Sentiment Analysis within Reddit

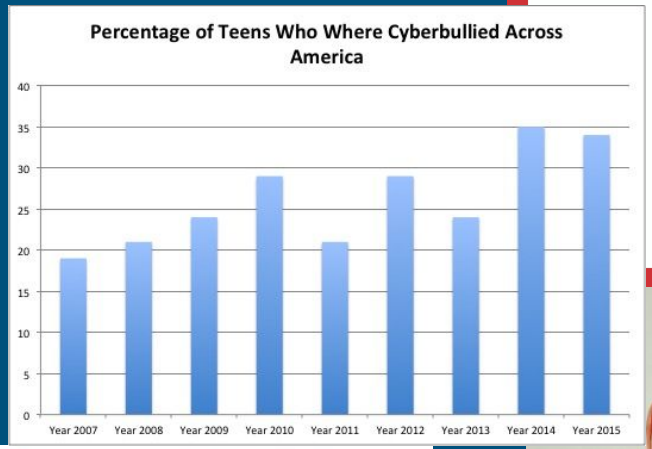
W266 - Natural Language Processing

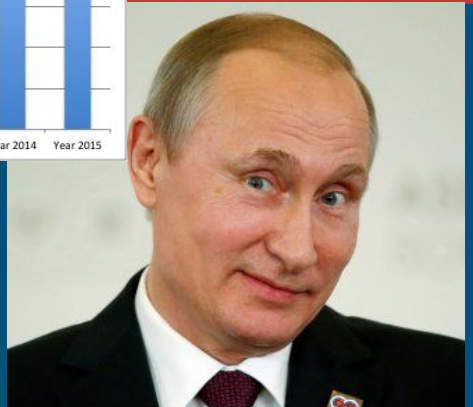
Matthew Nelson | Dan Wald

The Internet is Dark and Full of ~~Terrors~~ Trolls

Trolls have taken over social media

- Russian Interference
- Cyberbullying
- Political Influence
- Depression
- Lack of parental controls





Where to Start?

Design Question:

- Can we create a Sentiment Classifier which predicts the overall sentiment for a single Reddit page?
  - Could serve as a potential parental warning
    - Google Chrome Extension to block / warn users as they view a page
  - Could provide a useful overview for daily browsing activity
  - Ability to produce a visualized network showing how negative comments can breed more negative comments, and vice versa

Many papers published that analyze positive / negative sentiment of tweets

Train a model with tweets and move into sub-reddits that have clear and complex parent/child relationships, bounded by a title

Training Corpora

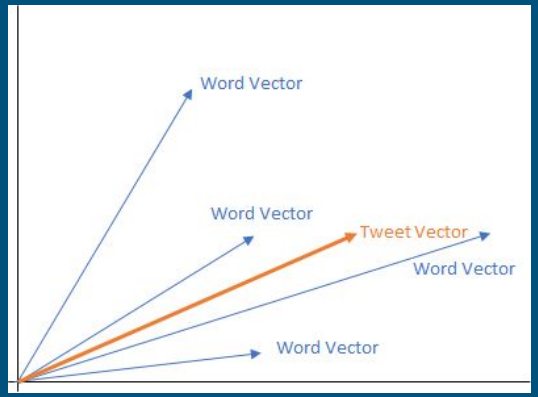
We attempted to leverage two different corpora in the Social Media domain

- Sentiment140 Corpus
  - 1.6 Million Tweet Corpus
  - Machine Annotated by linking positive emoticons [:), =)] to positive sentiment & negative emoticons [:(, =( ]to negative sentiment
  - Labeled Positive/Negative
  - Easily accessed in entirety
- SemEval Twitter Challenge Corpus (2012 through 2017)
  - Human annotated, but substantially smaller
  - 16667 Total Tweets
  - Labeled Positive/Neutral/Negative
  - Tweets had to be extracted via Twitter API with many having since been deleted

Considered the SemEval labels more accurate but required a larger dataset to improve accuracy, so both corpora were combined

Feature Engineering

1. Tokenizing & Filtering
   a. Filtered out some unnecessary links, mentions, retweets, handles, and emoticons
2. Convert Tokens into Vector Representation
   a. Leveraged GenSim's Word2Vec model to convert our vocabulary into a vocabulary of word vectors.
   b. 128 dimensional embeddings
3. Convert Tweets into Vector Representation
   a. Tweet Vector = Avg(Word Vector * TFIDF Weight)
4. Scaled Tweet Vectors



Model Selection / Application

Fit a Convolutional Neural Network:

- Layers = 2x ReLU with dropout, 1x softmax
- Features = Scaled Tweet Vectors
- Labels = Positive/Neutral/Negative
- Accuracy = 77.17%

Baseline model:

- Single layer CNN softmax
- Accuracy = 76.80%
- Multi-layer improves model by +0.36%

keras : making neural networks boring since 2015

Fit a Convolutional Neural Network Model using the Tweet Vectors as Features

Using Keras & Tensorflow backend.

In [17]:
```
# Keras Model w/ Two Affine Layers
model = Sequential()
model.add(Dense(64, activation='relu', input_dim=vector_dim))
model.add(Dropout(0.2))
model.add(Dense(16, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(3, activation='softmax')) # softmax for multi-class
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy', # categorical_crossentropy
              metrics=['accuracy'])

model.fit(X_train_tweet_vecs, y_train, epochs=3, batch_size=32, verbose=2)
```

```
Epoch 1/3
 - 78s - loss: 0.5285 - acc: 0.7549
Epoch 2/3
 - 78s - loss: 0.5199 - acc: 0.7619
Epoch 3/3
 - 79s - loss: 0.5190 - acc: 0.7639
```

Out[17]:
```
<keras.callbacks.History at 0x152f89ef0>
```

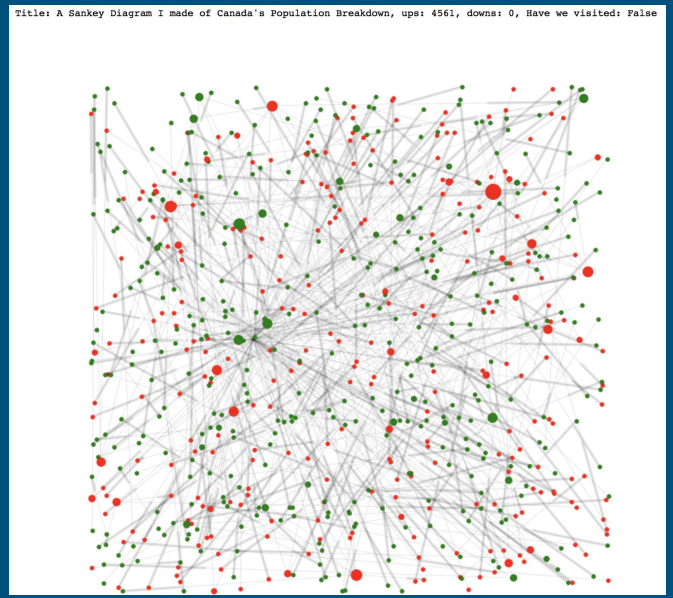
Evaluate Model on Test Set

In [18]:
```
accuracy = model.evaluate(X_test_tweet_vecs, y_test, batch_size=32, verbos
print("Accuracy: %.2f%%" % (accuracy[1]*100))
```

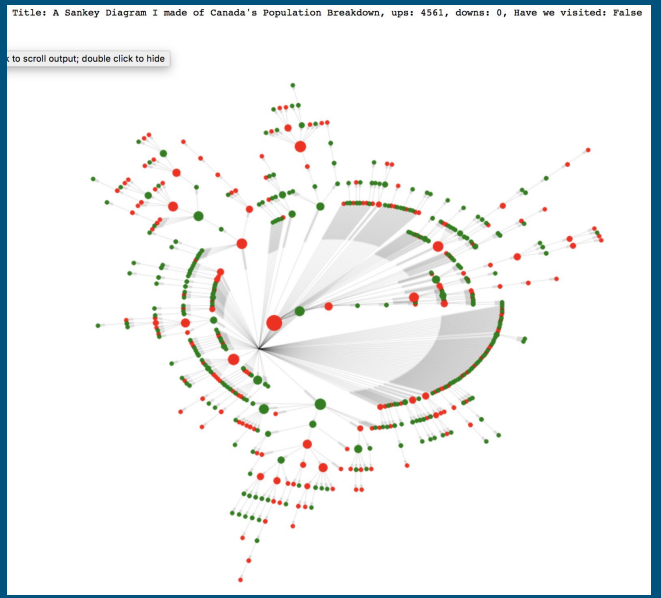
```
Accuracy: 77.17%
```

Model Output

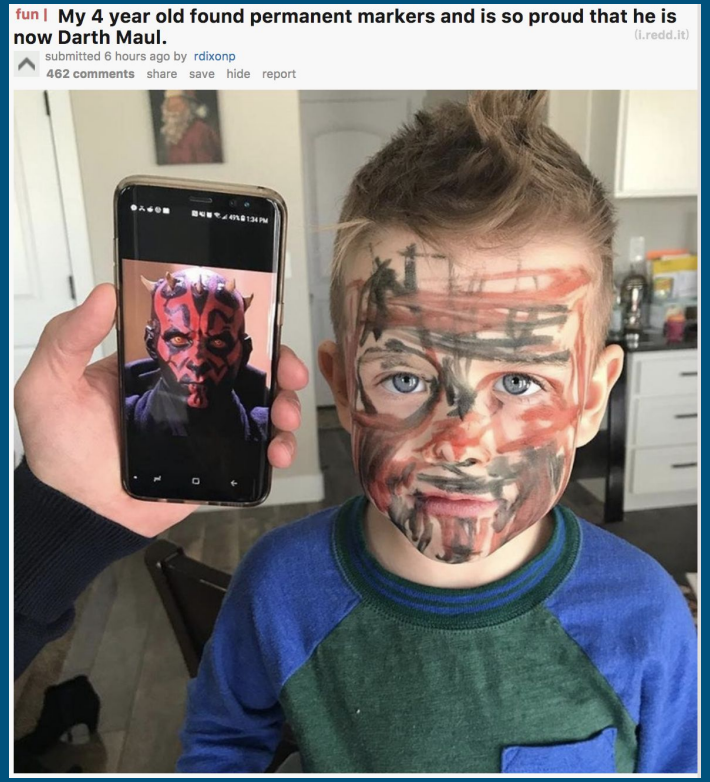
Title: A Sankey Diagram I made of Canada's Population Breakdown, ups: 4587, downs: 0, Have we visited: False

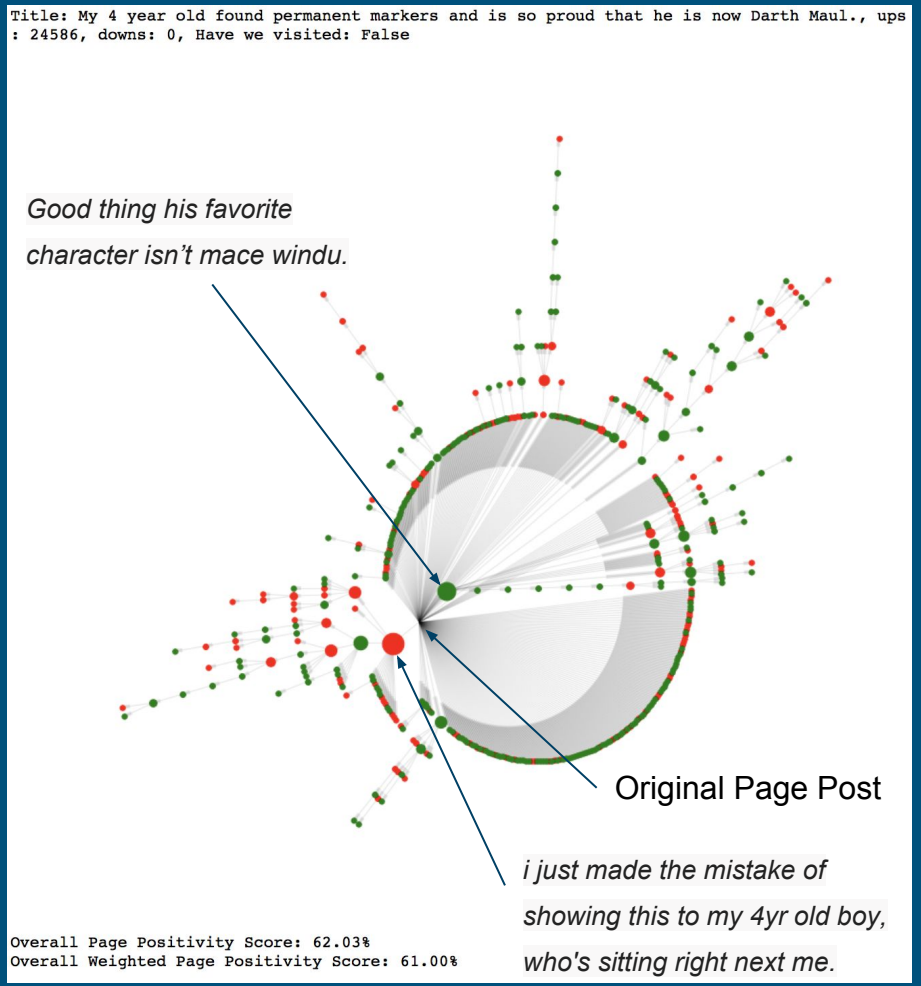


Tree Structure



Example 1: Positive





Example 2: Negative - people *hate* bedbugs

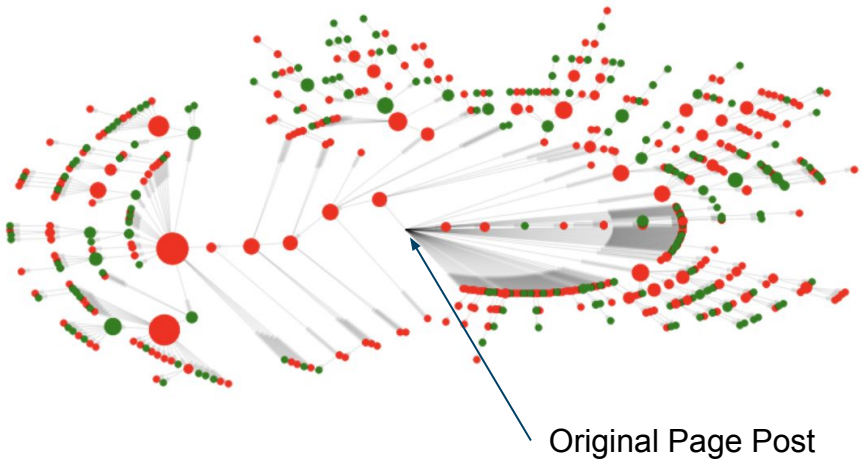
CBS/AP / *December 9, 2017, 5:32 PM*

Woman burns down Cincinnati, Ohio, home trying to kill bedbugs

Authorities said that three people were injured and 10 people were left homeless after a woman accidentally started a fire while trying to kill bed bugs in Cincinnati. / **WKRC**

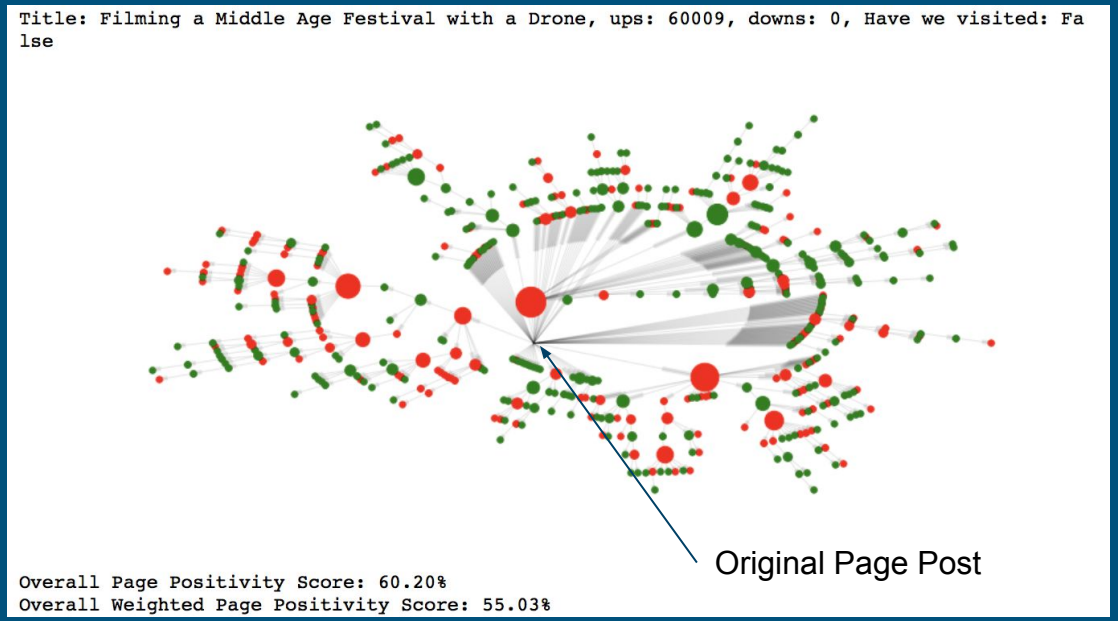
Share / Tweet / Reddit / Flipboard / Email

Title: Woman burns down home trying to kill bedbugs, ups: 14891, downs: 0, Have we visited: False



Overall Page Positivity Score: 37.05%
Overall Weighted Page Positivity Score: 34.13%

Example 3: Complex embedded positive / top comments negative



Model Accuracy / Exploration of Error

Corpus	Entries	NN Accuracy
SemEval2017	10,000	52%
SemEval2012-2017	23,000	45%
Sentiment 140	1,600,000	77%
Concatenated Corpuses	1,623,000	76.8%

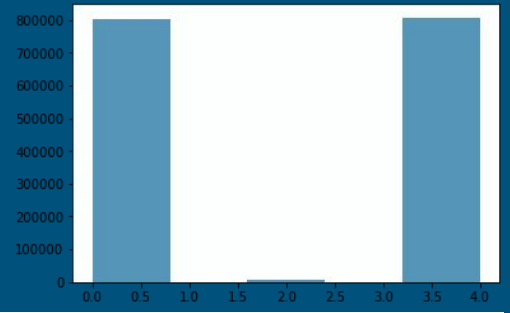
Model predicts corpus with 77.25% accuracy

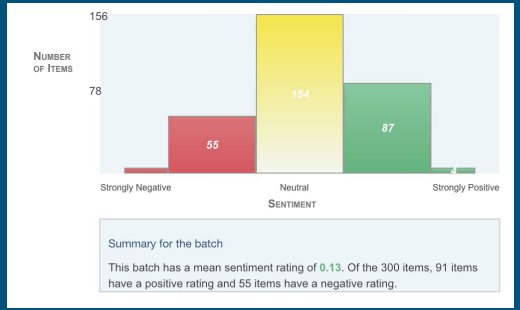
Mechanical Turk Analysis of 300 tweets/comments:

- Corpus: 69% accuracy (10% model loss)
- Sample Reddit (r/portland): 59% accuracy (15% overfit)

Sources of error:

- Model Loss
- Inaccurate corpus categorization
- Inaccurate categorization exposure (50/50)





Corpora Challenges

1. Mislabeling - Some Sentiment 140 tweets are miscategorized. For example: 'I love my sister a.k.a brianna!!! Missing her..we might be away by distance,never in heart..' is labeled Negative due to the presence of a :(in the original tweet. The accuracy of any model trying to predict Positive/Negative Sentiment on this corpus will consistently struggle to approach perfection due to these types of labels.
2. Domain adjustments - General tweet sentence structure is grammatically awful and doesn't necessarily translate well to Reddit where individuals tend to hold more long-form conversation.
3. Human biases reflected in positive/negative assessments - while averaging shows similar assessments, the ‘neutral’ category is in practice the most common. Ignoring ‘neutral’ from corpus training is one major source of model loss (10%)

Potential Future Improvements

- Corpus Improvements
  - Human annotate all 1.6million tweets into three categories instead of two
  - Randomize sample such that positive / neutral / negative tweets match frequency of occurrence in the wild
  - Create and annotate a large Reddit corpus to account for differences in media
  - Increase the sample size 10x
- Model Improvements
  - Incorporate a feed forward model that inherits context from parent to child (RNN)
  - Concatenate all word vectors from a tweet into a 2x2 matrix and feed into a CNN (similar to an image)
- Close the loop
  - Incorporate findings into a chrome extension for seamless reddit browsing (js->python->js)
  - Submit paper for publish - failing publication, self publish on medium

References

- Corpora
  - SemEval
  - Sentiment140
- Reference Papers
  - Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
  - Sentiment analysis of Twitter data
  - BB twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs
  - TwiSe at SemEval-2017 Task 4: Five-point Twitter Sentiment Classification ar Quantification
  - Twitter Sentiment Classification using Distant Supervision
- Reference Blogs
  - https://ahmedbesbes.com/sentiment-analysis-on-twitter-using-word2vec-and-keras.html
  - https://medium.com/@thoszymkowiak/how-to-implement-sentiment-analysis-using-word-embedding-and-convolutional-neural-networks-on-keras-163197aef623
- Github Repo
  - https://github.com/dswald/sentiment_analysis.git