„Speech and Language Processing”
Dan Jurafsky and James H. Martin

(3rd ed. draft)

https://web.stanford.edu/~jurafsky/slp3/

Precision, Recall, and F measure

Evaluation

Let's consider just binary text classification tasks

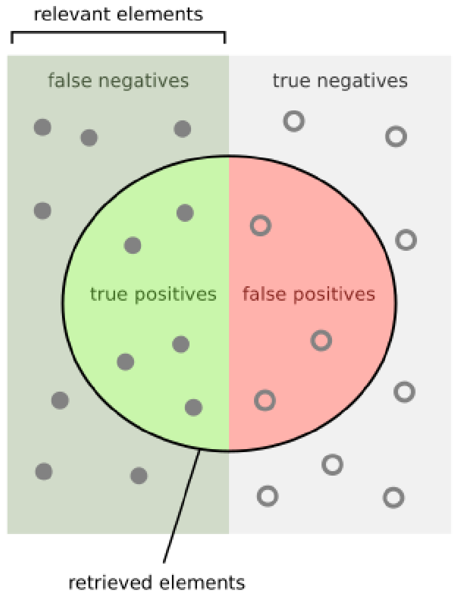
Imagine you're the CEO of Delicious Pie Company

You want to know what people are saying about your pies

So you build a "Delicious Pie" tweet detector

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

The 2-by-2 confusion matrix



		gold standard labels		
		gold positive	gold negative	
system output labels	system positive	**true positive**	**false positive**	**precision** $= \frac{\text{tp}}{\text{tp+fp}}$
	system negative	**false negative**	**true negative**	
		recall $= \frac{\text{tp}}{\text{tp+fn}}$		**accuracy** $= \frac{\text{tp+tn}}{\text{tp+fp+tn+fn}}$

Evaluation: Accuracy

Why don't we use **accuracy** as our metric?

Imagine we saw 1 million tweets

- 100 of them talked about Delicious Pie Co.
- 999,900 talked about something else

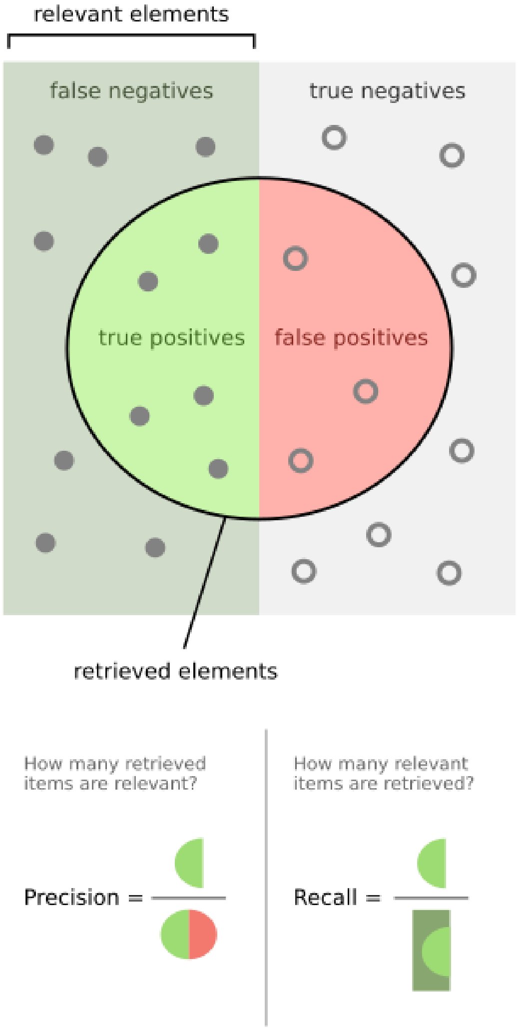
We could build a dumb classifier that just labels every tweet "not about pie"

- It would get 99.99% accuracy!!! Wow!!!!
- But useless! Doesn't return the comments we are looking for!
- That's why we use **precision** and **recall** instead

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)

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

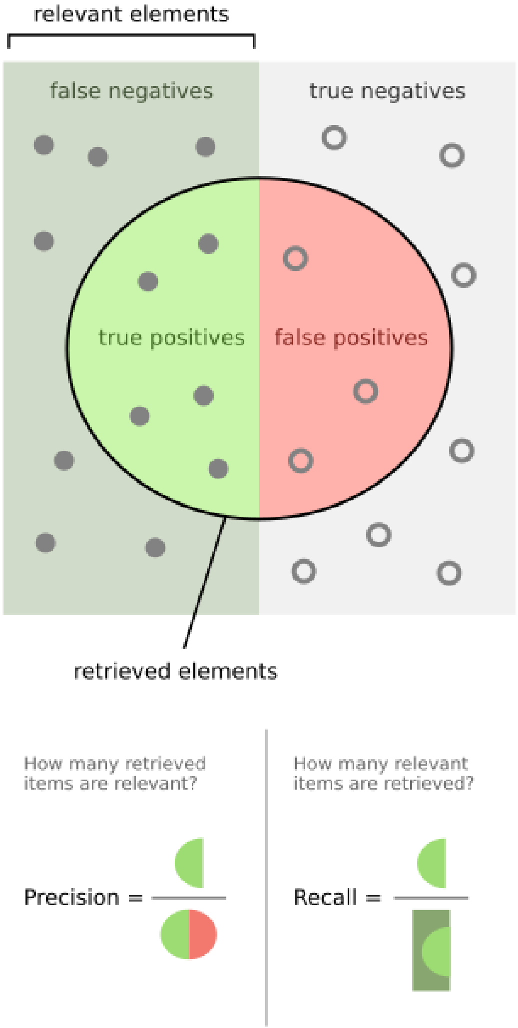


https://commons.wikimedia.org/wiki/File:Precisionrecall.svg

Evaluation: Recall

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

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



https://commons.wikimedia.org/wiki/File:Precisionrecall.svg

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., $\beta = 1$)

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

Development Test Sets ("Devsets") and Cross-validation

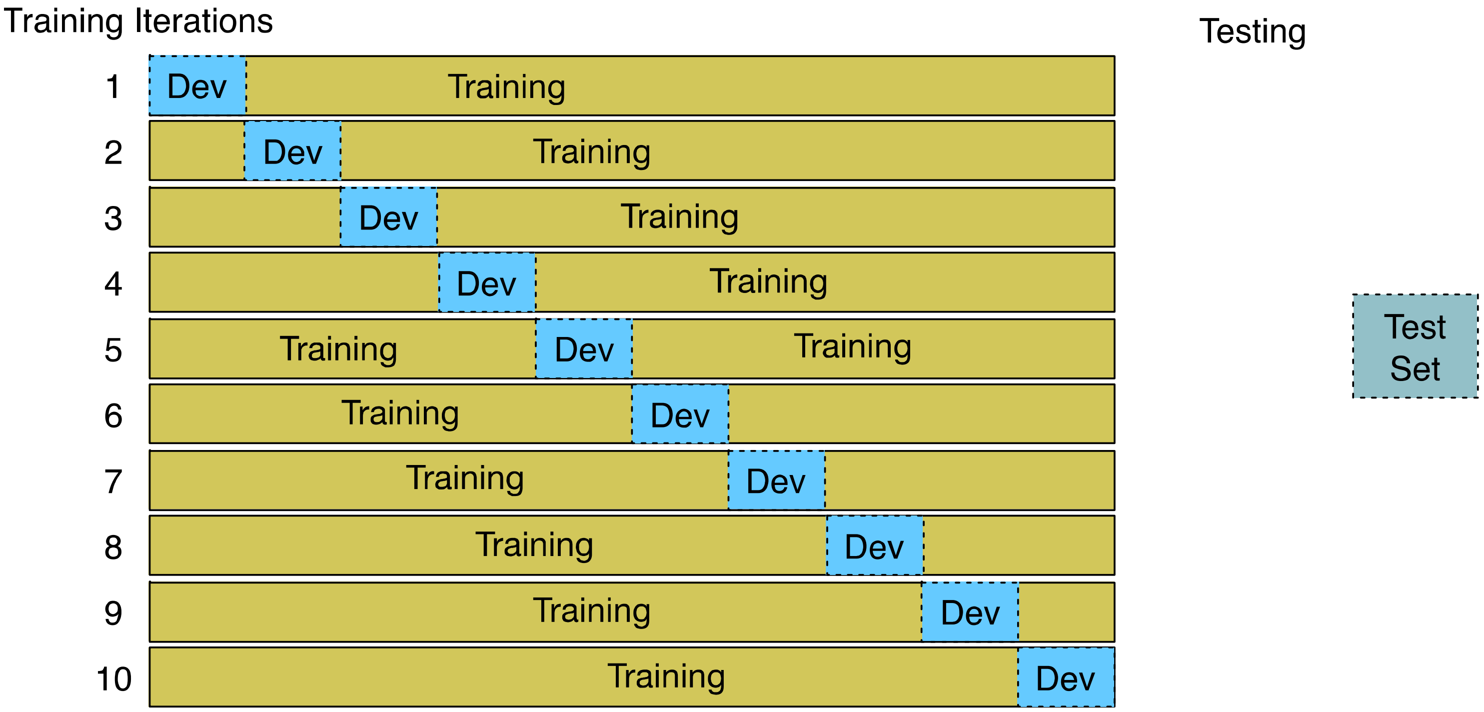
Training set | Development Test Set | Test Set

Train on training set, tune on devset, report on testset

- This avoids overfitting ('tuning to the test set')
- More conservative estimate of performance
- But paradox: want as much data as possible for training, and as much for dev; how to split?

Cross-validation: multiple splits

Pool results over splits, Compute pooled dev performance



Evaluation with more than two classes

Confusion Matrix for 3-class classification

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	**precision**u= $\frac{8}{8+10+1}$
	normal	5	60	50	**precision**n= $\frac{60}{5+60+50}$
	spam	3	30	200	**precision**s= $\frac{200}{3+30+200}$
		recallu = $\frac{8}{8+5+3}$	**recall**n = $\frac{60}{10+60+30}$	**recall**s = $\frac{200}{1+50+200}$	

How to combine P/R from 3 classes to get one metric

Macroaveraging:

- compute the performance for each class, and then average over classes

Microaveraging:

- collect decisions for all classes into one confusion matrix
- compute precision and recall from that table.

Macroaveraging and Microaveraging

	urgent	normal	spam
urgent	8	10	1
normal	5	60	50
spam	3	30	200

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

$$\begin{array}{c}\text{microaverage}\\\text{precision}\end{array} = \frac{268}{268+99} = \mathbf{.73}$$

$$\begin{array}{c}\text{macroaverage}\\\text{precision}\end{array} = \frac{.42+.52+.86}{3} = \mathbf{.60}$$