Sentiment movie analysis

DiWu
BowenLi

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

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Introduction

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

Introduction

- extract some specific patterns from context

	First Word	Second Word	Third Word
1	JJ	NN/NNS	anything
2	RB/RBR/RBS	JJ	not NN nor NNS
3	JJ	JJ	not NN nor NNS
4	NN/NNS	JJ	not NN nor NNS
5	RB/RBR/RBS	VB/VBD/VBG/VBN	anything
6	NN	NV/VBD/VBG/VBN	anything
7	NN	RB/RBR/RBS	VB/VBD/VBG/VBN

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
```

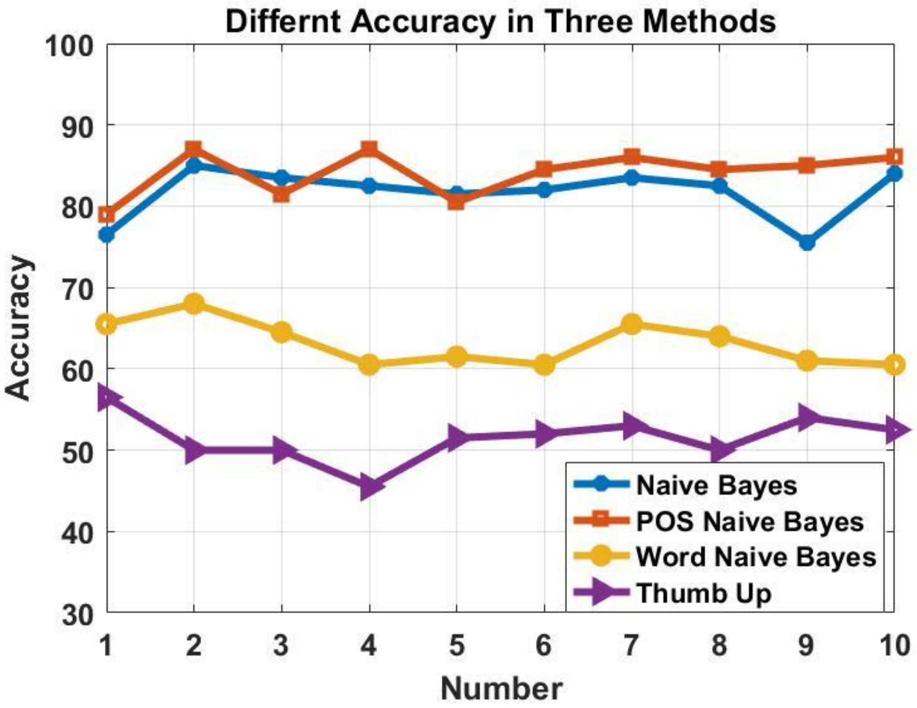
Evaluation

- Extract words

```
[INFO] Fold 0 Accuracy: 0.680000
[INFO] Fold 1 Accuracy: 0.645000
[INFO] Fold 2 Accuracy: 0.605000
[INFO] Fold 3 Accuracy: 0.615000
[INFO] Fold 4 Accuracy: 0.605000
[INFO] Fold 5 Accuracy: 0.655000
[INFO] Fold 6 Accuracy: 0.640000
[INFO] Fold 7 Accuracy: 0.610000
[INFO] Fold 8 Accuracy: 0.605000
[INFO] Fold 9 Accuracy: 0.655000
[INFO] Accuracy: 0.631500
```

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 sabatoge 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

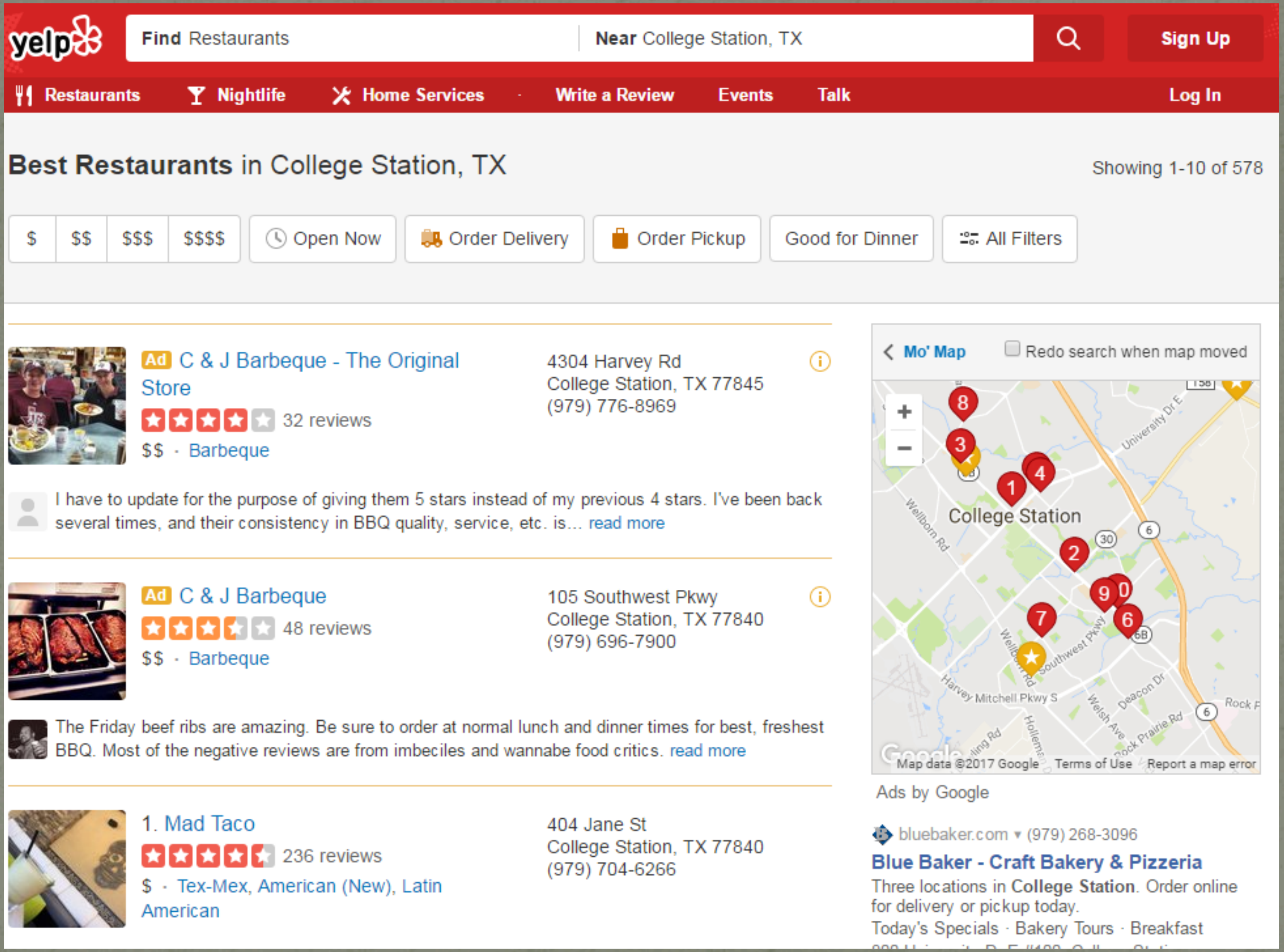
- We combine the POS and Naive Bayes method with better accuracy.
- The final accuracy is about 84.1%, better than the PA4(51%), Naive bayes(81%) and this paper(74%).
- We can analyze the sentiment of the real time input review into 5 different level.

Thank you!

Sentiment Analyzer on Yelp Restaurant Comments

Ruicong Cai
Zhe Zan

Yelp :



yelp
Find Restaurants
Near College Station, TX
Sign Up
Restaurants
Nightlife
Home Services
Write a Review
Events
Talk
Log In
Best Restaurants in College Station, TX
Showing 1-10 of 578
$ $$ $$$ $$$$
Open Now
Order Delivery
Order Pickup
Good for Dinner
All Filters
Ad C & J Barbeque - The Original Store
32 reviews
$$ · Barbeque
4304 Harvey Rd
College Station, TX 77845
(979) 776-8969
I have to update for the purpose of giving them 5 stars instead of my previous 4 stars. I've been back several times, and their consistency in BBQ quality, service, etc. is... read more
Ad C & J Barbeque
48 reviews
$$ · Barbeque
105 Southwest Pkwy
College Station, TX 77840
(979) 696-7900
The Friday beef ribs are amazing. Be sure to order at normal lunch and dinner times for best, freshest BBQ. Most of the negative reviews are from imbeciles and wannabe food critics. read more
1. Mad Taco
236 reviews
$ · Tex-Mex, American (New), Latin American
404 Jane St
College Station, TX 77840
(979) 704-6266
Mo' Map
Redo search when map moved
College Station
Map data ©2017 Google
Terms of Use
Report a map error
Ads by Google
bluebaker.com · (979) 268-3096
Blue Baker - Craft Bakery & Pizzeria
Three locations in College Station. Order online for delivery or pickup today.
Today's Specials · Bakery Tours · Breakfast

Yelp Comments:

Mark S.
San Diego, CA
6 friends
88 reviews

★★★★★ 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.

Was this review ...?

Useful | Funny | Cool

Katie V.
Philadelphia, PA
0 friends
16 reviews

★★★★★ 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!

Was this review ...?

Useful 1 | Funny | Cool

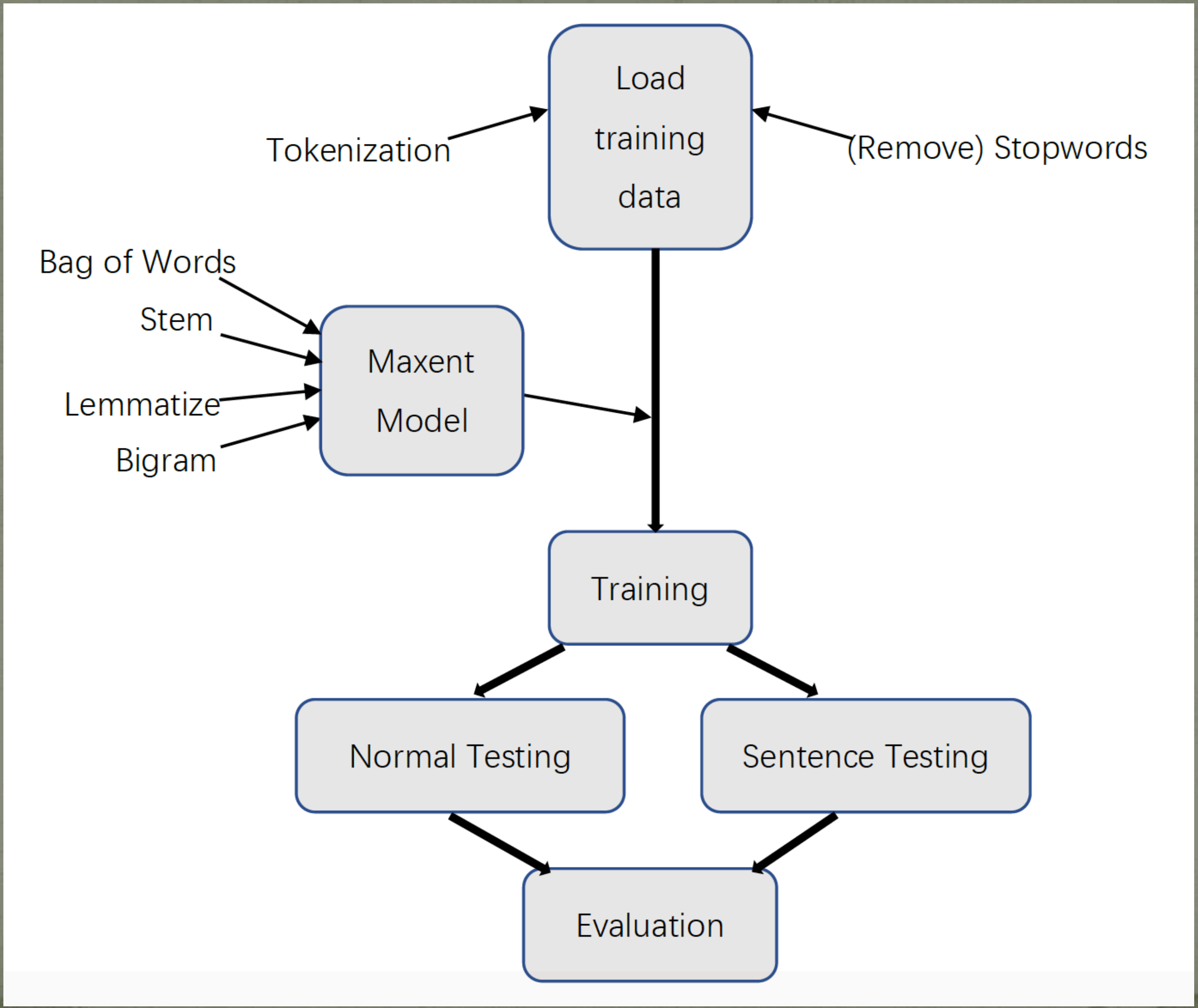
Data:

- 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.

How the Data Looks


```
1, "date": "2011-05-07", "text": "If I could give this place less than one star, I would. I have no idea who gave
1, "date": "2011-01-12", "text": "Take it from me; avoid this place at all cost.  The only time I go is when I am
1, "date": "2015-06-08", "text": "I use to order here fairly often.  The past 2 years their food has been getting
1, "date": "2011-03-14", "text": "Terrible service.  Food unremarkable.  Waiter disappeared for 45 minutes to serv
1, "date": "2013-12-26", "text": "I have been to this restaurant twice and was disappointed both times. I won't go
1, "date": "2014-08-03", "text": "We stopped at Papa J's last Friday night (8/1) for a round of drinks. There were
1, "date": "2014-11-21", "text": "Food was NOT GOOD at all! My husband & I ate here a couple weeks ago for the fi
1, "date": "2015-07-12", "text": "Had dinner with a friend. My friend ordered veal and they brought him sausage. T
1, "date": "2015-12-04", "text": "We visited on 11/15 with a party of 15.  While I know a party of 15 can be overw
1, "date": "2015-05-04", "text": "I've never posted a yelp review before.  This meal was so horrible that I downlo
1, "date": "2010-06-02", "text": "This is the absolute WORST Steak N Shake I've ever been to. \n\nThe bf and I got
1, "date": "2010-12-19", "text": "I went here at 3 PM between the lunch rush and the dinner rush, and the restaura
1, "date": "2011-12-20", "text": "The only thing worse than the food is the service.", "type": "review", "business
1, "date": "2011-12-29", "text": "This was the most horrible experience at a restaurant I have had in years!!!!!!
1, "date": "2012-03-13", "text": "Terrible wait staff  couldn't even seat us. Before we, and another party walked
1, "date": "2013-01-25", "text": "Food is good, what you'd expect from Steak n Shake. THE SERVICE IS AWFUL. so inc
1, "date": "2013-04-03", "text": "The service was fast but the food was terrible and so was the service. I had a b
1, "date": "2013-05-14", "text": "I should have known better than to stop here, but I was nursing a hangover and j
1, "date": "2013-06-17", "text": "You know what you're getting with a Steak N Shake: it's about one rung up from a
1, "date": "2013-06-28", "text": "I love Steak N Shake. This one, however, leaves a lot to be desired. The food of
1, "date": "2014-05-03", "text": "I like the occasional steak and shake stop.....    but this one has to be the sl
1, "date": "2014-05-13", "text": "Every time we come here the service is laughably bad. On this visit a tabe which
1, "date": "2014-06-04", "text": "Wow. Dirty and slow.  The floors felt like they had the days burger grease spil
1, "date": "2014-07-29", "text": "The staff is very rude at the drive thru to the point of telling me at 2:02 pm \
1, "date": "2014-08-09", "text": "This location is terrible. The drive-thru workers are rude and they give you cra
1, "date": "2014-08-28", "text": "Awful in every category. The service is the worst I've ever seen. We were waitin
1, "date": "2015-04-04", "text": "I really don't know how this place stays open. I've been here a couple of times.
1, "date": "2015-04-06", "text": "If could give toys cunt of a human being \"Sue\" a manager negative 1,000,000 ne
1, "date": "2015-06-18", "text": "The hostess (Jenn of Jess, I'm not sure) is atrocious. I am autistic and asked t
```

Structure


Load
training
data
Tokenization
(Remove) Stopwords
Bag of Words
Stem
Lemmatize
Bigram
Maxent
Model
Training
Normal Testing
Sentence Testing
Evaluation

Features

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 $\Sigma\lambda_i f_i(c,d)$

$$P(c \mid d,\lambda) = \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c',d)}$$

Makes votes positive

Normalizes votes

  - P(LOCATION|*in Québec*) = $e^{1.8}e^{-0.6}/(e^{1.8}e^{-0.6} + e^{0.3} + e^{0}) = 0.586$
  - P(DRUG|*in Québec*) = $e^{0.3} /(e^{1.8}e^{-0.6} + e^{0.3} + e^{0}) = 0.238$
  - P(PERSON|*in Québec*) = $e^{0} /(e^{1.8}e^{-0.6} + e^{0.3} + e^{0}) = 0.176$
- The weights are the parameters of the probability model, combined via a "soft max" function

Results and Analysis

- Single – category feature: (Baseline)

N-FOLD CROSS VALIDATION RESULT

accuracy: 0.81775 precision 0.838529623691

recall 0.81821424837 f-measure 0.815084552685

- Single – category feature: (Without Stopwords)

N-FOLD CROSS VALIDATION RESULT

accuracy: 0.81725 precision 0.837560216013

recall 0.817450440755 f-measure 0.81468628399

Results and Analysis

- 2 - category features: (+ stemmed words)

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

- 3 - 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

Results and Analysis

- Quite a few comments are combination of both positive and negative sentences.

Sarah S.
Somerville, TX
0 friends
7 reviews
3 photos

3/18/2017

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

Was this review ...?

Useful Funny Cool

Thank You

Aspect Based Sentiment Analysis

Divyesh Tekale(923004428)
Mragank Kumar Yadav(625005280)

TEXAS A&M UNIVERSITY

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”

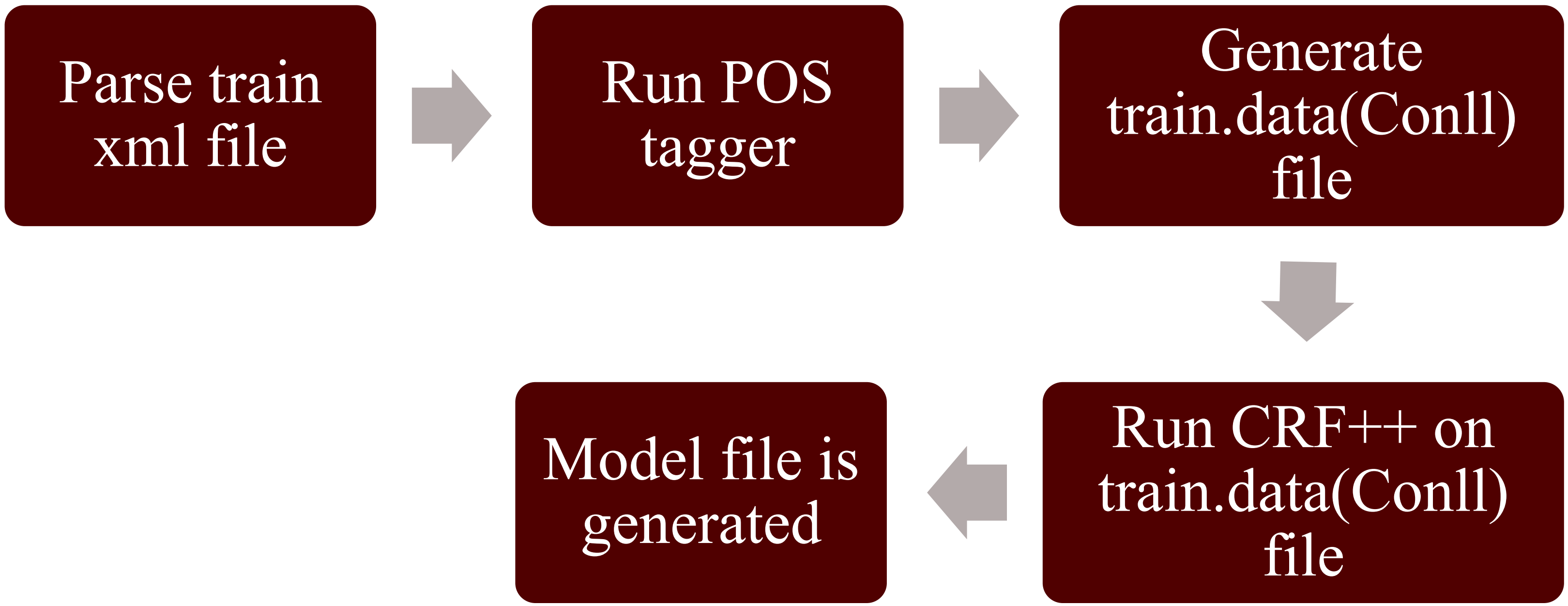
Aspect=food Polarity 

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.

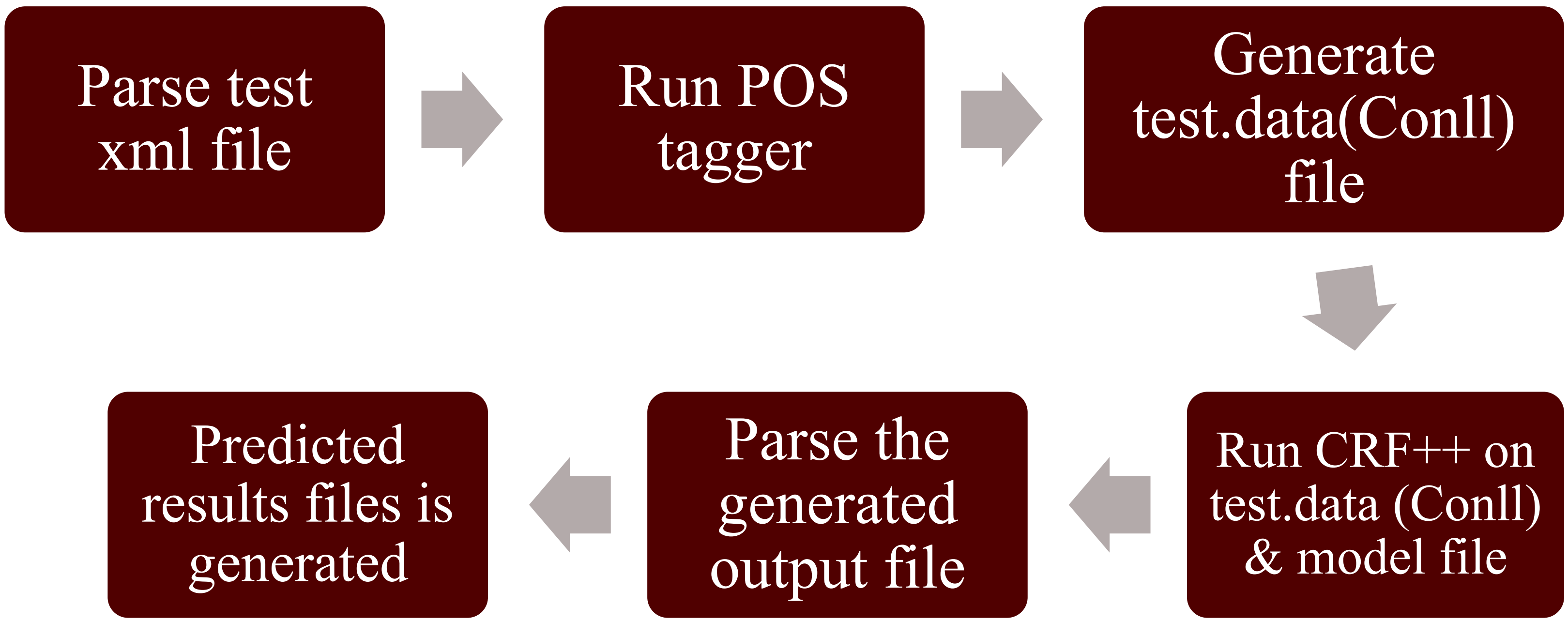
Aspect Extraction Training Phase



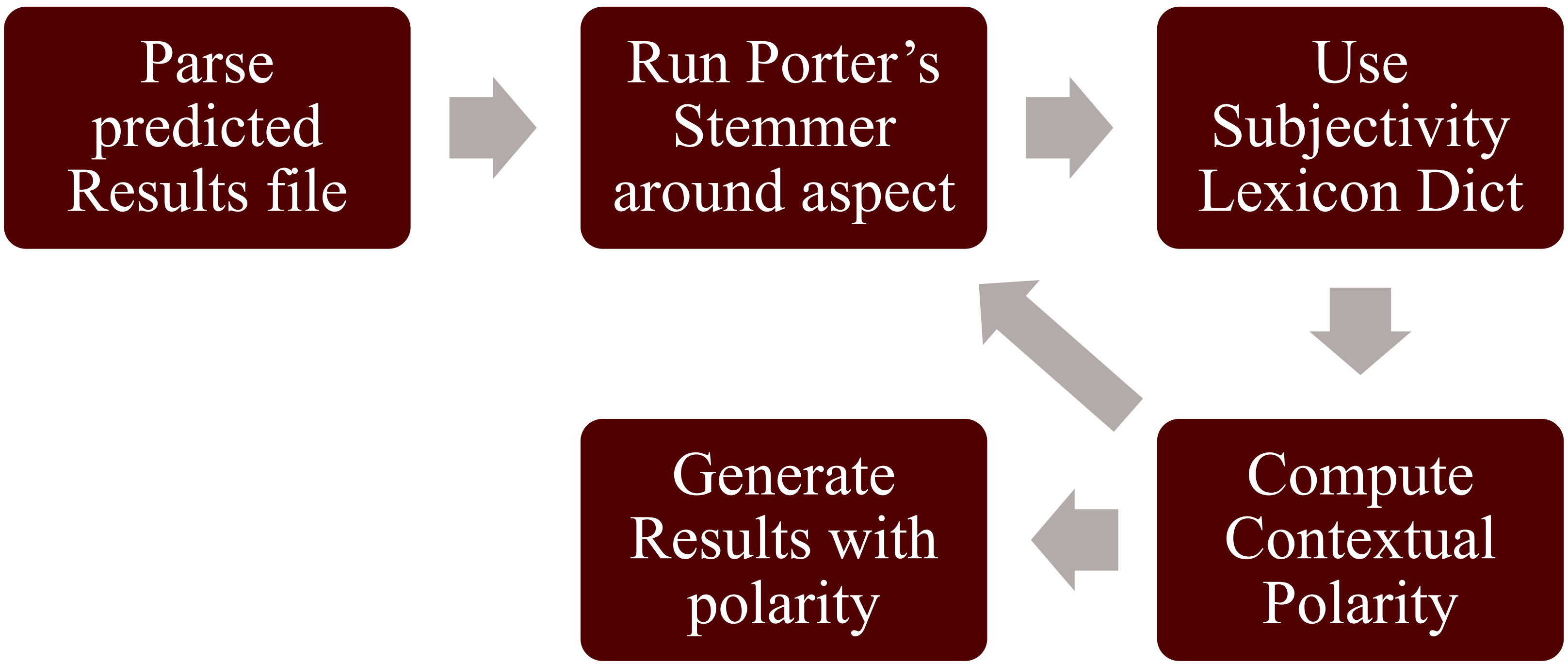
Sample Train.data(Conll) file

Word	POS	Chunk	Is-Aspect
But	CC	B-O	False
the	DT	B-NP	False
staff	NN	I-NP	True
was	VBD	B-VP	False
so	RB	B-ADJP	False
horrible	JJ	I-ADJP	False

Aspect Extraction Testing Phase



Polarity Computation of the Predicted Aspects



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

TEXAS A&M UNIVERSITY

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

- Obtained the TripAdvisor JSON data from http://times.cs.uiuc.edu/~wang296/Data/index.html
- 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

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

Naive Bayes Results (3-Class)

Accuracy : 0.812883

Confusion Matrix :

```
[[172  61  17]
 [ 49 197   4]
 [  8   9 233]]
```

Classification Report :

	precision	recall	f1-score	support
avg	0.75	0.69	0.72	250
neg	0.74	0.79	0.76	250
pos	0.92	0.93	0.92	250
avg / total	0.80	0.80	0.80	750

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 :

	precision	recall	f1-score	support
average	0.62	0.61	0.61	250
awesome	0.80	0.75	0.77	250
fair	0.51	0.64	0.57	250
good	0.62	0.62	0.62	250
poor	0.77	0.63	0.69	250
avg / total	0.66	0.65	0.65	1250

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 : 0.769333333333

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

Classification Report :

	precision	recall	f1-score	support
pos	0.94	0.80	0.87	500
avg	0.65	0.72	0.68	500
neg	0.76	0.79	0.77	500
avg / total	0.78	0.77	0.77	1500

NBSVM results (5-Class)

Accuracy : 0.6396

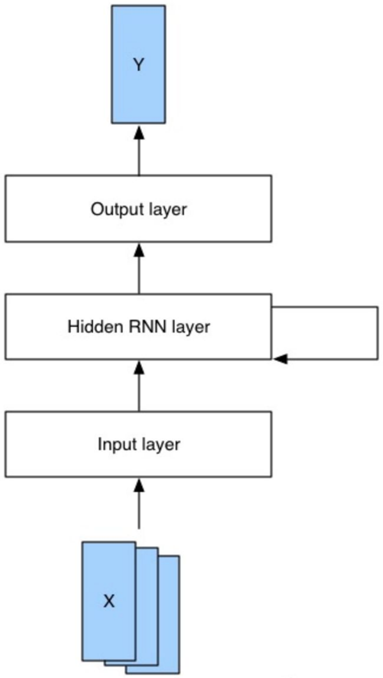
```
Confusion Matrix :
[[342 141  14   3   0]
 [ 95 287  97  16   5]
 [ 15 216 228  35   6]
 [  6  13 135 282  64]
 [  2   2   2  34 460]]
```

Classification Report :

	precision	recall	f1-score	support
1	0.74	0.68	0.71	500
2	0.44	0.57	0.50	500
3	0.48	0.46	0.47	500
4	0.76	0.56	0.65	500
5	0.86	0.92	0.89	500
avg / total	0.66	0.64	0.64	2500

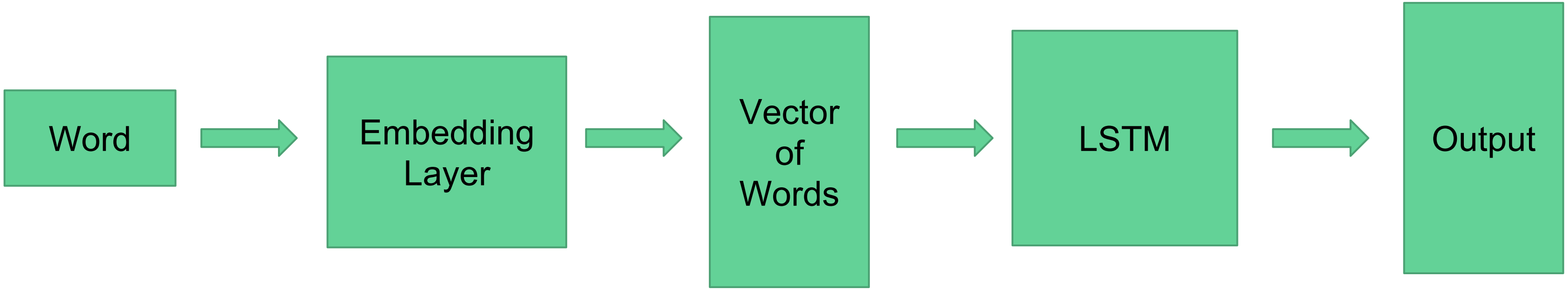
RNN

- 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

Classification Report

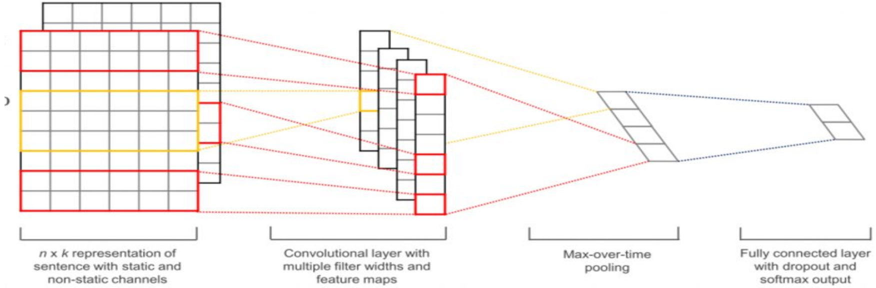
	precision	recall	f1-score	support
0	0.56	0.46	0.50	4541
1	0.00	0.00	0.00	4043
2	0.76	0.96	0.85	19089
avg / total	0.62	0.74	0.67	27673

5-class Results

	precision	recall	f1-score	support
1.0	0.48	0.44	0.45	2198
2.0	0.47	0.02	0.04	2343
3.0	0.35	0.19	0.24	4043
4.0	0.42	0.45	0.44	9016
5.0	0.53	0.72	0.61	10073

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
             precision    recall  f1-score   support

          0       0.64      0.44      0.52      4541
          1       0.90      0.95      0.92     23132

avg / total       0.85      0.87      0.86     27673
```

Evaluation

	2-class		3-class		5-class	
	Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
Naive Bayes	-	-	81.28%	80%	65.32%	65%
NBSVM	-	-	77%	77%	64%	64%
RNN	-	-	74%	67%	48%	44%
CNN	86.8%	86%	-	-	-	-

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

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,"""\xa0HAHAHAHAH, 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

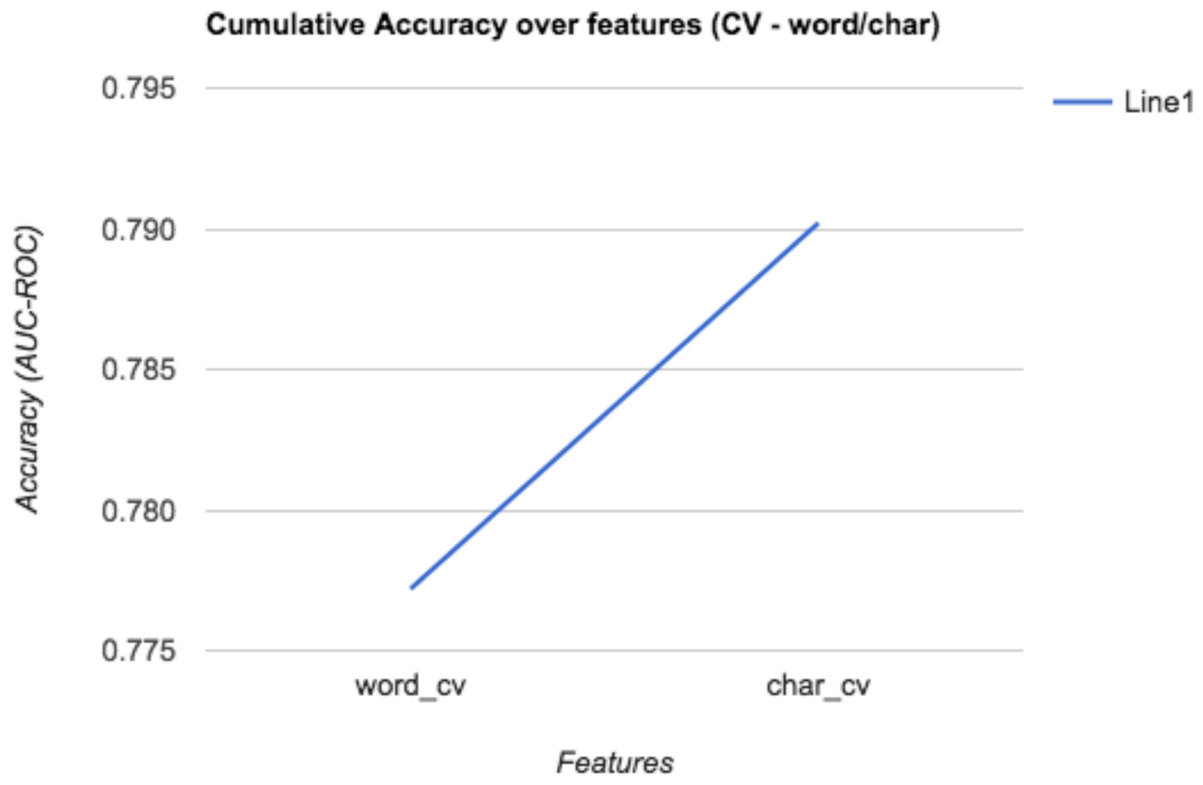
Micro

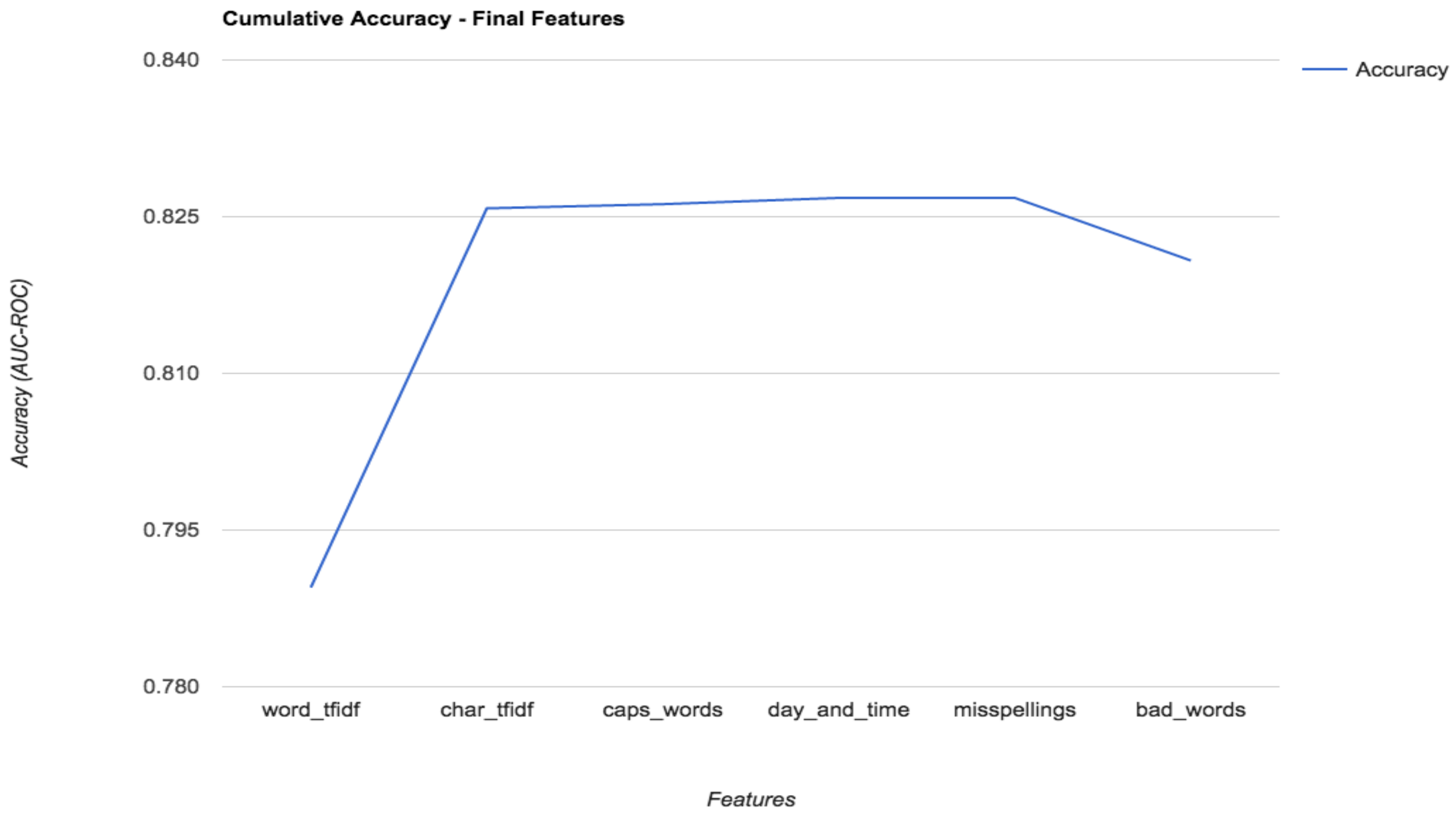
- Precision 0.743
- Recall 0.743
- F-score 0.743

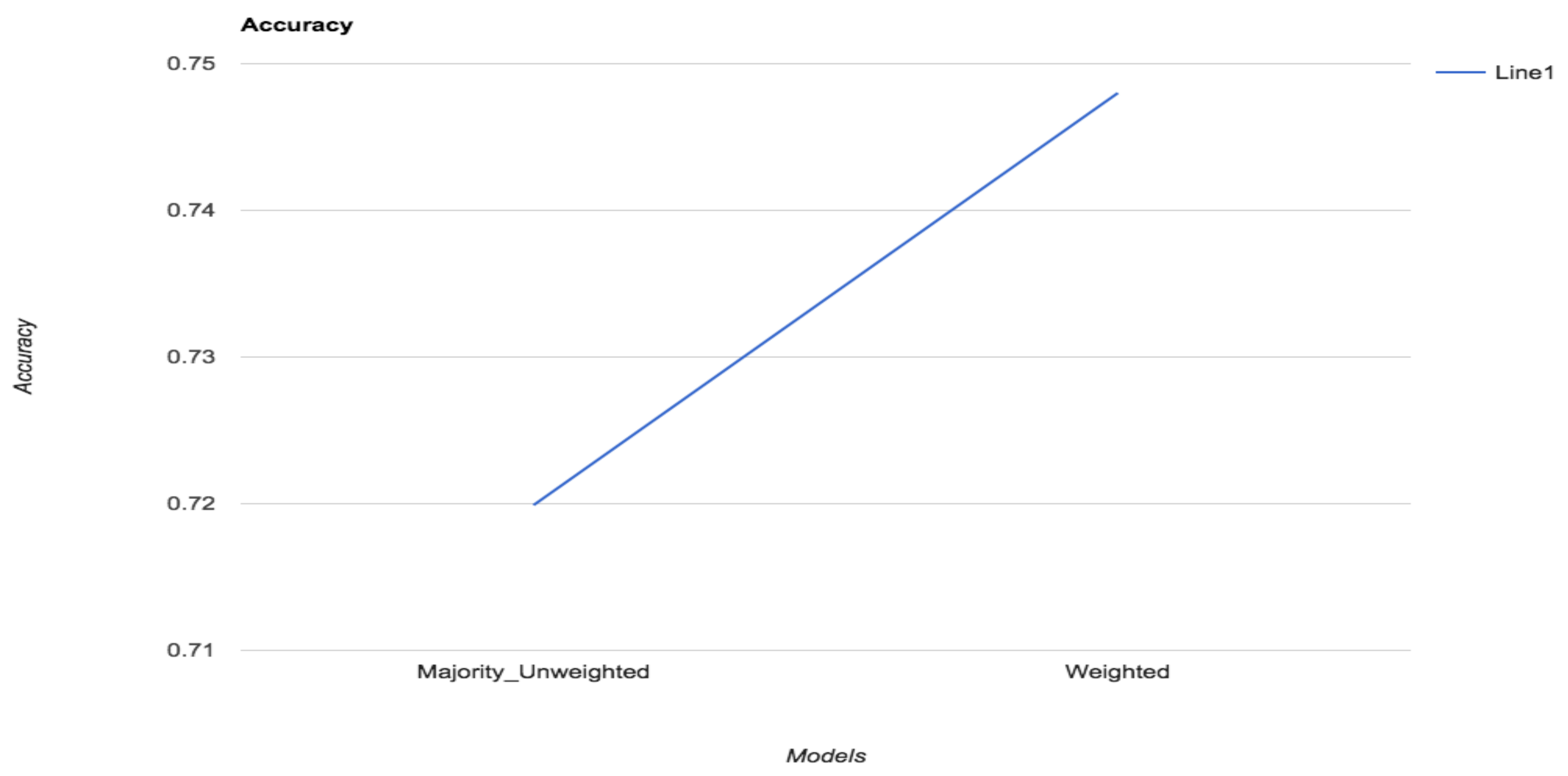
Class wise

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

Results Graphs




Cumulative Accuracy - Final Features
Accuracy
0.840
0.825
0.810
0.795
0.780
Accuracy (AUC-ROC)
word_tfidf
char_tfidf
caps_words
day_and_time
misspellings
bad_words
Features


Accuracy
0.75
0.74
0.73
0.72
0.71
Accuracy
Majority_Unweighted
Weighted
Models
Line1

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

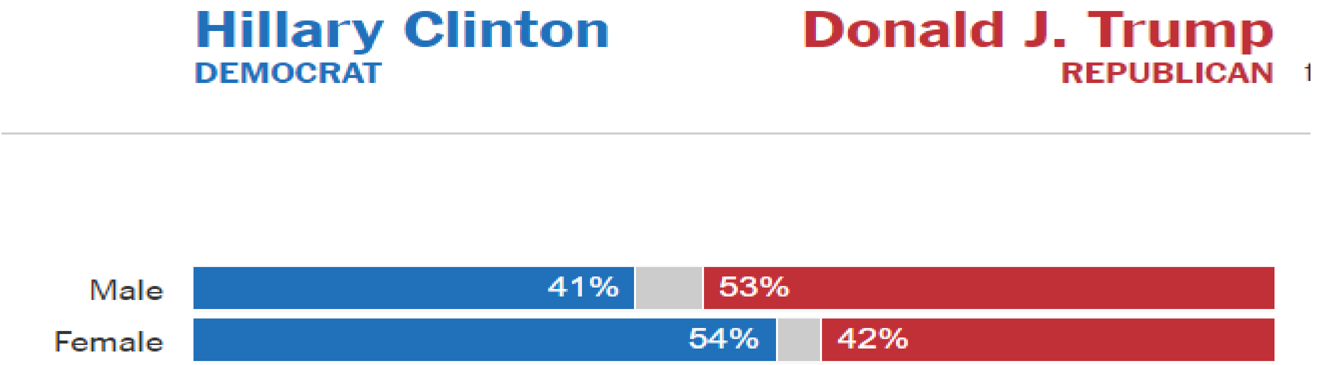
- Abusive Language Detection in Online User Content, Chikashi Nobata et al., WWW’16 Proceedings of the 25th International Conference.
- Data - https://www.kaggle.com/c/detecting-insults-in-social-commentary
- Article - https://www.overleaf.com/articles/detecting-insults-in-social-commentary/gkvrrwryjxhr/viewer.pdf
- Code - https://github.com/navgupta14/abuse-detector

Analysis in Twitter Gender Classification

Chuong Trinh

Motivation

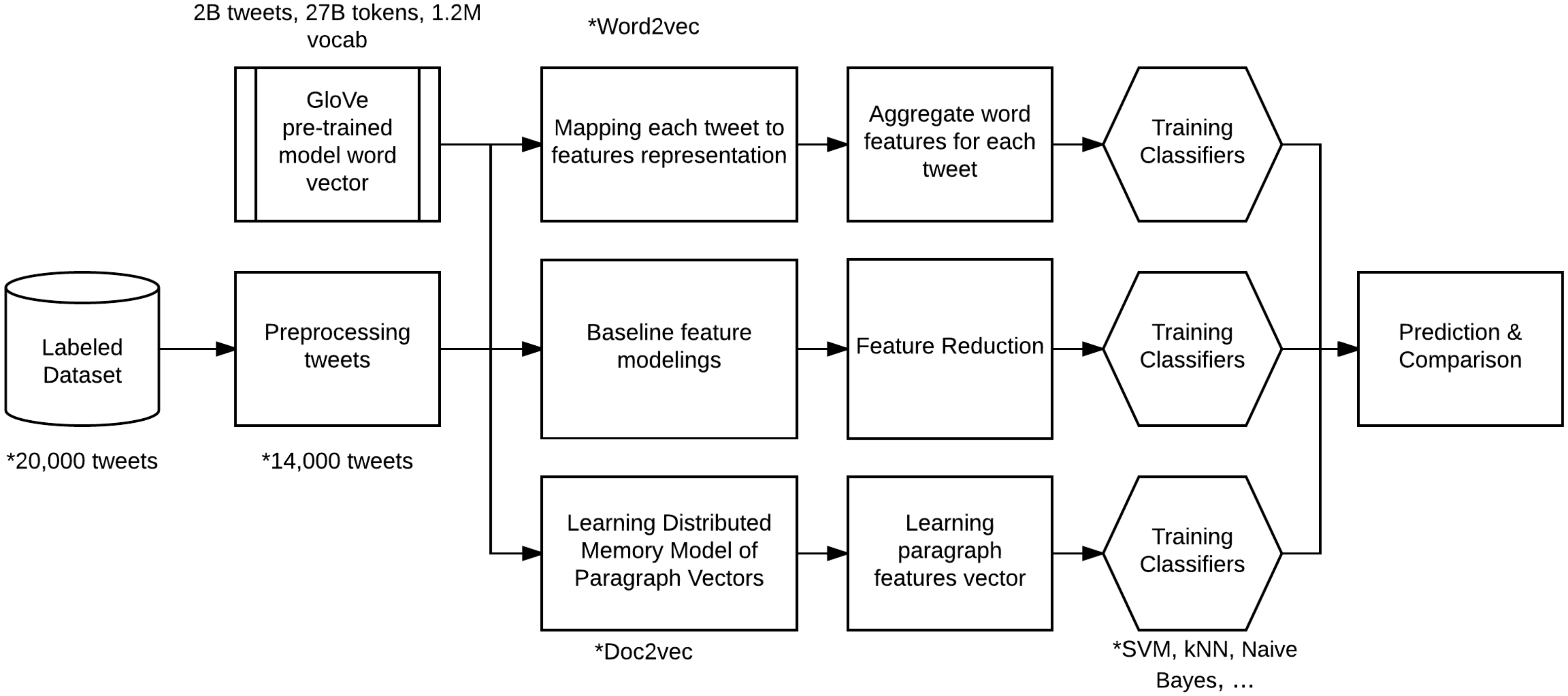
- Growing interest in automatically predicting the gender of authors from texts:
  - Opinions, political stances, styles, and preferences may be unique to each gender
  - Useful to individuals, companies, and governments for personal recommendation, customization, targeted advertising, political analysis,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



Why Gender Classification from Tweets is Hard!

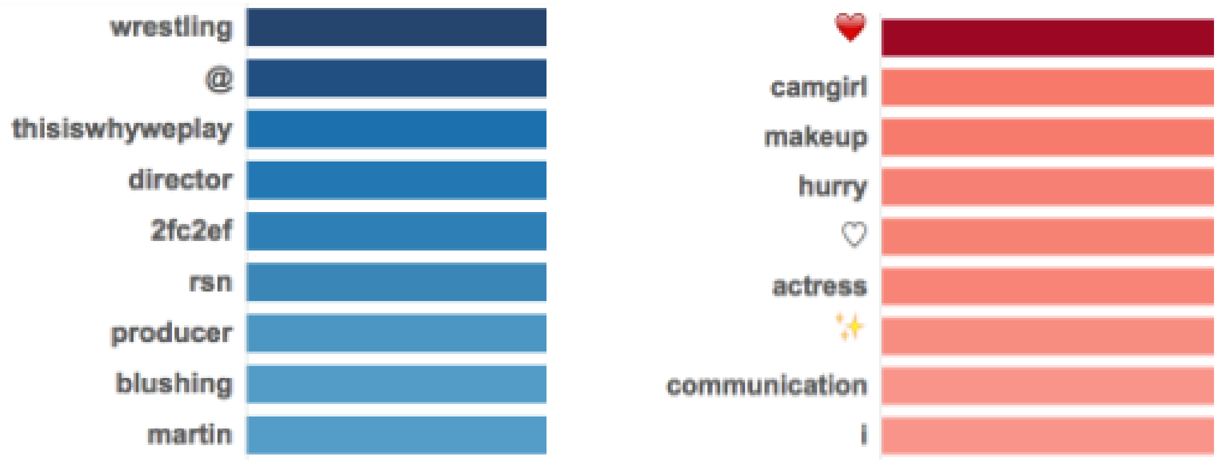
- Limited characters (140) per tweet
- Lots of spamming, advertising accounts, media sources, bots, etc.
- User’s profile privacy
- Users construct their identity through interacting with other users! (Marwick and boyd, 2011) – all depend on the context
- For example
  - Tweet 1: I’m walking on sunshine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

Pipeline


2B tweets, 27B tokens, 1.2M vocab
GloVe pre-trained model word vector
*Word2vec
Mapping each tweet to features representation
Aggregate word features for each tweet
Training Classifiers
Labeled Dataset
*20,000 tweets
Preprocessing tweets
*14,000 tweets
Baseline feature modelings
Feature Reduction
Training Classifiers
Prediction & Comparison
Learning Distributed Memory Model of Paragraph Vectors
*Doc2vec
Learning paragraph features vector
Training Classifiers
*SVM, kNN, Naive Bayes, …

Dataset & Baseline

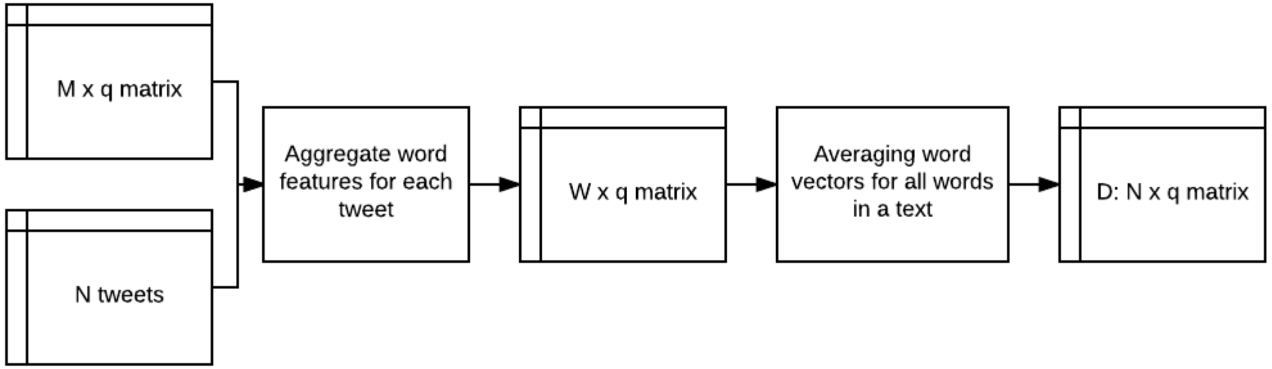
- CrowdFlower (kaggle – data challenge site)
  - 20,000 tweets – collected in 2015
    - Human Amazon Turker labeling + CrowdFlower's labeling system
    - ~ 14,000 tweets can be used (non-English, low confidence, or unreadable is ignored)
    - Labels: male + female + brand



- Men are more likely to talk at another account
- Women are more likely to use emoji
- Current accuracy: ~60%

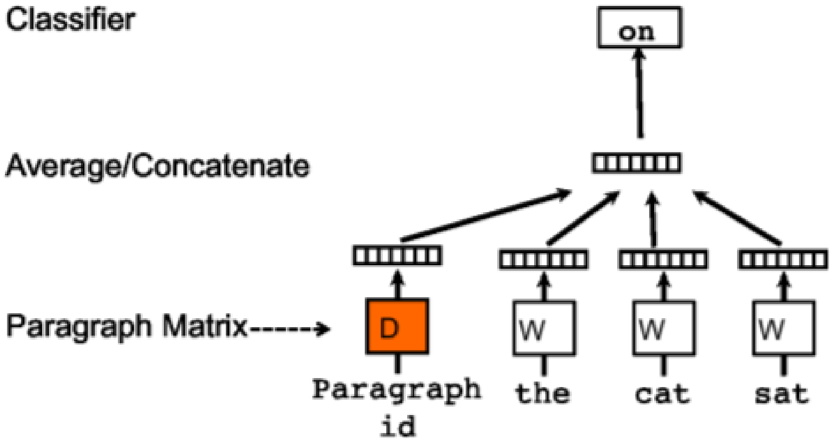
GloVe: Global Vectors for Word Representation

- 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.



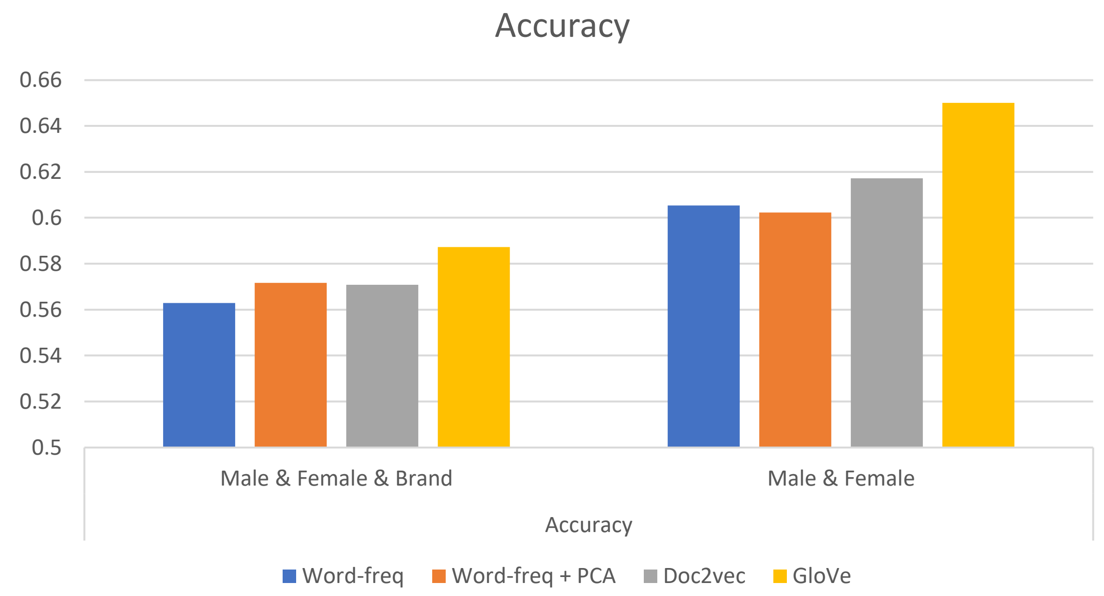
D: N x p matrix paragraph vector (each paragraph is mapped to p-dimensional features vector)

W: M x q matrix word vector (each word is mapped to q-dimensional features vector)

Q. Le, T. Mikolov. 2014. Distributed Representations of Sentences and Documents. In Proceedings of ICML 2014

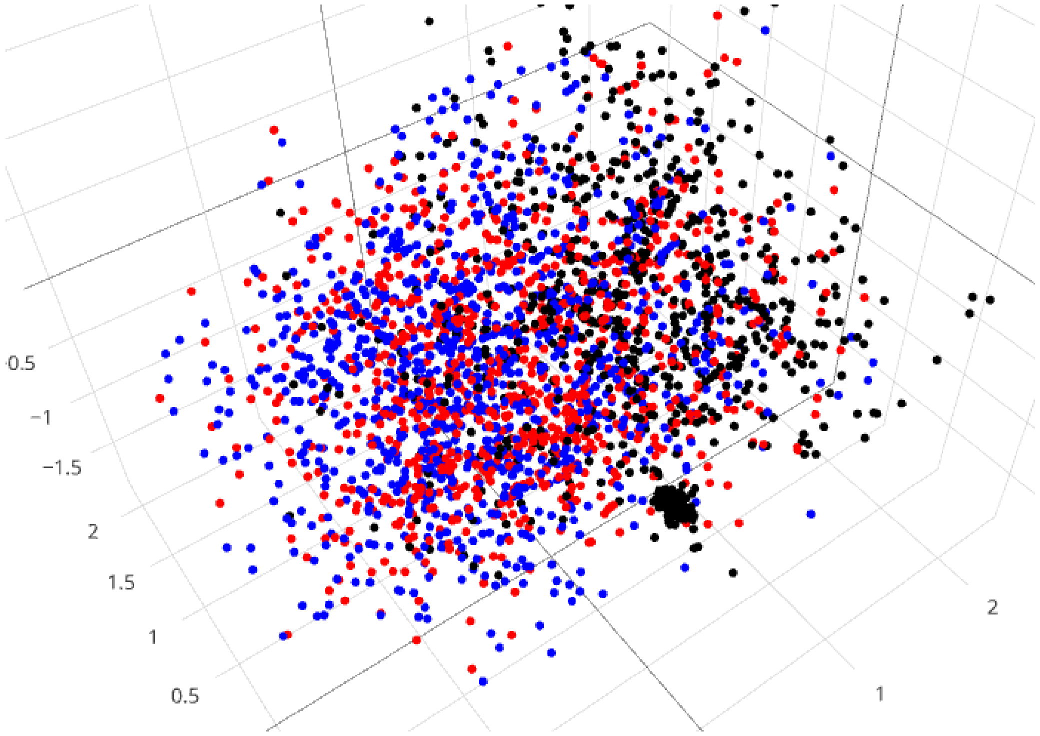
Analysis & Evaluation

		Word-freq	Word-freq + PCA	Doc2vec	GloVe
Accuracy	Male & Female & Brand	0.5629	0.5716	0.5708	0.5872
	Male & Female	0.6054	0.6023	0.6172	0.6500



Analysis & Evaluation

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



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>

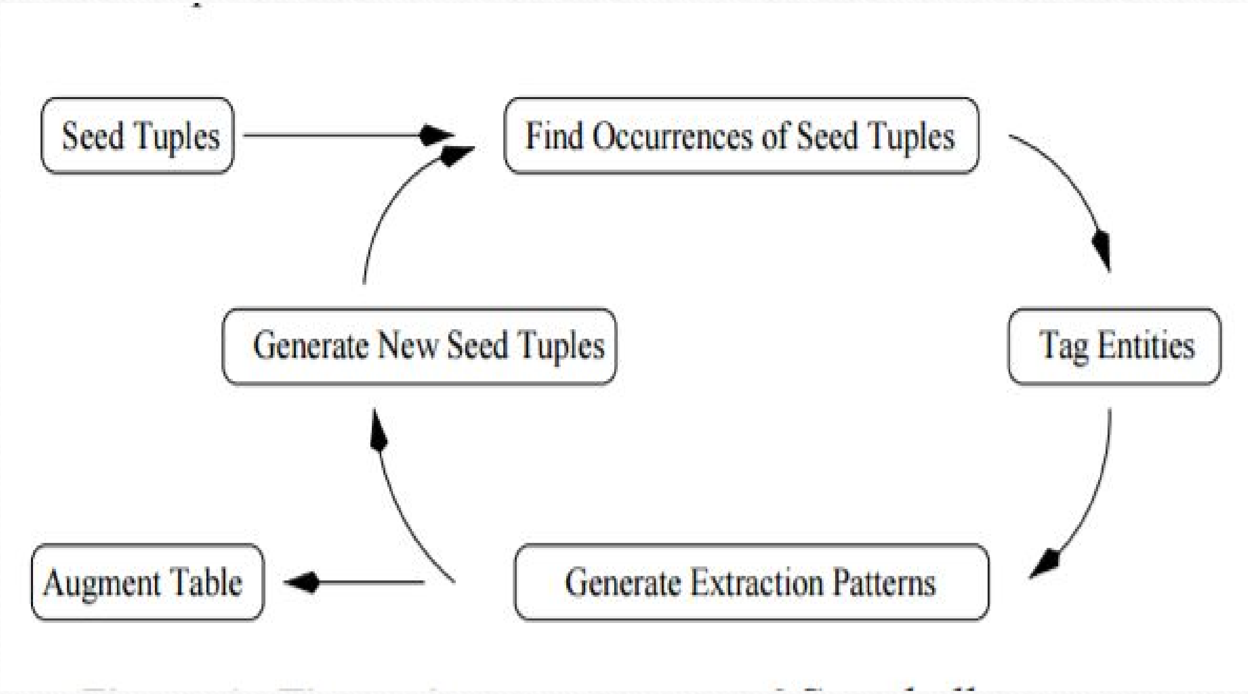
Snowball is a bootstrapped relation extraction method.

Seeds + Data = Relations!

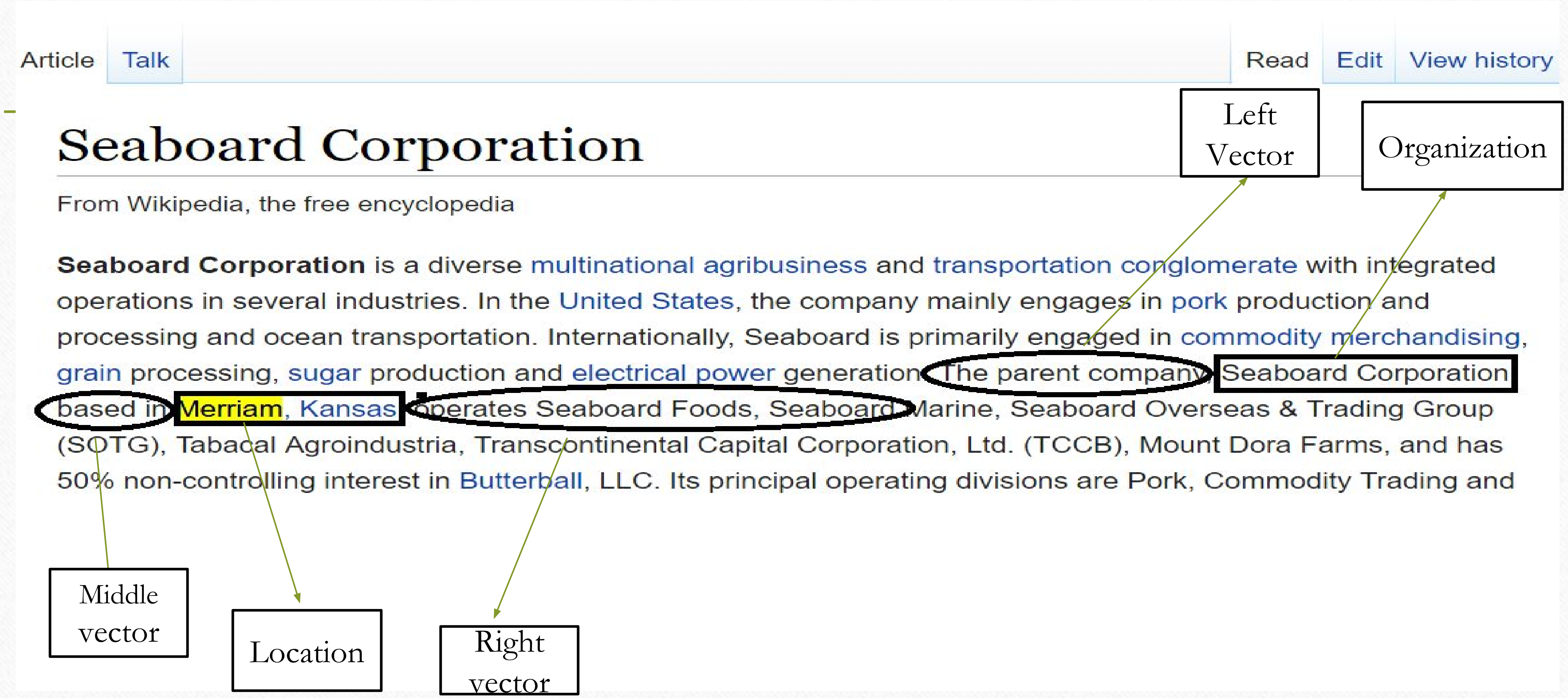
Snowball Algorithm: Terminology

- 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



Snowball Matches



Article Talk Read Edit View history

Seaboard Corporation

From Wikipedia, the free encyclopedia

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** (/ˈniːsɑːn/ or UK /ˈnɪsæn/; Japanese: [nisːaɴ]), 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

- E. Agichtein and L. Gravano. Snowball: extracting relations from large plain-text collections. In ICDL, 2000
- 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……!