Basic Sentiment Analysis

Peter Dolog
dolog@cs.aau.dk
http://people.cs.aau.dk/~dolog

Based on Christopher Potts' tutorial from the Sentiment Analysis Symposium, San Francisco, Nov. 8-9, 2011 and some slides from Bo Thiesson

IMDB reviews

when _star wars_ came out some twenty years ago , the image of traveling throughout the stars has become a commonplace image . [...]

when han solo goes light speed , the stars change to bright lines , going towards the viewer in lines that converge at an invisible point .

cool .

october sky offers a much simpler image–that of a single white dot , traveling horizontally across the night sky .   [. . .]

" snake eyes " is the most aggravating kind of movie : the kind that shows so much potential then becomes unbelievably disappointing .

it's not just because this is a brian depalma film , and since he's a great director and one who's films are always greeted with at least some fanfare .

and it's not even because this was a film starring nicolas cage and since he gives a brauvara performance , this film is hardly worth his talents .

Why is sentiment analysis important?

- Sentiment (opinions, attitudes, emotions, perspective, etc) from others are key influencers of our beliefs and behaviors
- When we need to make a decision we often seek out advice from others
- <u>In the past</u>:
  - Individuals: from friends, family (and experts)
  - Organizations: from user surveys, focus groups, opinion polls, consultants
- <u>Now:</u>
  - user-generated opinion content on the Internet has risen exponentially
    - Reviews, blogs, discussions, news, comments, feedback, …
  - ⇒ nearly all our decision-making is social
  - before buying products (attending events, trying services, visiting specialists. …), we see what our peers are saying about them.
  - fate of a offerings often sealed by those evaluations

Examples: User generated reviews

"Centralt og pænt rent."

Anmeldt 3 dage siden

NY

Dejligt hotel med central beliggenhed.Super hyggeligt nyt og rent.Ligger i gåafstand til indre by og nemt at parkere. Pæne værelser og dejligt roligt selvom det ligger midt i byen. Gode priser og gode tilbud på weekend ophold.

Var denne anmeldelse nyttig? Ja

TripAdvisor

5 of 5 people found the following review helpful

Halo is still great January 27, 2014

By Peter Anderson

Verified Purchase

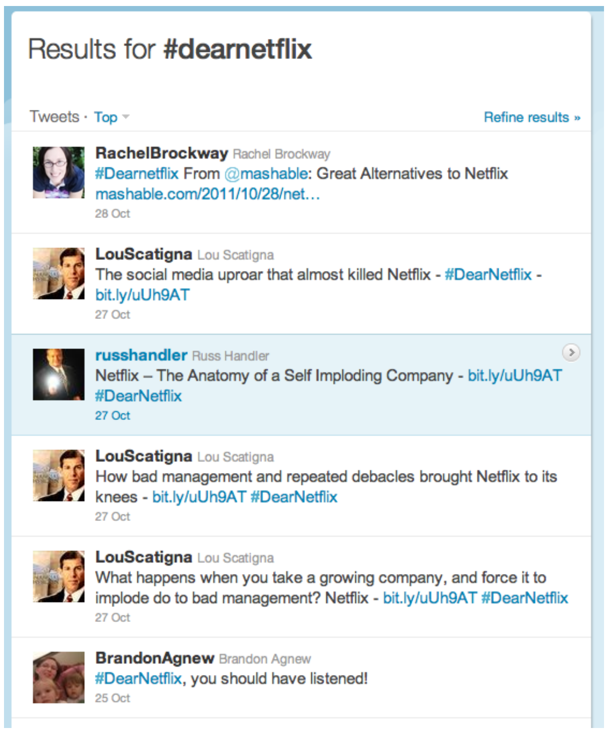
This is definitely the best looking Halo game ever made. Game play is great, Halo has always been and still is the gold standard when it comes to FPS gameplay mechanics. The only complaints and reasons that this is not 5 stars is some issues with the multiplayer. While still very fun to play online, they took Halo 4 the way of COD in that you have to level up to unlock new weapons to use. This makes it so that it becomes more about who has played the most hours instead of who is most skilled. Also, I have a lot of connectivity issues when playing Big Team Battles...this has been an issue since Halo 3 and Halo Reach and it gets annoying at times.

Comment | **Was this review helpful to you?** Yes No

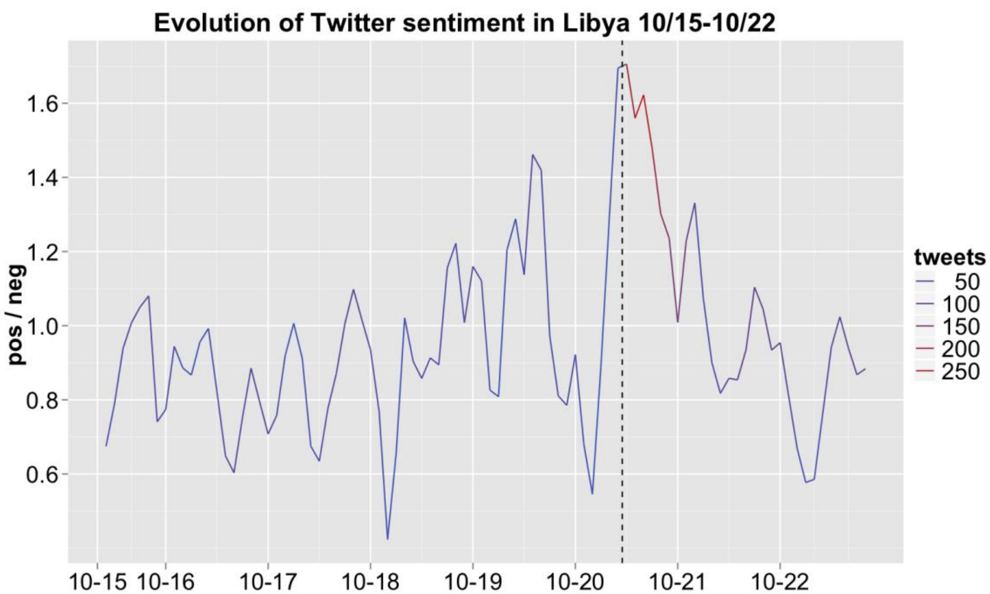
Amazon

Example: Twitter discussion and effect on stock

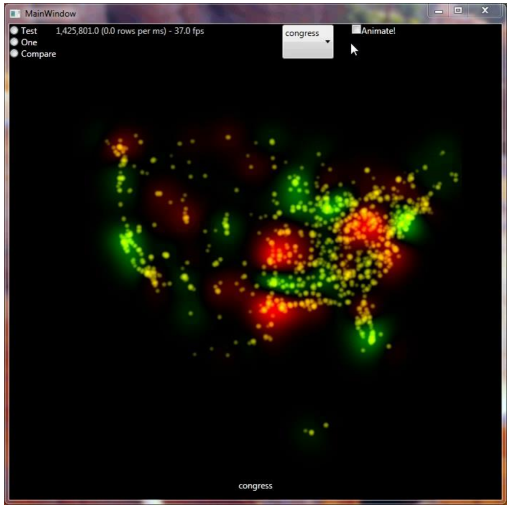
Netflix' announced that it would be increasing its subscription prices!



Example: Political sentiment from tweets



Location based sentiment towards US Congress



Baseline Algorithm

- Tokenization
- (Stemming - usually not good for sentiment analysis)
  - Sometimes destroys the Positive/Negative distinction
- Feature Extraction
- Classification using different classifiers
  - Naïve Bayes
  - MaxEnt
  - …

Tokenization

Sentiment Tokenization

Issues:

- Mark-ups (e.g. # in Twitter, informative html tags strong,b,em,i)
- Capitalization
  - "SHOUTING" – so preserve
- Emoticons!!
- Masked curses (****, s***t) and swears ($#!@).
- Lengthening (I reeeeealy liked…)

Good tokenizer design especially important for sentiment analysis, because many of these issues carry significant sentiment clues

Potts’ emoticons

- Emoticons are extremely common in many forms of social media, and they are reliable carriers of sentiment
- The following regular expression captures 96% of the emoticon tokens occurring on Twitter
- Only captures just 36% of the emoticon types, but most are extremely rare and highly confusable with other chunks of text

<table>
<tr><td>[<>]?</td><td># optional hat/brow</td></tr>
<tr><td>[:;=8]</td><td># eyes</td></tr>
<tr><td>[\-o*\']?</td><td># optional nose</td></tr>
<tr><td>[\)\]\(\[dDpP/\:\}\{@\|\\]</td><td># mouth</td></tr>
<tr><td>|</td><td>#### reverse orientation</td></tr>
<tr><td>[\)\]\(\[dDpP/\:\}\{@\|\\]</td><td># mouth</td></tr>
<tr><td>[\-o*\']?</td><td># optional nose</td></tr>
<tr><td>[:;=8]</td><td># eyes</td></tr>
<tr><td>[<>]?</td><td># optional hat/brow</td></tr>
</table>

Punctuation

Basic strategy (Potts):

1. Identify all word-internal punctuations first and create tokens
   - (emoticons, Twitter and HTML markups, and masked curses)
2. Tokenize sequences of characters that are obvious words, numbers, '…'-variations
3. Remaining punctuation kept as separate tokens
   - E.g. progression from ! to !! is somewhat additive (until, e.g. !!!)
   - Becomes useful for later handling of negation

Potts's sentiment aware tokenizer

http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py

(Linguistic) Feature Extraction

Extracting Features: Issues

- Which words to use?
  - Only adjectives
  - Only from positive lists/dictionaries
  - All words (in many cases turns out to work better)
    - Should we combine with Part-Of-Speech (POS) tags?
      - Using <u>(word,tag)</u> pairs instead of just <u>words</u> as the features
- How do we handle negation?
  - I didn't like this product vs I really like this product

Handling of negation

Effect of negation depends on the negated word(s)

- Weak (mild) words behave like their opposites when negated:
  bad ≈ not good; good ≈ not bad.
- Strong (intense) words have very general meanings under negation:
  not superb ~ everything from horrible to just-shy-of-superb

Diverse negation expressions and influences can be far-reaching (syntactically speaking).

Examples for “neg-enjoy”:

- I didn't enjoy it.
- I never enjoy it.
- No one enjoys it.
- I have yet to enjoy it.
- I don't think I will enjoy it

Can be handled by

- semantic parsers (expensive)
- KISS principle (cheap and works well)

Handling of negation (KISS principle)

(Das and Chen 2001; Pang, Lee & Vaithyanathan 2002)

Append a _NEG suffix to every word appearing between a negation and a clause-level punctuation mark

Definition: Negation

A negation is any word matching the following regular expression:

```
(?:
    ^(?:never|no|nothing|nowhere|noone|none|not|
        havent|hasnt|hadnt|cant|couldnt|shouldnt|
        wont|wouldnt|dont|doesnt|didnt|isnt|arent|aint
    )$
)
|
n't
```

Definition: Clause-level punctuation

A clause-level punctuation mark is any word matching the following regular expression:

```
^[.:;!?]$
```

http://sentiment.christopherpotts.net/lingstruc.html

Handling of negation: Examples

No one enjoys it.

no
one_NEG
enjoys_NEG
it_NEG
.

I don't think I will enjoy it: it might be too spicy.

i
don't
think_NEG
i_NEG
will_NEG
enjoy_NEG
it_NEG
:
it
might
be
too
spicy
.

Classification

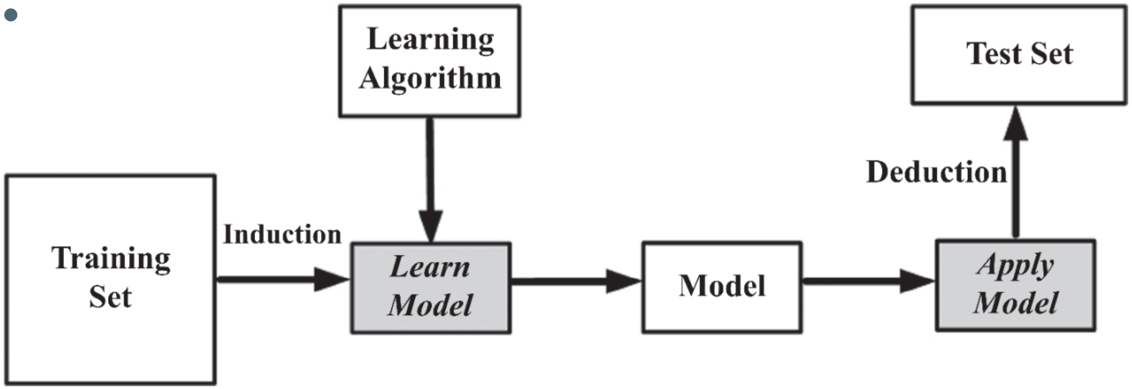
Machine Learning - The three basic tasks

- **Predictive tasks**: Predict the value for an (unobserved) target variable given observed values for input variables
  - Classification, if target is discrete (i.e. sentiment classes)
  - Regression, if target is continuous (i.e. sentiment scores)
  - Learn from labeled data -- both input (i.e. review) and target variables (i.e. sentiment) are observed
  - Predict the target for un-labeled data (only input variables are observed)

Today's focus

- **Descriptive task**:
  - Clustering, identify coherent clusters (subgroups) in the data.
  - Learn from un-labeled data (no target variable)
  - Explore, find structure, or compress information in the data

Classification: The process



- We are given a set of labeled examples in the format (x, c) where
  - x is a feature vector (from review), and
  - c is the class attribute (the known sentiment)
- The supervised learning task is to build a model that maps x to c (find a mapping m such that $m(x) = c$)
- Given an unlabeled instances $(x', ?)$, we compute $m(x')$
  - E.g., positive/negative review
- More than two classes is possible
  - E.g., positive / negative / neutral -or- 1 / 2 / 3 / 4 / 5 -or- (1,2) / 3 / (4,5)

Training data (how do we get the labeled data)

Implicit labeled class c = 4 (out of 5)

-or-

Explicit labeling (e.g., by mechanical turks)

Feature vector x (tokenization and linguistic feature construction)

Boolean "bag-of words"

Classification Algorithms

- Naïve Bayes
- Maximum Entropy Modeling
  - Aka. Logistic Regression
- (Linear) Discriminant Analysis (LDA)
- Support Vector Machines (SVM)
- Decision Trees
- …

widely used in natural language processing

"Funny" name for classification algorithm

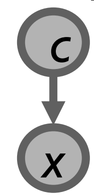
Features

- Supervised learning classifiers can use any sort of feature
  - URL, email address, punctuation, capitalization, dictionaries, network features, words, emoticons,…
- In the simplest (boolean) "bag of words" view of documents
  - We use **only** word features
  - We use **all** of the words in the text (not a subset)

Naïve Bayes Classifier

Bayes Theorem: (for two random variables/vectors C and X)

$$p(C|X) = \frac{p(X|C)p(C)}{p(X)}$$



class variable

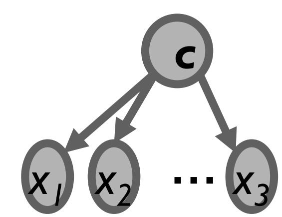
feature vector

The score for class $c \in C$ given review x is:

$$score(x,c) = p(c|x) = \frac{p(x|c)p(c)}{\cancel{p(x)}}$$

Same for all c !

We assume that features in the feature vector $X = (X_1, \dots, X_n)$ are independent given the class attribute ("bag-of-words" model)



$$score(x,c) \propto \left(\prod_{i=1}^{n} p(x_i|c)\right)p(c)$$

The "naïve" assumption

x_i is either the appearance or non-appearance of the i'th word from the vocabulary

Naïve Bayes Classifier – Prediction

Sometimes we just want the most likely sentiment for a given review x:

$$\hat{c} = \underset{c \in C}{\arg\max}\, score(x, c)$$

At other times we want the whole distribution (*tells us how close runner-ups are*)

$$\hat{p}(c|x) = \frac{score(x, c)}{\sum_{c \in C} score(x, c)}$$

NBC – Tricks of the trade

Trick 1 (computational efficiency):

- Most words from the vocabulary are **not** present in a given review (reviews are short)
- Compute the score for the "empty" review
  - $s^*(\text{empty}, c) = \left(\prod_{i=1}^{n} p(not\ x_i|c)\right)p(c)$
- Adjust for the words that actually appear. Say, $x_1, \ldots, x_k$ actually appear in the review x
  - $s(x, c) = s^*(\text{empty}, c)\left(\prod_{j=1}^{k} \frac{p(x_j|c)}{p(not\ x_j|c)}\right)$

NBC – Tricks of the trade

Trick 2 (numerical stability):

- Multiplying many probabilities will often result in numerical instability (very small numbers)
- Switch to log-space
  - $\log s(x,c) = \log P(c) + \sum_{i=1}^{n} \log p(x_i|c)$

Naïve Bayes Classifier – Learning the Model

…is simple counting

Prediction model:

$$score(x, c) \propto \left(\prod_{i=1}^{n} p(x_i|c)\right)p(c)$$

- Estimate $p(c)$ for all $c \in C$:
  - Count the number of reviews: N
  - Count the number of reviews with sentiment c: $N(c)$

$$p(c) = \frac{N(c)}{N}$$

- Estimate $p(x_i|c)$ for all possible words in vocabulary $x_i \in X$ and all possible sentiment classes $c \in C$
  - Count the number of times the word x_i appears across all reviews with sentiment c: $N(x_i, c)$
  - Count all possible words in the reviews with sentiment c: *W(c)*

$$p(x_i|c) = \frac{N(x_i, c)}{W(c)}$$

$$p(not\ x_i|c) = 1 - p(x_i|c)$$

Naïve Bayes Classifier – Learning the Model

Prediction model:

$$score(x, c) \propto (\prod_{i=1}^{n} p(x_i|c))p(c)$$

...with Laplace smoothing

- Estimate $p(c)$ for all $c \in C$:
  - Count the number of reviews: N
  - Count the number of reviews with sentiment c: $N(c)$

$$p(c) = \frac{N(c) + 1}{N + |C|}$$

Number of classes (2)

- Estimate $p(x_i|c)$ for all possible words in corpus $x_i \in X$ and all possible sentiment classes $c \in C$
  - Count the number of times the word x_i appears across all reviews with sentiment c: $N(x_i, c)$

$$p(x_i|c) = \frac{N(x_i, c) + 1}{W(c) + |X|}$$

Size of vocabulary

Naive Bayes is Not So Naive

- Very fast learning and testing (basically just count words)
- Low storage requirements
- Very good in domains with many equally important and (conditionally) independent features
- More robust to irrelevant features than many learning methods
  - Irrelevant features tend to cancel each other without affecting results

Naive Bayes is Not So Naive

- Naive Bayes won 1st and 2nd place in KDD-CUP 97 competition out of 16 systems
  - Goal: Financial services industry direct mail response prediction: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.
- A good dependable baseline for text classification (but not the best)!

Conditional Maximum Entropy (MaxEnt) Classifier

Goal: estimate p(c|x) directly, making the minimum assumptions about unseen data

With some math…

$$p(c|x,\lambda) = \frac{e^{\sum_i \lambda_i f_i(c,x)}}{\sum_{c \in C} e^{\sum_i \lambda_i f_i(c,x)}}$$

- $f_i(c, x)$ is 1 if the word (indexed by) i is in the review feature vector x and the class for the review is c; and 0 otherwise.
- λ_i is the weight for feature f_i

- (Note: non-apparent features do not affect the expression)

MaxEnt – Learning the Model

To find the parameters $\lambda_1, \lambda_2, \lambda_3, \ldots$

- Write out the conditional log-likelihood of the training data and maximize it

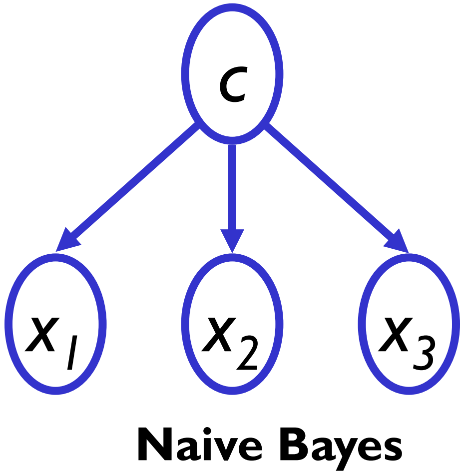
$$LogLikelihood = \sum_{n=1}^{N} \log p(c^n | x^n, \lambda)$$

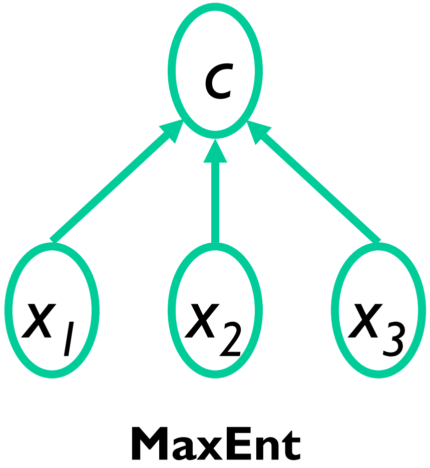
  The log-likelihood is concave and has a single maximum; use your favorite numerical optimization package.
  - E.g. L-BFGS, stochastic gradient ascent, Newton-Raphson,…

--or–

- Use one of many open source MaxEnt modeling packages

Naïve Bayes vs MaxEnt





Refined feature selection

Iterative feature selection process

Select base features
(e.g. boolean term-vector)

Learn classifier (on training data)

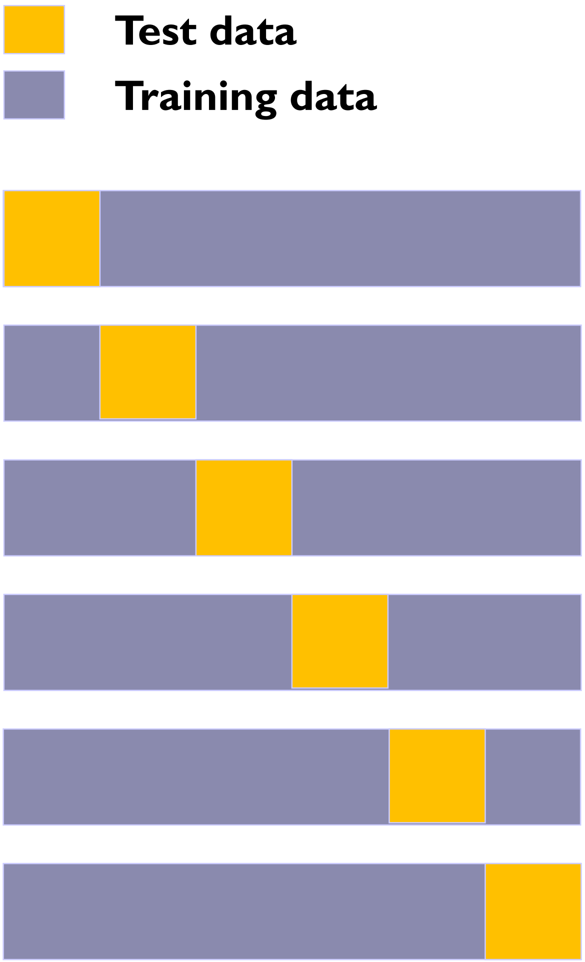
Test data (on validation data)

Investigate classification errors

Adjust feature set

Cross Validation

- Break up data into 10 folds
  - (Same pos/neg rate in each fold)
- For each fold
  - Choose the fold as a temporary test set
  - Train on 9 folds, compute performance on the test/validation set
- Report average performance of the 10 runs



Quality metrics – Categorical target

Consider a binary classification problem.

- Positive / Negative (Sentiment)
- Spam / Ham
- +1 / -1
- …

Algorithms predict either a

- Class of the example
  - E.g. Naïve Bayes or MaxEnt classifier assigns the dominant class of the node where the example falls; or
- Score of a class for the example
  - The higher the score, the greater is the probability/likelihood of the data example being positive.

We want to avoid…

false positive
(Type I Error)

false negative
(Type II Error)

Confusion matrix

	Predicted Positive	Predicted Negative
Labeled Positive	True Positives (TP)	False Negatives (FN)
Labeled Negative	False Positives (FP)	True Negatives (TN)

Confusion matrix: Accuracy & Error rate

	Predicted Positive	**Predicted Negative**
Labeled Positive	True Positives (TP = 400)	False Negatives (FN = 100)
Labeled Negative	False Positives (FP = 200)	True Negatives (TN = 800)

Accuracy = Fraction of data classified correctly

$$= \frac{TP+TN}{TP+FN+FP+TN} = \frac{400+800}{1500} = 0.80$$

Error rate = 1 – Accuracy

Confusion matrix: Precision & Recall

	Predicted Positive	Predicted Negative
Labeled Positive	True Positives (TP = 400)	False Negatives (FN = 100)
Labeled Negative	False Positives (FP = 200)	True Negatives (TN = 800)

Precision (of positive predictions)

$$=\frac{TP}{TP+FP}=\frac{400}{400+200}=0.67$$

Recall (of positive labels)

$$=\frac{TP}{TP+FN}=\frac{400}{400+100}=0.80$$

F-score $=2\frac{precision \cdot recall}{precision+recall}=2\frac{0.67 \cdot 0.80}{0.67+0.80}=0.73$

Summary

- Tokenization
- Linguistic Feature Extraction
- Classification using different classifiers
  - Naïve Bayes
  - MaxEnt
- Refined feature selection
- Evaluation measures

What makes reviews hard to classify?

Subtlety:

Perfume review in *Perfumes: the Guide*:

- "If you are reading this because it is your darling fragrance, please wear it at home exclusive, and tape the windows shut."

Thwarted expectations and ordering effect:

- "This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."

Irony and sarcasm:

Review of e-reader:

- "Great idea, now try again with a real product development team."

Review on book:

- "Love the cover"