Emojify: Prediction Emoji from Sentence
Boyu (Bill) Zhang, Chen Huang, Xueying (Shirley) Xie

Introduction

Motivation
- The human brain processes images 40,000 times faster than text, and 90% of information transmitted to the brain is visual.
- Add visual information to the content you’re trying to deliver to your user would help capture their attention.

Emoji have become a new language that can more effectively express an idea or emotion.

Goal
- Emoji: to predict emoji from sentence

Difficulties
- Weak semantic connection between sentence and emoji.
- Ambiguity: one emoji can express multiple feelings, e.g. 😥 😳 😹
- Multi-label: multiple emojis share the same semantic meaning, e.g. 😥 😳 😹
- Weak semantic connection between sentence and emoji.

Data
Twitter dataset originally contains 1,678,685 <sentence-emoji> pairs.

With original 1791 classes to predict in the dataset, these three data pre-processing techniques reduced the number of classes.

Data Pre-processing
- Stop emoji: Remove high frequent emojis which is everywhere and do not have specific semantic meaning.
- Noise removal: Filter out emojis which has less than 1000 corresponding sentences.
- Multi-label: Multiple emojis share same semantic meaning, e.g. 😥 😳 😹.

Methods (Traditional)
- GLoVe: generated sets of learned vectors for emojis.
- GLoVe: An unpre-trained model which maps words into a meaningful space where distance between words is related to semantic similarity. In this project, we used the pre-trained model GLoVe-50 and GLoVe-300 from Stanford.

Methods (Deep Learning)
STM Architecture
- LSTM
- Conv 1D
- Bi-LSTM
- Softmax

Word Embedding

Bags of Words (BoW) representation is a sparse matrix representation, where each item is in one row and each word is in the vocabulary is on a column. The dictionary size is 1834 after stopwords and stemming. The sentence is represented by TF-IDF.

Word2Vec: pre-trained shallow, two-layer networks that are trained to reconstruct linguistic contexts of words. Word2Vec maps each unique word into a corresponding vector in feature space such that words that have common contexts are located close to each other.

Methods (Deep Learning)

LSTM Architecture
- LSTM
- Conv 1D
- BI-LSTM
- Softmax

Emojis

Experiments

Dataset Summary
- 1232 valid examples after data-processing
- 1708 sentence per emoji in training dataset

Evaluation Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Word Embedding</th>
<th>Bi-LSTM</th>
<th>Deep GRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multinomial Naive Bayes</td>
<td>BoW + TF-IDF</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>GLoVe-50</td>
<td>19.530%</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>GLoVe-300</td>
<td>16.376%</td>
<td>14.966%</td>
</tr>
</tbody>
</table>

Prediction Example
- I’m angry 😥
- I need sleep 😌
- I love you 😍
- I feel pretty sad 😥 (failure case)

Discussion
1. Stop emoji is essential to handle uneven distribution. One emoji may dominate the dataset.
2. Emoji and sentence only have weak semantic relations. Many examples share the same emoji but express totally opposite emotion.
3. There often isn’t a 1:1 mapping between an emoji and a sentence or expression. Sometimes, if one has a user decide on which emojis to use for a similar sentence, the emoji selection would vary quite a bit. Therefore, having more than 1 prediction with decreasing confidence may be a better way to solve this problem.
4. In addition, when we calculate accuracy, it may be best to have weighted penalties for determining accuracy. The correlation matrix portrays a lot of the emoji overlap with some good reasoning, so we can penalize less misclassified emojis as opposed to.

References