Graphical Feelings: GIF Sentiment Learning

Overview
To improve GIF search, we implement a deep learning system to predict real-valued sentiment scores for GIFs using two network architectures: a MaxPool fully connected model and an RNN using LSTM units. The network takes a GIF as input and outputs 17 real values corresponding to different emotions. Both models achieve moderate loss but make conservative predictions close to the class means. Different GIF encoders may offer improvements.

Data and Features
Data consisted of 6,143 GIFs from the Giphy API and 2,758 pairwise emotion comparisons of GIF sentiment by internet users from MIT Media Lab’s GIFFIT[1]. These were transformed into 17 normalized, real-valued scores per GIF using the Bradley-Terry model. Figure 1 shows the distribution of these scores by sentiment class. The dataset was split into 80/10/10 train/dev/test portions. Frames from GIFs were sampled and encoded as 2048 dimensional vectors using the ResNet101 CNN[2] pre-trained on the Kinetics dataset of videos of human actions. Two sampling methods and data augmentation yielded three training data sets: (1) Sparse Small, (2) Dense Small, and (3) Sparse Augmented. As performance on all training sets was comparable, the Dense Small results are reported here.

Results
A two-layer RNN with 64 hidden units trained with dropout probability 0.1765 marginally outperformed all other models and the baselines.

Discussion and Future Directions
Even the best RNN model does not substantially outperform the baseline. All models trained appear to be too conservative, making predictions very close to the mean value for each sentiment class. As seen in Figure 3, dev and train loss were comparable, suggesting that the problem was one of bias, not variance. Qualitatively, GIFs with low prediction error tended to lack strong emotions while GIFs with high error had salient facial emotion markers such as smiles, tears, or wide eyes. These results suggest that the CNN encoder may not be capturing key emotional features. Future work may use a different image encoder optimized for sentiment-specific features such as SentIBank[3] or use transfer learning to tune the last few layers of the encoder on GIF sentiment. The augmented dataset could be more densely sampled, combining the benefits of more frames per training example and more examples. Finally, an alternate loss function that more heavily weights correct predictions of strong emotions may also improve the model’s ability to capture variance in the data.

Figure 1. Distribution of sentiment scores

Figure 2. FC-MaxPool and RNN architectures

Figure 3. Train/dev loss of best model over training epochs

Table 1. Performance of best models

<table>
<thead>
<tr>
<th>Model</th>
<th>Train RMSE</th>
<th>Dev RMSE</th>
<th>Test RMSE</th>
<th>Test EV</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinReg</td>
<td>0.07471</td>
<td>0.07848</td>
<td>0.08047</td>
<td>0.343</td>
</tr>
<tr>
<td>Mean Only</td>
<td>0.07336</td>
<td>0.07432</td>
<td>0.07511</td>
<td>0.406</td>
</tr>
<tr>
<td>FC-MaxPool-2</td>
<td>0.07336</td>
<td>0.07432</td>
<td>0.07527</td>
<td>0.406</td>
</tr>
<tr>
<td>FC-MaxPool-4</td>
<td>0.07382</td>
<td>0.07453</td>
<td>0.07562</td>
<td>0.406</td>
</tr>
<tr>
<td>RNN-2</td>
<td>0.07220</td>
<td>0.07399</td>
<td>0.07610</td>
<td>0.408</td>
</tr>
<tr>
<td>RNN-4</td>
<td>0.07341</td>
<td>0.07446</td>
<td>0.07613</td>
<td>0.406</td>
</tr>
<tr>
<td>RNN-8</td>
<td>0.07344</td>
<td>0.07431</td>
<td>0.07547</td>
<td>0.405</td>
</tr>
</tbody>
</table>

Equation 1. Mean-Squared Error loss

\[
MSE(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2
\]

References

Images made by Brandon Lee from www.flickr.com