Comp 408 Advanced Topics in Artificial Intelligence

Lecture 5

Text Classification

6/3/ 2025 – 8/3/2025

Most of the Slides are by D. Jurafsky and J. M. Martin

Outline

- Introduction
- Text classification application
- Text classification definition
- Naïve Bayes
  - Intuition of Naïve Bayes
  - Formalizing the Naïve Bayes Classifier
  - Training the Naïve Bayes Classifier
  - Evaluation
    - Precision, Recall, F-measure

Introduction

- The goal of text classification is to take a document, extract some features, and then **classify** the document into one of a set of classes.
- Most cases of classification in NLP are done via supervised machine learning.

Supervised Machine Learning

- We have a data set of input observations, each associated with some correct output (a ‘supervision signal’). The goal of the supervised learning algorithm is to learn how to map from a new observation to a correct output.

Is this spam?

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

Gender Identification (Male or female author?)

M

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam…

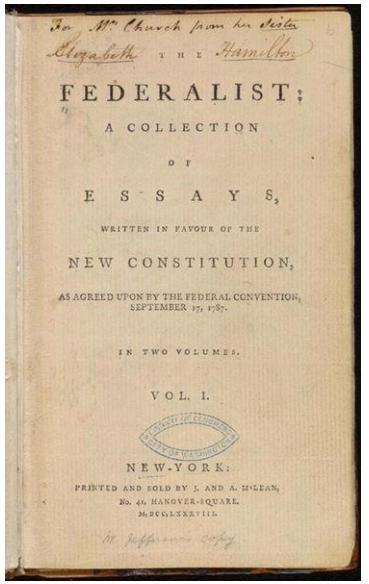
F

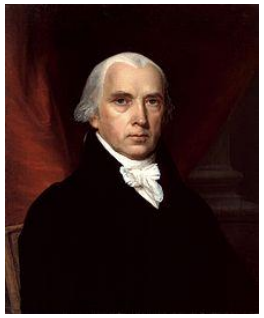
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets…

S. Argamon, M. Koppel, J. Fine, A. R. Shimoni, 2003. "Gender, Genre, and Writing Style in Formal Written Texts," Text, volume 23, number 3, pp. 321–346

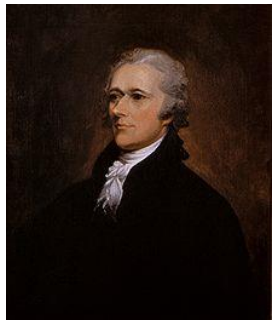
Who wrote which Federalist papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods





James Madison



Alexander Hamilton

What is the subject of this medical article?

MEDLINE Article

Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

?

MeSH Subject Category Hierarch

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Positive or negative movie review?

\+ ...zany characters and richly applied satire, and some great plot twists

− It was pathetic. The worst part about it was the boxing scenes...

\+ ...awesome caramel sauce and sweet toasty almonds. I love this place!

− ...awful pizza and ridiculously overpriced...

Positive or negative movie review?

\+ *...zany characters and **richly** applied satire, and some **great** plot twists*

– *It was **pathetic**. The **worst** part about it was the boxing scenes...*

\+ *...**awesome** caramel sauce and sweet toasty almonds. I **love** this place!*

– *...**awful** pizza and **ridiculously** overpriced...*

Why sentiment analysis?

- Movie: is this review positive or negative?
- Products: what do people think about the new iPhone?
- Public sentiment: how is consumer confidence?
- Politics: what do people think about this candidate or issue?
- Prediction: predict election outcomes or market trends from sentiment

Basic Sentiment Classification

- Sentiment analysis is the detection of **attitudes**
- Simple task we focus
  - Is the attitude of this text positive or negative?

Summary: Text Classification

- Sentiment analysis
- Spam detection
- Authorship identification
- Language Identification
- Assigning subject categories, topics, or genres
- …

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- *Output*: a predicted class $c \in C$

Classification Methods: Hand-coded rules

- Rules based on combinations of words or other features
  - spam: black-list-address OR (words like “dollars” AND “you have been selected”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

Classification Methods: Supervised Machine Learning

- Input:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of m hand-labeled documents $(d_1,c_1),....,(d_m,c_m)$
- Output:
  - a learned classifier $\gamma:d \rightarrow c$

Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression
  - Neural networks
  - k-Nearest Neighbors
  - …

Why Naive Bayes?

1. Bayesian models are important in AI
   - Naive Bayes is a simple introduction to them
2. Naive Bayes is relatively easy to **interpret**
   - It's possible to do the computation by hand (as with the n-gram model)
   - That makes the factors that play a role in the classification more transparent
   - Developing these kinds of intuitions is much harder for huge neural models

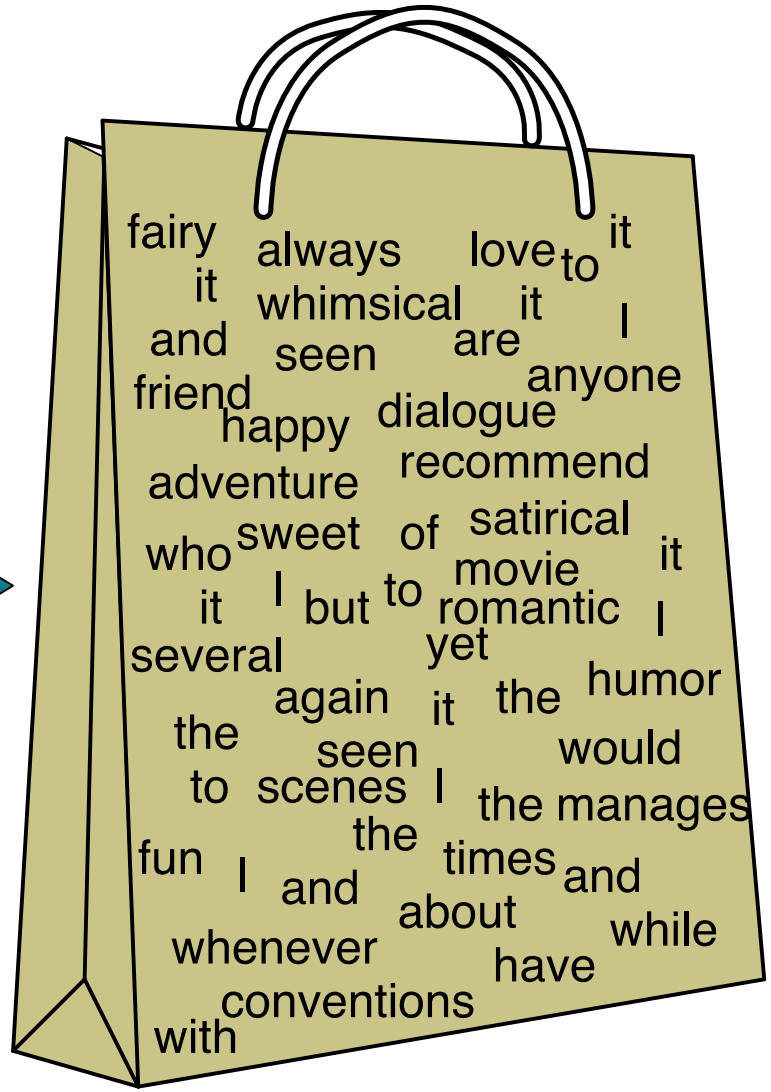
Naive Bayes Intuition

- Simple ("naive") classification method based on Bayes rule
- Relies on very simple representation of document
  - **Bag of words (BOW)**

The Bag of Words Representation

Document

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

The bag of words representation

$\gamma($

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

$)=c$

Bayes' Rule Applied to Documents and Classes

- For a document d and a class c

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Naive Bayes Classifier (I)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Dropping the denominator

Naive Bayes Classifier (II)

"Likelihood" "Prior"

$$c_{MAP} = \underset{c \hat{I} C}{\operatorname{argmax}} P(d \mid c) P(c)$$

Document d represented as features $x1..xn$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Naïve Bayes Classifier (III)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Naïve Bayes Classifier (IV)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

O($|X|^n \bullet |C|$) parameters

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Multinomial Naïve Bayes Independence Assumptions

$$P(x_1, x_2, ..., x_n \mid c)$$

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i|c_j)$ are independent given the class c.

$$P(x_1, ..., x_n \mid c) = P(x_1 \mid c) * P(x_2 \mid c) * P(x_3 \mid c) * ... * P(x_n \mid c)$$

Multinomial Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, ..., x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in position} P(xi \mid c)$$

Applying Multinomial Naive Bayes Classifiers to Text Classification

positions ← all word positions in test document

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

Problems with multiplying lots of probabilities

- There's a problem with this:

$$c_{NB} = \underset{c_j \hat{I} C}{\text{argmax}} P(c_j) \tilde{O}_{i \hat{I} positions} P(x_i | c_j)$$

- Multiplying lots of probabilities can result in floating-point underflow!

.0006 * .0007 * .0009 * .01 * .5 * .000008….

- Idea: Use logs, because $\log(ab) = \log(a) + \log(b)$

We'll sum logs of probabilities instead of multiplying probabilities!

We actually do everything in log space

Instead of this: $$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

This: $$c_{\mathrm{NB}} = \underset{c_j \in C}{\operatorname{argmax}} \left[\log P(c_j) + \sum_{i \in \mathrm{positions}} \log P(x_i | c_j) \right]$$

Notes:

1) Taking log doesn't change the ranking of classes!
   The class with highest probability also has highest log probability!
2) It's a linear model:
   Just a max of a sum of weights: a **linear** function of the inputs
   So naive bayes is a **linear classifier**

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates (MLE)
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

Parameter Estimation

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

fraction of times word w_i appears among all words in documents of topic c_j

V is the set of all vocabulary

- Create mega-document for topic *j* by concatenating all docs in this topic
  - Use frequency of *w* in mega-document يوخذ تكرار الكلمات في الحساب

Problem with Maximum Likelihood

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive.**

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{count(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0$$

$$c_{MAP} = \text{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Solution: Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i \mid c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} \left(\text{count}(w, c) + 1\right)}$$

$$= \frac{\text{count}(w_i, c) + 1}{\left(\sum_{w \in V} \text{count}(w, c)\right) + |V|}$$

Multinomial Naïve Bayes: Learning

1. From training corpus, extract *Vocabulary*

2. Calculate $P(c_j)$ terms:
   - For each c_j in C do
     $docs_j \leftarrow$ all docs with class $= c_j$

3. Calculate $P(w_k \mid c_j)$ terms
   - $Text_j \leftarrow$ single doc containing all $docs_j$
   - For each word w_k in *Vocabulary*
     $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

$$P(w_k \mid c_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$$

n is the number of words in the class c_j

Unknown words

- What about unknown words
  - that appear in our test data
  - but not in our training data or vocabulary?
- We **ignore** them
  - Remove them from the test document!
  - Pretend they weren't there!
  - Don't include any probability for them at all!
- Why don't we build an unknown word model?
  - It doesn't help: knowing which class has more unknown words is not generally helpful!

Stop words

- Some systems ignore stop words
  - **Stop words:** very frequent words like *the* and *a*.
    - Sort the vocabulary by word frequency in training set
    - Call the top 10 or 50 words the **stopword list**.
    - Remove all stop words from both training and test sets
      - As if they were never there!
- But removing stop words doesn't usually help
  - So in practice most NB algorithms use **all** words and **don't** use stopword lists

A worked sentiment example

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

A worked sentiment example with add-1 smoothing

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable ~~with~~ no fun

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

P(-) = 3/5
P(+) = 2/5

2. Drop "with" as it is not in training data

3. Likelihoods from training:

$$p(w_i|c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

|V| = 20, count of positive vocab = 14, count of negative vocab =9

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

4. Scoring the test set:

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Example 2

- Given the following short movie reviews, each labeled with a genre, either comedy or action:

	Documents	Label
1	fun, couple, love, love	comedy
2	fast, furious, shoot	action
3	couple, fly, fast, fun, fun	comedy
4	furious, shoot, shoot, fun	action
5	fly, fast, shoot, love	action

and a new document D: fast, couple, shoot, fly

compute the most likely class for D.

Assume a naive Bayes classifier and use add-1 smoothing for the likelihoods.

Example 2 (Cont.)

- $P(\text{class}|\text{feature})= \frac{P(\text{feature}|\text{class})P(\text{class})}{P(\text{feature})}$
- A calculation formula for the new document:
  - “fast couple shoot fly”

$P(\text{class}|\text{feature}) \propto P(\text{fast}|\text{class})P(\text{couple}|\text{class})P(\text{shoot}|\text{class})P(\text{fly}|\text{class})P(\text{class})$

$$P(\text{comedy}) = \frac{\text{count (comedy)}}{\text{count(all class)}} = \frac{2}{5}$$

$$P(\text{action}) = \frac{3}{5}$$

Example 2 (Cont.)

- The bag of words is {fly, fast, shoot, love, fun, couple, fast, furious} with length 7

P(feature |class)	Comedy	Action
fast	P(fast | comedy) $=\frac{count(fast)+1}{count(comedy)+\|V\|}=\frac{1+1}{9+7}=\frac{1}{8}$	P(fast |action) $=\frac{count(fast)+1}{count(action)+\|V\|}=\frac{2+1}{11+7}=\frac{3}{18}=\frac{1}{6}$
couple	P(couple |comedy) = $\frac{3}{16}$	$\frac{1}{18}$
shoot	P(shoot|comedy) = $\frac{1}{16}$	$\frac{5}{18}$
fly	P(fly|comedy) = $\frac{1}{8}$	$\frac{1}{9}$

Example 2 (Cont.)

- $P(\text{comedy}|D) \propto \frac{1}{8} * \frac{3}{16} * \frac{1}{16} * \frac{1}{8} * \frac{2}{5} \approx \frac{6}{81920} \approx .00007$
- $P(\text{action}|D) \propto \frac{1}{6} * \frac{1}{18} * \frac{5}{18} * \frac{1}{9} * \frac{3}{5} \approx \frac{15}{87480} \approx 0.00017$
- As a result, the most likely class for D is **action**.

H. W.

- Assume the following likelihoods for each word being part of a positive or negative movie review, and equal prior probabilities for each class.

P(pos) = P(neg)

	pos	neg
I	0.09	0.16
always	0.07	0.06
like	0.29	0.06
foreign	0.04	0.15
films	0.08	0.11

- What class will Naïve Bayes assign to the sentence:

I always like foreign films

Evaluating Classifiers: How well does our classifier work?

Consider the binary classifiers:

- Is this email spam?

  spam (+) or not spam (-)

- Is this post about Delicious Pie Company?

  about Del. Pie Co (+) or not about Del. Pie Co(-)

- Positive class: tweets about Delicious Pie Co
- Negative class: all other tweets

We'll need to know

1. What did our classifier say about each email or post?
2. What should our classifier have said, i.e., the correct answer, usually as defined by humans ("gold label")

First step in evaluation: The confusion matrix

		gold standard labels	
		gold positive	gold negative
system output labels	system positive	**true positive**	**false positive**
	system negative	**false negative**	**true negative**

Accuracy on the confusion matrix

gold standard labels

system output labels		gold positive	gold negative
	system positive	**true positive**	**false positive**
	system negative	**false negative**	**true negative**

$$\textbf{accuracy} = \frac{tp+tn}{tp+fp+tn+fn}$$

Why don't we use accuracy?

Accuracy doesn't work well when we're dealing with uncommon or imbalanced classes

Suppose we look at 1,000,000 social media posts to find Delicious Pie-lovers (or haters)

- 100 of them talk about our pie
- 999,900 are posts about something unrelated

Imagine the following simple classifier

Every post is "not about pie"

Why don't we use accuracy?

Accuracy of our "nothing is pie" classifier

999,900 true negatives and 100 false negatives

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} = \frac{999{,}900}{1000000}$$ = 99.99%!

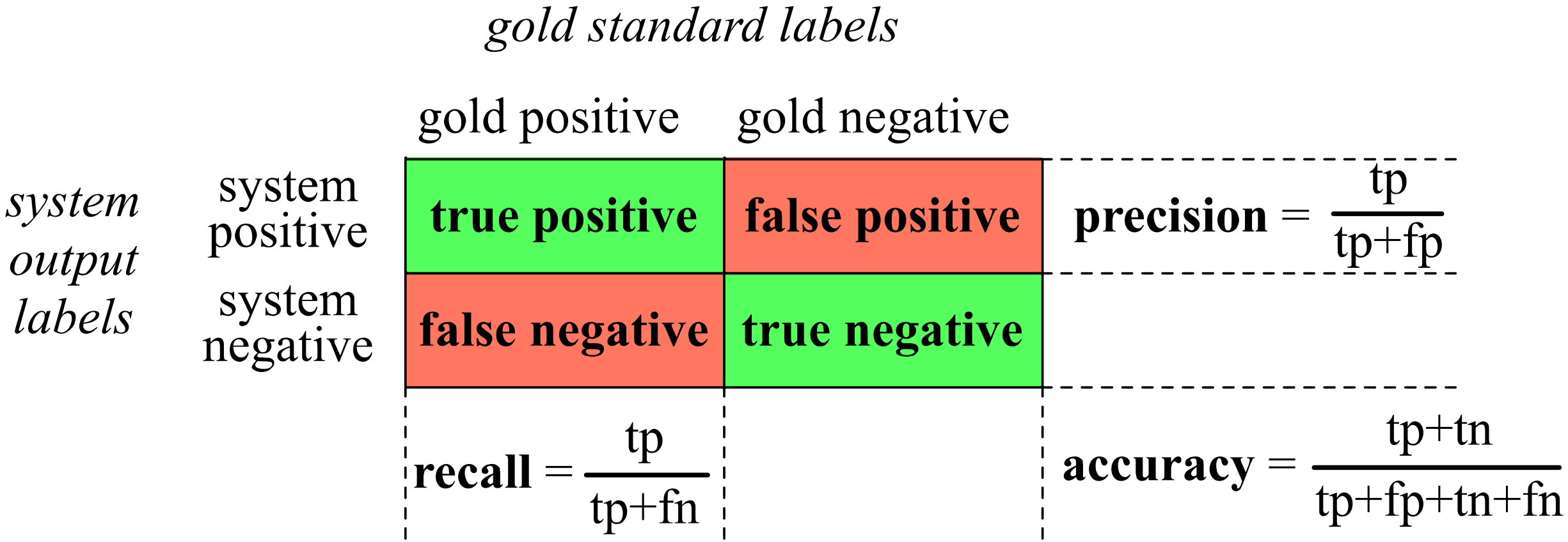
But useless at finding pie-lovers (or haters)!!

Which was our goal!

Accuracy doesn't work well for unbalanced classes

Most tweets are not about pie!

Instead of accuracy we use precision and recall



Evaluation: Precision

- % of items the system detected (i.e., items the system labeled as positive) that are in fact positive (according to the human gold labels)

$$\textbf{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Precision: % of selected items that are correct

Evaluation: Recall

- % of items actually present in the input that were correctly identified by the system.

$$\textbf{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Recall: % of correct items that are selected

Why Precision and recall

- Our dumb pie-classifier
  - Just label nothing as "about pie"

Accuracy=99.99%

but

Recall = 0

- (it doesn't get any of the 100 Pie tweets)

Precision and recall, unlike accuracy, emphasize true positives:

- finding the things that we are supposed to be looking for.

A combined measure: F

- F measure: a single number that combines P and R:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- We almost always use balanced F_1 (i.e., $\beta = 1$)

$$\mathrm{F}_1 = \frac{2PR}{P + R}$$