# TEXT CLASSIFICATION

Feng Gao, Feb. 26th, 2015

# Outline

- Definition
- Text Representation
- Feature selection
- Text Classification with Naive Bayes
- The Voted Perceptron
- Examples

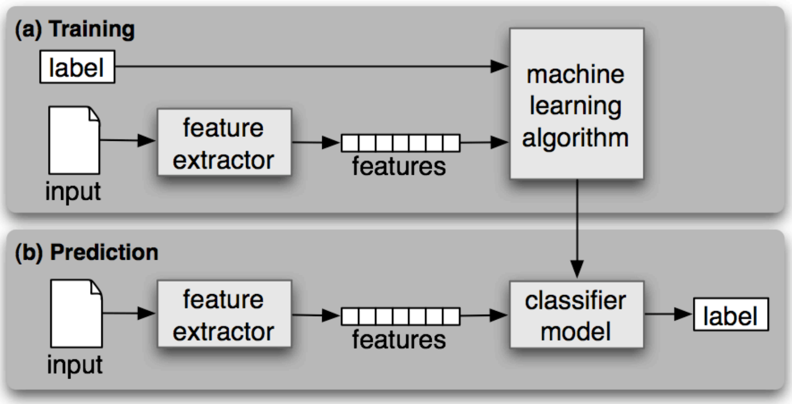
# What is classification?

- Classification or categorization is the task of assigning objects from a universe to two or more classes or categories.

- Classification is the task of choosing the correct class label for a given input.

# What is supervised Classification?

- A classifier is called supervised if it is built based on training corpora containing the correct label for each input.



# What is text classification?

- The classifier:
  - Input: a document x
  - Output: a predicted class y from some fixed set of labels $y_1, \ldots y_k$
- The learner:
  - Input: a set of m hand-labeled documents $(x_1, y_1), \ldots, (x_m, y_m)$
  - Output: a learned classifier f: $x \rightarrow y$

# Application

- Personal email sorting
- Automatic detection of *spam* pages
- Automatic detection of sexually explicit content
- Automatic classification of a review as positive or negative
- Topic-specific or *vertical* search

# Representation of text

# Text Representation

f(

> ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
> BUENOS AIRES, Feb 26
> Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:
> - Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
> - Maize Mar 48.0, total 48.0 (nil).
> - Sorghum nil (nil)
> - Oilseed export registrations were:
> - Sunflowerseed total 15.0 (7.9)
> - Soybean May 20.0, total 20.0 (nil)
>
> The board also detailed export registrations for subproducts, as follows....

)=y

?

What is the ~~**best**~~ **simplest useful** representation for the document *x* being classified?

# Text Representation

- Document is represented as a vector of attribute values
- Attributes:

"Bag of words" method: Use a set of words as attributes

# Text Representation

- Attribute values:

Method 1:

use 0 or 1 as attribute value

Method 2:

use the absolute or relative frequency of each word

Method 3:

use TF-IDF weight as the attribute value

# Method 1

Training data sets:

- Method 1:

| | **$word_1$** | **$word_2$** | **…** | **$word_m$** | **Class** |
|---|---|---|---|---|---|
| **$document_1$** | 0 | 1 | … | 1 | C1 |
| **$document_2$** | 1 | 0 | … | 1 | C2 |
| **…** | … | … | … | … | … |
| **$document_n$** | 1 | 0 | | 0 | C2 |

# Method 2

Training data sets:

- Method 2 with absoluate frequency:

| | $word_1$ | $word_2$ | ... | $word_m$ | Class |
|---|---|---|---|---|---|
| $document_1$ | 0 | 3 | … | 1 | C1 |
| $document_2$ | 2 | 0 | … | 3 | C2 |
| ... | … | … | … | … | … |
| $document_n$ | 5 | 0 | | 0 | C2 |

# Method 3

- TF: term frequency
  - Definition: TF = *tij*
  - frequency of term *i* in document *j*
  - Purpose: makes the frequent words *for the document more* important
- IDF: inverted document frequency
  - Definition: IDF = log(*N*/*ni*)
  - *ni : number of documents containing term i*
  - *N : total number of documents*
- TF-IDF value of a term *i in document j*
  - Definition: TF×IDF = *tij* * log(*N*/*ni*)

# Text Processing

- Word (token) extraction
- Stop words removal
- Stemming
- Feature Selection

# Text Processing

f(

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

)=y

f(

(argentine, 1986, 1987, grain, oilseed, registrations, buenos, aires, feb, 26, argentine, grain, board, figures, show, crop, registrations, ~~of,~~ grains, oilseeds, ~~and, their,~~ products, ~~to,~~ february, 11, ~~in,~~ …

)=y

**Common refinements: remove stopwords, stemming, collapsing multiple occurrences of words into one….**

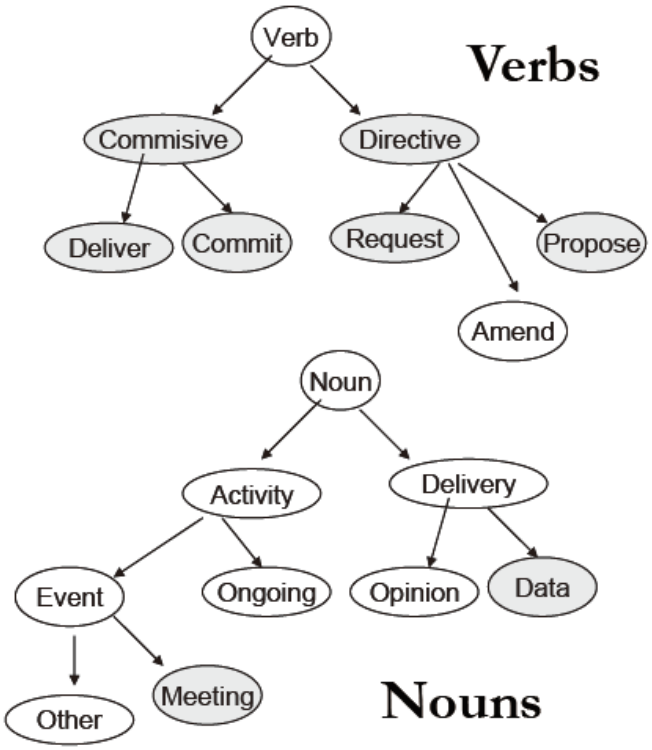
# Word (token) extraction

- Extract all the words in a document
- Convert them into lower cases

# Classifying Email into Acts

- From EMNLP-04, Learning to Classify Email into Speech Acts, Cohen-Carvalho-Mitchell
- An Act is described as a verb-noun pair (e.g., propose meeting, request information) - Not all pairs make sense. One single email message may contain multiple acts.
- Try to describe commonly observed behaviors, rather than all possible speech acts in English. Also include non-linguistic usage of email (e.g. delivery of files)

# Classifying Email into Acts



# Classifying Email into Acts

| Symbol | Pattern |
|---|---|
| [number] | any sequence of numbers |
| [hour] | [number]:[number] |
| [wwhh] | "why, where, who, what, or when" |
| [day] | the strings "Monday, Tuesday, ..., or Sunday" |
| [day] | the strings "Mon, Tue, Wed, ..., or Sun" |
| [pm] | the strings "P.M., PM, A.M. or AM" |
| [me] | the pronouns "me, her, him, us or them" |
| [person] | the pronouns "I, we, you, he, she or they" |
| [aaafter] | the strings "after, before or during" |
| [filetype] | the strings ".doc, .pdf, .ppt, .txt, or .xls" |

Table 1: Some PreProcessing Substitution Patterns

| 1-gram |
|---|
| ? |
| please |
| [wwhh] |
| could |
| do |
| can |
| of |
| [me] |

| 3-gram |
|---|
| [person] need to |
| [wwhh] do [person] |
| let [me] know |
| would [person] |
| do [person] think |
| are [person] meeting |
| could [person] please |
| do [person] need |

| 5-gram |
|---|
| [wwhh] do [person] think ? |
| let [me] know [wwhh] [person] |
| a call [number]-[number] |
| give [me] a call [number] |
| please give give [me] a call |
| [person] would be able to |
| take a look at it |
| [person] think [person] need to |

# Word (token) extraction for Email

| Request | Commit | Meeting |
|---|---|---|
| [wwhh] do [person] think | is good for [me] | [day] at [hour] [pm] |
| do [person] need to | is fine with [me] | on [day] at [hour] |
| and let [me] know | i will see [person] | [person] can meet at |
| call [number]-[number] | i think i can | [person] meet at [hour] |
| would be able to | i will put the | will be in the |
| [person] think [person] need | i will try to | is good for [me] |
| let [me] know [wwhh] | i will be there | to meet at [hour] |
| do [person] think ? | will look for [person] | at [hour] in the |
| [person] need to get | $[number] per person | [person] will see [person] |
| ? [person] need to | am done with the | meet at [hour] in |
| a copy of our | at [hour] i will | [number] at [hour] [pm] |
| do [person] have any | [day] is fine with | to go over the |
| [person] get a chance | each of us will | [person] will be in |
| [me] know [wwhh] | i will bring copies | let's plan to meet |
| that would be great | i will do the | meet at [hour] [pm] |

| dData | Propose | Deliver |
|---|---|---|
| – forwarded message begins | [person] would like to | forwarded message begins here |
| forwarded message begins here | would like to meet | [number] [number] [number] [number] |
| is in my public | please let [me] know | is good for [me] |
| in my public directory | to meet with [person] | if [person] have any |
| [person] have placed the | [person] meet at [hour] | if fine with me |
| please take a look | would [person] like to | in my public directory |
| [day] [hour] [number] [number] | [person] can meet tomorrow | [person] will try to |
| [number] [day] [number] [hour] | an hour or so | is in my public |
| [date] [day] [number] [day] | meet at [hour] in | will be able to |
| in our game directory | like to get together | just wanted to let |
| in the etc directory | [hour] [pm] in the | [pm] in the lobby |
| the file name is | [after] [hour] or [after] | [person] will be able |
| is in our game | [person] will be available | please take a look |
| fyi – forwarded message | think [person] can meet | can meet in the |
| just put the file | was hoping [person] could | [day] at [hour] is |
| my public directory under | do [person] want to | in the commons at |

# Stop words removal

- The most frequently used words in English
- Examples of stop words
  - the, of, and, to, a, …
- Typically about 400 to 500 such words
- Additional domain-specific stop words
- Stop words are usually removed

# Stemming

- find the root/stem of a word
- Reduce the number of words
- Improve effectiveness of text classification
- For example:
  - discussed
  - discusses
  - discussing
  - Discuss
  - Stem: discuss

# Example Stemming Rules

- **Remove ending**
  - If a word ends with *s*, preceded by a consonant other than an *s*, then delete the *s*.
- **Transform words**
  - If a word ends with “ies” but not “eies” or “aies”, then “ies” is replaced with “y”.

selection
regression
modeling
data-driven
variable
feature
variables
noise
complexity
data
problem
multicollinearity
real-world
robustness
importance
insight
scalable
non-linear
high-dimensional
features
focus
benefits
problems
Data

# Feature Selection

- Selecting the “bag of words” to represent documents

- Why do we need to select?
  - Leaning program may not be able to handle all possible features
  - Good features can result in higher accuracy

# Feature Selection Methods

- Class independent methods (Unsupervised)
  - Document Frequency (DF)
  - Term Strength (TS)
- Class-dependent methods (Supervised)
  - Information Gain (IG)
  - Mutual Information (MI)
  - $\chi^2$ statistic (CHI)

# Document Frequency (DF)

- Document frequency of a word
  - DF (w) = number of documents containing w
- Advantages
  - Can remove rare words (hence noise)
  - Easy to compute
- Disadvantages
  - Class independent
  - Some infrequent terms can be good discriminators, which cannot be selected by this method.

# Information Gain

- A measure of importance of the feature
- The number of "bits of information" gained by knowing the word is present or absent

$$\mathrm{Gain}(\omega) = -\sum_{i=1}^{k} P(C_i)\log P(C_i) + P(\omega)\sum_{i=1}^{k} P(C_i|\omega)\log P(C_i|\omega) + P(\overline{\omega})\sum_{i=1}^{k} P(C_i|\overline{\omega})\log P(C_i|\overline{\omega})$$

- Rank the words according to their information gain value
- Select the first *m words with high gain values*

# Information Gain

- Advantage
  - Consider the classes
- Disadvantage
  - computationally expensive
- Remove rare words (appears 1 or 2 times)
  - reduce the amount of computation, and
  - remove noisy words that have by-chance correlations with the classes.

# What Do People Do In Practice?

- Infrequent term removal
  - infrequent across the whole collection (i.e. DF)
  - met in a single document
- Most frequent term removal (i.e. removing stop words)
- Stemming. (*often*)
- Use a class-dependent method (e.g., the information gain method) to select features.

# Naive Bayes

$$\boxed{P(A|B)} = \boxed{P(B|A)}\, P(A) \,/\, P(B)$$

$$\boxed{P(B|A)} = \boxed{P(A|B)}\, P(B) \,/\, P(A)$$

Prior probability

Likelihood

Posterior probability

# Text Classification with Naive Bayes

- Represent document x as list of words w1,w2,…
- For each y, build a probabilistic model Pr(X|Y=y) of "documents" in class y
- To classify, find the y which was most likely to generate x—i.e., which gives x the best score according to Pr(x|y)

$$f(x) = \mathrm{argmax}_y \Pr(x|y) * \Pr(y)$$

# Text Classification with Naive Bayes

- How to estimate Pr(X | Y) ?
- Simplest useful process to generate a bag of words:
  - pick word 1 according to Pr(W | Y)
  - repeat for word 2, 3, ....
  - each word is generated independently of the others (which is clearly not true) but means

$$\Pr(w_1,\ldots,w_n \mid Y = y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y = y)}_{}$$

**How to estimate Pr(W|Y)?**

# Text Classification with Naive Bayes

- How to estimate Pr(X | Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1} \underbrace{\Pr(w_i \mid Y=y)}$$

Estimate *Pr(w|y)* by looking at the data...

$$\Pr(W=w \mid Y=y) = \frac{\text{count}(W=w \text{ and } Y=y)}{\text{count}(Y=y)}$$

- This gives score of zero if x contains a brand-new word $w_{new}$

# Text Classification with Naive Bayes

- How to estimate Pr(X | Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}$$

... and also **imagine** *m* examples with *Pr(w|y)=p*

$$\Pr(W=w \mid Y=y) = \frac{\text{count}(W=w \text{ and } Y=y)+mp}{\text{count}(Y=y)+m}$$

- This Pr(W | Y) is a *multinomial distribution*
- This use of *m and p is a Dirichlet prior for the multinomial*

# Text Classification with Naive Bayes

- Putting this together:
- for each document $x_i$ with label $y_i$

for each word $w_{ij}$ in $x_i$

- count[$w_{ij}$][$y_i$]++
- count[$y_i$]++
- count++

# Text Classification with Naive Bayes

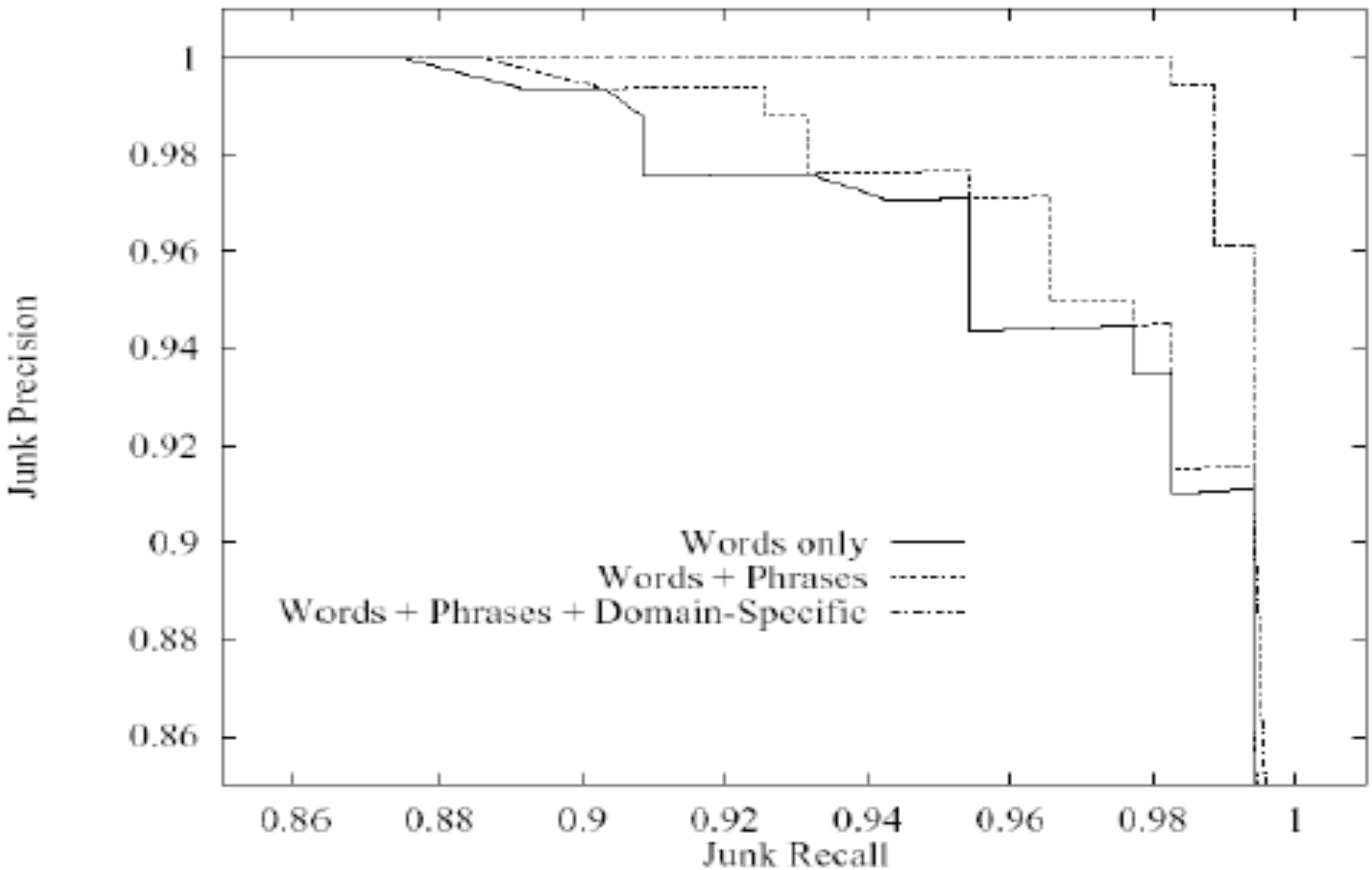
- to classify a new $x = w_1 \ldots w_n$, pick y with top score:

$$score(y, w_1 \ldots w_k) = \lg \frac{\text{count}[y]}{\text{count}} + \sum_{i=1}^{n} \lg \frac{\text{count}[w_i][y] + 0.5}{\text{count}[y] + 1}$$

**key point:** we only need counts for words that actually appear in *x*

# Naïve Bayes for SPAM filtering

- Sahami et al, 1998
- Used bag of words, + special phrases ( "FREE!" ) and + special features ( "from *.edu" , ...)



Junk Precision
Junk Recall
1
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.86
0.88
0.9
0.92
0.94
0.96
0.98
1
Words only
Words + Phrases
Words + Phrases + Domain-Specific

# Naïve Bayes for SPAM filtering

| | Classified Junk | Classified Legitimate | Total |
|---|---|---|---|
| Actually Junk | 36 (92.0% precision) | 9 | 45 |
| Actually Legitimate | 3 | 174 (95.0% precision) | 177 |
| Total | 39 | 183 | 222 |

# Naive Bayes Summary

- Pros:
  - Very fast and easy-to-implement
  - Well-understood formally & experimentally

- Cons:
  - Seldom gives the very best performance
  - “Probabilities” $\Pr(y|x)$ are not accurate

# The Voted Perceptron

**Training**

Input: a labeled training set $\langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\rangle$
number of epochs $T$

Output: a list of weighted perceptrons $\langle(\mathbf{v}_1, c_1), \ldots, (\mathbf{v}_k, c_k)\rangle$

- Initialize: $k := 0$, $\mathbf{v}_1 := \mathbf{0}$, $c_1 := 0$.
- Repeat $T$ times:
  - For $i = 1, \ldots, m$:
    - Compute prediction: $\hat{y} := \text{sign}(\mathbf{v}_k \cdot \mathbf{x}_i)$
    - If $\hat{y} = y$ then $c_k := c_k + 1$.
      else $\mathbf{v}_{k+1} := \mathbf{v}_k + y_i \mathbf{x}_i$;
      $c_{k+1} := 1$;
      $k := k + 1$.

# The Voted Perceptron

**Prediction**

Given: the list of weighted perceptrons: $\langle(\mathbf{v}_1, c_1), \ldots, (\mathbf{v}_k, c_k)\rangle$

an unlabeled instance: $\mathbf{x}$

compute a predicted label $\hat{y}$ as follows:

$$s = \sum_{i=1}^{k} c_i \operatorname{sign}(\mathbf{v}_i \cdot \mathbf{x}); \qquad \hat{y} = \operatorname{sign}(s) \ .$$

# Classifying Reviews as Favorable or Not

- Turney, ACL 2002
- Dataset: 410 reviews from Epinions
  - Autos, Banks, Movies, Travel Destinations

- Learning method:
  - Extract 2-word phrases containing an adverb or adjective (eg “unpredictable plot”)
  - Classify reviews based on average Semantic Orientation

# Classifying Reviews as Favorable or Not

$$\mathrm{SO}(\mathit{phrase}) = \mathrm{PMI}(\mathit{phrase}, \text{“excellent”}) - \mathrm{PMI}(\mathit{phrase}, \text{“poor”})$$

$$\mathrm{PMI}(\mathit{word}_1, \mathit{word}_2) = \log_2\left[\frac{\mathrm{p}(\mathit{word}_1 \ \& \ \mathit{word}_2)}{\mathrm{p}(\mathit{word}_1)\ \mathrm{p}(\mathit{word}_2)}\right]$$

**Computed using queries to web search engine**

# Classifying Reviews as Favorable or Not

| Extracted Phrase | Part-of-Speech Tags | Semantic Orientation |
|---|---|---|
| online experience | JJ NN | 2.253 |
| low fees | JJ NNS | 0.333 |
| local branch | JJ NN | 0.421 |
| small part | JJ NN | 0.053 |
| online service | JJ NN | 2.780 |
| printable version | JJ NN | -0.705 |
| direct deposit | JJ NN | 1.288 |
| well other | RB JJ | 0.237 |
| inconveniently located | RB VBN | -1.541 |
| other bank | JJ NN | -0.850 |
| true service | JJ NN | -0.732 |
| Average Semantic Orientation | | 0.322 |

# Classifying Reviews as Favorable or Not

Table 5. The accuracy of the classification and the correlation of the semantic orientation with the star rating.

| Domain of Review | Accuracy | Correlation |
|---|---|---|
| Automobiles | 84.00 % | 0.4618 |
| Honda Accord | 83.78 % | 0.2721 |
| Volkswagen Jetta | 84.21 % | 0.6299 |
| Banks | 80.00 % | 0.6167 |
| Bank of America | 78.33 % | 0.6423 |
| Washington Mutual | 81.67 % | 0.5896 |
| Movies | 65.83 % | 0.3608 |
| The Matrix | 66.67 % | 0.3811 |
| Pearl Harbor | 65.00 % | 0.2907 |
| Travel Destinations | 70.53 % | 0.4155 |
| Cancun | 64.41 % | 0.4194 |
| Puerto Vallarta | 80.56 % | 0.1447 |
| All | 74.39 % | 0.5174 |

# Classifying Reviews as Favorable or Not

Table 5. The accuracy of the classification and the correlation of the semantic orientation with the star rating.

| Domain of Review | Accuracy | Correlation |
|---|---|---|
| Automobiles | 84.00 % | 0.4618 |
| Honda Accord | 83.78 % | 0.2721 |
| Volkswagen Jetta | 84.21 % | 0.6299 |
| Banks | 80.00 % | 0.6167 |
| Bank of America | 78.33 % | 0.6423 |
| Washington Mutual | 81.67 % | 0.5896 |
| Movies | 65.83 % | 0.3608 |
| The Matrix | 66.67 % | 0.3811 |
| Pearl Harbor | 65.00 % | 0.2907 |
| Travel Destinations | 70.53 % | 0.4155 |
| Cancun | 64.41 % | 0.4194 |
| Puerto Vallarta | 80.56 % | 0.1447 |
| All | 74.39 % | 0.5174 |

# Classifying Reviews as Favorable or Not

- Pang et al, EMNLP 2002
- 700 movie reviews (ie all in same domain); Naïve Bayes, MaxEnt, and linear SVMs; accuracy with different representations x for a document
- Interestingly, the off-the-shelf methods work well… perhaps better than Turney's method.

# Classifying Movie Reviews

| | Features | # of features | frequency or presence? | NB | ME | SVM |
|---|---|---|---|---|---|---|
| (1) | unigrams | 16165 | freq. | **78.7** | N/A | 72.8 |
| (2) | unigrams | " | pres. | 81.0 | 80.4 | **82.9** |
| (3) | unigrams+bigrams | 32330 | pres. | 80.6 | 80.8 | **82.7** |
| (4) | bigrams | 16165 | pres. | 77.3 | **77.4** | 77.1 |
| (5) | unigrams+POS | 16695 | pres. | 81.5 | 80.4 | **81.9** |
| (6) | adjectives | 2633 | pres. | 77.0 | **77.7** | 75.1 |
| (7) | top 2633 unigrams | 2633 | pres. | 80.3 | 81.0 | **81.4** |
| (8) | unigrams+position | 22430 | pres. | 81.0 | 80.1 | **81.6** |

# Classifying Movie Reviews

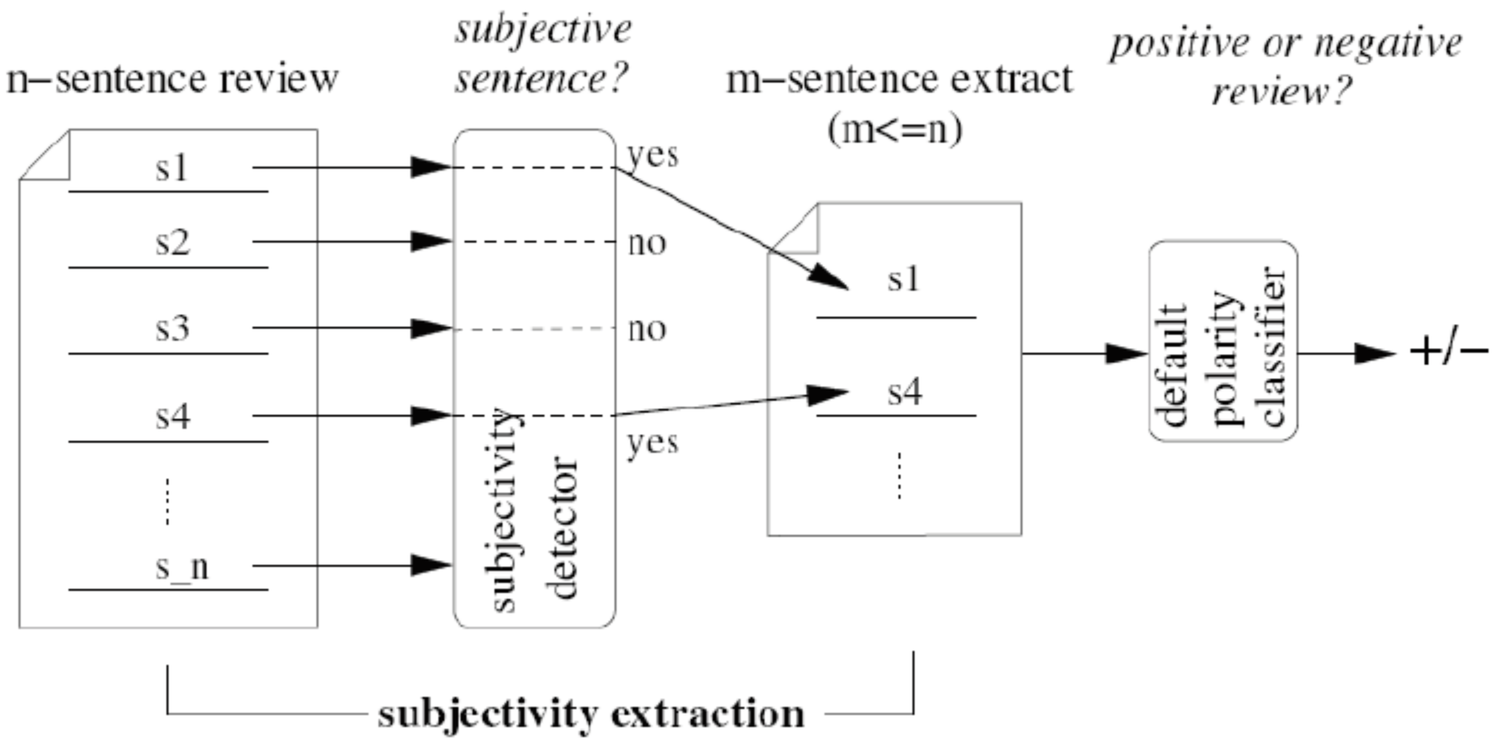
- Assume the classifier is same form as Naïve Bayes, which can be written:

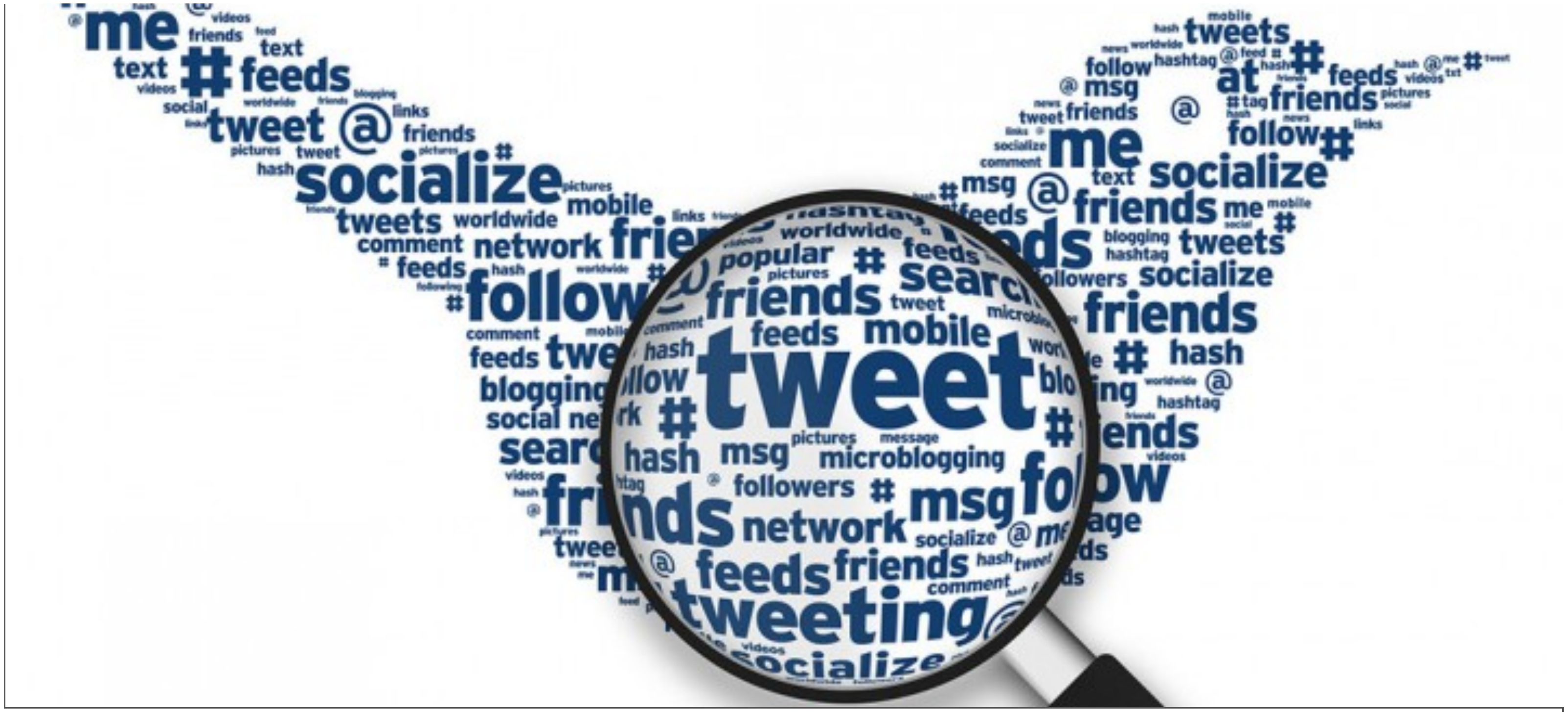
$$\Pr(y \mid w_1, w_2, \ldots, w_N) = \frac{1}{Z}\prod_i \lambda_i f(y, w_i)$$

- Set weights ( $\lambda$ 's) to maximize probability of the training data:

$$\prod_{(x_j, y_j) \in D} \Pr(y_j \mid x_j) + \underbrace{\Pr(\lambda \mid Q)}_{\text{prior on parameters}}$$

# MaxEnt classification





me
text
feeds
tweet
socialize
tweets
network
follow
feeds
blogging
social
friends
tweets
socialize
friends
hash
popular
worldwide
search
tweet
mobile
hash
msg
microblogging
followers
network
feeds
friends
tweeting
socialize


## Thank you!