Learning Emoji Embeddings Using Emoji Co-Occurrence Network Graph

1st International Workshop on Emoji Understanding and Applications in Social Media
Stanford University, California
25th June, 2018

Presenting Author
Anurag Illendula
aianurag09@iitkgp.com
Indian Institute of Technology (IIT) Kharagpur
https://aianurag09.github.io/

Co-author
Manish Reddy Yedulla
es15btech11012@iith.ac.in
Indian Institute of Technology (IIT) Hyderabad
Why Emoji Embeddings

- Every Deep Learning model requires us to represent an entity in numbers.
Why Emoji Embeddings

- Every Deep Learning model requires us to represent an entity in numbers.
- Emoji Embeddings have increase accuracies of many Emoji Understanding tasks.
Why Emoji Embeddings

- Every Deep Learning model requires us to represent an entity in numbers.

- Emoji Embeddings have increased accuracies of many Emoji Understanding tasks.

- Emoji Similarity, Emoji sense disambiguation, Sentiment Analysis.
Why Emoji Co-Occurrence

- Emoji Co-Occurrence helps us to understand context of use of multiple emojis in social media posts.
Why Emoji Co-Occurrence

- Emoji Co-Occurrence helps us to understand the context of use of multiple emojis in social media posts.

- For example:
Why Emoji Co-Occurrence

- How does emoji co-occurrence help us learn Emoji Embeddings?
Why Emoji Co-Occurrence

- How does emoji co-occurrence help us learn Emoji Embeddings?
- “I got betrayed by a 🧐, I want to kill you 🔫”
Why Emoji Co-Occurrence

- How does emoji co-occurrence help us learn Emoji Embeddings?
  - “I got betrayed by a 🕵️‍♂️, I want to kill you 🎃”
  - Here both emojis (🕵️‍♂️, 🎃) contain the same sentiment as the overall sentiment of the tweet which is **negative**.
We create the emoji network using 14.3 Million tweets, each of which have multiple emojis.

Each tweet generates a polygon of n edges where n is the number of emojis embedded in the tweet, each node represents one emoji and each edge represents the number of co-occurrences between the connecting emojis.
Emoji Co-Occurrence Network

EMOJI POLYGONS

I got betrayed by a 🙃 I would die for 💀arma
Emoji Co-Occurrence Network

EMOJI POLYGONS

Baby text back 😘 😅 😁 😘 😪 😪, waiting for your reply 😘
Emoji Co-Occurrence Network

MORE EXAMPLES

I got betrayed by a 😱 I would die for 😟 😤

I can’t trust so its Hi & Bye flow 😊👋🏻✌️ I still ❤️ them though

Baby text back 😘 😩 😘 , waiting for your reply 😞

Without 💯 respect there is no 😞 😘 😘

Illendula, Anurag et al. Learning Emoji Embeddings using Emoji Co-Occurrence Network Graph
Emoji Co-Occurrence Network

EMOJI CO-OCCURRENCE NETWORK

- By combining all the emoji polygons.
Emoji Co-Occurrence Network

EMOJI CO-OCCURRENCE NETWORK

- By combining all the emoji polygons.
Emoji Co-Occurrence Network

- By combining all the emoji polygons.

Emoji Embeddings
300 dimensions
Network Embedding Model
Model Parameters

- We define two types of measures which signify the proximity between nodes of the network.
Model Parameters

- We define two types of measures which signify the proximity between nodes of the network.

- **First Order Proximity**: This is the local pairwise proximity which can be related to the weight of the edge joining emoji nodes.
We define two types of measures which signify the proximity between nodes of the network.

- **First Order Proximity**: This is the local pairwise proximity which can be related to the weight of the edge joining two emoji nodes.

- **Second Order Proximity**: This can be considered as the similarity between neighbourhood network structures.
First Order Proximity

Let $u_i$, $u_j$ represent the emoji embeddings. The joint probability which signifies the first order proximity between emoji nodes $v_i$, $v_j$ is given by

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(\bar{u}_i^T \cdot \bar{u}_j)}$$
First Order Proximity

- Let $u_i, u_j$ represent the emoji embeddings. The joint probability which signifies the first order proximity between emoji nodes $v_i, v_j$ is given by

$$p_1(v_i, v_j) = \frac{1}{1 + exp(u_i^T \cdot u_j)}$$

- The empirical probability between the vertices $v_i, v_j$ is given by

$$\tilde{p}_1(i, j) = \frac{w_{ij}}{W}$$
First Order Proximity

- Let \( u_i, u_j \) represent the emoji embeddings. The joint probability which signifies the first order proximity between emoji nodes \( v_i, v_j \) is given by

\[
p_1(v_i, v_j) = \frac{1}{1 + \exp(u_i^T \cdot u_j)}
\]

- The empirical probability between the vertices \( v_i, v_j \) is given by

\[
\tilde{p}_1(i, j) = \frac{w_{ij}}{W}
\]

- Hence the objective function in this case is

\[
O_1(i, j) = d(\tilde{p}_1(i, j), p_1(i, j))
\]
First Order Proximity

Empirical Probability Curve

\[ P(1,2) = \frac{3.0}{25} \]
\[ P(1,3) = 0 \]
\[ P(1,4) = \frac{4.0}{25} \]
\[ P(2,3) = \frac{6.0}{25} \]
\[ P(2,4) = \frac{7.0}{25} \]
\[ P(3,4) = \frac{5.0}{25} \]
First Order Proximity

Illendula, Anurag et al. Learning Emoji Embeddings using Emoji Co-Occurrence Network Graph
Second Order Proximity

- The second order proximity of two nodes \((v_i, v_j)\) measure the similarity of the neighbourhood network structures of respective nodes (referred as “context”)

\[
p_2(v_j | v_i) = \frac{\exp(\overrightarrow{u'_j} \cdot \overrightarrow{u_i})}{\sum_{k=1}^{|V|} \exp(\overrightarrow{u'_k} \cdot \overrightarrow{u_i})}
\]
Second Order Proximity

- The second order proximity of two nodes \((v_i, v_j)\) measures the similarity of the neighbourhood network structures of respective nodes (referred as “context”)

\[
p_2(v_j | v_i) = \frac{\exp(u_j^T \cdot \bar{u}_i)}{\sum_{k=1}^{|V|} \exp(u_k^T \cdot \bar{u}_i)}
\]

- The empirical probability between the vertices \(v_i, v_j\) is given by

\[
\tilde{p}_1(i, j) = \frac{w_{ij}}{d_i} \quad \text{and} \quad d_i = \sum_{k \in N(i)} w_{ik}
\]
The second order proximity of two nodes \((v_i, v_j)\) measures the similarity of the neighbourhood network structures of respective nodes (referred as “context”)

\[
p_2(v_j | v_i) = \frac{\exp(u_j^T \cdot \bar{u}_i)}{\sum_{k=1}^{|V|} \exp(u_k^T \cdot \bar{u}_i)}
\]

The empirical probability between the vertices \(v_i, v_j\) is given by

\[
\tilde{p}_1(i, j) = \frac{w_{ij}}{d_i} \quad \text{and} \quad d_i = \sum_{k \in N(i)} w_{ik}
\]

The objective function in this case is

\[
O_2 = \sum_{v_i \in V} \lambda_i d(\tilde{p}_2(\cdot | v_i), p_2(\cdot | v_i))
\]

\(\lambda_i\) is the prestige of the vertex \(i\) which can is measured by PageRank
Model Optimization

- Further we use KL-Divergence for optimizing the distance between probability curves.
Model Optimization

- Further we use KL-Divergence for optimizing the distance between probability curves.

First Order Proximity

\[ O_1 = - \sum_{(i, j) \in E} w_{ij} \log p_1(v_i, v_j) \]
Model Optimization

- Further we use KL-Divergence for optimizing the distance between probability curves.

First Order Proximity

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

Second Order Proximity

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

- Further we use KL-Divergence for optimization.

First Order Proximity

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

Second Order Proximity

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

The model is trained using RMS Propagation gradient descent algorithm with learning rate as 0.025 and batch size as 128 using Tensorflow library on a cuda GPU.
Experiments

1. Sentiment Analysis Task:
   ○ How can emoji co-occurrence can help in Sentiment Analysis Task?
   ○ “I got betrayed by a 🧐, I want to kill you 🎃”
Experiments

1. **Sentiment Analysis Task:**
   - How can emoji co-occurrence help in Sentiment Analysis Task?
   - “I got betrayed by a 😎, I want to kill you ⚰️”
   - Here both emojis (😎, ⚰️) contain the same sentiment as the overall sentiment of the tweet which is negative.
Experiments

1. Sentiment Analysis Task:
   - How can emoji co-occurrence help in Sentiment Analysis Task?
   - “I got betrayed by a 🙄, I want to kill you 😡”
   - Here both emojis (🙄, 😡) contain the same sentiment as the overall sentiment of the tweet.
   - Hence we evaluate our emoji embeddings on Sentiment Analysis task.
1. **Sentiment Analysis Task:**
   - We report our accuracies on gold standard dataset developed by Novak et al.
Experiments

1. **Sentiment Analysis Task:**
   - We report our accuracies on gold standard dataset developed by Novak et al.
   - We use pre-trained FastText word embeddings on Wikipedia corpus to embed words into low dimensional space.
Experiments

1. **Sentiment Analysis Tasks:**

   - We calculate the bag of words vector for each tweet and then use this vector as a feature to train a SVM and a random forest model.
Experiments

1. Sentiment Analysis Tasks:

- We calculate the bag of words vector for each tweet and then use this vector as a feature to train a SVM and a random forest model.

<table>
<thead>
<tr>
<th>Word Embeddings</th>
<th>Classification Accuracy using RF</th>
<th>Classification Accuracy using SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the art results</td>
<td>60.7</td>
<td>63.6</td>
</tr>
<tr>
<td>First Order Embeddings</td>
<td>62.1</td>
<td>65.2</td>
</tr>
</tbody>
</table>
2. **Emoji Similarity Task:**
   - How does emoji co-occurrence help in Emoji Similarity task?
Experiments

2. Emoji Similarity Task:
   - How does emoji co-occurrence help in Emoji Similarity task?
   - We have observed that (❤️, 😘 😘) and (😘 😘, 😄 😄) emoji pairs have co-occurred most times.
2. **Emoji Similarity Task:**
   - How does emoji co-occurrence help in Emoji Similarity task?
   - We have observed that (❤️, 😘) and (😘, 😊) emoji pairs have co-occurred most times.
   - Hence we evaluate our emoji embeddings on Emoji Similarity task.
Experiments

2. Emoji Similarity Task:
   - The similarity measure between two different emojis is defined as the cosine similarity between two emoji embeddings.
   - We evaluate our similarity values with that of the similarity measures using the semantic embeddings (SEMANTIC SIMILARITY).

\[
\text{similarity}(e_1, e_2) = \frac{\vec{a} \cdot \vec{b}}{|a| \cdot |b|}
\]

Similarity Values can range from (0,1)
Experiments

- **Emoji Similarity Tasks:**

<table>
<thead>
<tr>
<th>Emoji Pair</th>
<th>Similarity Measure</th>
<th>Semantic Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(<strong>,</strong>)</td>
<td>0.921</td>
<td>0.442</td>
</tr>
<tr>
<td>(<strong>,</strong>)</td>
<td>0.916</td>
<td>0.598</td>
</tr>
<tr>
<td>(<strong>,</strong>)</td>
<td>0.911</td>
<td>0.623</td>
</tr>
<tr>
<td>(<strong>,</strong>)</td>
<td>0.909</td>
<td>0.546</td>
</tr>
</tbody>
</table>

*Emoji Similarity measured using first order embeddings*
Experiments

- **Emoji Similarity Tasks:**

<table>
<thead>
<tr>
<th>Emoji Pair</th>
<th>Similarity Measure</th>
<th>Semantic Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(👍, 😄)</td>
<td>0.646</td>
<td>0.662</td>
</tr>
<tr>
<td>(💰, 🎁)</td>
<td>0.606</td>
<td>0.598</td>
</tr>
<tr>
<td>(👍, 🎉)</td>
<td>0.596</td>
<td>0.623</td>
</tr>
<tr>
<td>(🗂️, 🔍)</td>
<td>0.556</td>
<td>0.622</td>
</tr>
</tbody>
</table>

Emoji Similarity measured using second order embeddings
Experiments

- Emoji To Emoji Analogy Tasks:
  - We extrapolate the analogical reasoning task in context of emojis by replacing words with emojis
Experiments

- **Emoji To Emoji Analogy Tasks:**
  
  - Consider the emoji analogy \((?(:) = (?, ?))\) where we fill the gap (represented by “?“) by finding an emoji from the complete list of emojis whose embedding (represented by \(\text{vec}(x)\)) is closest to

  \[
  \text{vec}(?(:)) - \text{vec}(?(:)) + \text{vec}(?(:))
  \]
Experiments

- **Emoji To Emoji Analogy Tasks:**
  - We extrapolate the analogical reasoning task in context of emojis by replacing words with emojis

<table>
<thead>
<tr>
<th>First Emoji Pair</th>
<th>Second Emoji Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>😊 : 😊</td>
<td>😃 : 😃</td>
</tr>
<tr>
<td>🇨🇦 : 🇺🇸</td>
<td>🇮🇳 : 🇨🇳</td>
</tr>
<tr>
<td>🚴🏼 : 🚴🏼</td>
<td>🚷 : 🚷</td>
</tr>
<tr>
<td>🐰 : 🐱</td>
<td>🙊 : 🙊</td>
</tr>
<tr>
<td>🍳 : 🍳</td>
<td>🍳 : 🍳</td>
</tr>
</tbody>
</table>
Future Work

- External knowledge has been proven to improve the accuracies of various NLP tasks.

- Using Jian et al.’s work as reference, we wish to work on incorporating external knowledge from Emojinet to our network embedding model to further improve the accuracies of sentiment analysis and emoji similarity tasks.
Acknowledgements

- We are grateful to Sanjaya Wijeratne and Dr. Amit P. Sheth for the thought-provoking discussions on this topic.

- We acknowledge support from the Indian Institute of Technology Kharagpur for this work.