Sentiment movie analysis
Introduction

- the Naive Bayes method shows the good accuracy and easy principle in classification method.
- However, it is acceptable that the Naive Bayes has some disadvantages to some extent.
  - Independence
  - Ignore relationship
  - Large computation
Introduction

- Aspired by “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”
- Select words by phrase pattern of POS

<table>
<thead>
<tr>
<th>First Word</th>
<th>Second Word</th>
<th>Third Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>JJ</td>
<td>NN or NNS</td>
<td>anything</td>
</tr>
<tr>
<td>RB, RBR, or RBS</td>
<td>JJ</td>
<td>not NN nor NNS</td>
</tr>
<tr>
<td>JJ</td>
<td>JJ</td>
<td>not NN nor NNS</td>
</tr>
<tr>
<td>NN or NNS</td>
<td>JJ</td>
<td>not NN nor NNS</td>
</tr>
<tr>
<td>RB, RBR, or RBS</td>
<td>VB, VBD, VBN, or VBG</td>
<td>anything</td>
</tr>
</tbody>
</table>
Introduction

- Select some specific words/phrases
  - Not long
  - Show perspective
  - Own sentiment degree
  - Follow some pattern
Introduction

- extract some specific patterns from context

<table>
<thead>
<tr>
<th></th>
<th>First Word</th>
<th>Second Word</th>
<th>Third Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>JJ</td>
<td>NN/NNS</td>
<td>anything</td>
</tr>
<tr>
<td>2</td>
<td>RB/RBR/RBS</td>
<td>JJ</td>
<td>not NN nor NNS</td>
</tr>
<tr>
<td>3</td>
<td>JJ</td>
<td>JJ</td>
<td>not NN nor NNS</td>
</tr>
<tr>
<td>4</td>
<td>NN/NNS</td>
<td>JJ</td>
<td>not NN nor NNS</td>
</tr>
<tr>
<td>5</td>
<td>RB/RBR/RBS</td>
<td>VB/VBD/VBG/VBN</td>
<td>anything</td>
</tr>
<tr>
<td>6</td>
<td>NN</td>
<td>NV/VBD/VBG/VBN</td>
<td>anything</td>
</tr>
<tr>
<td>7</td>
<td>NN</td>
<td>RB/RBR/RBS</td>
<td>VB/VBD/VBG/VBN</td>
</tr>
</tbody>
</table>

Table 1 phrase pattern
Introduction

- extract some specific words
  - Adjective
  - Adverb
  - verb
Method

● The first step of algorithm is to extract some specific patterns from context.
● The second method is use Naive Bayes method to all words that satisfy the pattern.
● The final step is to calculate accuracy.
● Compare with other methods.
Evaluation

- Extract pattern and Naive Bayes

```
[INFO]  Fold 0 Accuracy: 0.790000
[INFO]  Fold 1 Accuracy: 0.870000
[INFO]  Fold 2 Accuracy: 0.815000
[INFO]  Fold 3 Accuracy: 0.870000
[INFO]  Fold 4 Accuracy: 0.805000
[INFO]  Fold 5 Accuracy: 0.845000
[INFO]  Fold 6 Accuracy: 0.860000
[INFO]  Fold 7 Accuracy: 0.845000
[INFO]  Fold 8 Accuracy: 0.850000
[INFO]  Fold 9 Accuracy: 0.860000
    [INFO]  Accuracy: 0.841000
```
Evaluation

- Extract words

<table>
<thead>
<tr>
<th>INFO</th>
<th>Fold 0 Accuracy: 0.680000</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFO</td>
<td>Fold 1 Accuracy: 0.645000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 2 Accuracy: 0.605000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 3 Accuracy: 0.615000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 4 Accuracy: 0.605000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 5 Accuracy: 0.655000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 6 Accuracy: 0.640000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 7 Accuracy: 0.610000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 8 Accuracy: 0.605000</td>
</tr>
<tr>
<td>INFO</td>
<td>Fold 9 Accuracy: 0.655000</td>
</tr>
<tr>
<td>INFO</td>
<td>Accuracy: 0.631500</td>
</tr>
</tbody>
</table>
Evaluation

- POS Naive Bayes vs other methods
Function 2: analize input review

- We let people input a review of a movie and we will justify the degree of good and bad for this review.
- We set different thresholds and classify review into 5 different star degree.
- 1 star, 2 star, 3 star, 4 star, 5 star. 4~5 star means positive, 1~2 star means negative.
- The more star means more agreed degree, the fewer star means more dislike degree.
Function 2: analize input review

- Please input your review or input 'esc' to quit:
- I will say that the movie's idea that two best friends can't agree on a better solution than to have competing weddings on the same day because of their childhood dreams is silly. However with that said, I still found the movie entertaining. Some of the things Hathaway and Hudson do to sabotage the each others weddings are really funny. It would be nice though if movie studios would quit showing so many of the funny scenes in movie trailers. Overall, a cute movie!

- output:
- ****
Function 2: analize input review

- Please input your review or input 'esc' to quit:
  - Only bought this because my best friend & I got married on the same day. We both fell asleep but we did get a laugh as we could sympathize with the ridiculousness of planning a wedding. (And because while goofing around I accidentally busted her lip just one week before the wedding.)

- output:
  - **
Conclusion

- We combine the POS and Naive Bayes method with better accuracy.
- The final accuracy is about 84.1%, better than the PA4(51%), Naive bayes(81%) and this paper(74%).
- We can analyze the sentiment of the real time input review into 5 different level.
Thank you!
Sentiment Analyzer on Yelp Restaurant Comments

Ruicong Cai
Zhe Zan
Yelp Comments:

Mark S.
San Diego, CA

8/22/2016
Excellent BBQ, good prices, friendly staff, and the best peach cobbler I've ever had. Highly recommended for anyone looking for a great place to eat.

Katie V.
Philadelphia, PA

10/26/2016
The original and the best, most delicious barbecue I have eaten in the Brazos Valley. The brisket is perfect, as are the ribs, jalapeño cheddar sausage, and all of the sides. Banana pudding is perfect, the setting is so laid back Texas. We love this place!
Data:

- From the comments of top restaurants.
- The data consists of the following items:
  1. Vote of the comment (funny, useful, cool)
  2. User ID
  3. Comment ID
  4. Date
  5. Comments
- Shuffle the data.
"If I could give this place less than one star, I would. I have no idea who gave them that."

"Take it from me; avoid this place at all cost. The only time I go is when I am in the Swagger Lounge."

"I use to order here fairly often. The past 2 years their food has been getting worse."

"Terrible service. Food unremarkable. Waiter disappeared for 45 minutes to serve us."

"I have been to this restaurant twice and was disappointed both times. I won't go again."

"We stopped at Papa J's last Friday night (8/1) for a round of drinks. There were no tables open."

"Food was NOT GOOD at all! My husband & I ate here a couple weeks ago for the first time and we won't be back."

"Had dinner with a friend. My friend ordered veal and they brought him sausage."

"We visited on 11/15 with a party of 15. While I know a party of 15 can be overwhelming, you could have given us a table."

"I've never posted a Yelp review before. This meal was so horrible that I don't know where to begin."

"This is the absolute WORST Steak N Shake I've ever been to."

"I went here at 3 PM between the lunch rush and the dinner rush, and the restaurant was dead."

"The only thing worse than the food is the service."

"This was the most horrible experience at a restaurant I have had in years!!!!!!!"

"Terrible wait staff couldn't even seat us. Before we, and another party walked out."

"Food is good, what you'd expect from Steak n Shake. THE SERVICE IS AWFUL. so in 15 minutes we left."

"The service was fast but the food was terrible and so was the service. I had a big party."

"I should have known better than to stop here, but I was nursing a hangover and..."

"You know what you're getting with a Steak N Shake: it's about one rung up from a food court."

"I love Steak N Shake. This one, however, leaves a lot to be desired. The food was terrible."

"I like the occasional steak and shake stop... but this one has to be the worst."

"Every time we come here the service is laughably bad. On this visit a tabe which...

"Wow. Dirty and slow. The floors felt like they had the days burger grease spilt."

"The staff is very rude at the drive thru to the point of telling me at 2:02 pm to "Get out of here!"

"This location is terrible. The drive-thru workers are rude and they give you crap."

"Awful in every category. The service is the worst I've ever seen. We were waiting..."

"I really don't know how this place stays open. I've been here a couple of times..."

"If could give toys cunt of a human being "Sue1" a manager negative 1,000,000 not..."

"The hostess (Jenn of Jess, I'm not sure) is atrocious. I am autistic and asked..."
Structure

Load training data

Tokenization

(Remove) Stopwords

Bag of Words

Stem

Lemmatize

Bigram

Maxent Model

Training

Normal Testing

Sentence Testing

Evaluation
We used 4 categories of feature:

1. Bag of Word Model (Baseline)
2. Stemmed Words
3. Lemmatized Words
4. Bigram
Maxent Model

- Exponential (log-linear, maxent, logistic, Gibbs) models:
  - Make a probabilistic model from the linear combination $\Sigma \lambda_i f_i(c, d)$
  
  \[
P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c, d)}{\sum_{c'} \exp \sum \lambda_i f_i(c', d)}
  \]

  - Makes votes positive
  - Normalizes votes

  - $P(\text{LOCATION} \mid \text{in Québec}) = \frac{e^{1.8}e^{-0.6}}{(e^{1.8}e^{-0.6} + e^{0.3} + e^0)} = 0.586$
  - $P(\text{DRUG} \mid \text{in Québec}) = \frac{e^{0.3}}{(e^{1.8}e^{-0.6} + e^{0.3} + e^0)} = 0.238$
  - $P(\text{PERSON} \mid \text{in Québec}) = \frac{e^0}{(e^{1.8}e^{-0.6} + e^{0.3} + e^0)} = 0.176$

- The weights are the parameters of the probability model, combined via a “soft max” function
Results and Analysis

- Single – category feature: (Baseline)

<table>
<thead>
<tr>
<th>N-FOLD CROSS VALIDATION RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy: 0.81775    precision 0.838529623691</td>
</tr>
<tr>
<td>recall 0.81821424837    f-measure 0.815084552685</td>
</tr>
</tbody>
</table>

- Single – category feature: (Without Stopwords)

<table>
<thead>
<tr>
<th>N-FOLD CROSS VALIDATION RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy: 0.81725    precision 0.837560216013</td>
</tr>
<tr>
<td>recall 0.817450440755    f-measure 0.81468628399</td>
</tr>
</tbody>
</table>
Results and Analysis

- Category features: (+ stemmed words)

```
N-FOLD CROSS VALIDATION RESULT
accuracy: 0.81975    precision 0.837613443651
recall 0.819774668243 f-measure 0.817248510066
```

- Category features: (+ lemmatized words)

```
N-FOLD CROSS VALIDATION RESULT
accuracy: 0.81825    precision 0.83518704751
recall 0.818827434827 f-measure 0.816130188551
```
Results and Analysis

- 4 - category features: (+ bigrams)

N-FOLD CROSS VALIDATION RESULT
accuracy: 0.85125   precision 0.860374597787
recall 0.851806195242 f-measure 0.850226689286

- 4 - category features: (Sentence-based)

SENTENCES: N-FOLD CROSS VALIDATION RESULT
accuracy: 0.93425   precision 0.934524043769
recall 0.93428800695 f-measure 0.934207504259
Quite a few comments are combination of both positive and negative sentences.

We ordered this through aggiefood for delivery. I absolutely LOVED my ribs and my grandbaby tore up the mac and cheese and ranch potatoes! The only let down was my sweet heart's sliced beef sandwich. It was smallish and flattened, seemed to be a thrown together afterthought. We'll definitely order again! Just a bit more carefully
Thank You
Aspect Based Sentiment Analysis

Divyesh Tekale(923004428)
Mragank Kumar Yadav(625005280)
Sentiment Analysis

• Extract opinions, views, emotions from unstructured text.

• Examples:
  – “My goodness, everything from the fish to the rice to the seaweed was absolutely amazing”  🤩  Polarity
  – “The food was terrible and overly priced”  😞  Polarity
Aspect Level Sentiment Analysis

• Two phased procedure:
  – Aspect Extraction
  – Polarity computation of that Aspect.

• Example: “Anyway, the food is good, the price is right and they have a decent wine list”

  Aspect=food   Polarity 🤩
  Aspect=price  Polarity 🤩
Task Overview

- SemEval-2014 Restaurant data.
- CRF model (CRF++) to extract aspects.
- POS tagger using TagChunk by Hal.
- Porters Stemmer to stem the words.
- Subjectivity Lexicon dictionary to determine the stemmed word polarity.
Aspect Extraction Training Phase

Parse train xml file → Run POS tagger → Generate train.data(Conll) file

Model file is generated → Run CRF++ on train.data(Conll) file
Sample Train.data(Conll) file

<table>
<thead>
<tr>
<th>Word</th>
<th>POS</th>
<th>Chunk</th>
<th>Is-Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>But</td>
<td>CC</td>
<td>B-O</td>
<td>False</td>
</tr>
<tr>
<td>the</td>
<td>DT</td>
<td>B-NP</td>
<td>False</td>
</tr>
<tr>
<td>staff</td>
<td>NN</td>
<td>I-NP</td>
<td>True</td>
</tr>
<tr>
<td>was</td>
<td>VBD</td>
<td>B-VP</td>
<td>False</td>
</tr>
<tr>
<td>so</td>
<td>RB</td>
<td>B-ADJP</td>
<td>False</td>
</tr>
<tr>
<td>horrible</td>
<td>JJ</td>
<td>I-ADJP</td>
<td>False</td>
</tr>
</tbody>
</table>
Aspect Extraction Testing Phase

Parse test xml file → Run POS tagger → Generate test.data (Conll) file

Predicted results files is generated → Parse the generated output file → Run CRF++ on test.data (Conll) & model file
Polarity Computation of the Predicted Aspects

1. Parse predicted Results file
2. Run Porter’s Stemmer around aspect
3. Use Subjectivity Lexicon Dict
4. Generate Results with polarity
5. Compute Contextual Polarity
Sample Results

- **Text:** In addition, the food is very good and the prices are reasonable.
  
  **Aspect Terms**
  - **Aspect**=food  **Polarity**=positive
  - **Aspect**=prices  **Polarity**=positive

- **Text:** Their calzones are horrific, bad, vomit-inducing, YUCK.
  
  **Aspect Terms**
  - **Aspect**=calzones  **Polarity**=negative
Challenges faced

- Handling punctuations while generating training data (Conll file) for CRF model.
- Handling different forms of words while searching in subjectivity lexicon dictionary. Eg: "fishing", "fished", and "fisher".
- Getting a balance between recall and precision values.
Results

• Aspect Extraction Metrics:
  Precision = 98 %
  Recall     = 65 %
  F-Score    = 78 %

• Polarity Metrics(5 word search around the extracted aspect term):
  Precision = 76 %
Questions
Sentiment Analysis : TripAdvisor

Savinay Narendra
Surya Akella
Problem Statement

- Analyzing Trip Advisor reviews of hotels
- **Sentiment Analysis**
  - Analyze an individual’s opinion or mood
  - Get insights into customer opinions
  - Predict Buying Signals
- **Multiclass Classification (Why?)**
  - 3-class: Positive(> 3), Negative(== 3), Average(< 3)
  - 5-class: Awesome(5), Good(4), Average(3), Fair(2), Poor(1)
Overview of Approach

Breadth of Techniques Explored:

- Naive Bayes (Baseline)
- Naive Bayes - Support Vector Machines (NBSVM)
- Deep Learning
  - Recurrent Neural Networks
  - Convolutional Neural Networks
Dataset

Data Preprocessing

- For NB and NBSVM, extracted 5000 examples belonging to each class into .txt files.
- For RNN and CNN, extracted data from 1325 files into a .csv file.
Naive Bayes

- Probabilistic classifier
- Baseline for evaluation
- Used unigram + bigram word features
- Binarized version of NB with add-1 Laplace smoothing.
- 2500 examples of each class - 10 fold cross validation

\[ c_{NB} = \arg \max_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j) \]
Naive Bayes Results (3-Class)

Accuracy : 0.812883

Confusion Matrix :
[[172 61 17]
 [49 197 4]
 [8 9 233]]

Classification Report :

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>0.75</td>
<td>0.69</td>
<td>0.72</td>
<td>250</td>
</tr>
<tr>
<td>neg</td>
<td>0.74</td>
<td>0.79</td>
<td>0.76</td>
<td>250</td>
</tr>
<tr>
<td>pos</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>250</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>750</td>
</tr>
</tbody>
</table>
Naive Bayes Results (5-Class)

Accuracy : 0.653180

Confusion Matrix :
[[152 2 66 17 13]
 [11 187 2 50 0]
 [33 2 161 22 32]
 [45 44 5 154 2]
 [6 0 83 4 157]]

Classification Report :

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>average</td>
<td>0.62</td>
<td>0.61</td>
<td>0.61</td>
<td>250</td>
</tr>
<tr>
<td>awesome</td>
<td>0.80</td>
<td>0.75</td>
<td>0.77</td>
<td>250</td>
</tr>
<tr>
<td>fair</td>
<td>0.51</td>
<td>0.64</td>
<td>0.57</td>
<td>250</td>
</tr>
<tr>
<td>good</td>
<td>0.62</td>
<td>0.62</td>
<td>0.62</td>
<td>250</td>
</tr>
<tr>
<td>poor</td>
<td>0.77</td>
<td>0.63</td>
<td>0.69</td>
<td>250</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.66</td>
<td>0.65</td>
<td>0.65</td>
<td>1250</td>
</tr>
</tbody>
</table>
NBSVM

- Binary linear classifier - Adapted from “Sida Wang and Christopher D. Manning”.
- Novel SVM variant using NB log-count ratios as feature values.
- Interpolation between MNB and SVM: Trust NB unless the SVM is very confident.
- Adapted to work for Multi-class:
  - OnevsRest classification - N binary classifiers - For each, need real-valued confidence score.
  - OnevsOne classification - $N(N-1)/2$ binary classifiers - Voting scheme to choose best.
NBSVM results (3-Class)

Accuracy : 0.76933333333333

Confusion Matrix :
[[401 91 8]  
 [ 23 360 117]  
 [ 1 106 393]]

Classification Report :

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos</td>
<td>0.94</td>
<td>0.80</td>
<td>0.87</td>
<td>500</td>
</tr>
<tr>
<td>avg</td>
<td>0.65</td>
<td>0.72</td>
<td>0.68</td>
<td>500</td>
</tr>
<tr>
<td>neg</td>
<td>0.76</td>
<td>0.79</td>
<td>0.77</td>
<td>500</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.78</td>
<td>0.77</td>
<td>0.77</td>
<td>1500</td>
</tr>
</tbody>
</table>
# NBSVM results (5-Class)

**Accuracy**: 0.6396

### Confusion Matrix:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>342</td>
<td>141</td>
<td>14</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>287</td>
<td>97</td>
<td>16</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>216</td>
<td>228</td>
<td>35</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>13</td>
<td>135</td>
<td>282</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>34</td>
<td>460</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>2500</td>
</tr>
</tbody>
</table>

### Classification Report:

<table>
<thead>
<tr>
<th>Class</th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.74</td>
<td>0.68</td>
<td>0.71</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>0.44</td>
<td>0.57</td>
<td>0.50</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>0.48</td>
<td>0.46</td>
<td>0.47</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>0.76</td>
<td>0.56</td>
<td>0.65</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>0.86</td>
<td>0.92</td>
<td>0.89</td>
<td>500</td>
</tr>
</tbody>
</table>

**avg / total**: precision 0.66, recall 0.64, f1-score 0.64, support 2500
RNN

- Like FeedForward Networks
- Has multiple layers combined into one
- Result of one time step supplements the next layer
- Problem
  - Vanishing Gradient
- Hence, we use LSTM architecture of RNN
- LSTM helps overcome this problem
RNN (Word Embeddings)

- Maps words to vectors
- Each vector has multiple dimensions
- Stores information about the word
- Finds relations in text
Results (RNN Classifier)

- **Accuracy**
  - 3 class ≈ 74%
  - 5 class ≈ 48%

### 3-class Results

<table>
<thead>
<tr>
<th>Classification Report</th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0.56</td>
<td>0.46</td>
<td>4541</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>4043</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.76</td>
<td>0.96</td>
<td>19089</td>
</tr>
<tr>
<td><strong>avg / total</strong></td>
<td>0.62</td>
<td>0.74</td>
<td>0.67</td>
<td>27673</td>
</tr>
</tbody>
</table>

### 5-class Results

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.48</td>
<td>0.44</td>
<td>0.45</td>
<td>2198</td>
</tr>
<tr>
<td>2.0</td>
<td>0.47</td>
<td>0.02</td>
<td>0.04</td>
<td>2343</td>
</tr>
<tr>
<td>3.0</td>
<td>0.35</td>
<td>0.19</td>
<td>0.24</td>
<td>4043</td>
</tr>
<tr>
<td>4.0</td>
<td>0.42</td>
<td>0.45</td>
<td>0.44</td>
<td>9016</td>
</tr>
<tr>
<td>5.0</td>
<td>0.53</td>
<td>0.72</td>
<td>0.61</td>
<td>10073</td>
</tr>
</tbody>
</table>
**CNN**

Our Model

- **First layer** - embeds words into low-dimensional vectors
- **Second layer** - Performs convolutions over the embedded word vectors
- **Max-pool** the result of the convolutional layer into a long feature vector
- **Classify** the result using a softmax layer
Results (CNN)

CNN Classifier's Accuracy: 0.86821

('Confusion Matrix:', array([[ 1990, 2551],
[ 1096, 22036]]))

Classification Report

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.64</td>
<td>0.44</td>
<td>0.52</td>
<td>4541</td>
</tr>
<tr>
<td>1</td>
<td>0.90</td>
<td>0.95</td>
<td>0.92</td>
<td>23132</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.85</td>
<td>0.87</td>
<td>0.86</td>
<td>27673</td>
</tr>
</tbody>
</table>
## Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>2-class (Accuracy)</th>
<th>2-class (F-score)</th>
<th>3-class (Accuracy)</th>
<th>3-class (F-score)</th>
<th>5-class (Accuracy)</th>
<th>5-class (F-score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>-</td>
<td>-</td>
<td>81.28%</td>
<td>80%</td>
<td>65.32%</td>
<td>65%</td>
</tr>
<tr>
<td>NBSVM</td>
<td>-</td>
<td>-</td>
<td>77%</td>
<td>77%</td>
<td>64%</td>
<td>64%</td>
</tr>
<tr>
<td>RNN</td>
<td>-</td>
<td>-</td>
<td>74%</td>
<td>67%</td>
<td>48%</td>
<td>44%</td>
</tr>
<tr>
<td>CNN</td>
<td>86.8%</td>
<td>86%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Conclusion

- Much better accuracy than majority classifier (5 classes - 20%, 3 classes - 33%)
- Bag of features models are still strong performers on snippet sentiment classification tasks.
- Naive Bayes giving the best performance on this dataset. (Not so Naive!)
- NBSVM performance very close to NB.
- Using bigram and trigram features improved performance.
- For RNNs, word embeddings improved performance - complementary to tf-idf, bigram and trigram features.
- RNNs seem to perform better for longer text reviews. Accuracy will be increased with more training data. (Currently only 10%)
Thank You!!
Insult Detection in Social Media Text Content

- Aditya Nanjangud, 625007600
- Navneet Gupta, 226000691
Table of Contents

• The need for abuse detection
• Methodology
• Results
• Observations
• Challenges (f)aced
• References
Intro

• Anonymity allows people to post insulting comments.
  • Example: kill yrslef a$$hole
• Common in Facebook, Twitter, Blogs
• Huge content makes manual classification infeasible.
• Rule based engine cannot scale with growing forms of abuse and vocabulary.
• ML and NLP algorithms can help to automate the classification task.
Data

• Provided by Kaggle as a part of a competition

• Training Data:
  • 6594 sentences
  • Ex: (Insult, Date, Comment)
  • 1,20120502173553Z,""""Either you are fake or extremely stupid...maybe both..."""
  • 0,20120612052926Z,""""But how would you actually get the key out?"""

• Test Data:
  • 2235 sentences
  • Ex: (id,Insult,Date,Comment,Usage)
  • 12,1,20120602124231Z,""""\xa0HAHAHAHAHAH, you are a delusional moron."""",PrivateTest
Preprocessing

• Removal of HTML tags
• Removal of URLs
• Correction of words like em, yo, u, d etc.
• Basic custom stemming
• Replace custom abuses like "f***" with "xexp”
• Normalizing unicode data like replacing \xc2, \xa0 with non-breaking space
• Replace some punctuations to clean up the text
Feature Extraction

- Word CountVectorizer
- Char CountVectorizer
- Word Tfidf (n-grams)
- Char Tfidf (n-grams)
- Number of uppercase words
- Ratio of uppercase words
- Day and Time
- Misspellings
- Number of bad words
- Ratio of bad words
- Number of times Addressing (@) used.
- Number of "xexp" ~ f***
- Mean and maximum word length
Feature Selection

• To Select the best features out of 100s of thousands of features.

• Chi-Squared Test : Selecting features with the highest dependence on the occurrence of the classes it has to be classified into.

• Earlier combined all the features and then ran feature selection.

• But running chi-squared test after each feature extraction led to better results.
Classification

- Support Vector Machines
- Naïve Bayes
- Stochastic Gradient Descent
- Logistic Regression
- Used a VotingClassifier to combine different combinations.
- Weighted averaging of SVM and LR gave the best results.
Parameter Tuning

• Used GridSearchCV to tune parameters and features.

• Cross validation scores to decide the weights for the classifiers in the voting classifier.
Results

Accuracy : 0.74
AUC (ROC ): 0.826
AUC (Recall vs Precision) : 0.83

Macro

  Precision  0.768
  Recall  0.737
  F-score  0.734

Micro

  Precision  0.743
  Recall  0.743
  F-score  0.743

Class wise

  Precision [0.6956, 0.8405]
  Recall [0.8981, 0.5775]
  F-score [0.7840, 0.6846]
Results Graphs

Cumulative Accuracy over features (CV - word/char)

Accuracy (AUC-RCC)

Features

word_cv
char_cv
Observations

• Data Preprocessing didn’t help much.

• In terms of features, TfIdf scores of n-gram characters mattered most. (perhaps the reason was weird spellings and grammar)

• Initially we selected the best features from a combined feature set. But later did the feature selection for each type of features individually – better results.

• Simpler models such as SVM and LR gave best results. We employed a weighted ensemble of them.
Challenges

• Feature Extraction
  • Preprocessing
• Feature Selection
• Choice in Classifiers
• Parameter Tuning
References

• Abusive Language Detection in Online User Content, Chikashi Nobata et al., WWW’16 Proceedings of the 25th International Conference.

• Data - https://www.kaggle.com/c/detecting-insults-in-social-commentary


• Code - https://github.com/navgupta14/abuse-detector
Analysis in Twitter Gender Classification

Chuong Trinh
Motivation

• Growing interest in automatically predicting the gender of authors from texts:
  • Opinions, political stances, styles, and preferences may be unique to each gender
  • Useful to individuals, companies, and governments for personal recommendation, customization, targeted advertising, political analysis, and policy formulation.
Why Gender Classification from Tweets is Hard!

- Limited characters (140) per tweet
- Lots of spamming, advertising accounts, media sources, bots, etc.
- User’s profile privacy
- Users construct their identity through interacting with other users! (Marwick and boyd, 2011) – all depend on the context
- For example
  - Tweet 1: I’m walking on sunshine <3 #and don’t you feel good
  - Tweet 2: lalaloveya <3
  - Tweet 3: @USER loveyou ;D
Dataset & Baseline

• CrowdFlower (kaggle – data challenge site)
  • 20,000 tweets – collected in 2015
    • Human Amazon Turker labeling + CrowdFlower’s labeling system
    • ~ 14,000 tweets can be used (non-English, low confidence, or unreadable is ignored)
    • Labels: male + female + brand

• Men are more likely to talk at another account
• Women are more likely to use emoji
• Current accuracy: ~60%
GloVe: Global Vectors for Word Representation

- Unsupervised learning algorithm for obtaining vector representations for words
- Ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning
- Pre-trained matrix model: Twitter – 2 billions tweets, 27 billions tokens, 25 to 200 dimensional features
Doc2Vec - Distributed Memory Model of Paragraph Vectors (PV-DM)

- **Word2vec**: Converts a word into a vector → losing ordering of the words
- **Doc2vec**: Learn word features + aggregate all the words in a sentence into a vector
  - Unsupervised algorithm that converts variable-length text to fixed-length feature representation.


D: \(N \times p\) matrix paragraph vector (each paragraph is mapped to \(p\)-dimensional features vector)  
W: \(M \times q\) matrix word vector (each word is mapped to \(q\)-dimensional features vector)
## Analysis & Evaluation

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Word-freq</th>
<th>Word-freq + PCA</th>
<th>Doc2vec</th>
<th>GloVe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male &amp; Female &amp; Brand</td>
<td>0.5629</td>
<td>0.5716</td>
<td>0.5708</td>
<td>0.5872</td>
</tr>
<tr>
<td>Male &amp; Female</td>
<td>0.6054</td>
<td>0.6023</td>
<td>0.6172</td>
<td>0.6500</td>
</tr>
</tbody>
</table>

Accuracy

![Accuracy Chart]

- **Word-freq**: Blue
- **Word-freq + PCA**: Orange
- **Doc2vec**: Cyan
- **GloVe**: Yellow
## Analysis & Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Word-freq</th>
<th>Word-freq + PCA</th>
<th>Doc2vec</th>
<th>GloVe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Precision</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>0.4888</td>
<td>0.5131</td>
<td>0.4898</td>
<td>0.5342</td>
</tr>
<tr>
<td>Female</td>
<td>0.5678</td>
<td>0.5838</td>
<td>0.6043</td>
<td>0.5930</td>
</tr>
<tr>
<td>Brand</td>
<td>0.6341</td>
<td>0.5961</td>
<td>0.6027</td>
<td>0.6294</td>
</tr>
<tr>
<td><strong>Recall</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>0.4359</td>
<td>0.3564</td>
<td>0.4183</td>
<td>0.4312</td>
</tr>
<tr>
<td>Female</td>
<td>0.6060</td>
<td>0.6132</td>
<td>0.6050</td>
<td>0.6798</td>
</tr>
<tr>
<td>Brand</td>
<td>0.6580</td>
<td>0.7770</td>
<td>0.7096</td>
<td>0.6477</td>
</tr>
<tr>
<td><strong>F1 score</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>0.4608</td>
<td>0.4203</td>
<td>0.4512</td>
<td>0.4771</td>
</tr>
<tr>
<td>Female</td>
<td>0.5862</td>
<td>0.5981</td>
<td>0.6046</td>
<td>0.6334</td>
</tr>
<tr>
<td>Brand</td>
<td>0.6457</td>
<td>0.6745</td>
<td>0.6516</td>
<td>0.6383</td>
</tr>
</tbody>
</table>

First 3 principal components
Black: brand; Red: female; Blue: Male
Conclusion

• After all, we’re not all that much different. We use a lot of the same words

• GloVe performs best because its underlying concept that distinguishes man from woman, i.e. sex or gender, or king and queen.

• Doc2vec performs weaker than GloVe because it could be the lack of its pre-trained model from very large corpus (only unsupervised learning on training data)
Thank you
Information Extraction from Wikipedia

Bhavik Ameta(225008988), Shobhit Jain(625007846)
Introduction

Relation Extraction can improve the question answering and information retrieval.

Eg. <Person, BornIn>, <Org., HQ>

**Snowball** is a bootstrapped relation extraction method.

Seeds + Data = Relations!
Snowball Algorithm: Terminology

• **Snowball Pattern:** `<left_vector, ORG, mid_vector, LOC, right_vector>`

• **Tags:** ORG (organization) and LOC (headquarter location)

• Vectors have TF of words as weights

• **Snowball Relation:** `<ORG_name, LOC_name>`

• **Seed Tuples:** (Microsoft, Redmond), (Facebook, Menlo Park)……
Snowball Algorithm
Seaboard Corporation is a diverse multinational agribusiness and transportation conglomerate with integrated operations in several industries. In the United States, the company mainly engages in pork production and processing and ocean transportation. Internationally, Seaboard is primarily engaged in commodity merchandising, grain processing, sugar production and electrical power generation. The parent company, Seaboard Corporation, based in Merriam, Kansas, operates Seaboard Foods, Seaboard Marine, Seaboard Overseas & Trading Group (SOTG), Tabacal Agroindustria, Transcontinental Capital Corporation, Ltd. (TCCB), Mount Dora Farms, and has 50% non-controlling interest in Butterball, LLC. Its principal operating divisions are Pork, Commodity Trading and...
Approach and Challenges

• Wikipedia data: Can use infobox for evaluation.
• Original Snowball paper uses Newspaper data.
• XML clean-up to obtain plain text.
• First used Stanford NER Tagger (days for tagging…)
• Switched to Spacy Tagger: less accurate but quicker
• Co-reference tools are lot less accurate and slower still...!
Approach and Challenges

- **Dataset changes everything.** Typical Wikipedia line:

  **Nissan Motor Company Ltd** (Japanese: 日産自動車株式会社 Hepburn: Nissan Jidōsha Kabushiki-gaisha?), usually shortened to **Nissan** (/niːsən/ or UK /nɪsən/; Japanese: [nis an]), is a Japanese multinational automobile manufacturer headquartered in Nishi-ku, Yokohama. The company sells its cars under the

- Challenge: Characters other than English, meta tags, HTML symbols
- **Solution:** Use Unicode

- Challenge: Lot of unrelated words between Company and Location.
- **Solution:** Use log TF over contexts instead of raw count and remove low frequency words
Approach and Challenges:

- Raw counts can work on Newspaper dataset taken by original Snowball paper.
- Middle window words are more useful than left and right windows. Use higher window size to capture ORG, LOC in Wikipedia sentences.
Results

- Captured 230 <company, HQ> pairs from around 1082 articles.
- 118 correct relations
- Precision: 51.34 %
- Some relations missed due to Tagger and shorter articles.
- Negative matches due to <company, branch location> and <company, Founding location> pairs. Occur in same pattern as <company, HQ>
Conclusion

• Co-Reference resolution almost necessary for good relation extraction.
• Just NER not enough.
• Base form required for location and company
• More data for better results
References

- YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia, Fabian M. Suchanek, Gjergji
- Wikipedia data from: https://dumps.wikimedia.org/enwiki
- For cleaning wikipedia: https://github.com/attardi/wikiextractor
- spaCy Tagger: https://spacy.io/
Thank You......!