All The Feels

By Melanie C & Priya G
Advisor: James Kunz

Though the length and depth of personal communication may be decreasing, the need to communicate complex emotion remains the same.

Data Preparation

Tweet Data
N = 21,051

Word Embeddings

Preprocessed data by removing punctuation, convert to lowercase, and remove whitespaces

Used GloVe word embeddings from Stanford [1,34]

Step 1

Polarity Classifier

Sentiment	Count
Positive	12,811
Negative	8,240

Step 2

Emotion Classifier

Sentiment	Count
:: surprise	3,849
:: fear	2,816
:: joy	8,240
:: sadness	3,830
:: disgust	761
:: anger	1,555

Challenges: Incorporating Polarity

Plan: Use TfidfVectorizer with scikit implementation of SVM to predict polarity and feed this into LSTM Model → How do we do this?

1. Add into word Embedding

This would extend each matrix by 1 to shape of [1,35]. However, this appended "0" and "1" which is just a word lookup

2. Predicted class + LSTM logits

This added a third classified that would bridge together the predicted class and logits from LSTM model

3. Append "Negative" & "Positive"

This appended "negative" and "positive" to the front of tweets for training and predicted class for test set. This created a shape of [1,35]

Challenges: Disgust/Anger

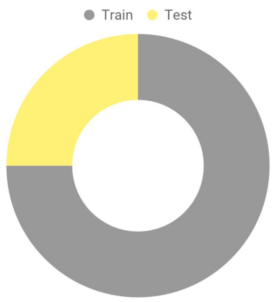
Anger & disgust had smallest volumes in data set

Train & test were balanced

Bias terms were strongly negative

Oversampling increased F1 for both emotions

Sentiment	Count
:: surprise	3,849
:: fear	2,816
:: joy	8,240
:: sadness	3,830
:: disgust	761
:: anger	1,555



Emotion	Bias
:: disgust	-0.319
:: anger	-0.101

Disgust: +10%
Anger: +18%

Model Iterations

Model Iteration	**Type/Description of Model**	**Polarity as Feature**	**F-Score**
Blind Classifier	Predict same class (joy) for all		22%
Baseline 1	SVM with linear kernel		55%
Baseline 2	SVM + Logistic Regression	✘	55%
Iteration 1	1 LSTM		42%
Iteration 2	SVM + 1 LSTM (Polarity added in embedding layer)	✘	46%
Iteration 3	SVM + 2 LSTM (Polarity added in embedding layer)	✘	48%
Iteration 4	SVM + 2 LSTM (Polarity appended to tweet as word “positive” or “negative”)	✘	44%

Future Work

- Get more data!
    - While the data set we used was larger than those in papers we referenced, it wasn’t large enough to fully benefit from the power of neural networks
- Try different approaches to data cleansing
    - Could try stemming or parsing hashtags into actual words for additional context
- Use different word vector embeddings
    - Used GloVe but could try word2vec or create something specific to Twitter with representation of acronyms like “LOL” or “TLDR”