

Future Emoji Entry Prediction Using Neural Networks

Helen Wilfred Raj and Santhi Balachandran

School of Computing, SASTRA Deemed University, Thanjavur, Tamilnadu, India

Article history

Received: 07-07-2019

Revised: 02-08-2019

Accepted: 09-08-2019

Corresponding Author:

Helen Wilfred Raj
School of Computing,
SASTRA Deemed University,
Thanjavur, Tamilnadu, India
Email: helen@cse.sastra.edu

Abstract: In today's world, textual data has made momentous progress in social media. The rise of digital communication via text has paved the way to emoji, a pictographically represented way of expressing emotions. In digital communication, Emoji gives a visual appeal to the text, which improves communication and new vistas of exchange and creativity. While emoji entry prediction based on text is well optimized, based on the neural network model, predicting the future emojis from images is not so easy due to lack of knowledge on the same. While effective models already exist for generating text descriptions of images, less attention has been given to models of symbolic description. We have used two models for predicting emoji from images, convolutional neural network architecture for image classification and an emoji2vec embedding into word2vec model. We have also done a sentiment analysis of the text for predicting future emoji labels. Our model captures the relation between emojis in an optimized way. This model has optimized the search time for future emoji entry predictions from images.

Keywords: Emoji, Emoji Entry, Images, Neural Network, Sentiment Analysis

Introduction

In today's world, information creation, sharing of information, incubating ideas, expressing feelings are fostered through social media. The rise of social media over the past years has dramatically changed the way people communicate with each other through the internet. Due to digital communication, textual data has become a principal aspect of this area. But textual data is not primarily restricted to text. Instead, it extended into new fields such as emoticons. Emoticons are nothing but characters put together to form an expression depicting their actual meaning by a complete shape like structure. Some emoticons require a large number of characters to exhibit expressions, which increases the complexity of emoticons usage (Janssens *et al.*, 2013). One of the examples is Shrugging emoticon (-\ (") _/-), which is slightly complicated using a large number of characters to form the expression (Weir *et al.*, 2014). Such emoticons take up a lot of users' time to indicate a single expression. That's the reason most people do not use emoticons in their text or social status compared to others.

This desire to express beyond the text paved the way to Emoji. While emoticons use characters to form pictures, emoji is itself pictographic. While emoticons are still text, emoji is represented as a visual icon. With the help of emojis, digital communication can be done effectively by converting the emotions to facial expressions. This led to the massive growth of emojis in popularity and number,

over recent years. Emojis solve the problem of the inability to express feeling, emotions and gestures as the communication can be done not only by words but also through picture characters.

Different companies represent emojis in different formats respective to the platforms. They use different styles to represent the same emoji in different types for attracting users. There are different patterns of usage of emoji-like emotional use for communicating feelings, decorative use for adding visual appeal to text or reaction use, etc. Text entry is well-optimized based on language and touch models, with the use of soft keyboards. But, such an outline is not available for designing emojis. As the number of emojis is increasing, there is a need to optimize the search of emojis in a faster manner for the end user's usage.

Deep learning has proven to be effective in predicting the appropriate emojis through classification (Comanicu and Najafizadeh, 2018). Future emoji entry prediction, based on text, is made with neural network models such as Skip gram or emoji Similarity model by similarity measures. But, predicting emojis from images is difficult due to the lack of similar information (Cappallo *et al.*, 2019). This work builds a NN model to predict future emoji labels from images. There are two models employed in this study. One model is used for image classification and the other one is word vector representations based on emoji embeddings. Emoji embeddings are Standard Unicode embeddings for each

emoji. The Unicode Standardization defines a mapping of a character code to an abstract emoji description.

Though traditional NNs can scale images, they require a huge number of parameters to represent images, which leads to overfitting. But deep neural networks consist of more than two hidden layers, which increase the level of abstraction to scale the images. Among numerous methodologies of deep neural networks, Convolutional Neural Network (CNN) has a major influence in the field of image classification. In this study, CNN is used to classify the image with its interleaved set of feed-forward layers.

This model shows detailed relationships between emojis by similarity models such as Cosine similarity or Euclidean distance. This model uses MSCOCO 2017 dataset for image classification, which consists of 5k images and Google-News-vectors for word vector representations. Visualization of emojis is done using t-Stochastic Neighbor Embedding (TSNE), which helps us to identify the cluster of emojis and outliers. High dimensional datasets cannot be plotted directly. TSNE reduces the number of dimensions to 2 and captures the local structure of data effectively. It also clusters the data which are distinguishably visible. Further, sentiment analysis of the text is done to predict future emoji entries. It also predicts the sentiment (mood or emotion) of the text using RNN.

Emoji Prediction

While the growth of emoji is fast, there is a need to optimize the emoji for easy access. While data and techniques are available for non-emoji (text) such as prediction or classification, there are no such methods available for emoji entry.

Several methods have been proposed for prediction of future emoji from text such as tweets etc., by Similarity Modeling or Skipgram model but not for prediction of emoji entry from images (Chablani, 2017). The work deals with the prediction of the next emoji entry label from images. The method proposes CNN for image classification and W2V for word vector representations. Also, this model describes the similarity between images by Cosine similarity and similarity between emojis by Euclidian distance. The similarity between emojis is visualized by T-SNE plot. Also, sentiment analysis of the text is performed to predict future emoji inputs using RNN.

The main principle of this model is to optimize the search time for future emoji entry predictions from images. This model gives faster access time to users for entering emojis.

Literature Review

Several prediction methods have been employed initially for future entry prediction of emoji by different researchers. Several methods are found in the literature for

the above. A mathematical model was suggested for visual search and selection time in linear menus (Bailly *et al.*, 2014). They used Search Decision Pointing (SDP), a regression model to determine the position and total selection time of the item. SDP uses four predictors and seven parameters to determine selection time. One of the parameters, the Pointing component, is based on Fitts' law, which predicts the items closer to the top for faster to select.

Total selection time, T is defined by the formula (1) as:

$$T(\theta) = T_s(\theta) + T_d(\theta) + T_p(\theta) \quad (1)$$

Where:

$\Theta = (l, t, P)$ = a vector containing: Length of the menu l , position of the target t and vector giving the number of previous encounters with the items

T_s = The time spent in serial search
 T_d = The time spent in directed search
 T_p = The pointing time

Fitts' law was used to predict the top first five objects and to minimize the selection time (Bi *et al.*, 2012). A new optimization technique for keyboard layouts based on Pareto front optimization was demonstrated (Dunlop and Levine, 2012). They used two important metrics for this optimization: Minimizing the finger travel distance to maximize the entry speed by using Fitts' law and Maximizing familiarity through similarity function such as Euclidean distance or cosine similarity etc.

Fitts' law predicts the time taken by users to select the particular target. For example, Fitts' law implies that the nearer and bigger a target is quicker to tap. It determines how long the user takes to move his/her finger to a position above the key and how long it takes to tap the key. The mathematical formula (2) for Fitts' law (M) is represented as:

$$M = a + b \log_2(D/W + 1) \quad (2)$$

Where:

D = The distance of the target from the starting position
 W = The width of the target
 a and b = Are the constants dependent on physical characteristics
 M = The Fitts' law rate

However, it fails to express the semantic relationship among items. Also, multi-dimensional analysis can't be done. Selection time occupies more time in case of a large number of items.

The optimization of emojis has been described by Similarity modeling (Domin *et al.*, 2017). They suggested two models for predicting next emoji entry from the text (tweets). One is the Jaccard similarity model from annotations and the other one is Skip gram implementation using corpus dataset.

The first method was based on deriving emoji similarity from annotations (tags) by using Jaccard Similarity (Han and Kamber, 2011) as a quantifier. Since all posts or tweets contain a tag, they categorized the set of emojis by entering the tag and selecting the matching emoji. The annotations of emojis are used to find out the similarity between them.

The Jaccard similarity coefficient is defined as (3):

$$J(T \text{ Emoji} A, T \text{ Emoji} B) = \frac{|T \text{ Emoji} A \cap T \text{ Emoji} B|}{|T \text{ Emoji} A \cup T \text{ Emoji} B|} \quad (3)$$

Where:

- J = The Jaccard similarity coefficient
- T = $\text{Emoji } A$ is the set of annotations for $\text{Emoji } A$
- T = $\text{Emoji } B$ is the set of annotations for $\text{Emoji } B$

In the second method, the emoji model was constructed using Skip gram NN. They trained the model using corpus dataset with one input, hidden and output layer. Any input vector such as word or emoji passed to input layer where hidden layer consists of a matrix made up of words from tweets as rows. The output layer is a soft max regression classifier, which predicts the top ten emoji labels which have the highest probability. Zero shot emoji prediction by Image2Emoji model and emoji scoring was conceptualized and demonstrated (Cappallo *et al.*, 2015).

Conceptual Model

For the prediction of future emoji entries from images, two NN models are implemented. A NN is an interconnected group of neurons that uses a mathematical or computational model for processing the information. Neural networks are used in various fields such as Sequence recognition, pattern recognition, compression, filtering, clustering, image recognition, etc., a convolutional neural network architecture (Convnets) for image classification and Word2Vec model for word vector representation with emoji2vec embeddings. The overall architecture for this model is given in Fig. 1.

Convolutional NN Architecture, ConvNets, takes COCO images as the input layer. Convnets are made up of neurons that have learnable weights and biases. Unlike a regular NN, the layers of ConvNet have neurons arranged in 3 dimensions, namely width, height, depth. ConvNet architecture has four important layers, Convolutional Layer, Normalization Layer, Pooling Layer and ReLU layer shown in Fig. 2. Max Pooling layer is the output layer which gives descriptions about images in context with image id of COCO dataset in CNN.

This model consists of stacked instances of sequential layers where each instance has four main CNN layers, whereas the last instance of a sequential layer consists of average pooling layer instead of max- pooling layer.

However, the final Convolutional layer uses a dropout of 0.5. The Convolutional layer has a convolutional filter to produce a feature map for inputs. ReLU layer applies a maximum function to all values in the input volume. Max pooling layer uses that maximum value from each cluster of neurons at the prior layer. The descriptions about images obtained from CNN are passed as an input to the W2V model. Word2Vec model is a three-layer neural network model with one hidden layer. W2V trains a linguistic context of words and produce word embeddings for the data.

W2V was trained using Google-News-word vectors and E2V embeddings. E2V consists of Unicode embeddings for all available standard emojis. The hidden layer consists of google news vectors, where input words are trained accordingly and mapped to Unicode embeddings to predict emoji labels. The output layer of E2V is a Soft max regression classifier which predicts the top five emoji labels which have the highest value. This is followed by a sentiment analysis of the text for predicting future emoji input labels. Sentiment analysis of the text or captions is done using Recurrent Neural Network model. Recurrent Neural Networks are a powerful and robust type of Neural Networks for sequential data. Unlike other NNs, RNN uses its internal state (memory) for the processing of inputs sequences.

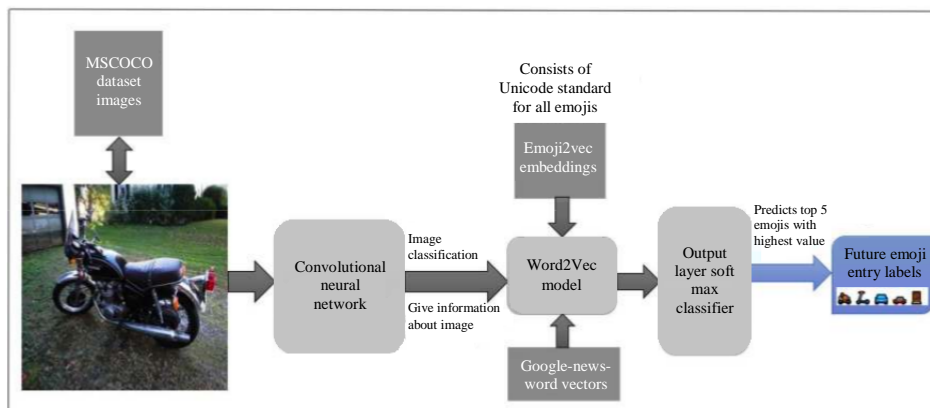


Fig. 1: Overall architecture for future emoji entry prediction from images

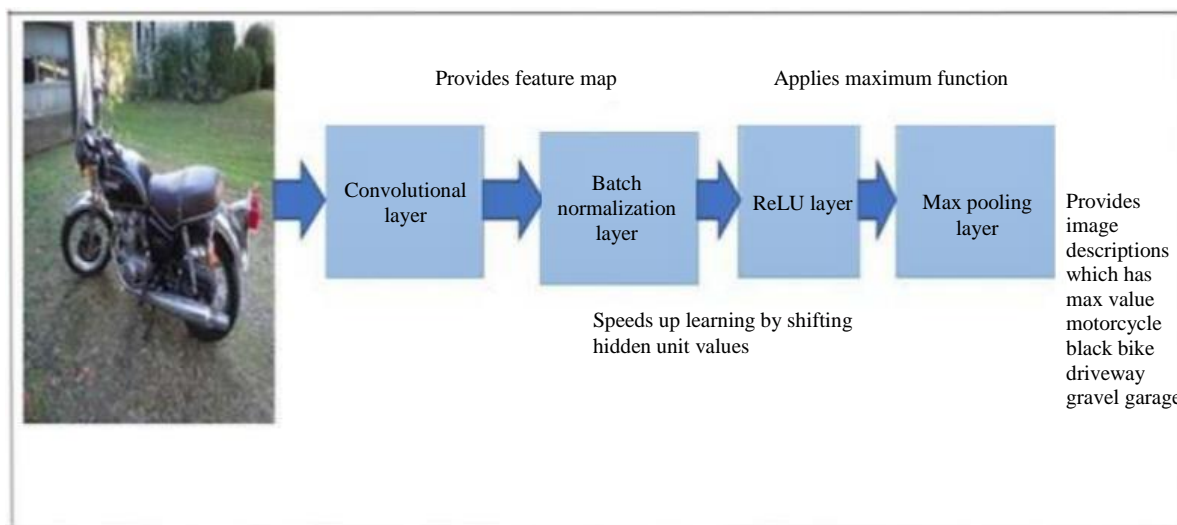


Fig. 2: Convolutional neural network

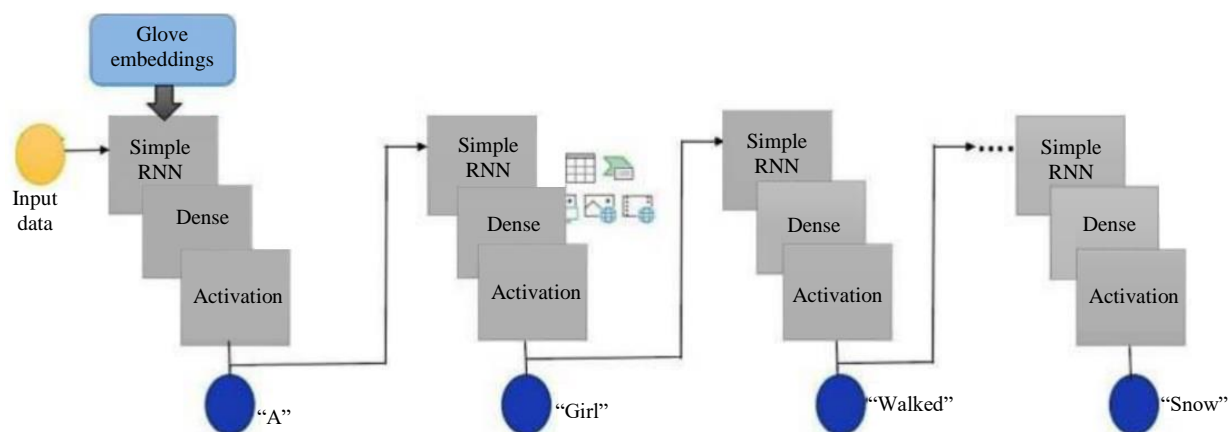


Fig. 3: Recurrent neural network

In RNN, the information cycles through a loop. Decision making is based on the current input and also the memory content, which it has learned from the past inputs. Our model trains the textual data using GloVe vector embeddings, which is used for training the sentiment analysis of the text. GloVe, coined from Global Vectors, is a model for distributed word representation.

This model uses Keras Sequential API for building the Simple RNN up one layer at a time (Brownlee, 2019). Input data, along with glove embeddings, train the network model. RNN consists of Simple RNN layer followed by dense layer, Dropout layer and Activation layer. Simple RNN layer takes the embedding matrix of input data and sets up the output dimensionality unit of

the space. Dense layer adds additional representational capacity to the network. Dropout layer prevents overfitting of the data. Activation layer produces probability for every word in the vocab using softmax activation. The highest probability words are output and emojized in python where it maps output word to the corresponding Unicode character which gives emoji label. RNN architecture for sentiment analysis of the text is shown in Fig. 3.

Methodology and Approach

Image Preprocessing

Data preprocessing is an important part of data mining, because irrelevant information, noisy and

unreliable data can lead to misleading results during the training phase. Hence it is necessary to preprocess the data before training. Due to the irregularity of dimensionality of images in the data set, it was necessary to resize every image to conform to the same size and dimensions. Hence every image is resized to 256×256 pixels at default by using ANTIALIAS in Python Imaging Library (PIL).

Dataset Used

We used MSCOCO 2017 dataset for image inputs, which consists of 15k training and 5k testing images (cocodataset.org). Google-News-vectors is used for word vector representations (GoogleNews-vectors-negative300.bin.gz).

CNN Model

After preprocessing of the image into 256×256 pixels, resized images are subjected to the CNN model. CNN is a Neural Network model made up of neurons that have learnable weights and biases. CNN is a multi-class, multi-label image classifier. As for implementation, Pytorch was used to build CNN.

The network architecture of CNN is made up of piled instances of specialized sequential layers, where each sequential layer contains a max-pooling layer, a ReLU activation layer, a convolutional layer and a batch normalization layer (Goodfellow *et al.*, 2016).

The Convolutional layer in CNN takes image input and produce a feature map using the convolution filter. By sliding the filter over input at every pixel location, element-wise matrix multiplication is done and the sum is calculated. Sum goes into feature map. This layer is followed by Batch Normalization layer which speeds up the training and learning phase by normalizing the feature map inputs by shifting the hidden unit values and subtracting the batch mean and dividing by the standard deviation.

The ReLU layer applies the maximum function to all the values in the input volume. All negative activations are changed to 0. This increases the model's nonlinear properties and also that of the overall network. There are a series of layers followed in every sequential layer. However, the final sequential layer consists of average pooling layer instead of max-pooling layer, which gives information about the image in textual data. It gives fine information about the image as an output. Following this main convolutional portion of the network, there is a 3-layer neural network (W2V) where the last layer (output layer) contains the same number of nodes as the number of emoji classes to predict emoji labels.

W2V Model

The information obtained from the CNN about the image is passed to another neural network model, Word2Vec to predict future emoji entry. For this task, Emoji2vec, a vector space model of emojis, instead of words, is used. E2V is an emoji embedding which consists of standard Unicode representation for all emojis. W2V based on E2V provided a semantically efficient model for relating textual data obtained from CNN to emoji characters.

Gensim and NLTK were used to construct W2V. The word vectors used for training are Google-News word vectors, which have three million 300-dimension word vectors. W2V makes use of Part-of-speech tag for each of the information obtained from CNN about the given image (e.g., noun, adjective, verb, etc.) by NLTK. Also, it removes redundant and unreliable information. NLTK selects appropriate noun, adjectives and verbs because these constitute necessary information about image rather all words in sentences. Gensim vectorizes the remaining words in textual data using word2vec. For every word in the textual data obtained from CNN, W2V obtains five number of emojis that are closest or similar to the word in W2V space.

W2V is a three-layer NN model where the input layer is a word in textual information obtained from CNN. The hidden layer consists of a matrix which has hidden weights of word vectors where word vectors form a row and every unique word in Google-News dataset forms a column. The output layer is a Soft max regression classifier, which is a multi-class classifier. It predicts the top five similar words for input and compares it with emoji2vec, which maps the top five words to its relevant Unicode characters (emoji) to give the top five future emoji input labels.

Visualization of emojis is done using TSNE to find the relationship between emojis and similarity between one another. TSNE uses dimensionality reduction to plot the emojis. Also, the similarity between images is done by Cosine similarity to find similar images and the similarity between emojis is done by Euclidian distance to find the closest emoji for accurate prediction.

Sentiment Analysis of the Text Using RNN

Sentiment analysis describes the emotion of the textual data such as happy, surprise, angry, sad, food, etc. We predict emoji by performing sentiment analysis on text (i.e., we classify textual data into five sentiments like happy, angry, love, play and food). By performing sentiment analysis, one can predict the mood or interests of users.

Sentiment analysis on the text to predict future emoji labels is done using RNN model. RNN takes input data and Glove vector embeddings for training the model. Simple RNN layer in this model takes the embedding matrix of input along with glove vectors and creates the output dimensionality space. The activation layer in RNN uses Soft max activation to predict the top most probabilities of the word for the given text. The output word is then classified according to the five sentiments described above and mapped to a corresponding Unicode character respective to sentiment by emoji parameter in python.

Experimental Observations

Image Preprocessing was done to remove the non-uniform dimension in every image. It is difficult for CNN to analyze the images with different sizes. So, a default dimension of 256x256 is set for every image for prediction.

After preprocessing of the input image, the resized image is subjected to CNN for image classification. CNN classifies the image by training the model and gives fine textual information about the given image. This information is passed to W2V for predicting the top five future emoji entry labels for a given image. W2V is based on Emoji2Vec where words in textual information are mapped to Unicode embeddings by E2V for prediction of next emoji labels.

To explore the relationship between emojis and to find the cluster of emojis by a similarity measure, visualization of emojis is implemented using the TSNE plot. TSNE captures the local structure very effectively and the clustered data is also highly distinguishable.

To find the similarity between emojis and to find similar emojis with one another, Euclidian distance is used. This helps in predicting and analyzing the next emoji labels efficiently. The sample output of the Euclidian distance measure of emojis is given in Table 1.

To find out the similarity between images, Cosine Similarity measure is used. Cosine similarity measure is used because it is a categorical measure rather than using only two variables. Cosine Similarity between images is given in Table 2.

By analyzing the sentiment or emotions of the text, this model predicts future emoji input labels. The whole textual data is classified under five sentiments such as happy, love, angry, sports and food. Here, the text is fitted under these sentiments to predict the emoji labels according to the emotions.

A graphical representation of the similarity measure between the two images is shown in Fig. 4. Similarity between the images is depicted by a low

distance value and dissimilar images show a wide gap in the similarity values.

To predict future emoji entries by sentiment analysis is implemented using Recurrent Neural Network. Recurrent means the output of the current step layer becomes the input to the next time layer. This model stores the current output as well as the preceding elements in its memory. This memory allows the network to learn long-term dependencies in a sequence which means it can take the entire context into account when making a prediction, whether that is the next word in a sentence or a sentiment classification.

Table 1: Euclidian distance measure of emojis

























Emoji1	Emoji2	Euclidean measure
		4.3499
		8.0196
		3.6697
		4.5518
		0.7677
		1.0983
		0.6882
		12.4731

Table 2: Cosine similarity between images

Image 1	Image 2	Score 1	Score 2	Cosine measure
		17.1973782, 12.8474092, 9.1777968, 5.7461844, 4.6259586	9.2134804, 8.179123, 5.4239813, 5.3930944, 5.1179015	0.97
		9.0948797, 7.9985015, 7.2192261, 6.5310003, 4.5374035	6.3266201, 6.1758306, 6.0407052, 5.9844337, 5.3388104	0.98
		15.6261262, 12.4556565, 9.39398360, 8.63604784, 8.03876137	12.4556565, 11.7424795, 10.0484517, 9.82780873, 9.02650415	0.99
		24.4456513, 17.3847130, 11.6502200, 4.65163254, 2.88344985	8.2942614, 7.5282688, 7.4762148, 7.2715449, 7.1243519	0.86
		7.9008179, 7.7935742, 7.1243519, 7.1025280, 5.7988606	23.8433782, 19.6132086, 9.39730453, 7.93009114, 7.77615022	0.9

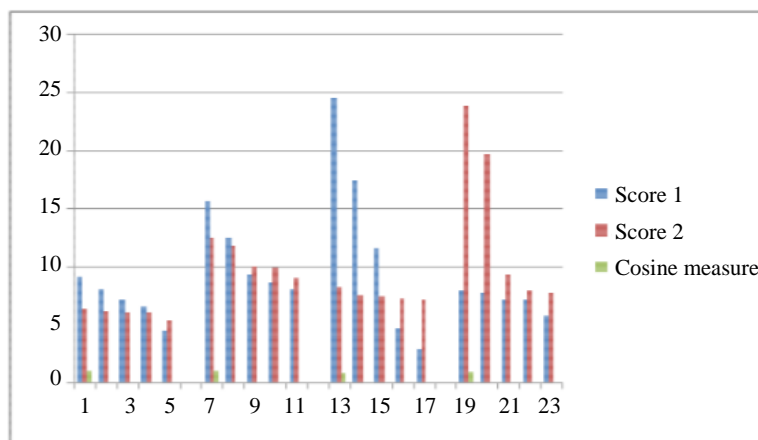


Fig. 4: Comparison of similarity measure

Discussion

Two models are constructed for predicting future emoji inputs from images. To represent information about images, textual data along with emoji is used in media. So, images can be used in the place of text for effective prediction of future emoji entry when compared to text. Thus, this model helps us to predict the next entry from images in minimal time and optimizing the speed and performance.

The two models used in constructing the Emoji Model are CNN and Word2Vec. W2V is a good model for predicting future emoji entry input from word vector representations when compared to any categorical similarity measure such as Jaccard Similarity or Cosine Similarity or any other distance norms. Because, W2V consists of soft max regression classifier, a multi-class classifier, unlike other regression models which takes only two variables. Also, it consists of E2V for standard Unicode emoji embeddings for all emojis.

Visualization of emojis using TSNE gives a clear-cut picture of analyzing a similar set of emojis by reducing and choosing appropriate dimensions to represent the plot. It helps us to find the relative emojis closer to each other for effective analysis for prediction of future emoji input labels. It also helps us to visualize the cluster of emojis and outliers in the plot. Visualization of TSNE gives better solution when compared to PCA because PCA cares little enough about local neighbors.

Sentiment analysis of text describes the emotion or mood of the end-users. In addition to the prediction of emoji labels, it also describes the emotion of the users by sentiment analysis, which can be used for psychometric analysis. By Psychometric analysis, we can predict if the user is happy or sad or undergoing depression. Comparing the two models, the prediction from images is the effective model for predicting future emoji labels because sentiment analysis gives less accuracy in prediction.

Future Work

Until now, emoji arrangement such as prediction of future emoji input from images has been focused, yet Unicode decoding has a wider range in this area. One of them are representing emoticons in Unicode, which is slightly complex when compared to emojis because of a large number of characters for representing emoticons while the others are different stylish fonts for different applications. We focused primarily on finding the next emoji input labels from images. This can be extended by predicting future emoji labels from videos, GIFs, animation, graphics, etc.

Our model can predict future emoji labels for COCO dataset only. This can be extended to predict output for real-time images. Recently, real-time facial recognition has a greater impact on social media. Real-time emotion recognition detects faces in video and images, classifies the emotion on each face and then replaces each face with the correct emoji for that emotion. Any person can take images through their camera or webcam and immediately find the next emoji input label by analyzing the sentiment of the facial expression.

We can predict emoji labels from animated gifs and videos by taking several screenshots depicting different scenes and ensuring all those screenshots predicts similar emojis. Nowadays, people have the desire to express text through emojis in social media to give colorful attire and visual description about their living, mood, or any interest. Analyzing such emojis for prediction of positive and negative thoughts tells people mentality which can be useful for giving counseling to people who undergoes depression.

Acknowledgement

The authors would like to record their appreciation for Dhanya Gandhirajan for her contribution towards dataset collection and drafting the report.

Author's Contributions

Both the authors have equally contributed towards conceptualization, system design, implementation, and result analysis.

Ethics

This article is original and its contents are unpublished. The corresponding author confirms that the other author has read and approved the manuscript and there are no ethical issues involved.

References

- Bailly, G., P.B. Duncan, R. Howes and A. Oulasvirta, 2014. Model of visual search and selection time in linear menus. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Apr. 26-26, Toronto, Ontario, Canada, pp: 3865-3874. DOI: 10.1145/2556288.2557093
- Bi, X., A.S. Barton and S. Zhai, 2012. Multilingual touchscreen keyboard design and optimization. *Hum. Comput. Interact.*, 27: 352-382.
- Brownlee, J., 2019. Machine learning mastery. Making Developers Awesome at Machine Learning.
- Cappallo, S., S. Svetlichnaya, P. Garrigues, T. Mensink and C.G.M. Snoek, 2019. New modality: Emoji challenges in prediction, anticipation and retrieval. *IEEE Trans. Multi.*, 21: 2-2. DOI: 10.1109/TMM.2018.2862363
- Cappallo, S., T. Mensink and C.G.M. Snoek, 2015. Image2Emoji: Zero-shot emoji prediction for visual media. Proceedings of the 23rd ACM International Conference on Multimedia, Oct. 15-17, Brisbane, Australia, pp: 1311-1314. DOI: 10.1145/2733373.2806335
- Chablani, M., 2017. Towards data science-sharing concepts, ideas and codes. cocodataset.org
- Comaniciu, A. and L. Najafizadeh, 2018. Enabling Communication for locked-in syndrome patients using deep learning and an emoji- based brain computer interface. Proceedings of IEEE Biomedical Circuits and Systems Conference, Oct. 17-19, IEEE Xplore Press, Cleveland, OH, USA. DOI: 10.1109/BIOCAS.2018.858 4821
- Domin, C., H. Pohl and M. Rohs, 2017. Beyond just text: Semantic emoji similarity modeling to support expressive communication. *ACM Trans. Comput. Hum. Interact.*, 24: 42-42. DOI: 10.1145/3039685
- Dunlop, M. and J. Levine, 2012. Multidimensional pare to optimization of touch screen keyboards for speed, familiarity and improved spell checking. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, May 5-10, Austin, Texas, USA, pp: 2669-2678. DOI: 10.1145/2207676.2208659
- Goodfellow, I., Y. Bengio and A. Courville, 2016. Deep Learning. 1st Edn., MIT Press, ISBN-10: 0262035618, pp: 775.
- GoogleNews-vectors-negative300.bin.gz
- Han, J. and M. Kamber, 2011. Data Mining: Concepts and Techniques. 3rd Edn., Elsevier, ISBN-10: 0123814804, pp: 774.
- Janssens, O., M. Slembrouck, S. Verstockt, S. Van Hoecke and R. Van de Walle, 2013. Real-time emotion classification of tweets. Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Aug. 25-28, IEEE Xplore Press, Niagara Falls, ON, Canada. DOI: 10.1145/2492517.2492577
- Weir, D., H. Poh, S. Rogers, K. Vertanen and P.O. Kristensson, 2014. Uncertain text entry on mobile devices. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Apr. 26-26, Toronto, Ontario, Canada, pp: 2307-2316. DOI: 10.1145/2556288.2557412