I 202: Information Organization & Retrieval Fall 2025

Class 15: Text Classification

Today's Outline

Automating Classification

Classification Process

Evaluation

Naïve Bayes Classification

What is Text Classification For?

Topic Classification

MEDLINE Article

?

MeSH Subject Category Hierarchy

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

Sentiment Analysis

By John Neal

This review is from: Accoutrements Horse Head Mask (Toy)

When I turned State's Witness, they didn't have enough money to put me in the Witness Protection Program, so they bought me this mask and told me to go to certain places to meet [illegible]. Since then I've lived my life in peace and safety knowing that my old identity is forever obscured by this life-saving item.

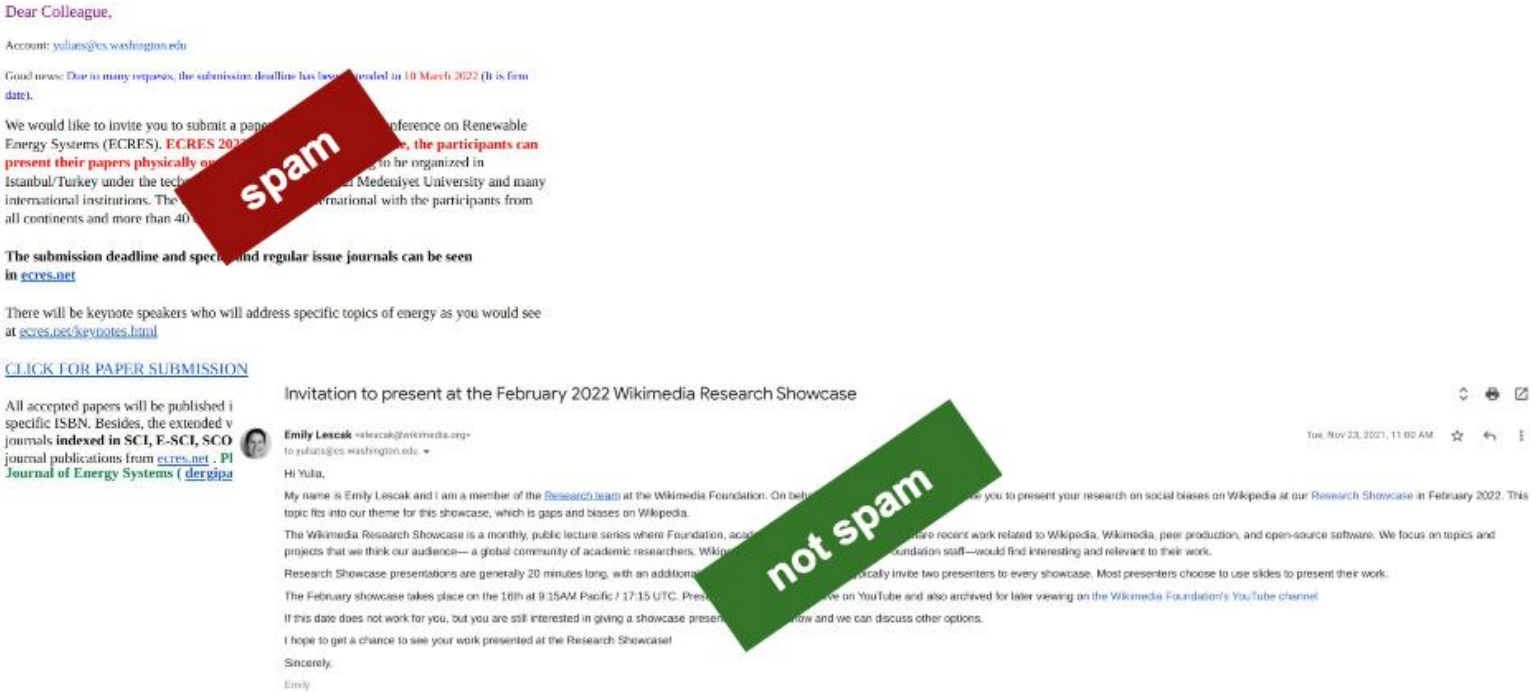
By Christine E. Torok

Verified Purchase (What's this?)

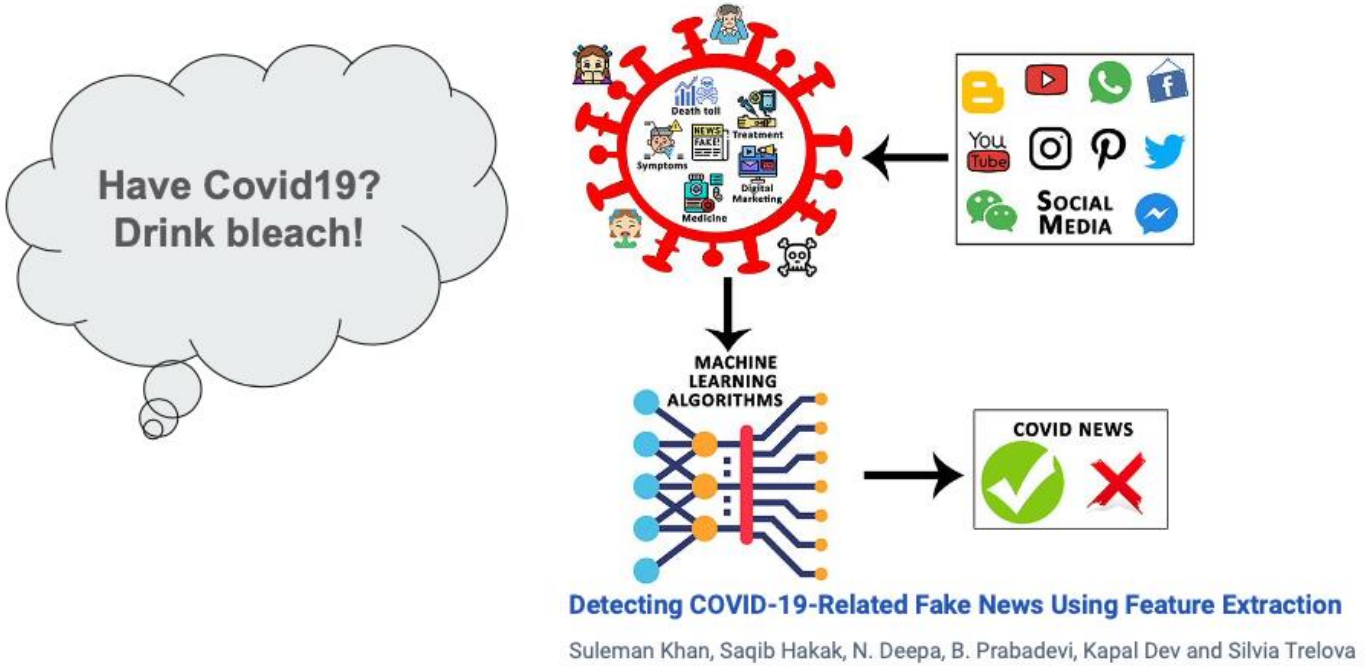
First of all, for taste I would rate these a 5. So good. Soft, true-to-taste fruit flavors like the sugar variety...I was a happy camper.

BUT (or should I say BUTT), not long after eating about 20 of these all hell broke loose. I had a gastrointestinal experience like nothing I've ever imagined. Cramps, sweating, bloating beyond my worst nightmare. I've had food poisoning from some bad shellfish and that was almost like a skip in the park compared to what was going on inside

Spam Classification



Fact Verification



Let's do Some Sentiment Analysis

Are these positive or negative movie reviews?

Still, this flick is fun and host to some truly excellent sequences.

Their computer-animated faces are very expressive.

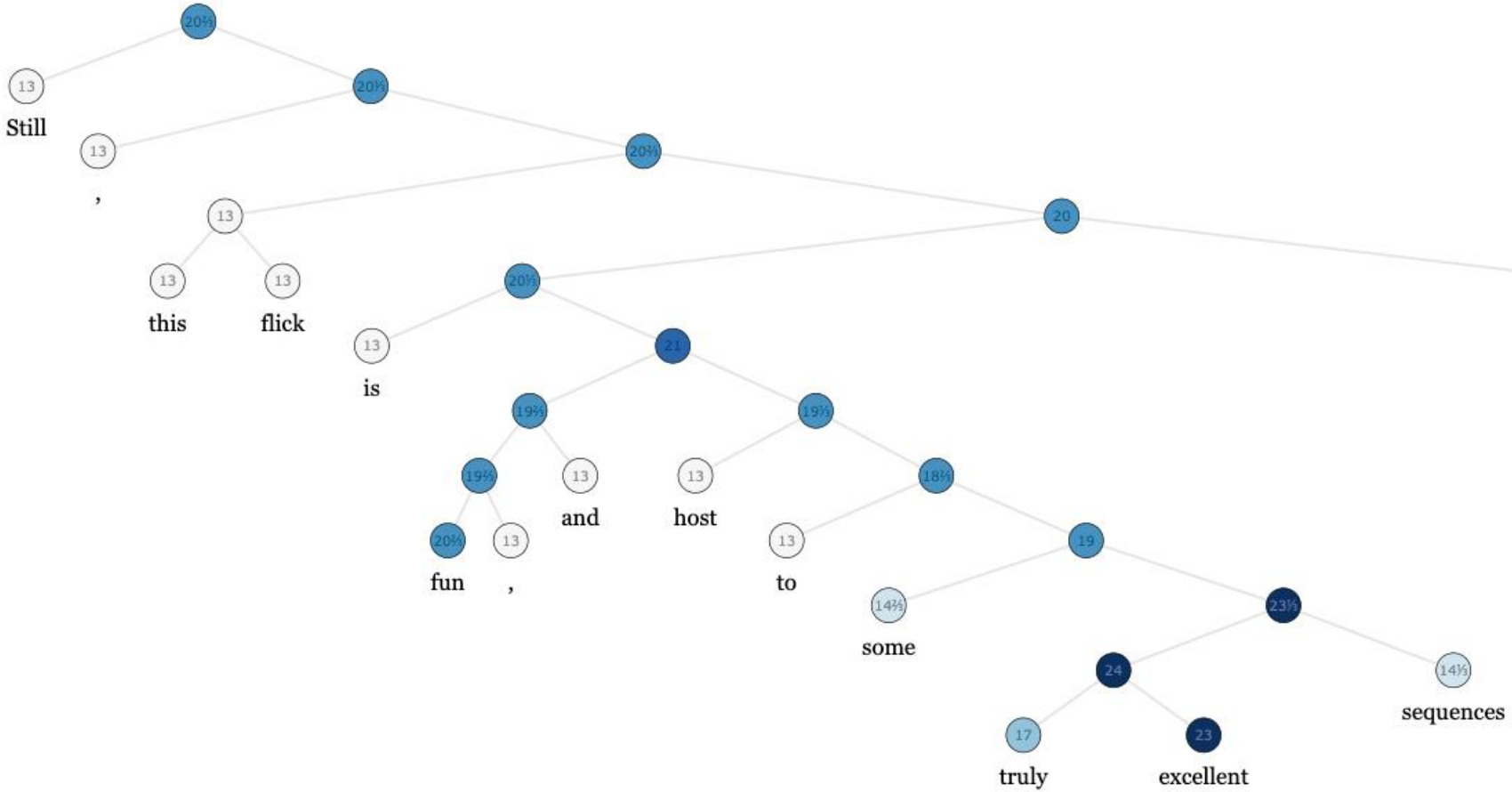
It's not life affirming -- it's vulgar and mean, but I liked it.

You walk out of The Good Girl with mixed emotions – disapproval of Justine with a tinge of understanding for her actions.

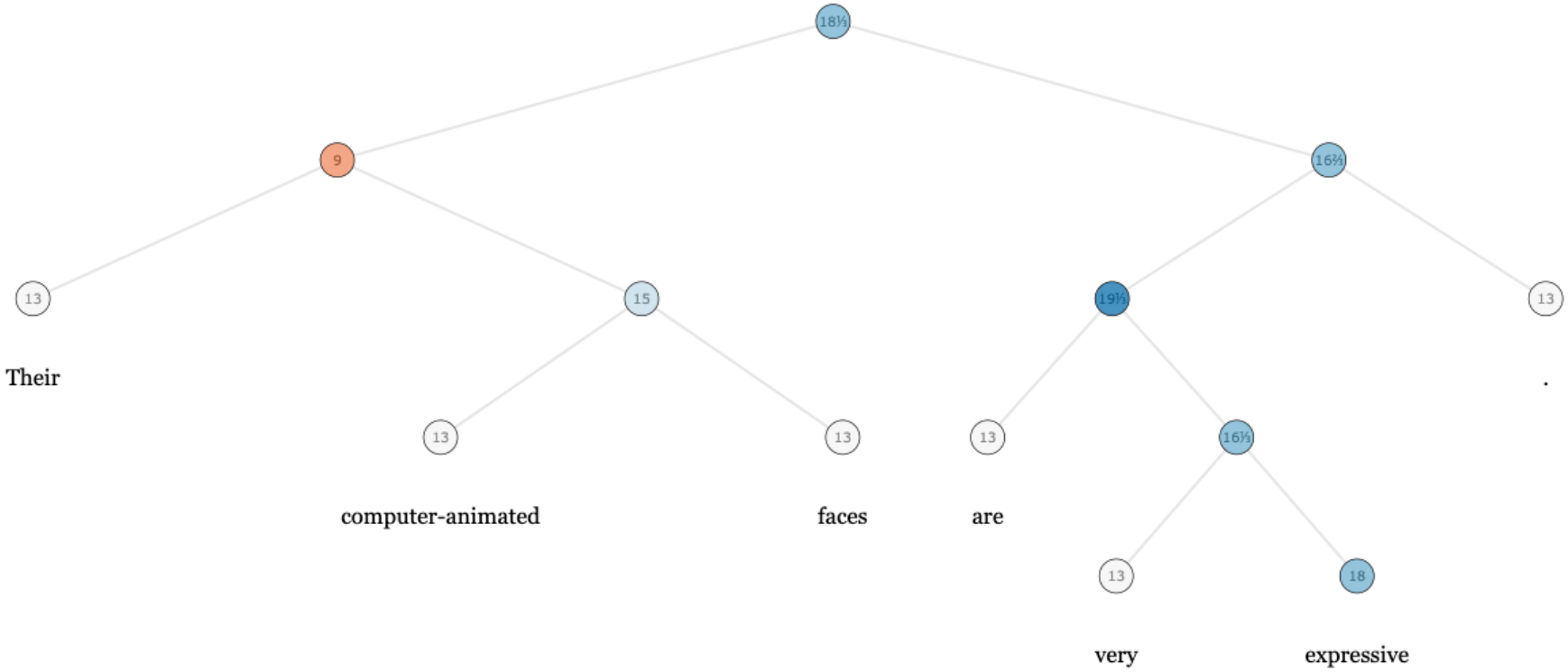
Coping with Sentiment Nuance

- Most sentiment prediction systems work just by looking at words in isolation, giving positive points for positive words and negative points for negative words and then summing up these points.
- Instead, we can build up a representation of whole sentences based on the sentence structure.
- We compute the sentiment based on how words compose the meaning of longer phrases.

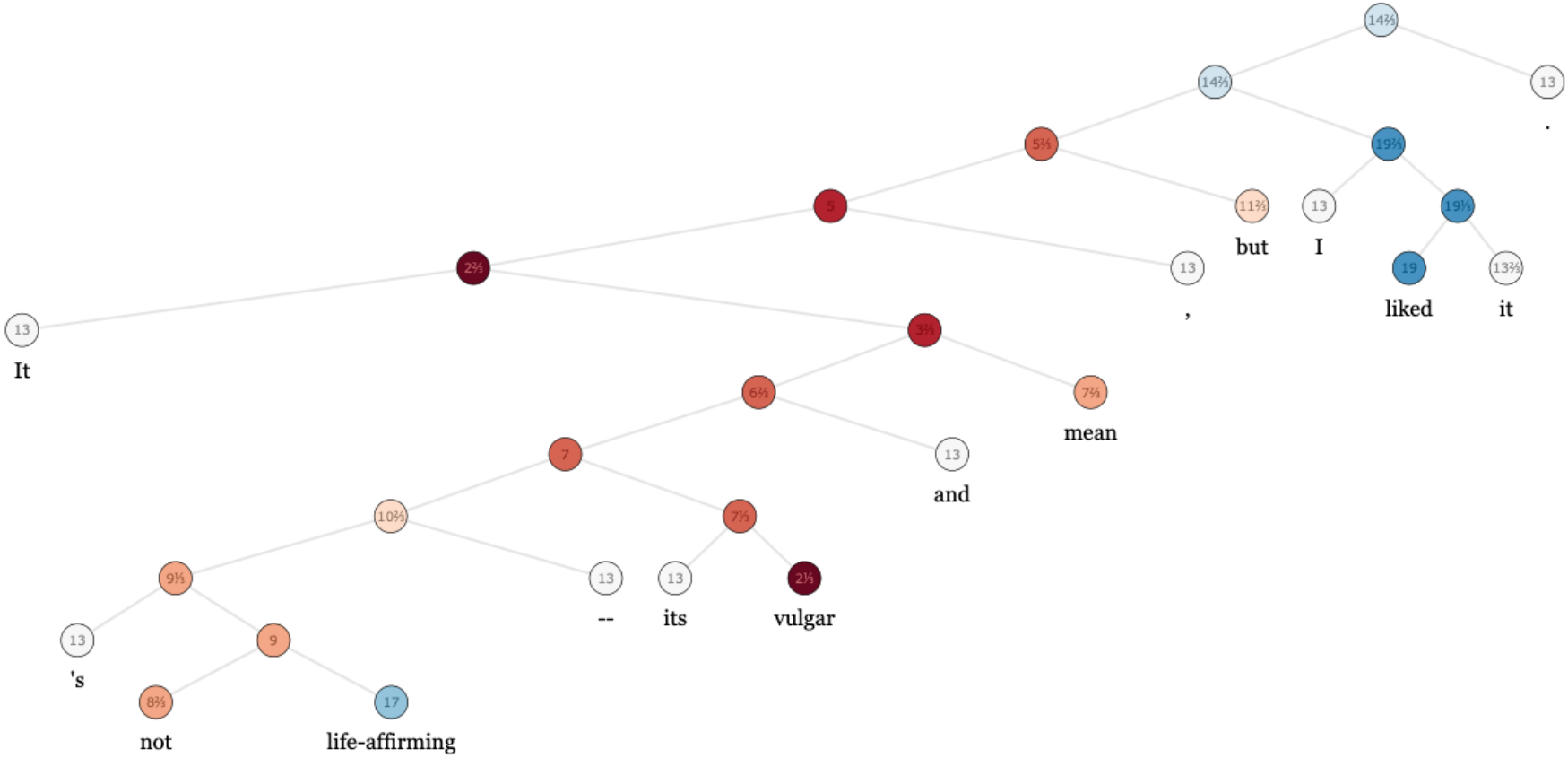
Socher et al, EMNLP 2013, https://nlp.stanford.edu/sentiment/treebank.html



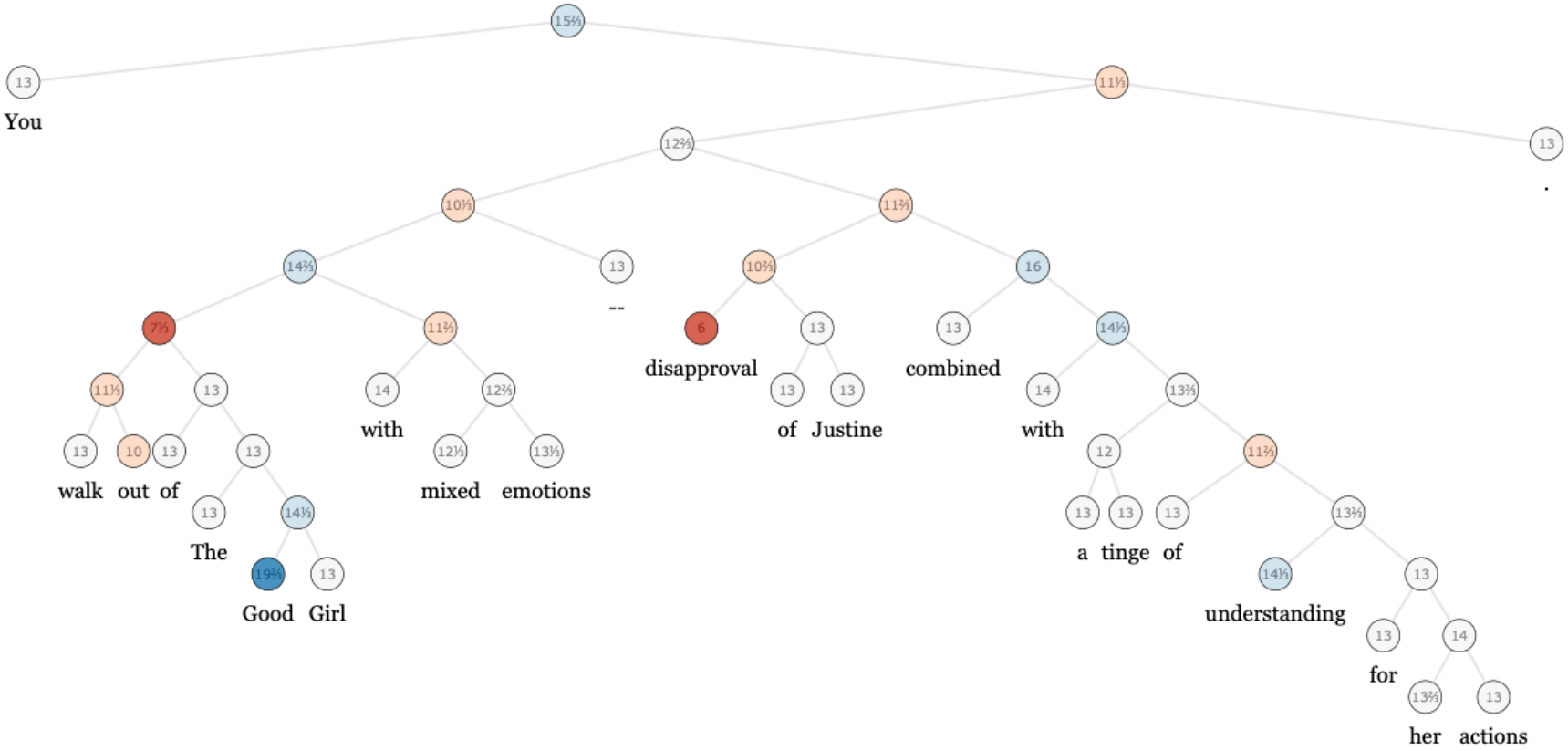
Socher et al, EMNLP 2013, https://nlp.stanford.edu/sentiment/treebank.html



Socher et al, EMNLP 2013, https://nlp.stanford.edu/sentiment/treebank.html



Socher et al, EMNLP 2013, https://nlp.stanford.edu/sentiment/treebank.html



Socher et al, EMNLP 2013, https://nlp.stanford.edu/sentiment/treebank.html

Why is Sentiment Analysis Difficult?

- Sentiment is a measure of a speaker's private state, which is unobservable.
- Sometimes words are a good indicator of sentiment (love, amazing, hate, terrible); many times it requires deep world + contextual knowledge

> "*Valentine's Day* is being marketed as a Date Movie. I think it's more of a First-Date Movie. If your date likes it, do not date that person again. And if you like it, there may not be a second date."
>
> Roger Ebert, Valentine's Day

WHAT IS AUTOMATIC CLASSIFICATION?

- Uses a *dataset* with *labeled classes* to learn which items belong to which classes

- Learns f(x) = y, where x is features of an item and y is the class it belongs to

- Ex: x = movie review, y = +/-

CLASSIFICATION

A mapping *h* from input data x (drawn from instance space $\mathcal{X}$) to a label (or labels) y from some enumerable output space $\mathcal{Y}$

$\mathcal{X}$ = set of all documents

$\mathcal{Y}$ = {english, mandarin, greek, …}

x = a single document

y = ancient greek

TEXT CATEGORIZATION PROBLEMS

task	x	y
language ID	text	{english, mandarin, greek, …}
spam classification	email	{spam, not spam}
authorship attribution	text	{jk rowling, james joyce, …}
genre classification	novel	{detective, romance, gothic, …}
sentiment analysis	text	{positive, negative, neutral, mixed}

CLASSIFICATION

$h(x) = y$

$h(\text{μῆνιν ἄειδε θεὰ}) = \text{ancient grc}$

CLASSIFICATION

Let h(x) be the “true” mapping. We never know it. How do we find the best $\hat{h}(x)$ to approximate it?

One option: rule based

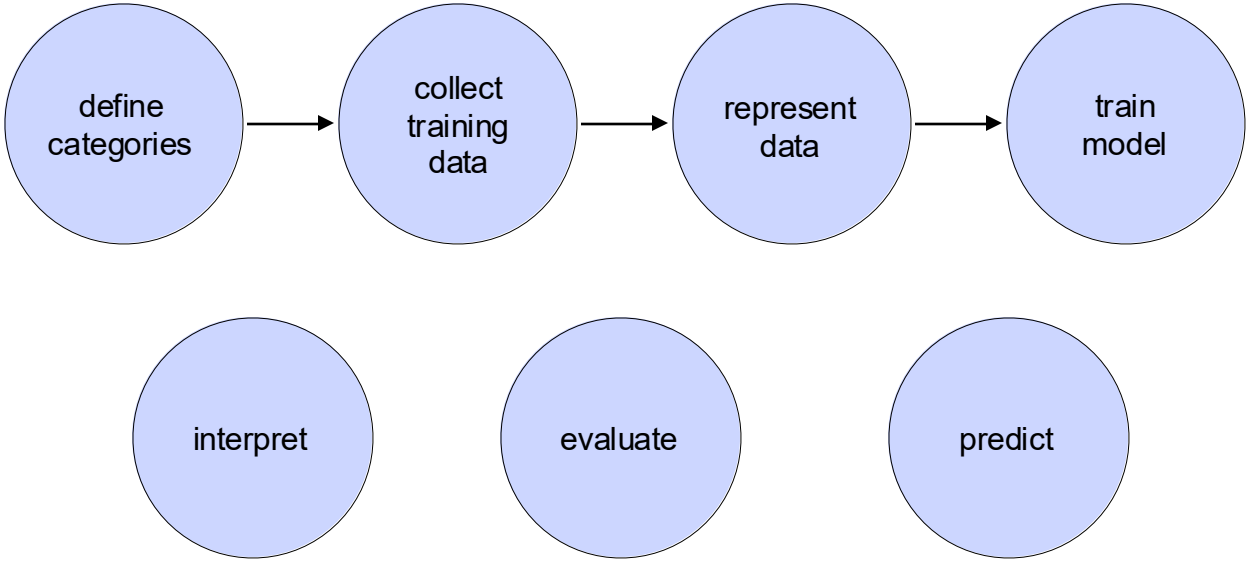
if x has characters in unicode point range 0370-03FF:
$\hat{h}(x)$ = greek

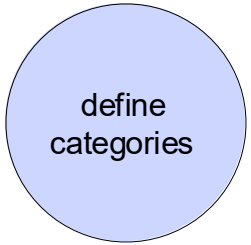
CLASSIFICATION

Supervised learning

Given training data in the form of <x, y> pairs, learn $\hat{h}(x)$

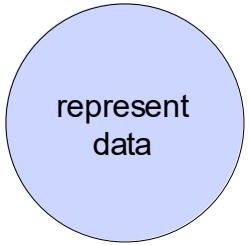
PROCEDURE


define categories
collect training data
represent data
train model
interpret
evaluate
predict



☞ categorization

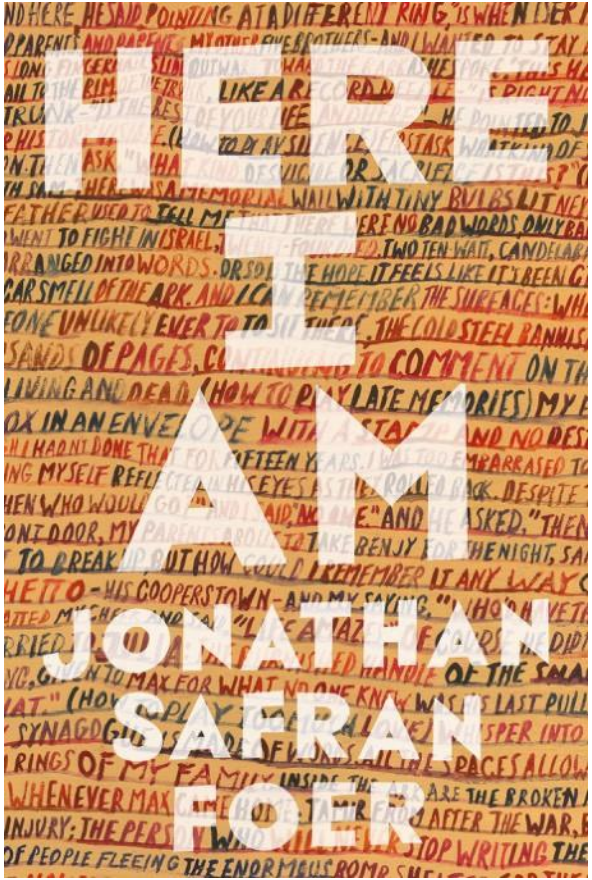
- Crucial step, since everything you learn to predict is only predictable with respect to these categories.
- Your categorization may reflect institutional, individual, or cultural categories is always biased.
- Important thing is to be aware of the source of that bias and its consequences downstream



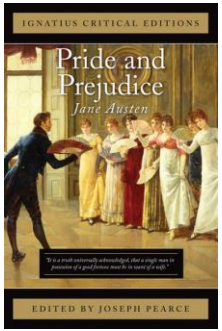
- Decide on what descriptions of the resource you want the algorithm to have access to.

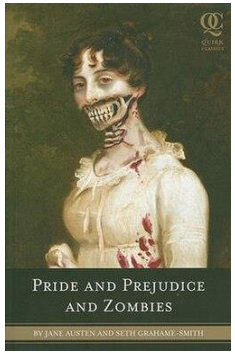
- You're critically not deciding what's important at this stage (the algorithms often determine this), but only rather what information is allowed to be learned to be important or not

author: foer	TRUE
author: austen	FALSE
pub year	2016
height (inches)	9.2
weight (pounds)	2
contain: the	TRUE
contains: zombies	FALSE
amazon rank @1 month	159



collect training data





author=foer	author=austen	pubdate	“zombies"	fiction
TRUE	FALSE	2016	FALSE	TRUE
TRUE	TRUE	1816	FALSE	TRUE
FALSE	FALSE	2016	FALSE	FALSE
FALSE	FALSE	2011	TRUE	TRUE

☞ resource selection

- Resources paired with their categories, as a result of:
    - human classification
    - being found nature

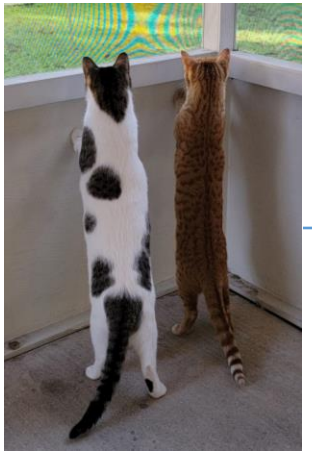
EXERCISE:

- You want to train a model to predict whether there is a dog in an Instagram image.
- 1. How would you collect data using code? The dataset should include (Image, Label) pairs.
- 2. If you could incorporate human-labeling, how would your data collection process change?

Image

Label

Yes



No

DATA COLLECTION

- We can collect data:
- (1) automatically via existing sources (e.g. via a web scraper + hashtags)
- (2) with the aid of human labeling

- Sometimes we can combine these two approaches, e.g. having humans confirm or correct automatically-collected examples.

- How much data to collect depends on your classifier! Range from dozens to millions of examples.

Uses

Computational classification serves two primary uses:

Prediction: automatically assign categories to new data points

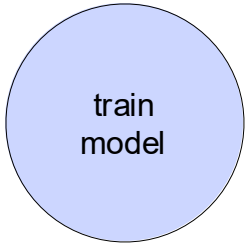
Insight: understanding what aspects of resource description are most informative for determining category membership

PREDICTION

- Replacement for expensive human classification, especially for repetitive tasks (e.g., mail sorting)
- Like all classifications, provide structure to support other interactions (e.g., dewey decimal)

INSIGHT

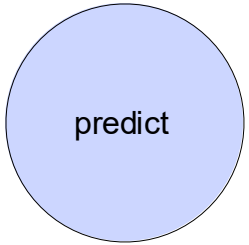
- Understanding the strongest indicators for category membership.
- For human classification, the principles for defining categories (enumeration, properties, similarity, family resemblance, etc.) are embodied in the classifications that use these principles
- Can lead to challenging the initial categorization system.



THERE ARE MANY METHODS IN MACHINE LEARNING/DATA SCIENCE TO PERFORM CLASSIFICATION

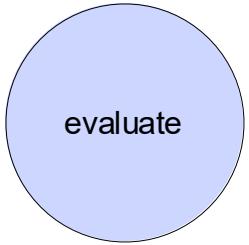
-
  - Decision trees (e.g., random forests)
  - Probabilistic models (e.g., Naive Bayes)
  - Nearest neighbor similarity (KNN)
  - Neural models (RNNs, LSTMs, Transformers, LLMs)

- All define a mapping from some input representation x to a category label y

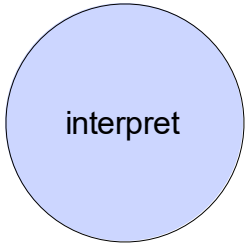


☞ classification

- Use that trained model to make predictions about the category membership of new data points.



- Assess the accuracy of the trained model by comparing the predictions it makes to some notion of “truth” for those same data points
- Cross-validation: train a model on some fraction of the data, and evaluate its performance on the remaining data.



- Understand what the model is learning about the data
- Some methods are more amenable to interpretation than others (very much method-dependent)

CLASSIFICATION

Supervised learning

Given training data in the form of <x, y> pairs, learn $\hat{h}(x)$

x	y
loved it!	positive
terrible movie	negative
not too shabby	positive

Representation for Sentiment Analysis

- Only positive/negative words in sentiment dictionaries (e.g., MPQA)
- Only words in isolation (bag of words)
- Conjunctions of words (sequential, skip ngrams, other non-linear combinations)
- Higher-order linguistic structure (e.g., syntax)

Bag of Words Representation

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
…	…

Jurafsky & Martin slides

Bag of words

Representation of text only as the counts of words that it contains

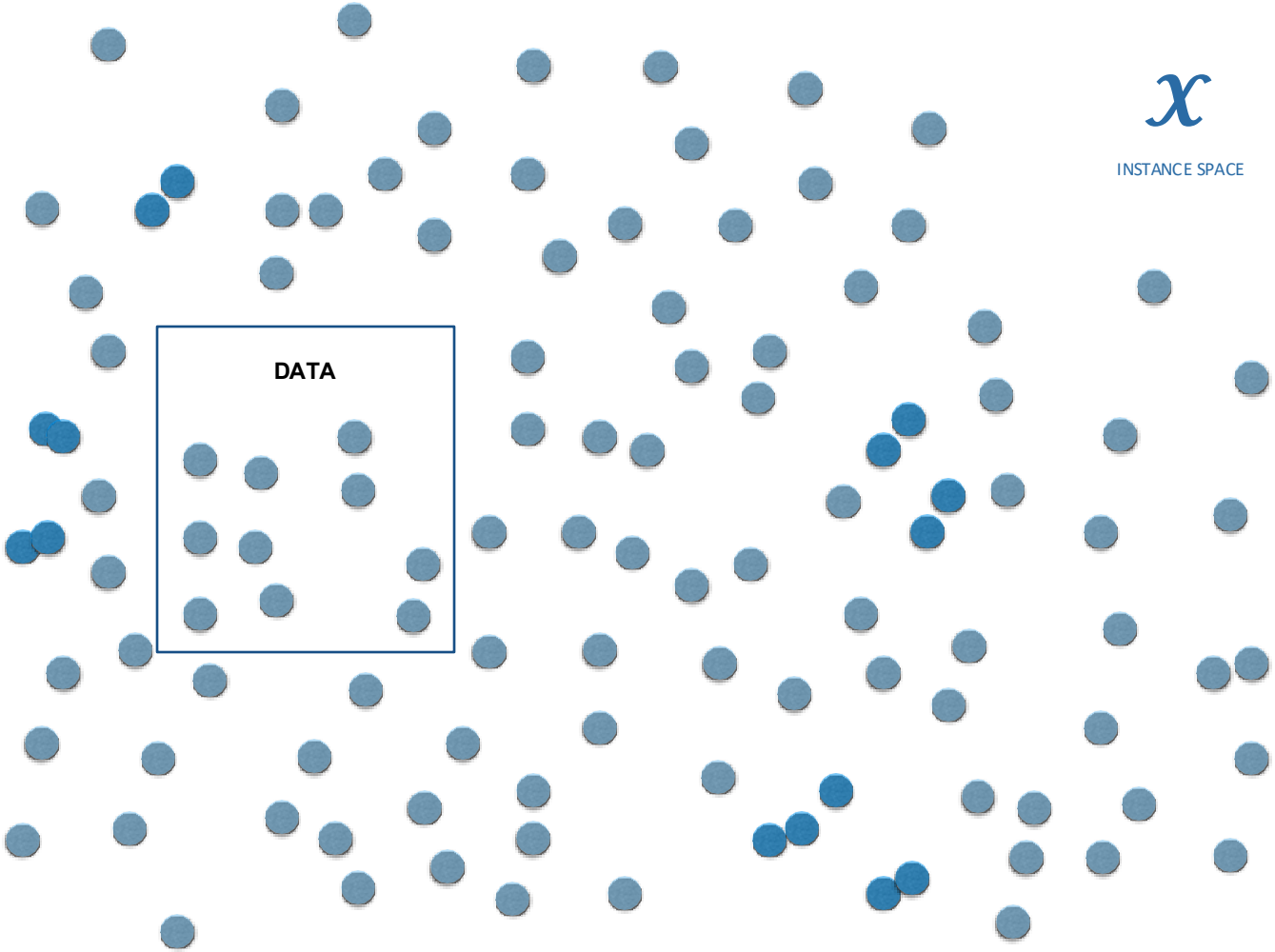
	Apocalypse now	North
the	1	1
of	0	0
hate	0	9
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
…		

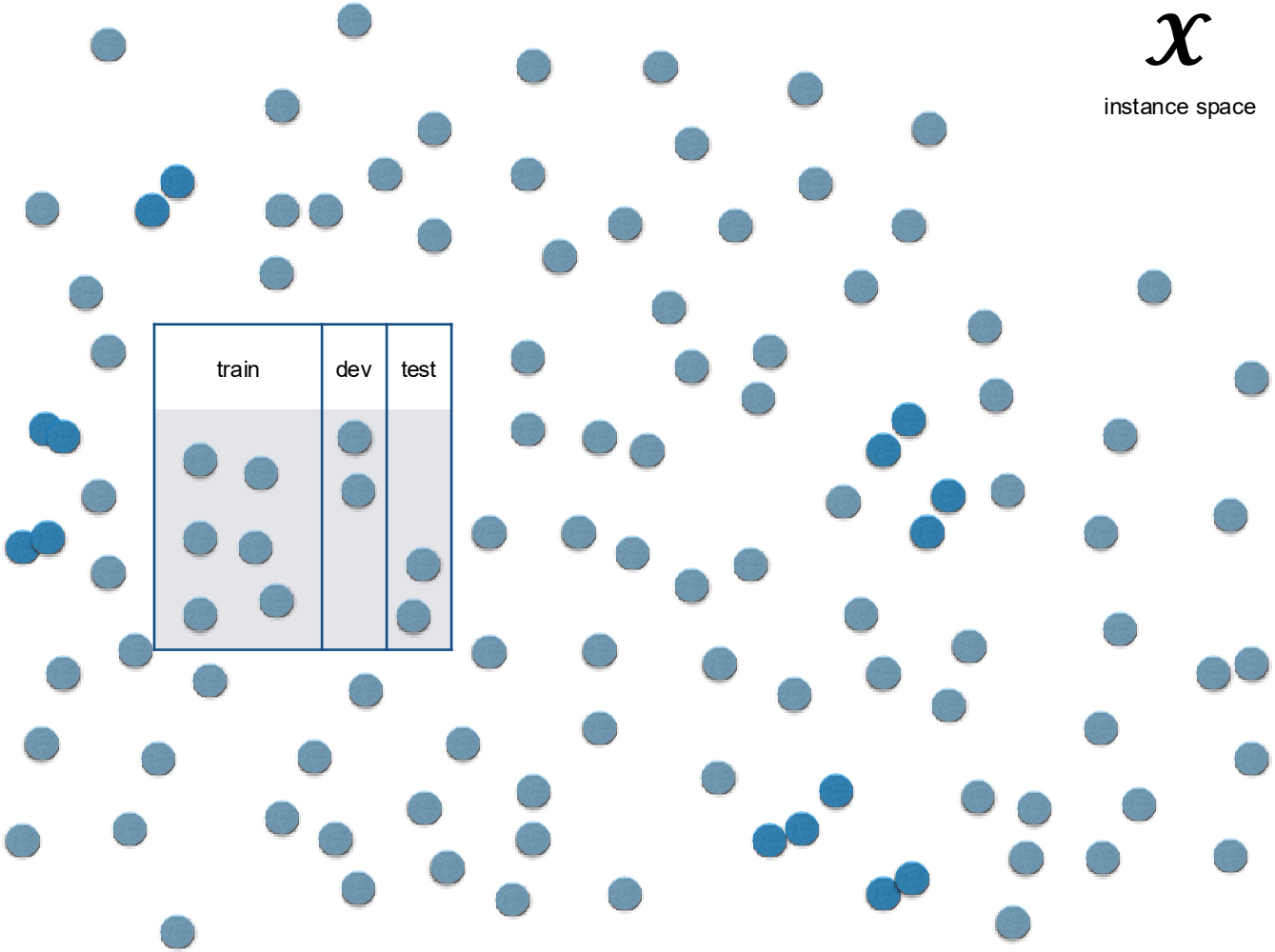
Other Types of Features

- Syntactic features
- - Part-of-speech information
- - Subject/Verb/Object information
- Length of Document
- Punctuation used

EVALUATION

- For all supervised problems, it's important to understand how well your model is performing
- What we try to estimate is how well you will perform in the future, on new data also drawn from $\mathcal{X}$
- Trouble arises when the training data <x, y> you have does not characterize the full instance space.
    - n is small
    - sampling bias in the selection of <x, y>
    - x is dependent on time
    - y is dependent on time (concept drift)


$\mathcal{X}$
INSTANCE SPACE
DATA


$\mathcal{X}$
instance space
train
dev
test

EXPERIMENT DESIGN

	training	development	testing
size	80%	10%	10%
purpose	training models	model selection	evaluation; never look at it until the very end

MAJORITY CLASS BASELINE

- Pick the label that occurs the most frequently in the training data. (Don't count the test data!)
- Predict that label for every data point in the test data.

Classification Model: Naive Bayes

- Simple model, but works well in a lot of instances!
- Relies on the probability of a certain class given features.
- Based on probability and Bayes theorem

Probability Concepts

- Individual event's probability: P(A)

  P(coin=heads) = 0.5

- Conditional probability: P(A|B)

  P(coin1=heads|coin2=tails)=0.5

  = P(coin1=heads), meaning these events are independent

  P(I take the bus to work | it is raining) = 0.9

  I'm more likely to take the bus when it rains, so these events are not independent

Bayes Theorem Introduction

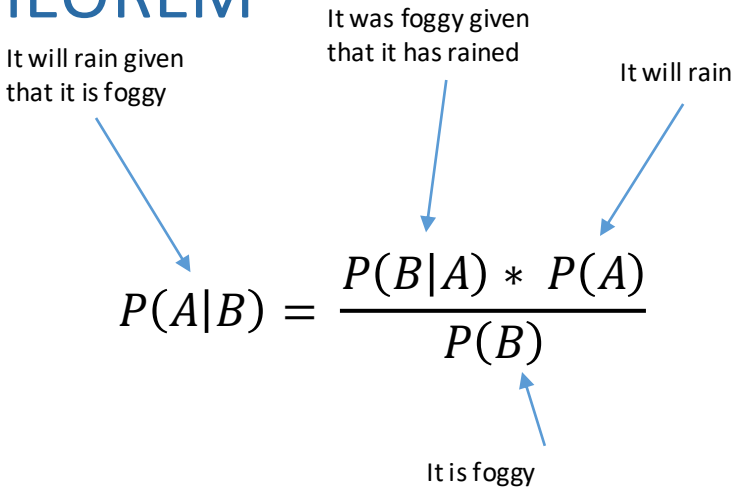
- How can we calculate the probability of Event A given that we have observed Event B?

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

- A lot of the time, we don't have direct access to P(A|B), so we can use Bayes theorem to calculate this!

Bayes Theorem

- You observe that the sky is foggy this morning. In Berkeley, it rains 5% of all days. You know that rainy days start off with foggy mornings 20% of the time. You also know that it is foggy 25% of the time. What is the probability that it will rain today given that it is foggy?



$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

$$P(A|B) = \frac{0.20 * 0.05}{0.25}$$

$$P(A|B) = 0.04 = 4.0\%$$

Event A: It will rain

Event B: You observe that it is foggy

NAÏVE BAYES CLASSIFIER

- Assumes we have a dataset with labeled features
- Generative classifier – assumes data is created by sampling class and then generating text of document.
- Leverages Bayes Rule to calculate probability of a specific class given the features present in the example.

NAÏVE BAYES CLASSIFIER

$y = class\ (+, -)$

$X = (x_1, x_2, \dots, x_n)$

For each document we classify, X is our featurized representation!

"Good movie" →

$x_1 = "good"$

$x_2 = "movie"$

Probability of the features given the class

Prior probability of each class

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

Prior probability of the features

We calculate the probability of a class (+/-)
given the features observed in the example.

NAÏVE BAYES CLASSIFIER

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

$y = class\ (+, -)$

$X = (x_1, x_2, \dots, x_n)$

$x_1 = "good"$

$x_2 = "movie"$

$$P(y|X) = \frac{P(x_1|y) * P(x_2|y) * \cdots * P(x_n|y) * P(y)}{P(x_1) * P(x_2) * \cdots * P(x_n)}$$

$$P(y|X) \propto P(x_1|y) * P(x_2|y) * \cdots * P(x_n|y) * P(y)$$

In practice, we drop the denominator!

$$P(y|X) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Want to find the most likely class

https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c?gi=1784c8032896

How do we train a Naïve Bayes Classifier?

- We can estimate these probabilities via our dataset!
- Let's revisit sentiment analysis with a *standard* bag-of-words featurization.

P("good" |+) = $\frac{\#\ of times\ \text{“}good\text{”}\ \text{appears in} + \text{docs}}{\#\ of words\ in\ all + docs}$

$\frac{215}{550} = 0.39$

P("good" |-) = $\frac{\#\ of times\ \text{“}good\text{”}\ \text{appears in} - \text{docs}}{\#\ of words\ in\ all - docs}$

$\frac{75}{750} = 0.10$

For our movie review dataset, 350 reviews are + and 650 reviews are -.

P(+) = 350 / 1000 = 0.35

P(-) = 650 / 1000 = 0.65

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Predicting a Class with Naïve Bayes

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

y = +

$$P(y = +) \prod_{i=1}^{n} P(x_i|y = +)$$

$$0.35 * 0.39 * 0.3$$

$$0.041$$

Good movie

y = -

$$P(y = -) \prod_{i=1}^{n} P(x_i|y = -)$$

$$0.65 * 0.10 * 0.3$$

$$0.020$$

- Our Naïve Bayes Classifier would predict this document belongs to the + class

LIMITATIONS OF NAÏVE BAYES

- Independence assumption – assumes each feature x is independent of all other features.
  - *This is not always true!*
  - *Likelihood of a document containing "good" increases once we've observed this word*
- Sparse data problem – what if the word "fantastic" never appears in our training dataset?

Midterm Instructions
Due Monday at 8pm; start no later than 5pm!

- This is an open-assigned-readings, open-note exam. However, you may not talk to other people about it, either in person or online. Since students are taking the exam a different times, please do not discuss it with others after you complete it until the instructors say it is ok to do so.
- You will have a maximum of 3 hours from starting to complete and submit this midterm. We suggest you type your answer for each question into a word processor and then copy-and-paste the answer into the quiz question field.
- Use of any kind of AI is **not permitted**. This includes grammar checkers. Please turn off any automated suggestions before you start, if you use a word processor.
- In case of difficulties with bCourses, please keep a copy of your answers and, should you not be able to submit online, send this copy to the instructional team in an email time-stamped within 3 hours of starting. Note that the due-date (8pm on Monday) does not take your starting time into account, so if you start after 5pm Sunday you won't have the full 3 hours.
- If you have any questions or run into any issues, please email the instructional team. As you may start the midterm at any time in the 3-hour period, we may not be continuously available to respond to your messages. Therefore, if you have started, do not wait for a response but continue working on the exam so you can complete it in the allotted time