CS11-711 Advanced NLP

Text Classification

Graham Neubig

Carnegie Mellon University
Language Technologies Institute

Site
https://phontron.com/class/anlp2022/

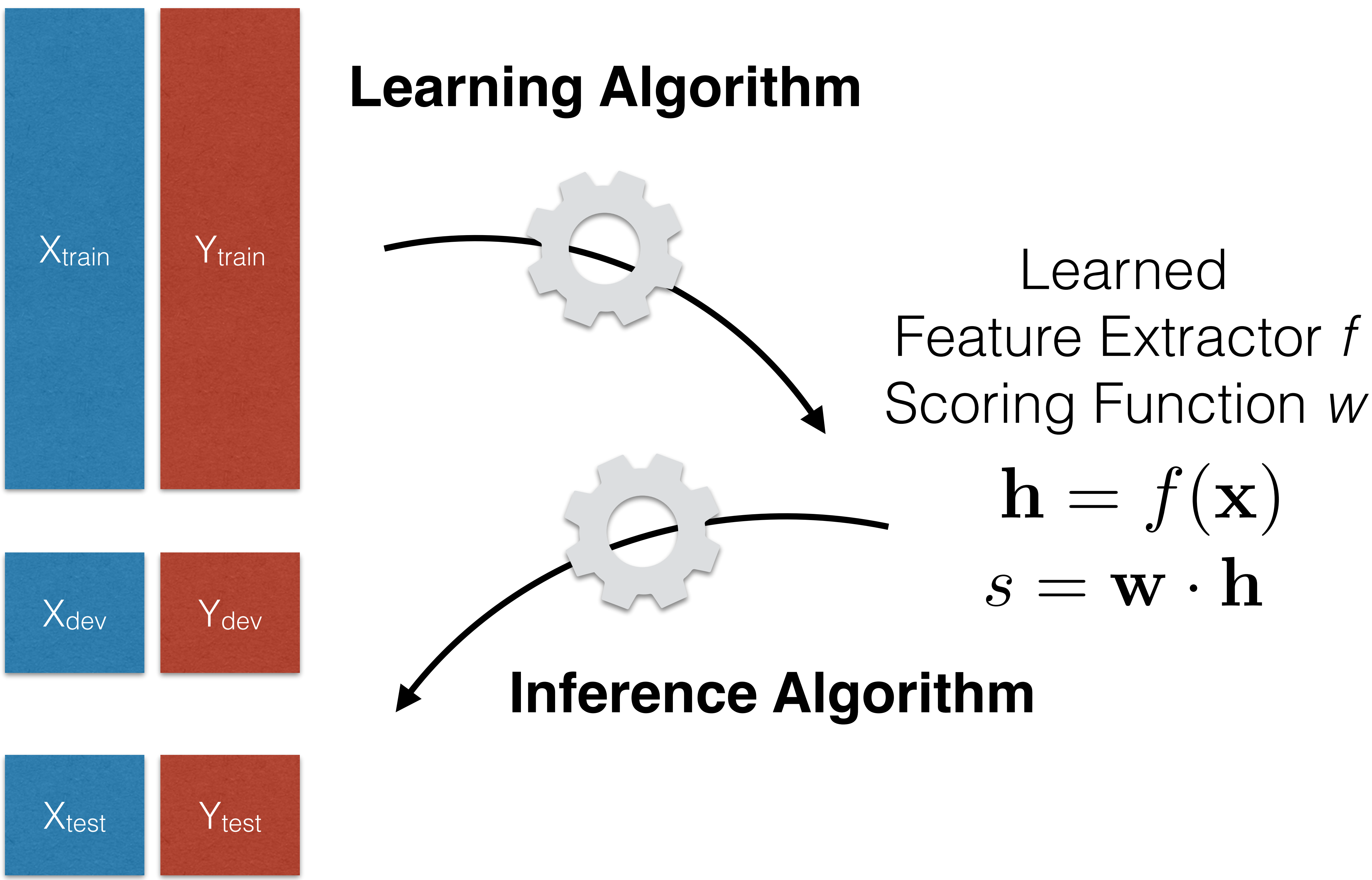
A General Framework for NLP Systems

- 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



Text Classification

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

I hate this movie

positive
neutral
negative

Generative and Discriminative Models

Generative vs. Discriminative Models

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

$$P(X) \qquad\qquad P(X, Y)$$

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

$$\hat{y} = \operatorname*{argmax}_{\tilde{y}} P(X, \tilde{y})$$

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

$$\hat{y} = \operatorname*{argmax}_{\tilde{y}} P(\tilde{y}|X)$$

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:**

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

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

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

- **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

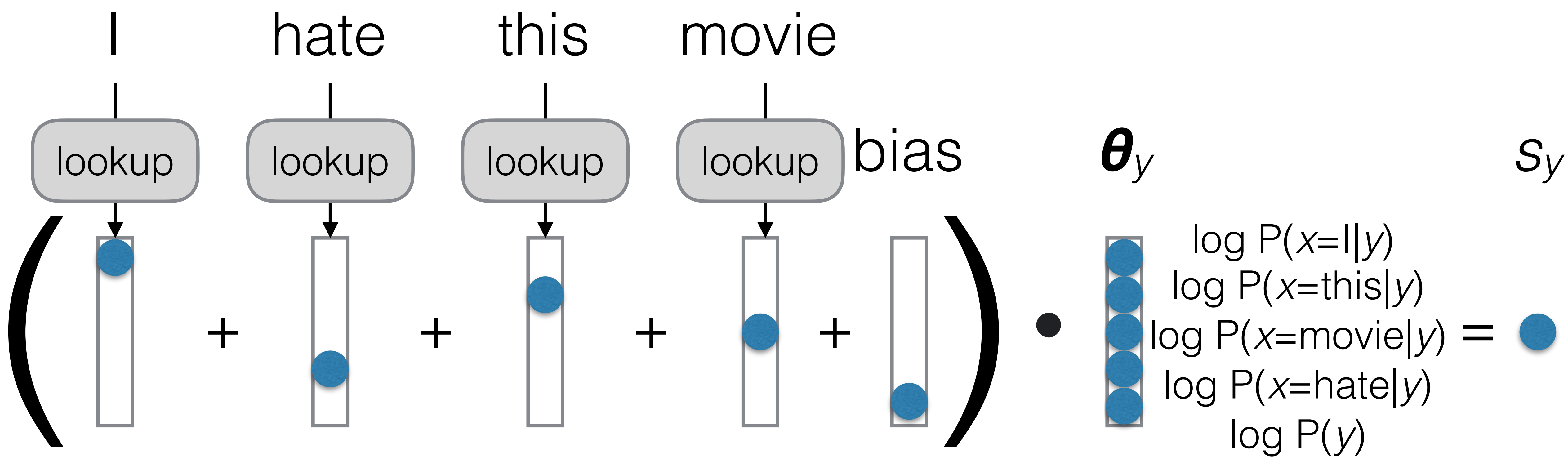
$$P(X, y) = \underbrace{P(X|y)}_{\text{class-conditional LM, trained on data associated with that class}} \underbrace{P(y)}_{\text{class prior probability (bias)}}$$

class-conditional LM, trained on data associated with that class

class prior probability (bias)

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

Bag-of-words Generative Classifier



Also called a "Naive Bayes" classifier more generally

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:

$$P(y|X) = \ ??$$

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

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

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

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

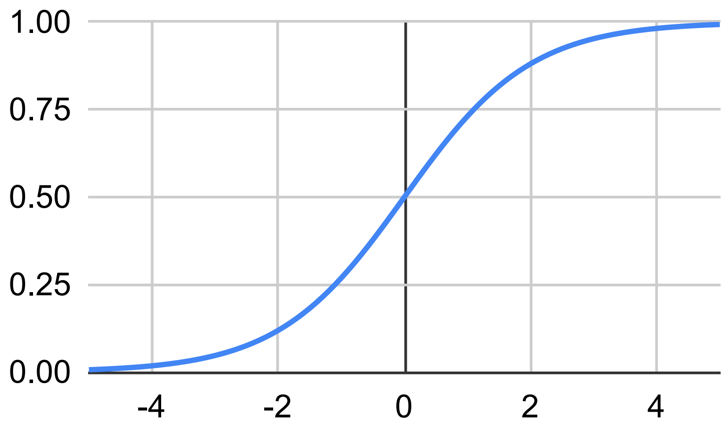
BOW Discriminative Model

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

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

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

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



Multi-class Classification: Softmax

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

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

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

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 *first-order* gradient descent
  - Others, e.g. Newton's method and L-BFGS, consider *second-order* (curvature) information and converge more quickly

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

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

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

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

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

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

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)

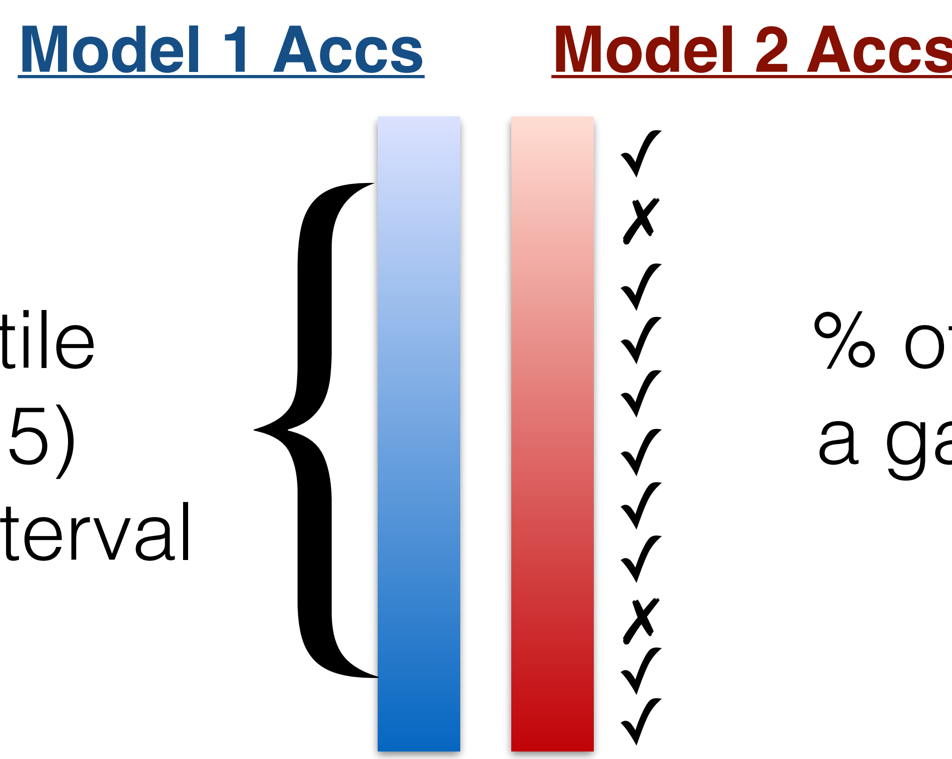
- **confidence interval:** what is the range under which we could expect another trial to fall?

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



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)

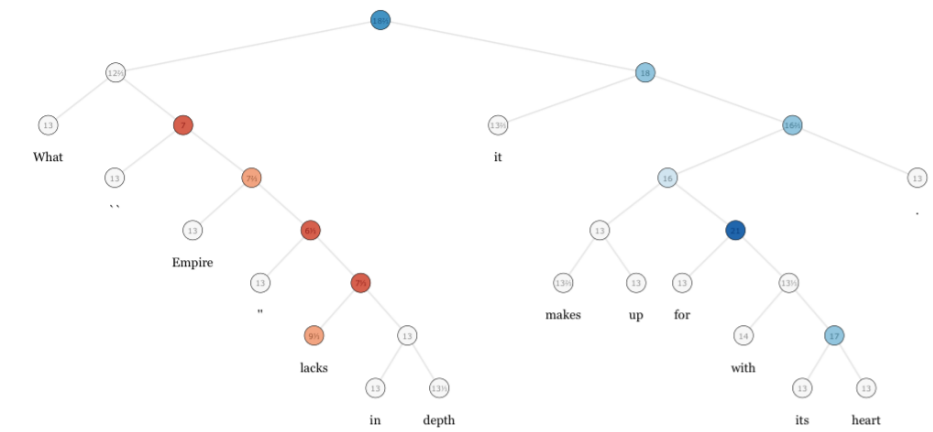
- **Easy** to implement, **applicable** to any evaluation measure, but somewhat **biased** on small datasets

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?