CS769 Advanced NLP

Text Classification

Junjie Hu

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Slides adapted from Graham, Luke
https://junjiehu.github.io/cs769-fall23/

Goals Today

- Generative Text Classifier (**Naive Bayes**)
- Discriminative Text Classifier (**Logistic Regression**)
- Classification Evaluation
- Dataset Curation

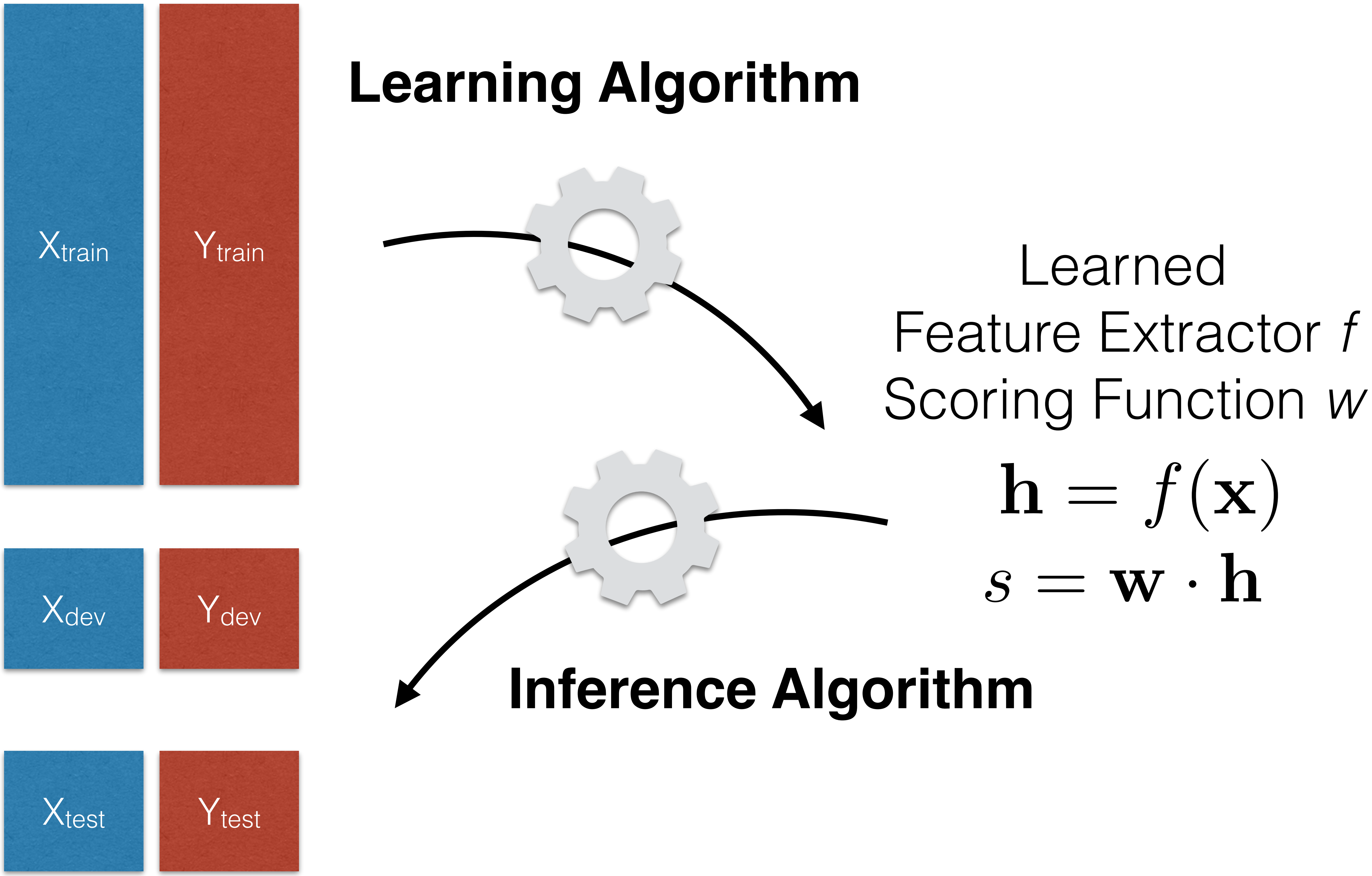
A General Framework for NLP Systems

- Formally, create a function to map an input *X (language)* into an output *Y*. Examples:

Input *X*	Output *Y*	Task
Text	Text in Other Language	Translation
Text	Response	Dialog
Text	Label	Text Classification
Text	Linguistic Structure	Language Analysis

- To create such a system, we can use
  - Manual creation of rules
  - Machine learning from paired data <*X*, *Y*>

Reminder: Machine Learning



X_{train} Y_{train}

X_{dev} Y_{dev}

X_{test} Y_{test}

Learning Algorithm

Learned
Feature Extractor *f*
Scoring Function *w*

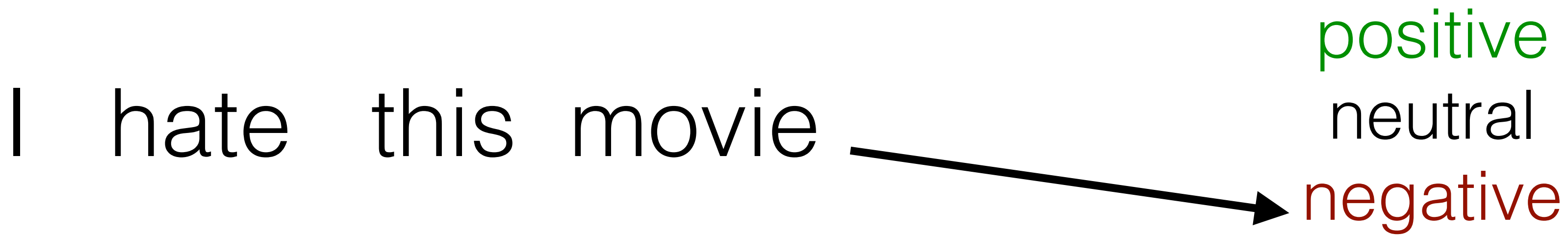
$$\mathbf{h} = f(\mathbf{x})$$

$$s = \mathbf{w} \cdot \mathbf{h}$$

Inference Algorithm

Text Classification

- Classify sentences according to various traits
- Topic, sentiment, subjectivity/objectivity, etc.



I hate this movie

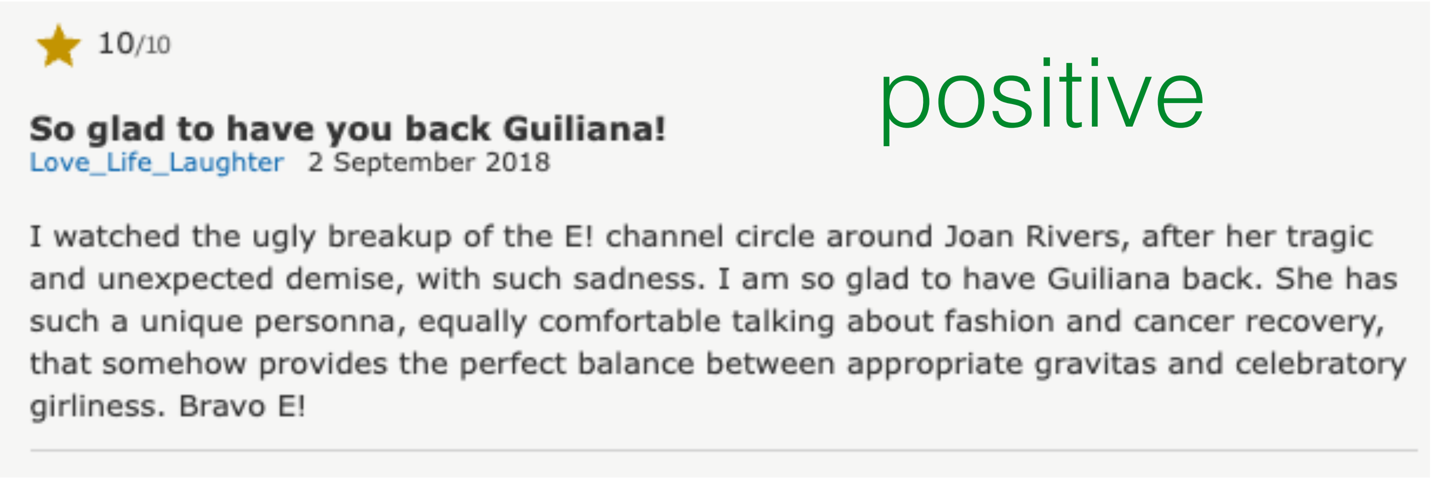
positive

neutral

negative

Example: Movie Review

- Predict sentiment from IMDB movie review: {positive, neural, negative}

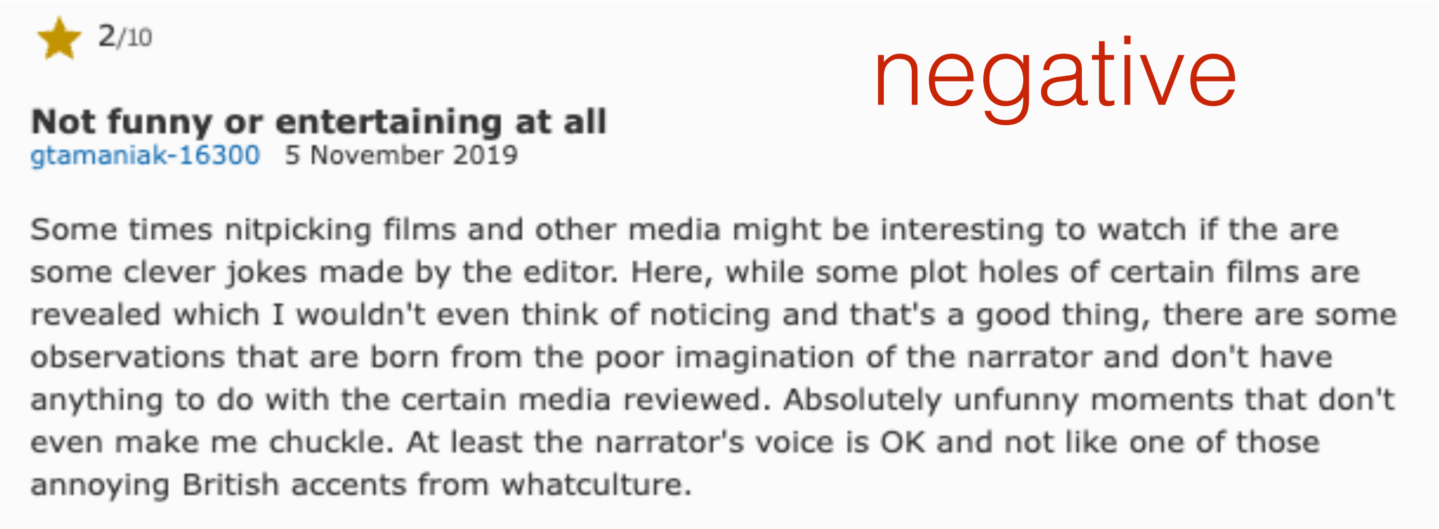


10/10

So glad to have you back Guiliana!
Love_Life_Laughter 2 September 2018

I watched the ugly breakup of the E! channel circle around Joan Rivers, after her tragic and unexpected demise, with such sadness. I am so glad to have Guiliana back. She has such a unique personna, equally comfortable talking about fashion and cancer recovery, that somehow provides the perfect balance between appropriate gravitas and celebratory girliness. Bravo E!

positive

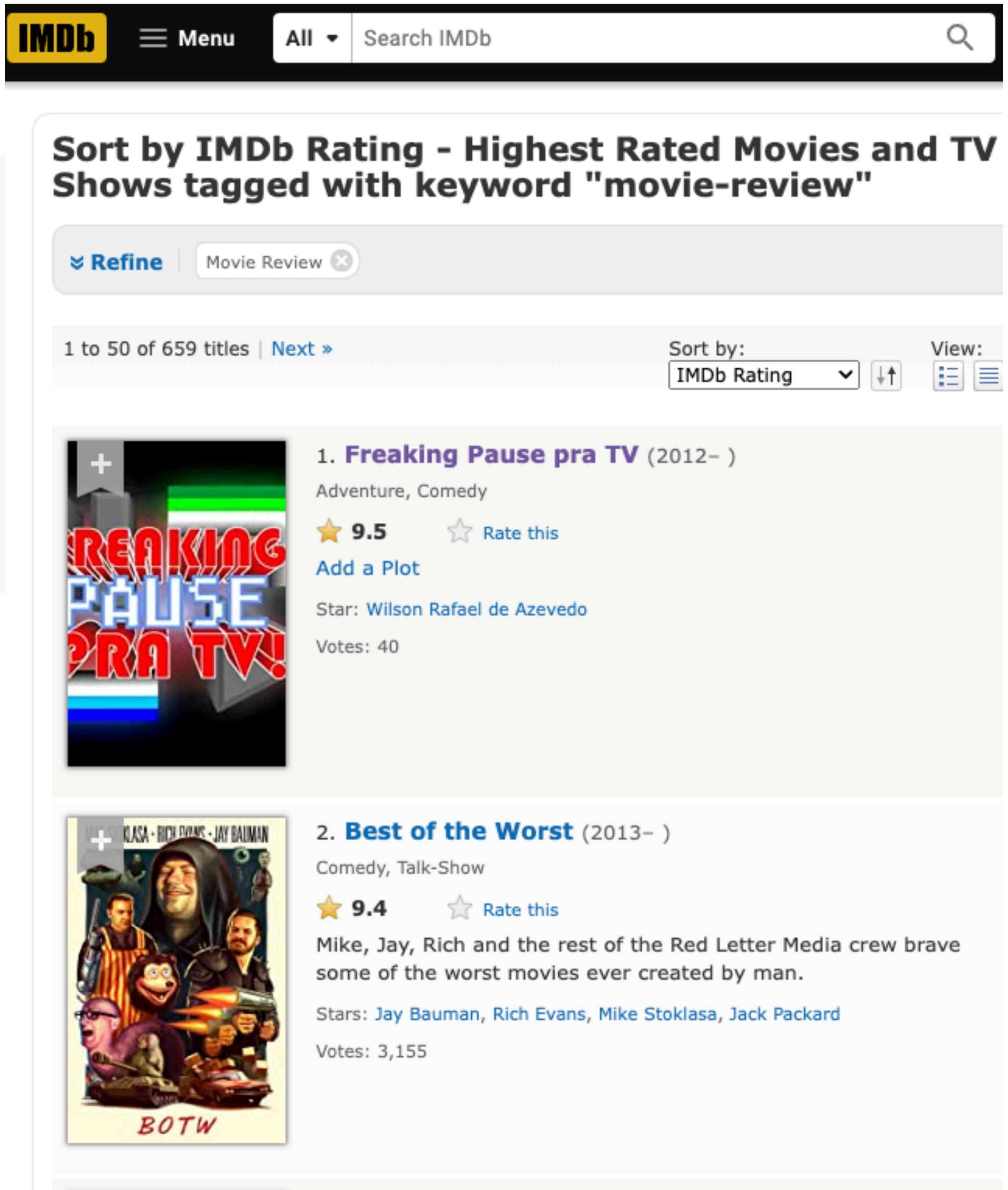


2/10

Not funny or entertaining at all
gtamaniak-16300 5 November 2019

Some times nitpicking films and other media might be interesting to watch if the are some clever jokes made by the editor. Here, while some plot holes of certain films are revealed which I wouldn't even think of noticing and that's a good thing, there are some observations that are born from the poor imagination of the narrator and don't have anything to do with the certain media reviewed. Absolutely unfunny moments that don't even make me chuckle. At least the narrator's voice is OK and not like one of those annoying British accents from whatculture.

negative



Sort by IMDb Rating - Highest Rated Movies and TV Shows tagged with keyword "movie-review"

Refine | Movie Review

1 to 50 of 659 titles | Next »

Sort by: IMDb Rating

View:

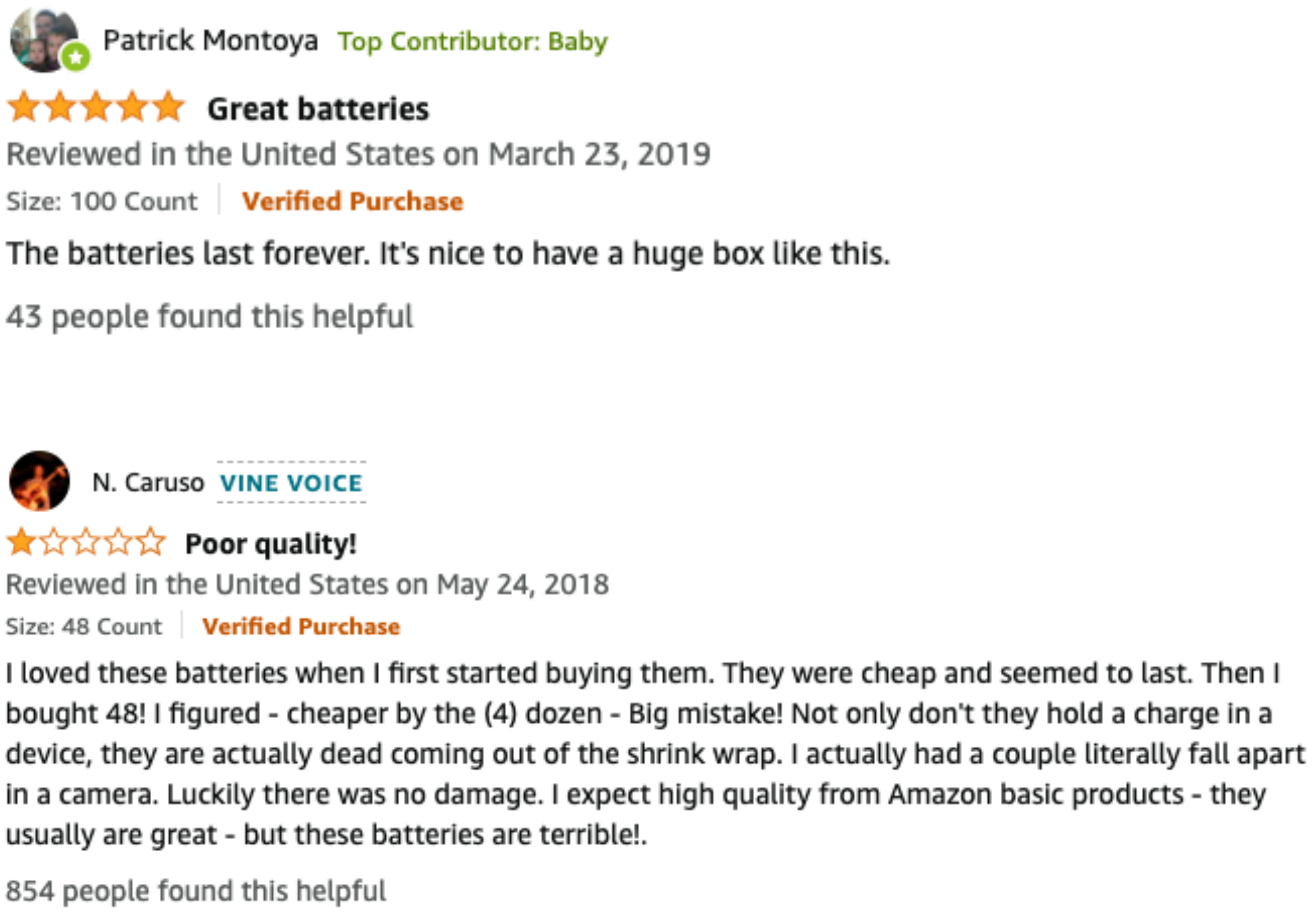
1. **Freaking Pause pra TV** (2012–)
Adventure, Comedy
9.5 Rate this
Add a Plot
Star: Wilson Rafael de Azevedo
Votes: 40

2. **Best of the Worst** (2013–)
Comedy, Talk-Show
9.4 Rate this
Mike, Jay, Rich and the rest of the Red Letter Media crew brave some of the worst movies ever created by man.
Stars: Jay Bauman, Rich Evans, Mike Stoklasa, Jack Packard
Votes: 3,155

Code: https://colab.research.google.com/github/bentrevett/pytorch-sentiment-analysis/blob/master/1%20-%20Simple%20Sentiment%20Analysis.ipynb
Dataset: http://ai.stanford.edu/~amaas/data/sentiment/

Example: Customer Rating

- Predict Amazon customer rating: {1, 2, 3, 4, 5}



Patrick Montoya Top Contributor: Baby

★★★★★ **Great batteries**

Reviewed in the United States on March 23, 2019

Size: 100 Count | Verified Purchase

The batteries last forever. It's nice to have a huge box like this.

43 people found this helpful

N. Caruso VINE VOICE

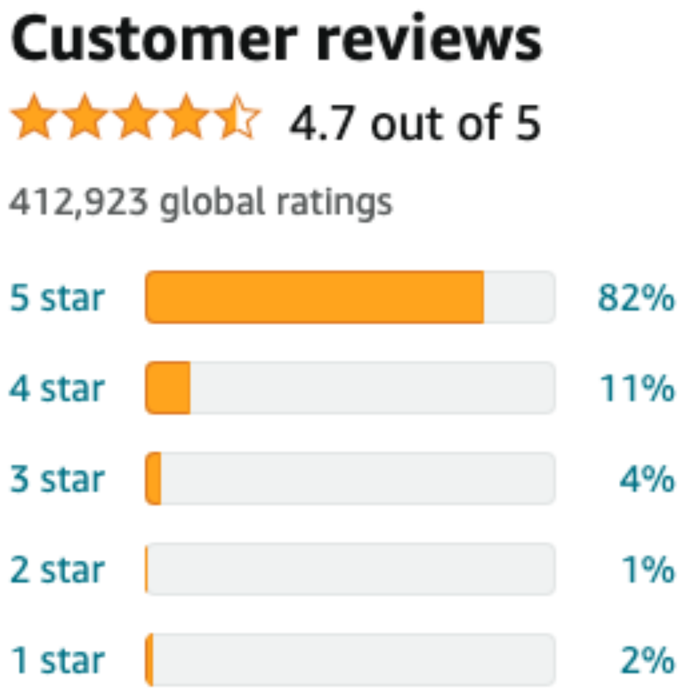
★☆☆☆☆ **Poor quality!**

Reviewed in the United States on May 24, 2018

Size: 48 Count | Verified Purchase

I loved these batteries when I first started buying them. They were cheap and seemed to last. Then I bought 48! I figured - cheaper by the (4) dozen - Big mistake! Not only don't they hold a charge in a device, they are actually dead coming out of the shrink wrap. I actually had a couple literally fall apart in a camera. Luckily there was no damage. I expect high quality from Amazon basic products - they usually are great - but these batteries are terrible!.

854 people found this helpful



Customer reviews

★★★★☆ 4.7 out of 5

412,923 global ratings

5 star	82%
4 star	11%
3 star	4%
2 star	1%
1 star	2%

Dataset: https://s3.amazonaws.com/amazon-reviews-pds/readme.html

Generative and Discriminative Models

Generative vs. Discriminative Models

- **Generative model:** a model that calculates the probability of the input data itself

$P(X)$ $P(X, Y)$

stand-alone *joint*

- **Discriminative model:** a model that calculates the probability of a latent trait given the data

$P(Y \mid X)$

conditional

Application to Text Classification

- **Generative text classification:** Learn a model of the joint P(*X*, *y*), and find

$$\hat{y} = \arg\max_{y} P(X, y)$$

- **Discriminative text classification:** Learn a model of the conditional P(*y* | *X*), and find

$$\hat{y} = \arg\max_{y} P(y|X)$$

Generative Text Classification

Language Modeling: Calculating the Probability of a Sentence

$$P(X) = \prod_{i=1}^{I} P(\underbrace{x_i}_{\text{Next Word}} \mid \underbrace{x_1, \ldots, x_{i-1}}_{\text{Context}})$$

The big problem: How do we predict

$$P(x_i \mid x_1, \ldots, x_{i-1})$$

?!?!

The Simplest Language Model: Count-based Unigram Models

- We'll cover more complicated models next class, so let's choose the simplest one for now!

- **Independence assumption:** $P(x_i|x_1,\ldots,x_{i-1}) \approx P(x_i)$

- **Count-based maximum-likelihood estimation:**

$$P_{\mathrm{MLE}}(x_i) = \frac{c_{\mathrm{train}}(x_i)}{\sum_{\tilde{x}} c_{\mathrm{train}}(\tilde{x})}$$

Handling Unknown Words

- If a word doesn't exist in training data becomes zero! $\frac{c_{\text{train}}(x_i)}{\sum_{\tilde{x}} c_{\text{train}}(\tilde{x})}$
- Need a distribution that assigns some probability to *all* words!
  - **Character/subword-based model:** Calculate the probability of a word based on its spelling.
  - **Uniform distribution:** Approximate by assuming fixed size vocabulary and defining: $P_{\text{unk}}(x_i) = 1/N_{\text{vocab}}$
- **Interpolate: Combine two probabilities w/ coefficient** λ_{unk}:

$$P(x_i) = (1 - \lambda_{\text{unk}}) * P_{\text{MLE}}(x_i) + \lambda_{\text{unk}} * P_{\text{unk}}(x_i)$$

Parameterizing in Log Space

- Multiplication of probabilities can be re-expressed as addition of log probabilities

$$P(X) = \prod_{i=1}^{|X|} P(x_i) \quad \longrightarrow \quad \log P(X) = \sum_{i=1}^{|X|} \log P(x_i)$$

- **Why?:** numerical stability, other conveniences

- We will define these parameters θ_{Xi}

$$\theta_{x_i} := \log P(x_i)$$

Generative Text Classifier

- Joint probability can be based on the following decomposition
- **Generative model**: Pick a **class**, then generate a input X using a **conditional language model** for that class

$$P(X, y) = P(X|y)P(y)$$

class-conditional LM, trained
on data associated with that class
(likelihood)

class prior probability
(bias)

$$P(y) = \frac{c(y)}{\sum_{\tilde{y}} c(\tilde{y})}$$

Naive-Bayes Models

- Naive-Bayes assumption: all words are **independent give the class label**

$$P(X, y) = P(X|y)P(y) = \left(\prod_{x_i} P(x_i|y)\right) P(y)$$

- Compared to a unigram language model

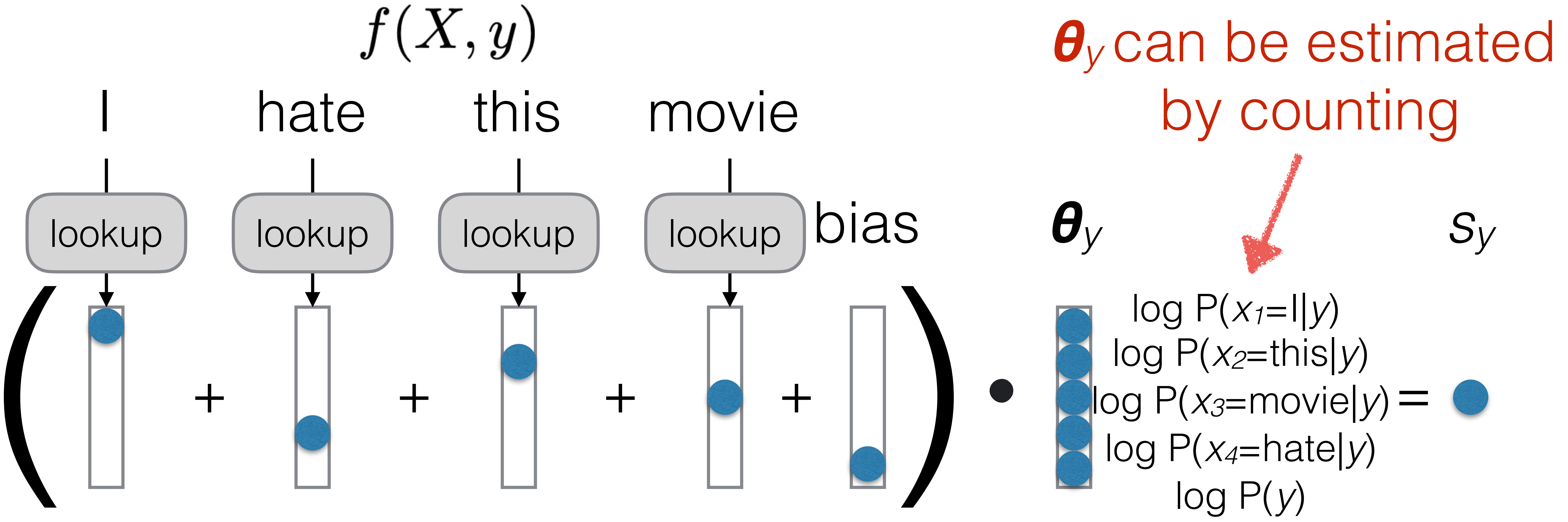
$$P(X) = \prod_{x_i} P(x_i)$$

- During training, estimate the likelihood and class probability parameterized by $\theta = \{\theta_1, \theta_2\}$, maximize the log of joint probability:

$$\log P_\theta(X, y) = \sum_{x_i} \log P_{\theta_1}(x_i|y) + \log P_{\theta_2}(y)$$

$$\theta^* = \arg\max_\theta \sum_{(X,y)\in\mathcal{D}_{\text{train}}} \log P_\theta(X, y)$$

Bag-of-words Naive Bayes Classifier



$\boldsymbol{\theta}_y$ can be estimated by counting

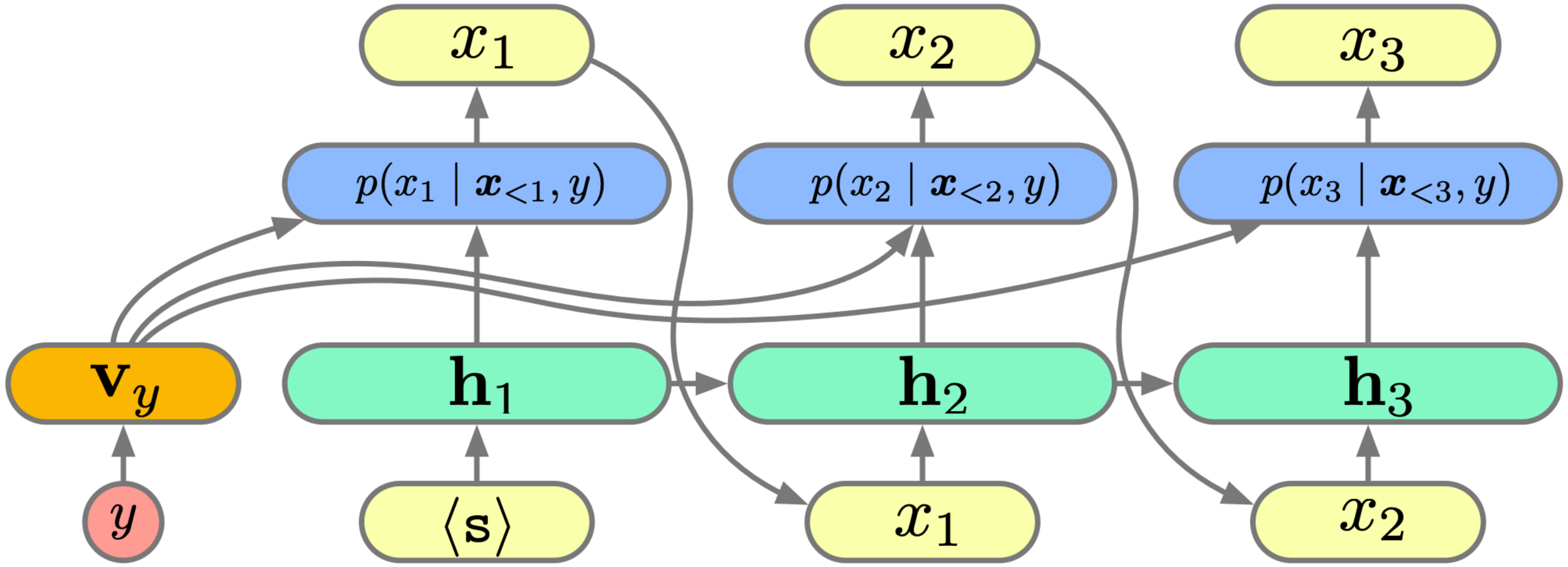
$$S_y = f(X,y) \cdot \theta_y = \log P(y) + \sum_{x_i} \log P(x_i|y)$$

$$y^* = \arg\max_{y \in \mathcal{Y}} S_y,$$

$$\mathcal{Y} = \{\text{“negative”}, \text{“positive”}, \text{“neural”}\}$$

Neural Network Generative Classifier

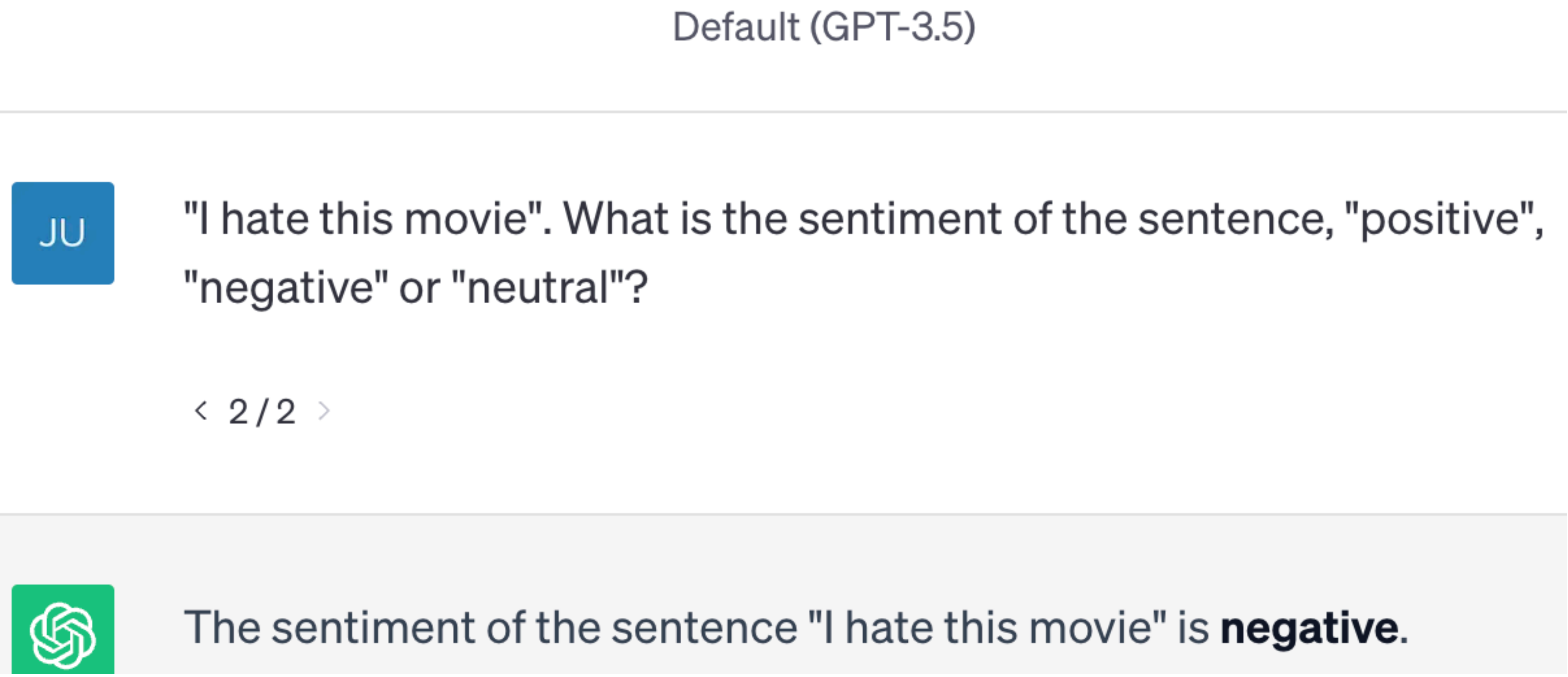
- Use a NN (e.g., LSTM) to model the context dependency $P(x_i \,|\, x_{<i}, y)$


x_1
x_2
x_3
$p(x_1 \mid \boldsymbol{x}_{<1}, y)$
$p(x_2 \mid \boldsymbol{x}_{<2}, y)$
$p(x_3 \mid \boldsymbol{x}_{<3}, y)$
$\mathbf{v}_y$
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_3$
y
$\langle \mathbf{s} \rangle$
x_1
x_2

$$P(X, y) = P(X|y)P(y) = \left(\prod_{x_i} P(x_i | x_{<i}, y) \right) P(y)$$

Prompt-based Classification by LLMs

- Formulate the text classification as a language modeling task, predicting the next tokens
- Use a template to construct a prompt (questions)
  - Example: [Sentence]. What is the sentiment of the sentence, “positive”, “negative” or “neutral”?



Default (GPT-3.5)

JU

"I hate this movie". What is the sentiment of the sentence, "positive", "negative" or "neutral"?

The sentiment of the sentence "I hate this movie" is **negative**.

Discriminative Text Classification

Why Discriminative Classifiers?

- Generative models are somewhat roundabout
  → spend lots of capacity modeling the input
- Discriminative models directly model the probability of the output → what we care about
- However, discriminative models **don't have an easy count-based decomposition**!

BOW Generative:

$$P(X, y) = P(y) \prod_{i=1}^{|X|} P(x_i|y) = \frac{c(y)}{\sum_{\tilde{y}} c(\tilde{y})} \prod_{i=1}^{|X|} \frac{c(x_i, y)}{\sum_{\tilde{x}} c(\tilde{x}, y)}$$

BOW Discriminative:

$$P(y|X) = \frac{P(y, X)}{P(X)} \quad ??$$

Sentence space is infinite!

Discriminative Model Training

- Instead, define model that calculates probability directly based on parameters θ

$$P(y|X;\theta)$$

- Define a **loss function** that is lower if the model is better, such as **negative log likelihood** over training data

$$\mathcal{L}_{\text{train}}(\theta) = -\sum_{\langle X,y\rangle \in \mathcal{D}_{\text{train}}} \log P(y|X;\theta)$$

- And **optimize the parameters directly** to minimize loss

$$\hat{\theta} = \underset{\tilde{\theta}}{\operatorname{argmin}}\ \mathcal{L}_{\text{train}}(\tilde{\theta})$$

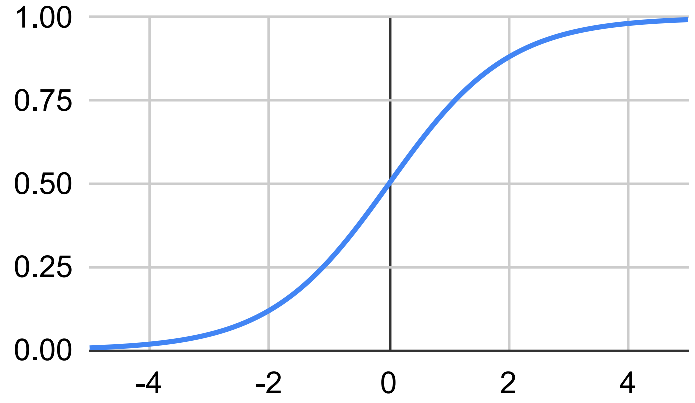
Logistic Regression

- For **binary classification** of positive/negative, first calculate score

$$s_{y|X} = \theta_{y|X} \cdot \underbrace{f(X)}_{\text{Learn a feature extractor}}$$

- Convert into a **probability**, e.g. using *sigmoid* function

$$P(y|X;\theta) = \text{sigmoid}(s_{y|X}) = \frac{1}{1+e^{-s_{y|X}}}$$



- **Learning**: maximize log likelihood of training data

$$\mathcal{L}_{\text{train}}(\theta) = \sum_{\langle X,y\rangle \sim \mathcal{D}_{\text{train}}} \log P(y|X;\theta), \quad \theta^* = \arg\max_{\theta} \mathcal{L}(\theta)$$

Multi-class Classification: Softmax

- Sigmoid can be used for binary decisions
- For multi-class decisions, calculate score for each class and use **softmax**

$$P(y|X;\theta) = \frac{e^{s_{y|X}}}{\sum_{\tilde{y}} e^{s_{\tilde{y}|X}}}$$

$$\text{s}=\begin{pmatrix} \text{-3.2} \\ \text{-2.9} \\ 1.0 \\ 2.2 \\ 0.6 \\ \ldots \end{pmatrix} \longrightarrow \text{p}=\begin{pmatrix} 0.002 \\ 0.003 \\ 0.329 \\ 0.444 \\ 0.090 \\ \ldots \end{pmatrix}$$

Gradient Descent

- Calculate the **gradient of the loss function** with respect to the parameters

$$\frac{\partial \mathcal{L}_{\text{train}}(\theta)}{\partial \theta}$$

  - How? Use the chain rule - more in later lectures.

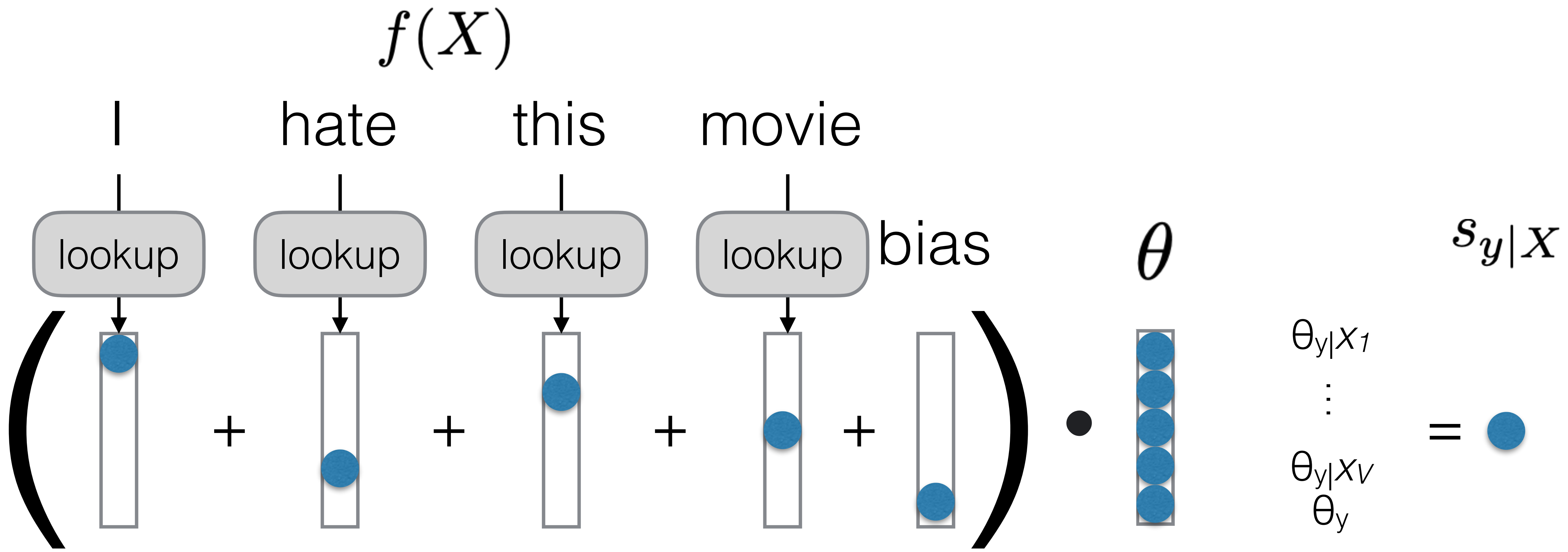
- **Update** to move in a direction that decreases the loss

$$\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}_{\text{train}}(\theta)}{\partial \theta}$$

  - α is a **learning rate** dictating speed of movement
  - This is the *first-order* gradient descent
  - Others, e.g. Newton's method, consider *second-order* (curvature) information and converge more quickly

BOW Discriminative Model

- Use BOX representations for $f(X)$



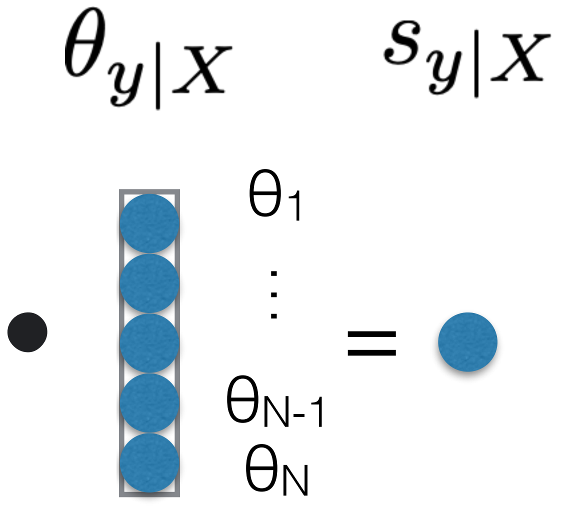
$$s_{y|X} = \theta_y + \sum_{i=1}^{|X|} \theta_{y|x_i}$$

Handcrafted linguistic features

- $f(X)$: unigram, bi-gram, POS, parsing tree, others,…

I hate this movie

$$\left(\begin{array}{l} f_1(X) = \delta(\text{“I” in } X) \\ f_2(X) = \delta(\text{“I hate” in } X) \\ f_3(X) = \delta(\text{“I hate this” in } X) \\ \quad \ldots \\ f_N(X) = \text{no. of negative words in } X \end{array} \right) \bullet \theta_{y|X} = s_{y|X}$$



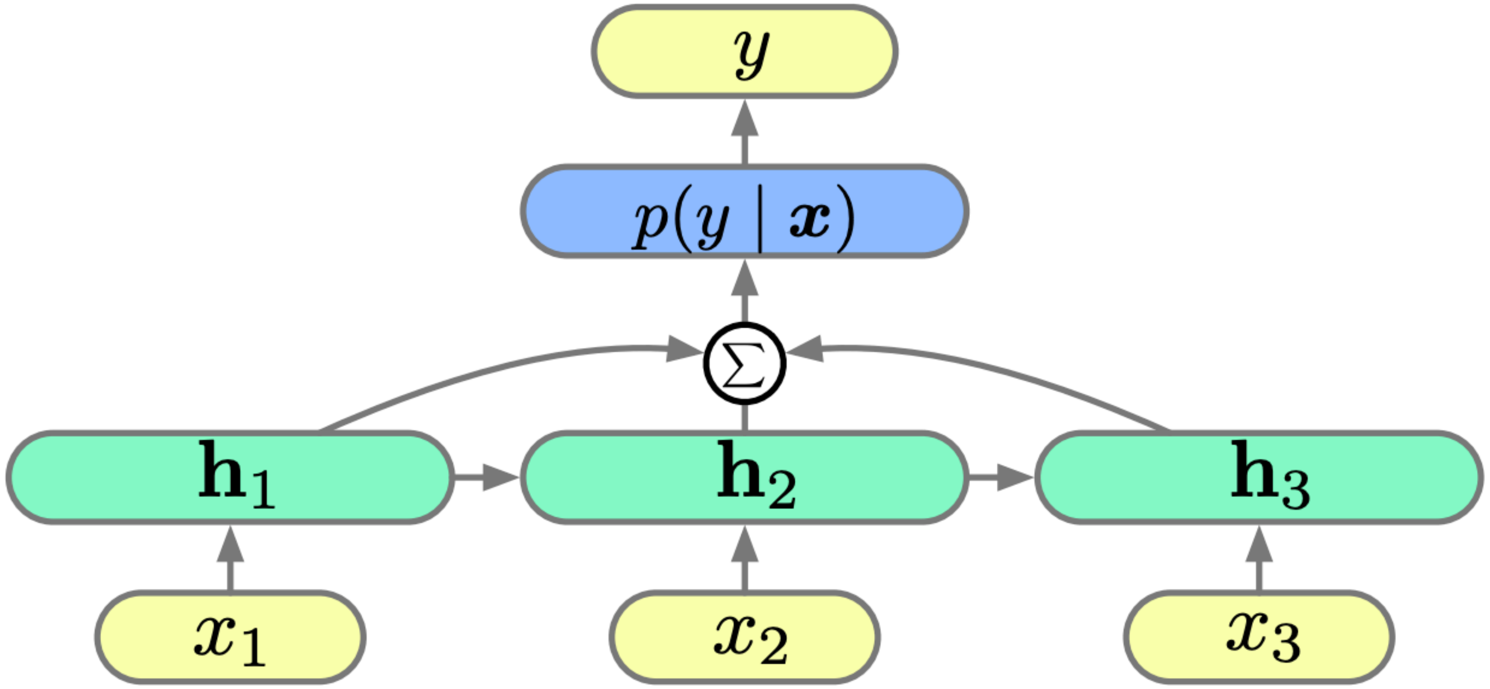
$\delta()$ is an indicator function that returns 1 if the condition is true o/w 0.

- Features can be extremely large and sparse, so is the weight $\theta_{y|X}$

$$\mathcal{L}_{\text{train}}(\theta) = \sum_{\langle X,y \rangle \sim \mathcal{D}_{\text{train}}} \log P(y|X;\theta) + \lambda \|\theta\|^2$$

Neural Network Discriminative Model

- Use neural network (e.g., LSTM) to learn features $f(X)$



$$s_{y|X} = \mathrm{NN}(X)$$

Evaluation

Model Comparison

- We've built two models (e.g. a generative and discriminative model), **how do we tell which one is better**?

- Train both on the same training set, **evaluate on a dev (test?) set**, and compare scores!

Accuracy

- Simplest evaluation measure, what percentage of labels do we get correct?

$$\mathrm{acc}(\mathcal{Y}, \hat{\mathcal{Y}}) = \frac{1}{|\mathcal{Y}|} \sum_{i=1}^{|\mathcal{Y}|} \delta(y_i = \hat{y}_i)$$

Precision/Recall/F1

- Often, we care about a particular (usually minority) class (e.g. "toxic SNS posts detected"), we'll call it "1"

- **Precision:** percentage of system output "1"s correct

$$\text{prec}(\mathcal{Y}, \hat{\mathcal{Y}}) = \frac{c(y = 1, \hat{y} = 1)}{c(\hat{y} = 1)}$$

- **Recall:** percentage of human-labeled "1"s correct

$$\text{rec}(\mathcal{Y}, \hat{\mathcal{Y}}) = \frac{c(y = 1, \hat{y} = 1)}{c(y = 1)}$$

- **F1 Score, F-measure:** harmonic mean of both

$$F_1 = \frac{2 \cdot \text{prec} \cdot \text{rec}}{\text{prec} + \text{rec}}$$

Statistical Testing

- We have two models with similar accuracies

	Dataset 1	Dataset 2	Dataset 3
Generative	**0.854**	**0.915**	0.567
Discriminative	0.853	0.902	**0.570**

- How can we tell whether the differences are due to consistent trends that hold on other datasets?
- **Statistical (significance) testing!**
- Covered briefly, see Dror et al. (2018) for a complete overview

Dror et al. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. ACL 2018

Significance Testing: Basic Idea

- Given a quantity, we test certain values of uncertainty with respect to the quantity, e.g.

- ***p*-value:** what is the probability that a difference with another quantity is by chance (lower = more likelihood of a significant difference. Typically, p<0.005 means a significant difference)

- **confidence interval:** what is the range under which we could expect another trial to fall?

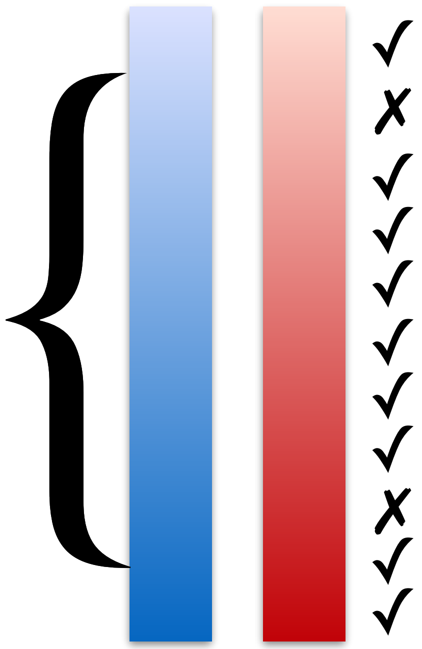
Unpaired vs. Paired Tests

- **Unpaired Test:** Compare means of a quantity on two unrelated groups
  - Example: test significance of difference of accuracies of **a model on two datasets**
- **Paired Test:** Compare means of a quantity on one dataset under two conditions
  - Example: test significance of difference of accuracies of **two models on one dataset**
- We are most commonly interested in the latter!

Bootstrap Tests

- A method that can measure p-values, confidence intervals, etc. by **re-sampling data**
- Sample many (e.g. 10,000) **subsets** from your dev/test set with replacement
- **Measure** accuracies on these many subsets

Model 1 Accs **Model 2 Accs**



The middle percentile range (e.g. 2.5-97.5) forms a confidence interval

% of wins is confidence that a gain in accuracy is *not* by chance (e.g. 1-*p*)

- **Easy** to implement, **applicable** to any evaluation measure, but somewhat **biased** on small datasets

Data Creation/Curation Basics

Task Definition

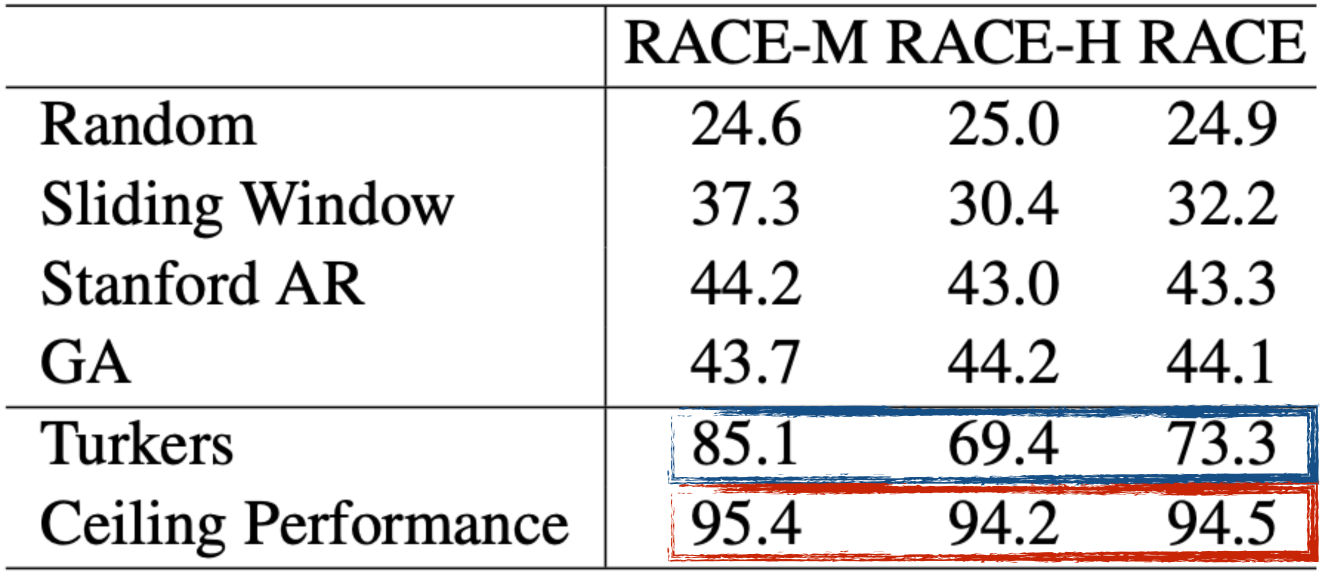
- What task do you want to perform and why?
- What are your classes?
- Creating an **annotation standard**:
    - Write down the classes and class definitions.
    - Try annotation yourself and note any hard examples.
    - Allow annotators to share hard examples with you, refine standard.

Source Data Collection

- Collect data textual data to annotate w/ labels
- Is the data:
  - **Appropriate:** Does it match the data you'll be expecting to process at test time?
  - **Representative:** Does it cover various demographics, languages, dialects, etc.?
  - **Broad:** Are you collecting data from a single domain or multiple ones?

Annotator Recruitment

- **Friends:** Good for small-scale, high-skill tasks
- **Freelancing sites:** Good for medium-scale, medium- or high-skill tasks
- **Crowdsourcing sites:** Good for large-scale, lower-skill tasks
- Can be *very* big difference in quality! (e.g. Lai et al. 2017)



	RACE-M	RACE-H	RACE
Random	24.6	25.0	24.9
Sliding Window	37.3	30.4	32.2
Stanford AR	44.2	43.0	43.3
GA	43.7	44.2	44.1
Turkers	85.1	69.4	73.3
Ceiling Performance	95.4	94.2	94.5

Random Turkers online

University Students

- Have multiple annotators annotate same data, measure agreement

Lai et al. RACE: Large-scale ReAding Comprehension Dataset From Examinations. EMNLP 2017.

Data Statements for NLP

(Bender and Friedman 2018)

- A checklist of things to document about your dataset, e.g.

- Curation rationale
- Language variety
- Speaker demographic
- Annotator demographic
- Speech situation
- Text characteristics
- Recording quality
- Other notes

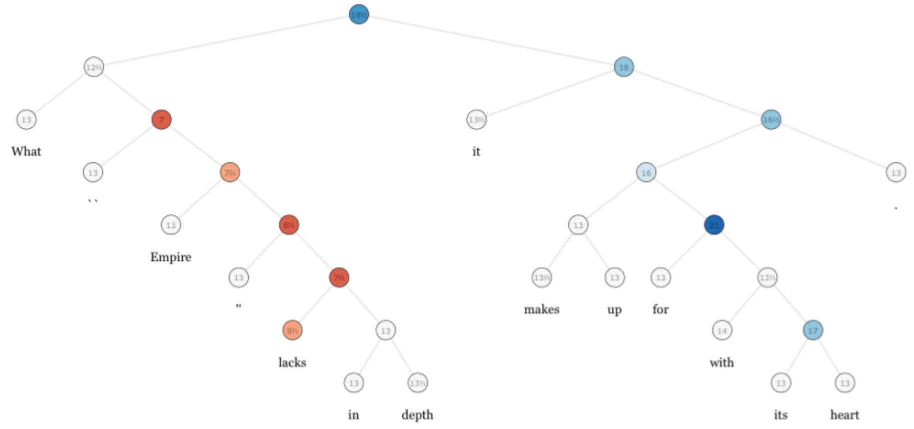
Bender and Friedman. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. TACL 2018.

Text Classification Datasets

Stanford Sentiment Treebank

(Socher et al. 2013)

- In addition to standard tags, each syntactic phrase tagged with sentiment



- **Data:** reviews from rottentomatoes.com collected by Pang and Lee (2004)
- **Annotator details:** People from MTurk

AG News

- News articles categorized into 4 classes
- **Data:** from an academic search engine (in 2004?)
- **Curation Rationale:** As a test bed for data mining and IR

- http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

Zhang et al. Character-level Convolutional Networks for Text Classification. NIPS 2016.

DBPedia

- Classification of Wikipedia entity description text into 9, 70, or 219 classes
- **Data:** from Wikipedia first sections
- **Curation rationale:** As a testbed for text categorization

https://www.kaggle.com/danofer/dbpedia-classes

Generative Classifiers

Discriminative Classifiers

Classification Eval

Data Creation

Example Datasets

Questions?