POLITÉCNICA
"Ingeniamos el futuro"
CAMPUS
DE EXCELENCIA
INTERNACIONAL

Text Mining 4
Text Classification

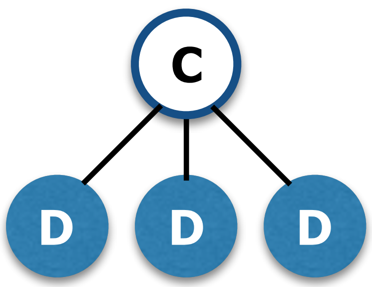
Madrid Summer School on
Advanced Statistics and Data Mining

Florian Leitner
Data Catalytics, S.L.
leitner@datacatytics.com

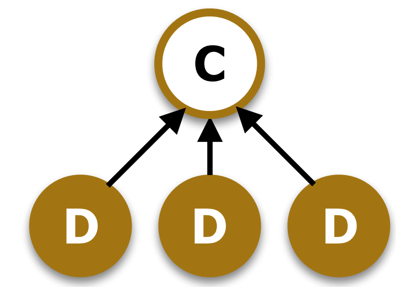
Incentive and applications

- Assign one or more “labels” to a collection of “texts”.

- Spam filtering
- Marketing and politics (**opinion mining**)
- Topic clustering
- …



Generative vs. discriminative models



- Generative models describe how the [hidden] labels "generated" the [observed] input as **joint probabilities**: $\mathrm{P}(class, data)$
  - They learn the distributions of each individual class.
  - Examples: Markov Chain, Naïve Bayes, Latent Dirichlet Allocation, Hidden Markov Model, …
  - Graphical models for detecting outliers or when there is a need to update models (change)
- Discriminative models predict ("discriminate") the [hidden] labels **conditioned** on the [observed] input: $\mathrm{P}(class \mid data)$
  - They ("only") learn the boundaries between classes.
  - Ex.: Logistic Regression, Support Vector Machine, Conditional Random Field, Random Forest, …
- Both can identify the most likely labels and their likelihoods
- **Only generative models**:
  - Most likely input value[s] and their likelihood[s]
  - Likelihood of input value[s] for some particular label[s]

$$P(H|D) = \frac{P(H) \times P(D|H)}{P(D)}$$

Maximum entropy (MaxEnt) intuition

The principle of maximum entropy

go	to	1/5	1/6	??
	up	1/5	1/6	??
	outside	1/5	1/6	??
	home	1/5	1/4	??
	out	1/5	1/4	??

$w \in W$

Observations:

$\sum_{w \in W} P(w \mid go) = 1$

$P(out \mid go) + P(home \mid go) = 0.5$

$P(to \mid go) + P(home \mid go) = 0.75$

Supervised MaxEnt classification

$$p(x) = \frac{1}{1 + exp(-(\lambda_0 + \lambda_1 x))}$$

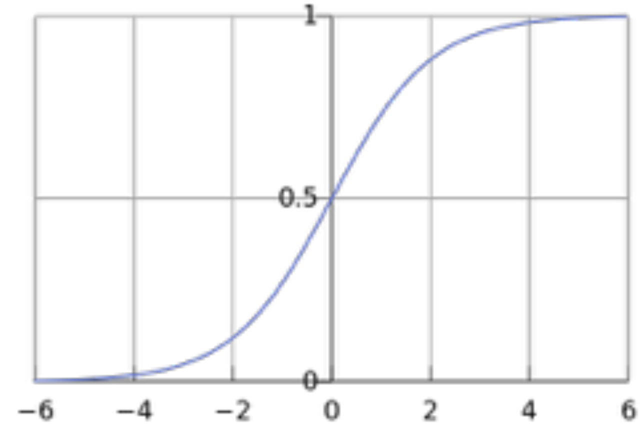
ln

$$ln \frac{p(x)}{1 - p(x)} = \lambda_0 + \lambda_1 x$$

e

odds-ratio!

$$\frac{p(x)}{1 - p(x)} = exp(\lambda_0 + \lambda_1 x)$$



logistic function p

Image Source: WikiMedia Commons, Qef

- a.k.a. multinomial logistic regression
- **Does not assume independence between the features**
- Can model **mixtures of** binary, discrete, and real **features**
- Training data are **per-feature-label probabilities**: P(F, L)
  - I.e., count(f_i, l_i) ÷ $\sum^{N}_{i=1}$ count(f_i, l_i)
  - ➡ words → very sparse training data (zero or few examples)
- Model parameters are commonly learned using gradient descent
  - Expensive if compared to Naïve Bayes, but efficient optimizers exist (**L-BFGS**)

Example feature functions for MaxEnt classifiers

- Examples of indicator functions (a.k.a. **feature functions**)
  - Assume we wish to classify the general polarity (positive, negative) of product reviews:
- $f(c, w) := \{c = \text{POSITIVE} \wedge w = \text{“}\textit{great}\text{”}\}$
  - Equally, for classifying words in a text, say to detect proper names, we could create a feature:
- $f(c, w) := \{c = \text{NAME} \wedge \text{isCapitalized}(w)\}$
- Note that while we can have multiple classes, we cannot require more than one class in the whole match condition of a single indicator (feature) function.

NB: typical text mining models can have a million or more features: unigrams + bigrams + trigrams + counts + dictionary matches + …

Maximizing the conditional entropy

- The **conditioned** (on X) version of Shannon's **entropy** H:

NB: the chain rule

$P(x,y) = P(x)\ P(y|x)$

$$H(Y|X) = -\sum_{x \in X} P(x)\ H(Y|X=x)$$

(swapped nom/denom

$$= -\sum_{x \in X} P(x) \sum_{y \in Y} P(y|x)\ log_2\ P(y|x)$$

to remove the minus)

$$= \sum_{x,y \in X,Y} P(x,y)\ log_2 \frac{P(x)}{P(x,y)}$$

- MaxEnt **training** then is about selecting the model p^* that maximizes H:

$$p^* = \underset{p \in P}{argmax}\ H(P) = \underset{p \in P}{argmax}\ H(Y|X)$$

Maximum entropy (MaxEnt 1/2)

- Some definitions:
  - The observed probability of y (the class) with x (the words) is:

$$\hat{P}(x,y) = count(x,y) \div N$$

  - An **indicator function** (“**feature**”) is defined as a binary valued function that returns 1 iff class and data match the **indicated** requirements (**constraints**):

$$f(x,y) = \begin{cases} 1 \; if \;\; y = c_i \wedge x = w_i \\ 0 \; otherwise \end{cases}$$

real/discrete/binary features now are all the same!

  - The probability of a feature with respect to the observed distribution is:

$$\hat{P}(f_i, X, Y) = E_{\hat{P}}[f_i] = \sum \hat{P}(x,y) f_i(x,y)$$

Getting lost? Reality check:

- I have told you:
  - MaxEnt is about maximizing "conditional entropy":
  - By multiplying binary (0/1) feature functions for observations with the joint (observation, class) probabilities, we can calculate the conditional probability of a class given its observations $H(Y=y|X=x)$
- We will still have to do:
  - Find weights (i.e., parameters) for each feature [function] that lead to the best model of the [observed] class probabilities.
- And you want to know:
  - How do we use all this to actually classify new input data?

Maximum entropy (MaxEnt 2/2)

- In a **linear** model, we'd use weights ("lambdas") that identify the most relevant features of our model, i.e., we use the following MAP to select a class:

$$\underset{y \in Y}{argmax} \sum \lambda_i f_i(X, y)$$

- To do **multinomial logistic** regression, expand with a **linear combination**:

$$\underset{y \in Y}{argmax} \frac{exp(\sum \lambda_i f_i(X, y))}{\sum_{y \in Y} exp(\sum \lambda_i f_i(X, y))}$$ *"exponential model"*

- Next: **Estimate** the λ weights (parameters) that **maximize** the conditional **likelihood** of this logistic model (**MLE**)

Maximum entropy (MaxEnt 2/2) [again]

- In summary, MaxEnt is about selecting the "maximal" model p*:

$$p^* = \underset{p \in P}{argmax} \; - \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \; log_2 \; p(y|x)$$

select some model that maximizes the conditional entropy...

- That obeys the following conditional equality constraint:

$$\sum_{x \in X} P(x) \sum_{y \in Y} P(y|x) \; f(x,y) = \sum_{x \in X, y \in Y} P(x,y) \; f(x,y)$$

...using a conditional model that matches the (observed) joint probabilities

- Next: Using, e.g., **Langrange multipliers**, one can establish the optimal λ parameters of the model that maximize the entropy of this probability:

$$p^*(y|X) = \frac{exp(\sum \lambda_i f_i(X,y))}{\sum_{y \in Y} exp(\sum \lambda_i f_i(X,y))}$$

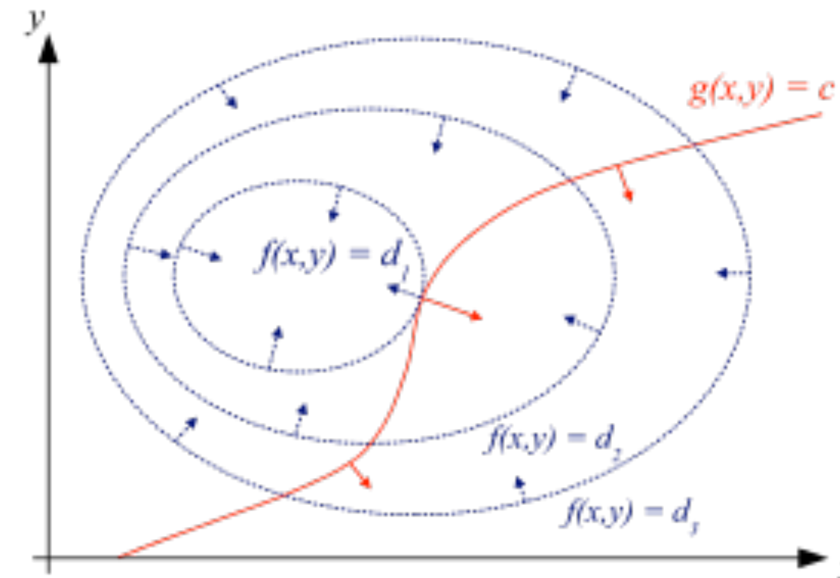


Image Source: WikiMedia Commons, Nexcis

Newton's method for paramter optimization

- Problem: find the λ parameters
  - an "**optimization problem**"
- MaxEnt surface is **concave**
  - one **single maximum**
- Using **Newton's method**
  - iterative, hill-climbing search for max.
  - the **first derivative** f' is zero at the [global] maximum (the "goal")
  - the **second derivative** f'' indicates rate of change: $\Delta\lambda_i$ (search direction)
  - takes the most direct route to the maximum *as opposed to gradient descent, which will follow a possibly curved path to the optimum*
- Using **L-BFGS**
  - a **heuristic** to simplify Newton's method *it is said to be "quasi-Newtonian"*
  - L-BFGS: **limited memory** **B**royden–**F**letcher–**G**oldfarb–**S**hanno
  - normally, the **partial second derivatives** would be stored in the **Hessian**, a matrix that **grows quadratically** with respect to the number of features
  - only uses the last few [partial] gradients to **approximate the search direction**

MaxEnt vs. naïve Bayes

MaxEnt model (as observed)

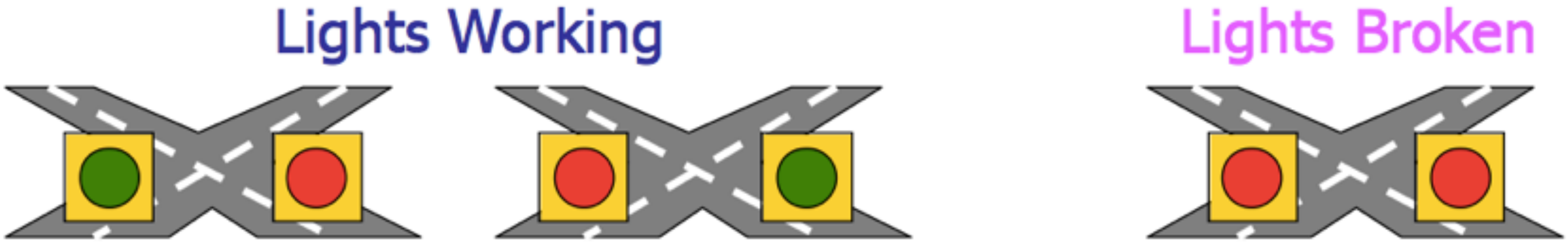


Image Source: Klein & Manning. Maxent Models, Conditional Estimation, and Optimization. ACL 2003 Tutorial

P(g,r,w) = 3/7   P(r,g,w) = 3/7   P(r,r,b) = 1/7

MaxEnt adjusts the Langrange multipliers (weights) to **model** the correct (observed) **joint probabilities**.

Note that the example has dependent features: the two stoplights!

Naïve Bayes model

- P(**w**) = 6/7
- P(r|**w**) = 1/2
- P(g|**w**) = 1/2

- P(**b**) = 1/7
- P(r|**b**) = 1
- P(g|**b**) = 0

- P(r,r,**b**) = (1/7)(1)(1) = 4/28
- P(r,g,**b**) = P(g,r,**b**) = P(g,g,**b**) = 0
- P(*,*,**w**) = (6/7)(1/2)(1/2) = 3/14

P(g,g,w) = 3/14??
P(r,r,w) = 3/14 > P(r,r,b) !?!?

But even MaxEnt cannot detect feature interaction

Empirical (joint) observations

A, B	a_r	a_g
b_r	1	3
b_g	3	0

Correct (target) distribution

A, B	a_r	a_g
b_r	1/7	3/7
b_g	3/7	0

2 feature model: A=r or B=r observed

A, B	a_r	a_g
b_r		
b_b		

only A=r

A, B	a_r	a_g
b_r	4/14	3/14
b_g	4/14	3/14

only B=r

A, B	a_r	a_g
b_r	4/14	4/14
b_g	3/14	3/14

A, B	a_r	a_g
b_r	16/49	12/49
b_g	12/49	9/49

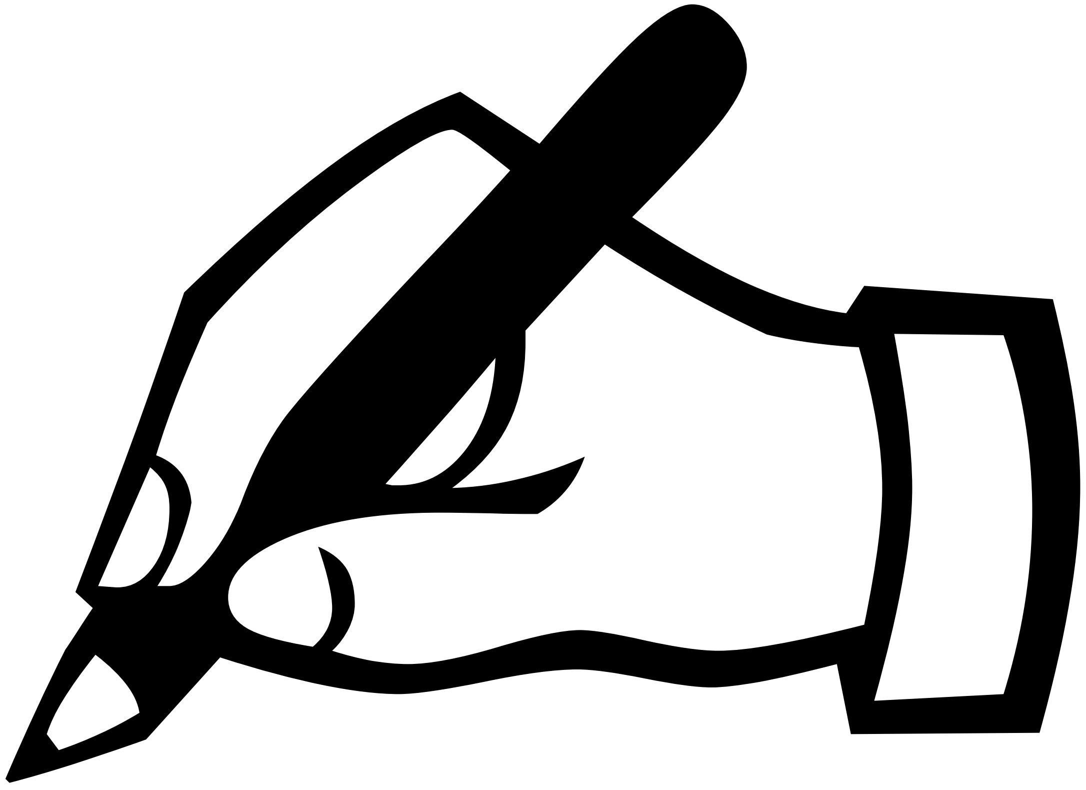
4 feature model: any a,b observed

A, B	a_r	a_g
b_r		
b_g		

A, B	a_r	a_g
b_r	1/7	3/7
b_g	3/7	0

Klein & Manning. MaxEnt Models, Conditional Estimation and Optimization. ACL 2003

Practical:
Classifying Wikipedia pages



A first look at probabilistic graphical models

- Latent Dirichlet Allocation: LDA
  - Blei, Ng, and Jordan. Journal of Machine Learning Research 2003
  - For assigning “topics” to “documents” *i.e., for text classification*
  - An **unsupervised**, **generative** model

Latent Dirichlet Allocation (LDA 1/3)

- Intuition for LDA
  - From: Edwin Chen. Introduction to LDA. 2011
  - "Document Collection"
    - I like to eat broccoli and bananas.
    - I ate a banana and spinach smoothie for breakfast.

      ➡ Topic A
    - Chinchillas and kittens are cute.
    - My sister adopted a kitten yesterday.

      ➡ Topic B
    - Look at this cute hamster munching on a piece of broccoli.

      ➡ Topic 0.6A + 0.4B

Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, …

Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, …

The Dirichlet process

A Dirichlet process is like drawing from an (infinite) "bag of dice" (with finite faces).

- A Dirichlet is a [possibly continuos] **distribution over** [discrete/multinomial] **distributions** (probability **masses**).

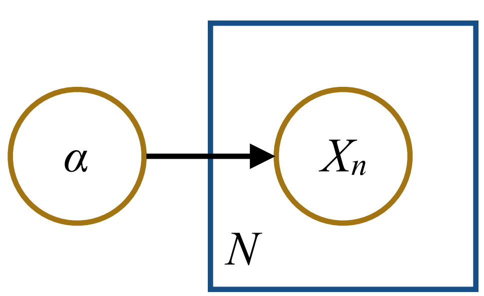
Gamma function -> a "continuous" factorial [!]

$$D(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \frac{\Gamma(\sum \alpha_i)}{\prod \Gamma(\alpha_i)} \prod \theta_i^{\alpha_i - 1}$$

α Dirichlet prior: ∀ $\alpha_i \in \alpha$: $\alpha_i > 0$

$\sum \theta_i = 1$; a Probability Mass Function

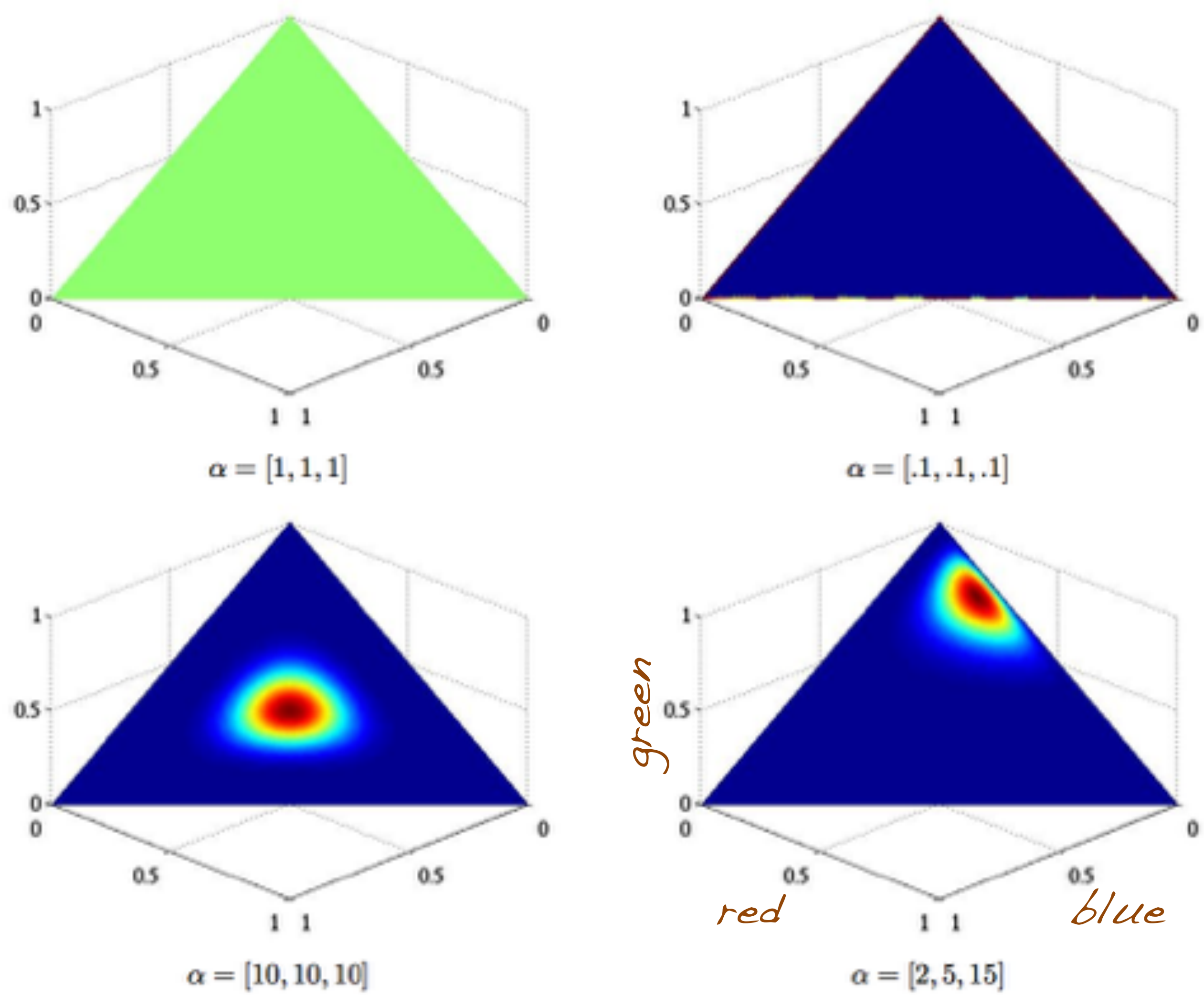
- The **Dirichlet Process samples** multiple independent, discrete **distributions** θ_i with repetition from $\boldsymbol{\theta}$ ("statistical clustering").



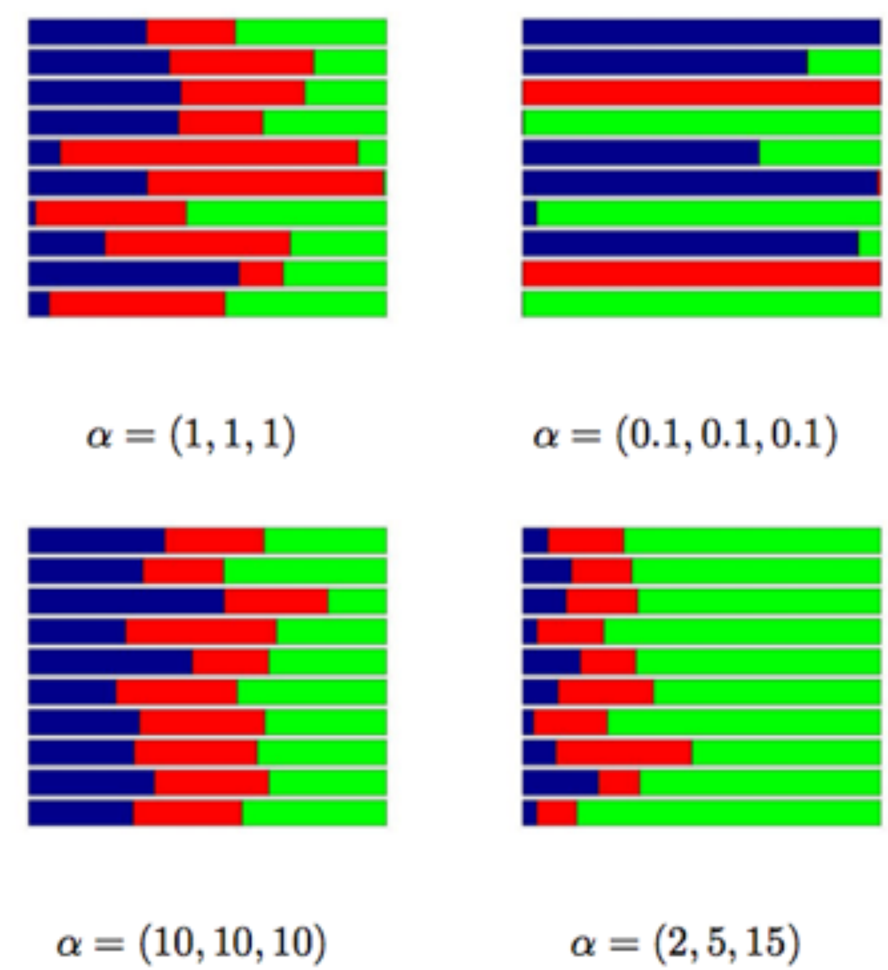
1. Draw a new distribution X from $D(\boldsymbol{\theta}, \boldsymbol{\alpha})$
2. With probability $\alpha \div (\alpha + n - 1)$ draw a new X
   With probability $n \div (\alpha + n - 1)$, (re-)sample an X_i from X

The Dirichlet prior α

" density plots over the probability simplex in R3"



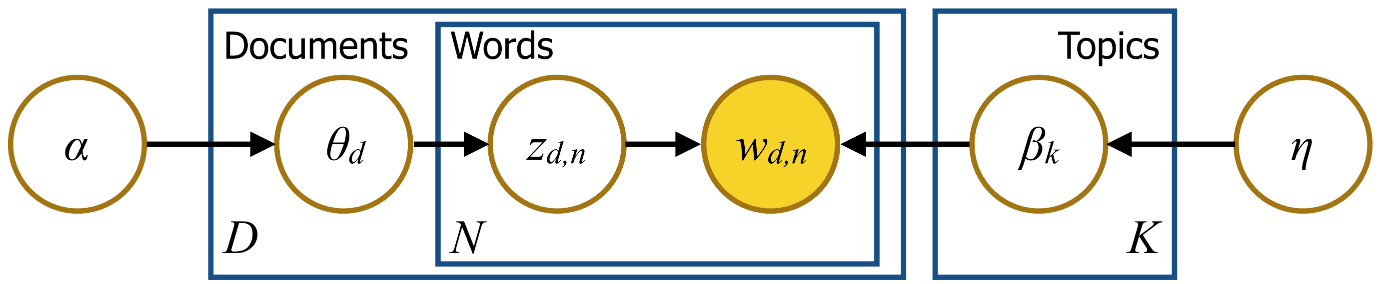
Documents and topic distributions (N=3)



- equal, =1 ➡ uniform distribution
- equal, <1 ➡ marginal distrib. ("choose few")
- equal, >1 ➡ symmetric, mono-modal distrib.
- not equal, >1 ➡ non-symmetric distribution

Frigyik et al. Introduction to the Dirichlet Distribution and Related Processes. 2010

Latent Dirichlet Allocation (LDA 2/3)



Joint Probability

P(Document-T.)

P(Word | Topics, Word-T.)

$$P(B,\Theta,Z,W) = \left(\prod_{k}^{K} P(\beta_k|\eta)\right)\left(\prod_{d}^{D} P(\theta_d|\alpha)\prod_{n}^{N} P(z_{d,n}|\theta_d)P(w_{d,n}|\beta_{1:K},z_{d,n})\right)$$

P(Topics)

P(Word-T. | Document-T.)

- α - per-document Dirichlet prior
- θ_d - topic distribution of document d
- $z_{d,n}$ - word-topic assignments
- $w_{d,n}$ - **observed** words
- β_k - word distrib. of topic k
- η - per-topic Dirichlet prior

dampens the topic-specific score of terms assigned to many topics

$$termscore_{k,n} = \hat{\beta}_{k,n}\,\boxed{log\frac{\hat{\beta}_{k,n}}{\left(\prod_j^K \hat{\beta}_{j,n}\right)^{1/K}}}$$

What Topics is a Word assigned to?

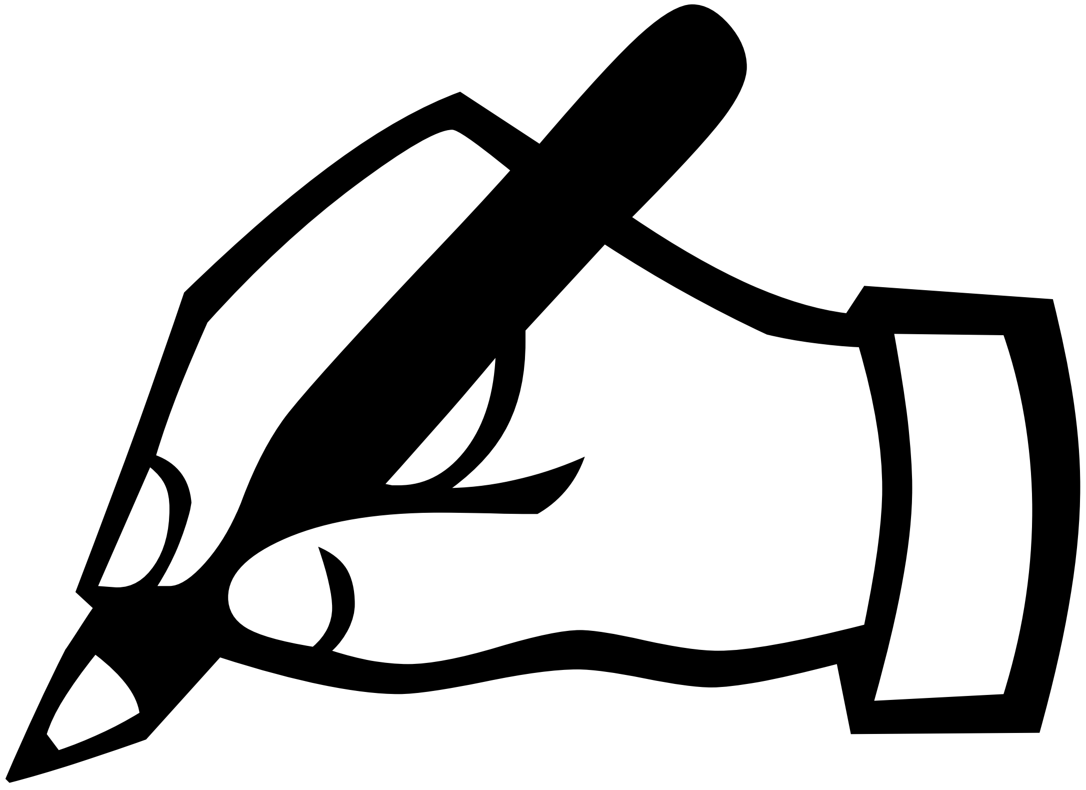
Latent Dirichlet Allocation (LDA 3/3)

- LDA inference in a nutshell
  - **Calculate the posterior probability that Topic t generated Word w.**
  - Initialization: Choose K, the number of Topics, and randomly assign one out of the K Topics to each of the N Words in each of the D Documents.
    - The **same word** can have different Topics **at different positions** in the Document.
  - Then, for each Topic:
    And for each Word in each Document:

1. Compute P(Word-Topic | Document): the proportion of [Words assigned to] Topic t in Document d
2. Compute P(Word | Topics, Word-Topic): the probability a Word w is assigned a Topic t (using the general distribution of Topics and the Document-specific distribution of [Word-] Topics)
   - Note that a Word can be assigned a different Topic each time it appears in a Document.
3. Given the prior probabilities of a Document's Topics and that of Topics in general, reassign
   P(Topic | Word) = P(Word-Topic | Document) * P(Word | Topics, Word-Topic)

- Repeat until P(Topic | Word) stabilizes (e.g., MCMC Gibbs sampling, Course 04)

Practical: Clustering Wikipedia pages



Evaluation metrics for classification tasks

Evaluations should answer questions like:

How to measure a change to an approach?

Did adding a feature improve or decrease performance?

Is the approach good at locating the relevant pieces or good at excluding the irrelevant bits?

How do two or more different methods compare?

Essential evaluation metrics: Accuracy, F-Measure, MCC Score

Patient / Doctor	has cancer	is healthy
diagnose cancer	TP	FP
detects nothing	FN	TN

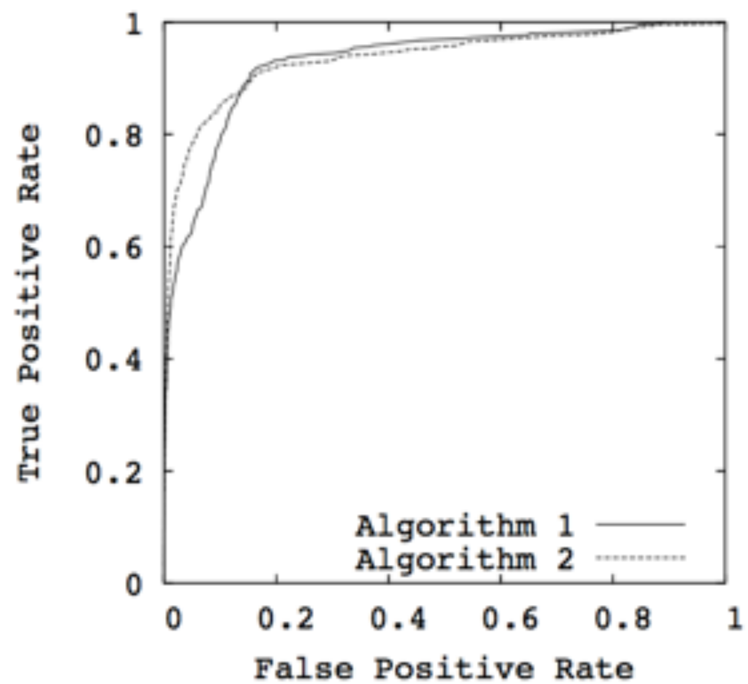
- **Precision** (P)
  - correct hits [TP] ÷ all hits [TP + FP]
- **Recall** (R; **Sensitivity**, TPR)
  - correct hits [TP] ÷ true cases [TP + FN]
- **Specificity** (True Negative Rate)
  - correct misses [TN] ÷ negative cases [FP + TN]

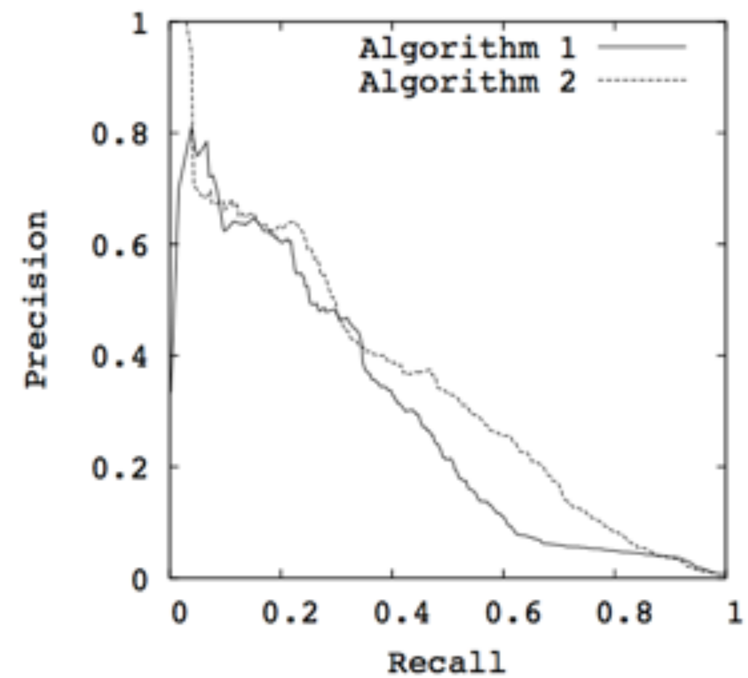
NB: no result order

- **Accuracy**
  - correct classifications [TP + TN] ÷ all cases [TP + TN + FN + FP])
  - highly **sensitive to** class **imbalance**
- **F-Measure** (F-Score)
  - the harmonic mean between P & R = 2 TP ÷ (2 TP + FP + FN) = (2 P R) ÷ (P + R)
  - does **not** require a **TN** count
- **MCC Score** (Mathew's Correlation Coefficient)
  - **χ^2-based**: (TP TN - FP FN) ÷ sqrt[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]
  - **robust against** class **imbalance**

Ranked evaluation results: AUC ROC and PR

Area Under the Curve
Receiver-Operator Characteristic
Precision-Recall



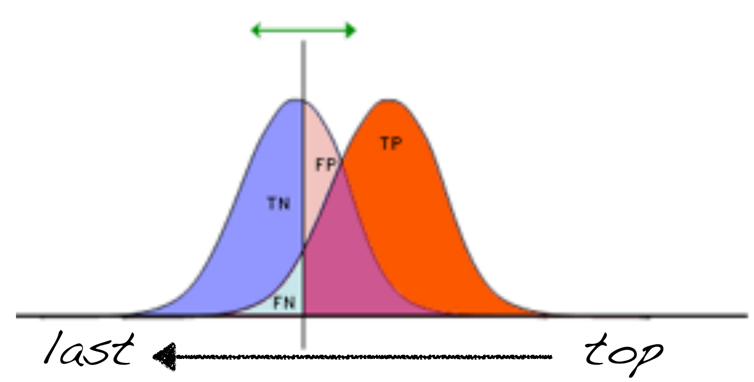


Davis & Goadrich.
ICML 2006

TPR / Recall *(aka. Sensitivity)*
TP ÷ (TP + FN)

FPR *(~~not~~ Specificity!)*
FP ÷ (TN + FP)

Precision
TP ÷ (TP + FP)



TP	FP
FN	TN
1	1

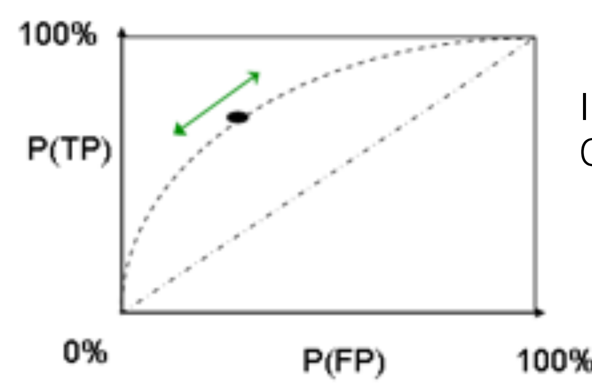


Image Source: WikiMedia Commons, kku ("kakau", eddie)

To ROC or to PR?

Curve I:
10 hits in the top 10, and 10 hits spread over the next 1500 results.

AUC ROC 0.813

Results: 20 T « 1980 N

Curve II:
Hits spread evenly over the first 500 results.

AUC ROC 0.875

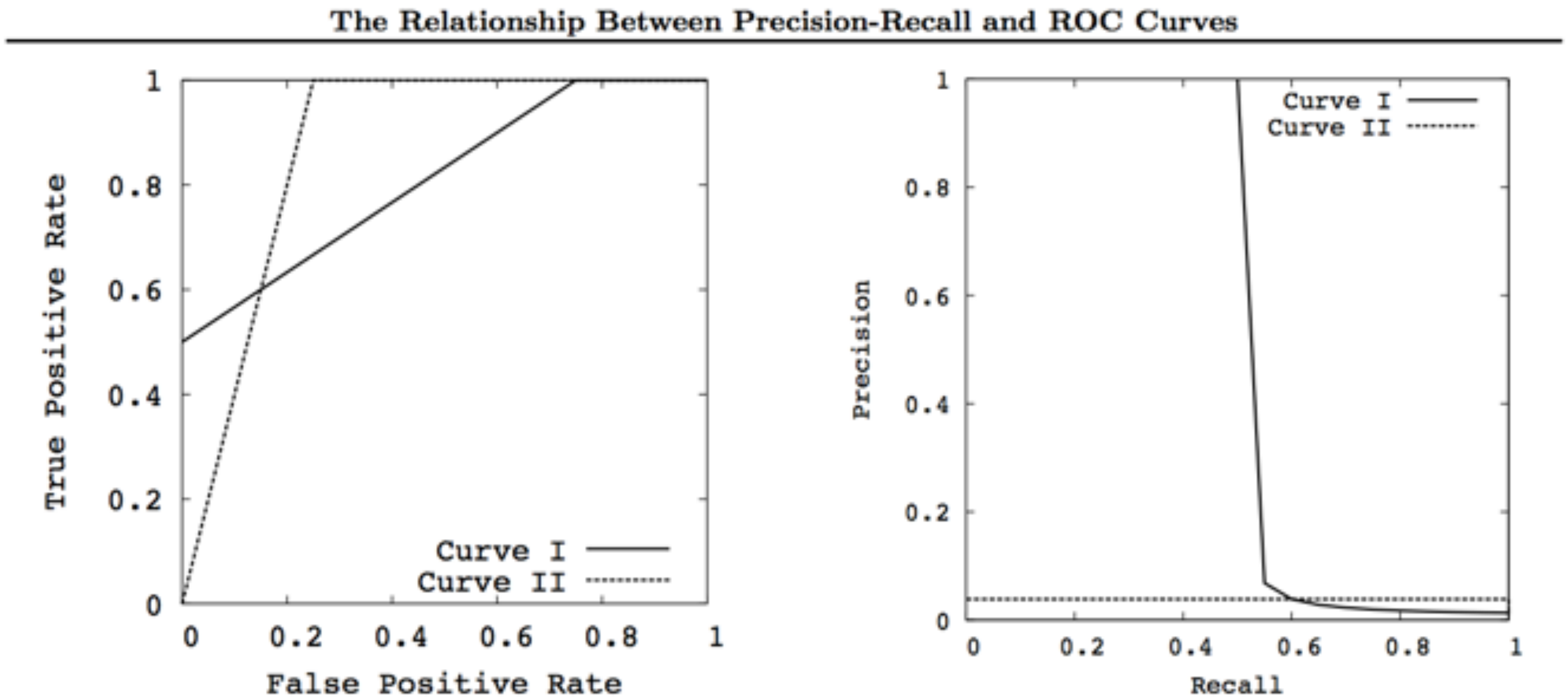


Figure 11. Comparing AUC-ROC for Two Algorithms

Figure 12. Comparing AUC-PR for Two Algorithms

"An algorithm which optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve."

Davis & Goadrich, 2006

- Davis & Goadrich. The Relationship Between PR and ROC Curves. ICML 2006
- Landgrebe et al. Precision-recall operating characteristic (P-ROC) curves in imprecise environments. Pattern Recognition 2006
- Hanczar et al. Small-Sample Precision of ROC-Related Estimates. Bioinformatics 2010

→ Use (AUC) PR for [imbalanced] ranking scenarios!

Sentiment Analysis

as an example **domain** for text classification

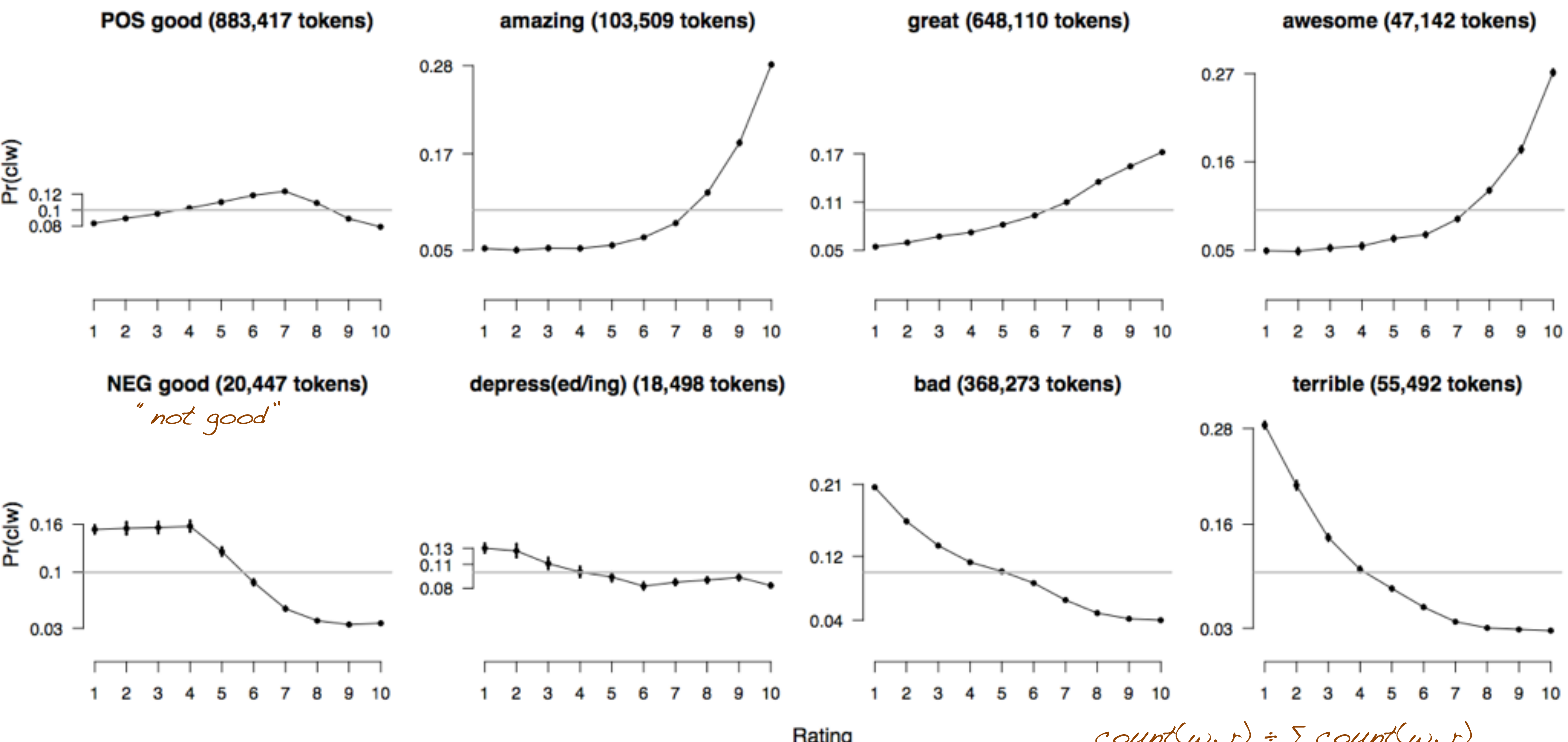
(only if there is time left after the exercises)

Cristopher Potts. Sentiment Symposium Tutorial. 2011
http://sentiment.christopherpotts.net/index.html

Opinion/Sentiment Analysis

- Harder than “regular” document classification
  - irony, neutral (“non-polar”) sentiment, negations (“not good”), syntax is used to express emotions (“!”), context dependent
- Confounding polarities from individual aspects (phrases)
  - e.g., a car company’s “customer service” vs. the “safety” of their cars
- Strong commercial interest in this topic
  - “Social” (commercial?) networking sites (FB, G+, …; advertisement)
  - Reviews (Amazon, Google Maps), blogs, fora, online comments, …
  - Brand reputation and political opinion analysis

Polarity of Sentiment Keywords in IMDB



Cristopher Potts. On the negativity of negation. 2011

count(w, r) ÷ ∑ count(w, r)

Note: P(rating I word) = P(word I rating) ÷ P(word)

5+1 Lexical Resources for Sentiment Analysis

Cristopher Potts. Sentiment Symposium Tutorial. 2011

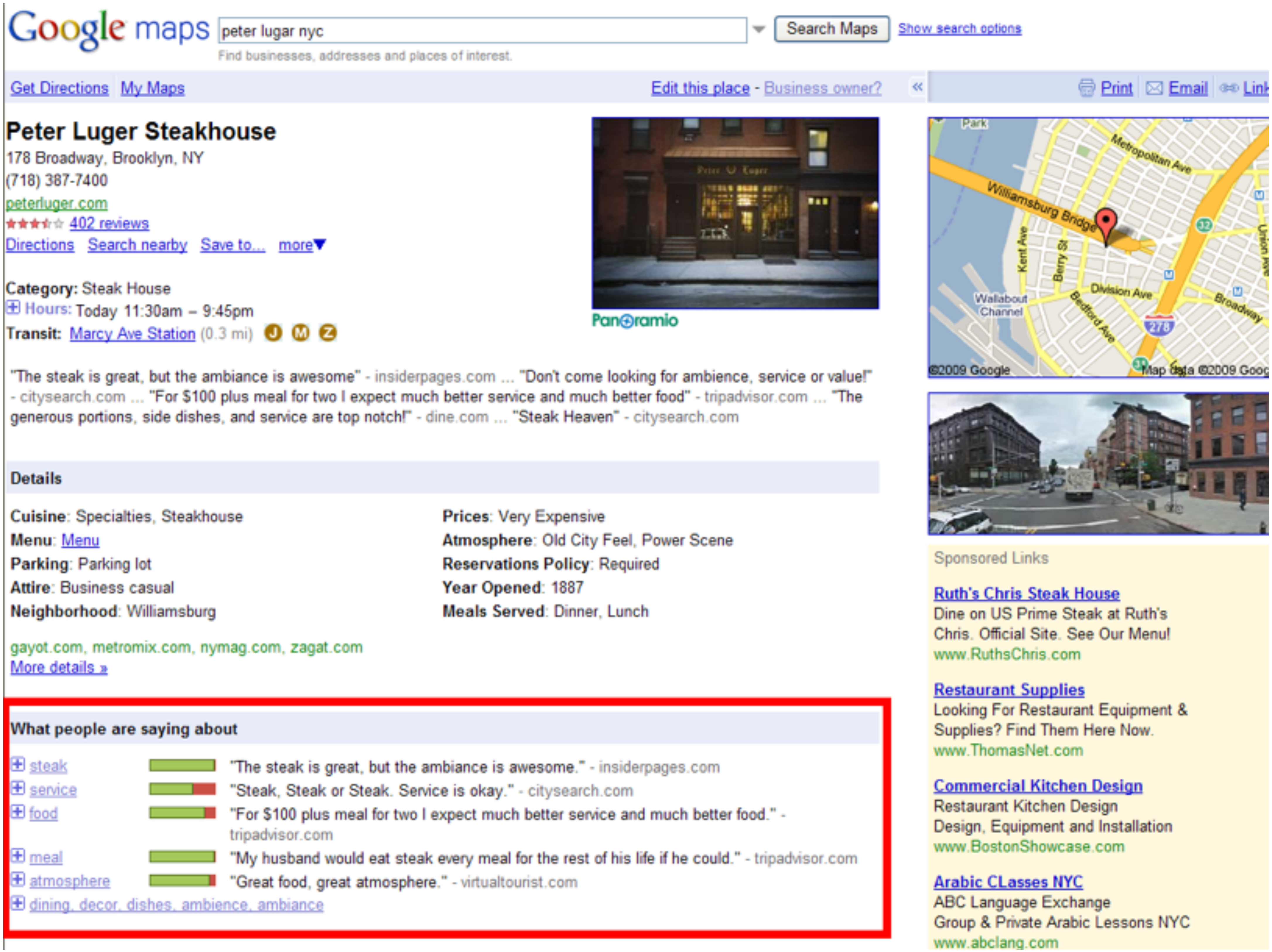
Disagree-ment	Opinion Lexicon	General Inquirer	SentiWordNet	LIWC
Subjectivity Lexicon	33/5402 (0.6%)	49/2867 (2%)	1127/4214 (27%)	12/363 (3%)
Opinion Lexicon		32/2411 (1%)	1004/3994 (25%)	9/403 (2%)
General Inquirer			520/2306 (23%)	1/204 (0.5%)
SentiWord Net				174/694 (25%)

MPQA Subjectivity Lexicon:	http://mpqa.cs.pitt.edu/
Liu's Opinion Lexicon:	http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
General Inquirer:	http://www.wjh.harvard.edu/~inquirer/
SentiWordNet:	http://sentiwordnet.isti.cnr.it/
LIWC (commercial, $90):	http://www.liwc.net/
NRC Emotion Lexicon (+1):	http://www.saifmohammad.com/ (➡Publications & Data)

Detecting the Sentiment of Individual Aspects

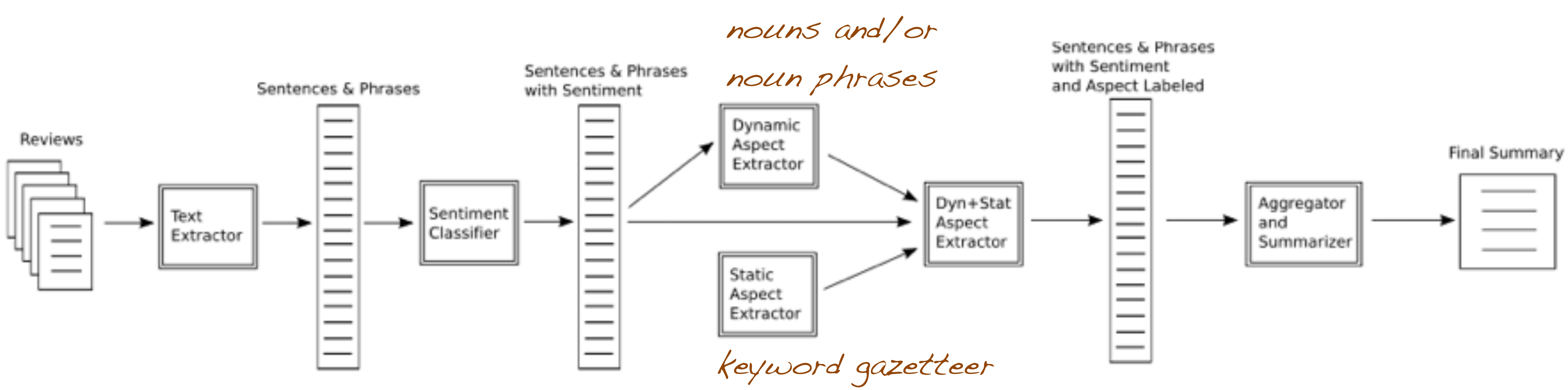
- Goal: Determine the sentiment for a particular aspect or establish their polarity.
  - An “aspect” here is a phrase or concept, like “customer service”.
  - “They have a **great**$_{+}$ customer service team, but the delivery **took ages**$_{-}$.”
- Solution: Measure the co-occurrence of the aspect with words of distinct sentiment or relative co-occurrence with words of the same polarity.
  - The “sentiment” keywords are taken from some lexical resource.

Google's Review Summaries



Using PMI to Detect Aspect Polarity

- **Polarity(aspect)** := PMI(*aspect*, pos-sent-kwds) - PMI(*aspect*, neg-sent-kwds)
  - Polarity > 0 = positive sentiment
  - Polarity < 0 = negative sentiment
- Google's approach:



- Blair-Goldensohn et al. Building a Sentiment Summarizer for Local Service Reviews. WWW 2008

Practical:
Twitter sentiment mining