

CS11-711 Advanced NLP

# Text Classification

Daniel Fried and Robert Frederking  
with slides from Graham Neubig



**Carnegie Mellon University**  
Language Technologies Institute

Site

<https://cmu-anlp.github.io/>

# A General Framework for NLP Systems

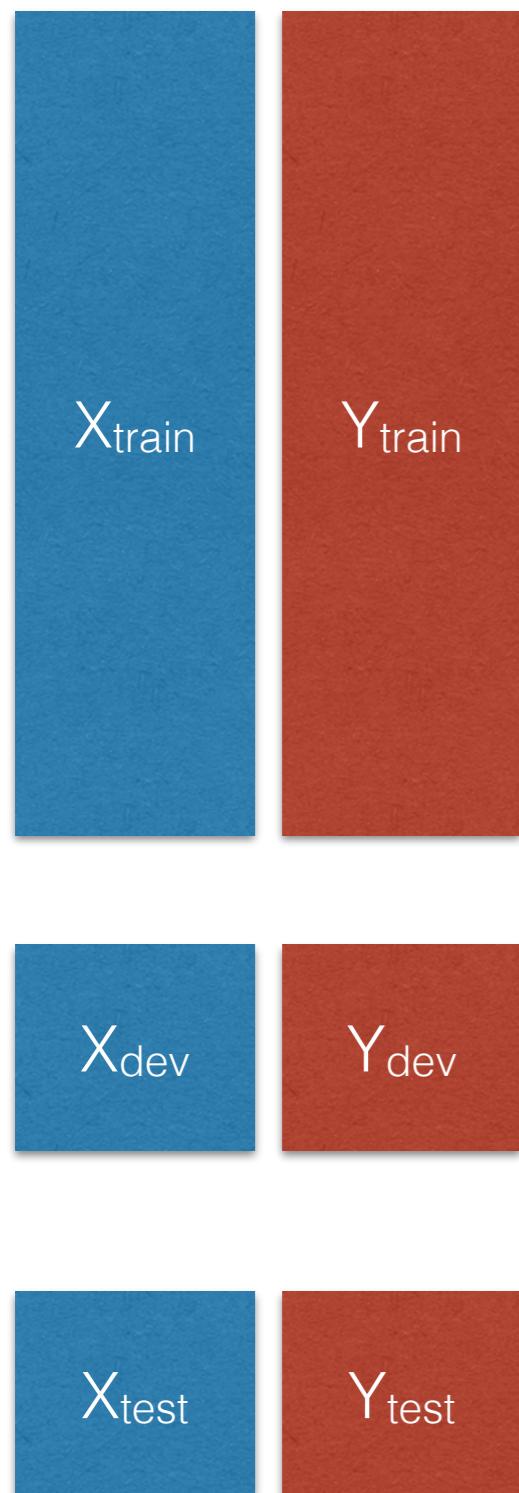
- Formally, create a function to map an **input  $X$  (*language*)** into an **output  $Y$** . Examples:

<u>Input <math>X</math></u>	<u>Output <math>Y</math></u>	<u>Task</u>
Text	Text in Other Language	Translation
Text	Response	Dialog
Text	Label	Text Classification
Text	Linguistic Structure	Language Analysis

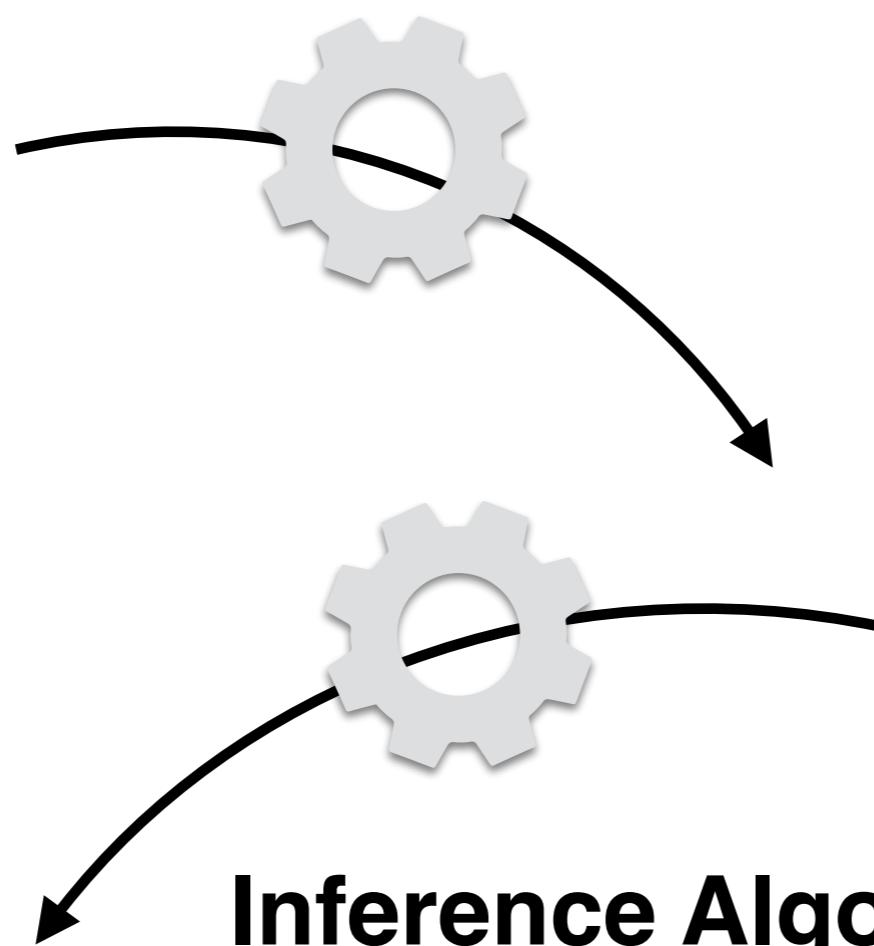
- To create such a system, we can use

- Manual creation of rules
- Machine learning from paired data  $\langle X, Y \rangle$

# Reminder: Machine Learning



## Learning Algorithm



Learned  
Feature Extractor  $f$   
Scoring Function  $w$

$$h = f(\mathbf{x})$$
$$s = \mathbf{w} \cdot \mathbf{h}$$

## Inference Algorithm

# Text Classification

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

I hate this movie →

positive  
neutral  
**negative**

# Generative and Discriminative Models

# Generative vs. Discriminative Models

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

$$P(\textcolor{blue}{X})$$

*stand-alone*

$$P(\textcolor{blue}{X}, \textcolor{red}{Y})$$

*joint*

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

$$P(\textcolor{red}{Y} | \textcolor{blue}{X})$$

*conditional*

# Application to Text Classification

- **Generative text classification:** Learn a model of the joint  $P(\textcolor{blue}{X}, \textcolor{red}{y})$ , and find

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

- **Discriminative text classification:** Learn a model of the conditional  $P(\textcolor{red}{y} | \textcolor{blue}{X})$ , and find

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

# Generative Text Classification

# Language Modeling: Calculating the Probability of a Text

$$P(X) = \prod_{i=1}^I P(x_i | x_1, \dots, x_{i-1})$$


The big problem: How do we predict

$$P(x_i | x_1, \dots, x_{i-1})$$

?!?!  
?

# 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:**  $P(x_i|x_1, \dots, x_{i-1}) \approx P(x_i)$
- **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!
- Need a distribution that assigns some probability to *all* words!
  - **Character/subword-based model:** Calculate the probability of a word based on its spelling.
  - **Add an `unknown` word (UNK-ing):** Approximate by replacing some rare words in your corpus with a special token, UNK. At test time, use the probability of this special token for any previously-unseen word.

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

# 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  $\theta_{xi}$

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

# Generative Text Classifier

- Joint probability can be based on the following decomposition

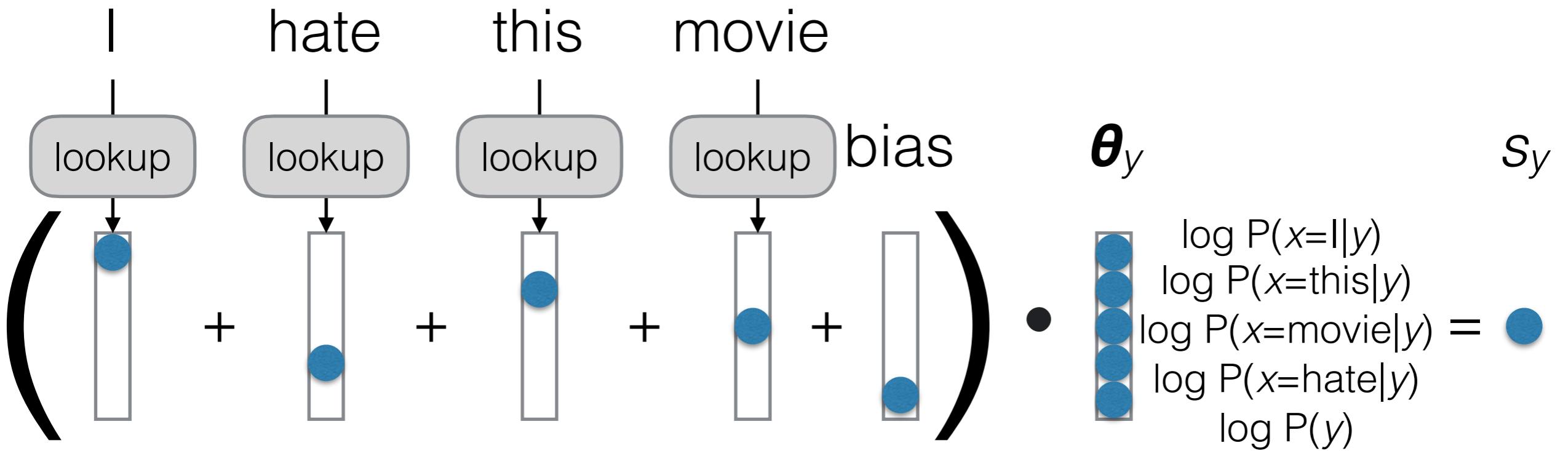
$$P(X, y) = P(X|y)P(y)$$


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

# Generative Beyond BoW

- But we're not limited to bag-of-words!

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

```
graph TD; A[P(X, y)] --> B[P(X|y)]; A --> C[P(y)]; B --> D["Could be a sophisticated neural model, e.g. GPT-3"]; C --> E["class prior probability (bias)"]
```

Could be a sophisticated  
neural model, e.g. GPT-3

class prior probability  
(bias)

# 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:**

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

See Klein and Manning, 2002  
<https://aclanthology.org/W02-1002.pdf>

# Discriminative Model Training

- Instead, define model that calculates probability directly based on parameters  $\theta$

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

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

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

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

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

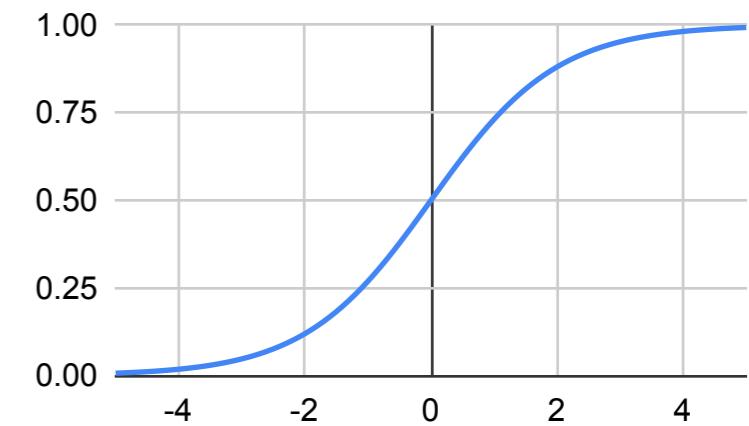
# BOW Discriminative Model

- 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**
- This is also known as logistic regression

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

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

# 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

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

- $\alpha$  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?

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

- Important to use chance rate as a baseline if you're using accuracy! (e.g. most emails are not spam)

# Precision/Recall/F1

- Often, we care about a particular (usually minority) class (e.g. "toxic 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	<b>0.854</b>	<b>0.915</b>	0.567
Discriminative	0.853	0.902	<b>0.570</b>

- 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

# Significance Testing: Basic Idea

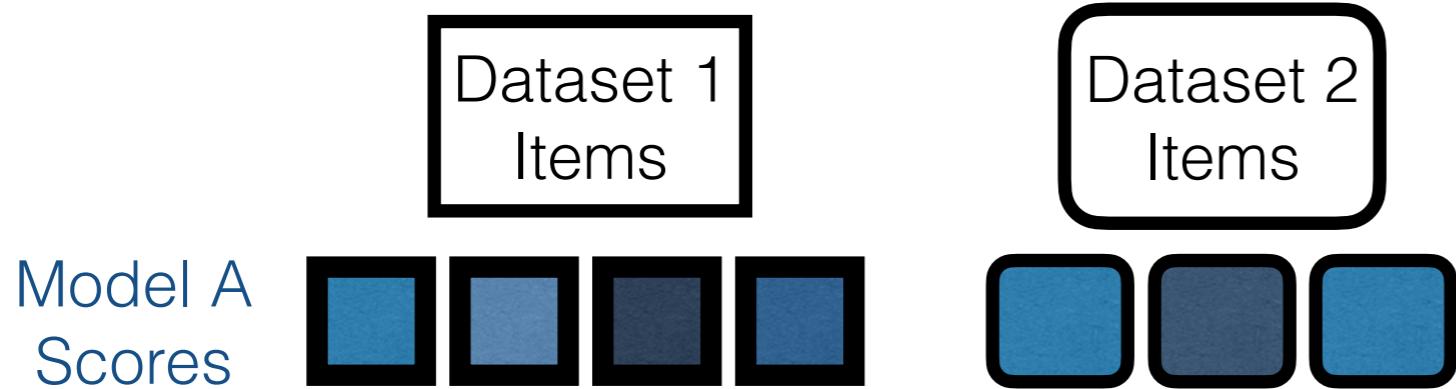
- Given a quantity (e.g. accuracy), we test certain values of uncertainty with respect to the quantity.  
Examples:
- **p-value:** what is the probability that a difference with another quantity is by chance (lower = more likelihood of a significant difference)
- **confidence interval:** what is the range under which we could expect another trial to fall?

# Unpaired vs. Paired Tests

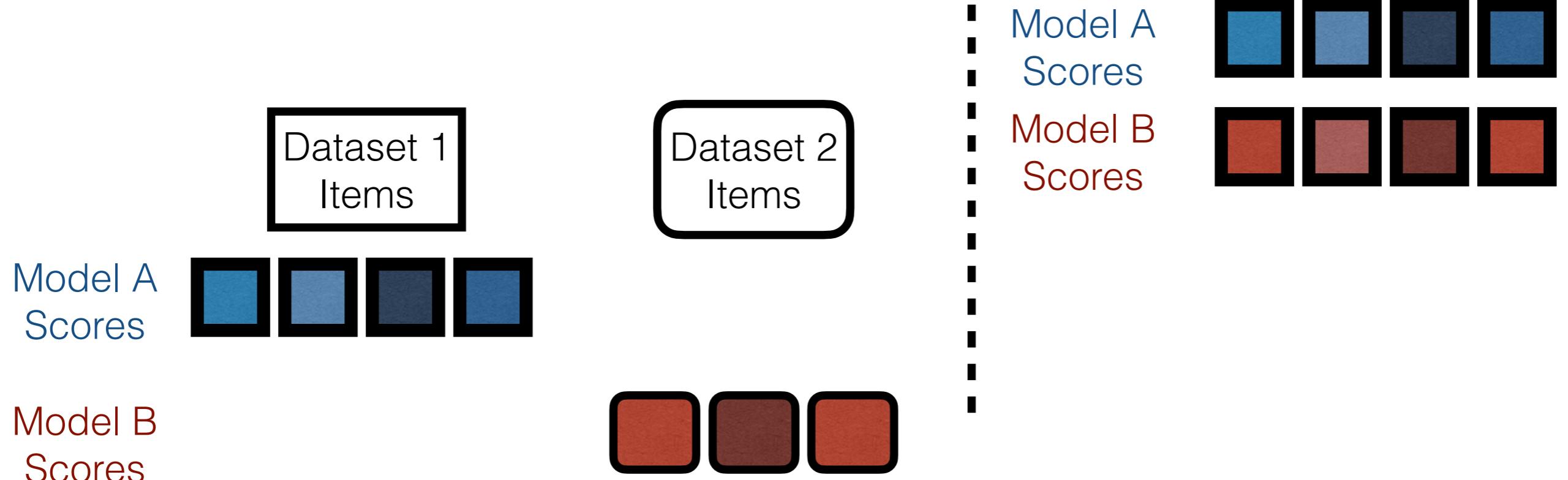
- **Unpaired Test:** Compare means of a quantity on two unrelated groups
  - Example: test significance of difference of accuracies of **a model on two datasets**
- **Paired Test:** Compare means of a quantity on one dataset under two conditions
  - Example: test significance of difference of accuracies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

# Example Conditions for Unpaired vs. Paired Tests

## Unpaired Tests



## Paired Tests



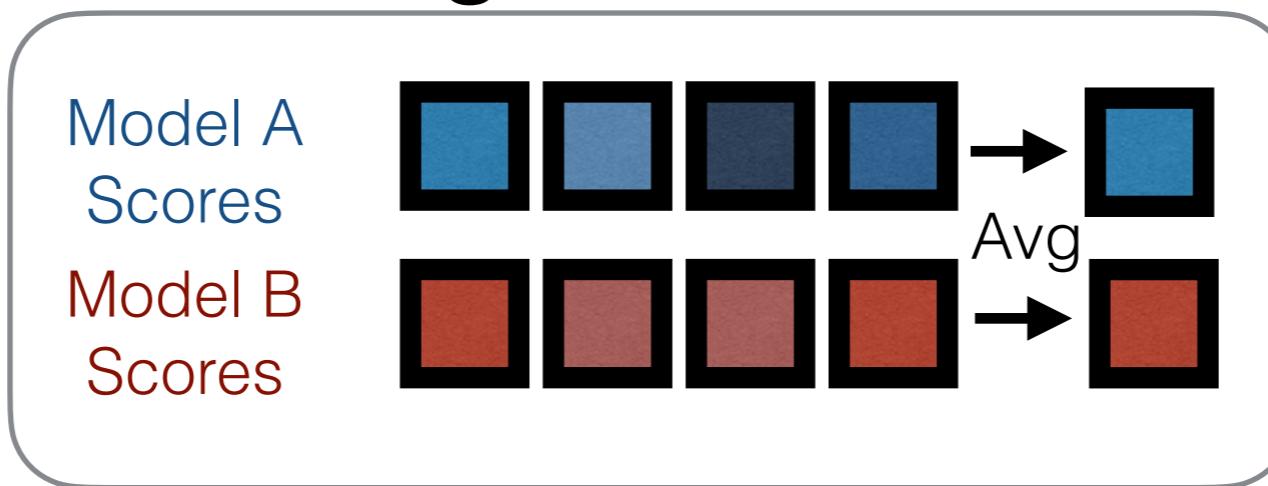
# What Type of Test to Use?

- **Parametric tests:** Make assumptions (e.g. Gaussian distributed) about the distributions of the test statistic/dataset (e.g. *t*-tests, *chi-squared test*)
  - A good fit if you have a standard scenario
- **Non-parametric tests:** Use the data that produces the test statistic, and random sampling (e.g. *bootstrap*, *jackknife*)
  - Pretty universally applicable, but might have bias with small data

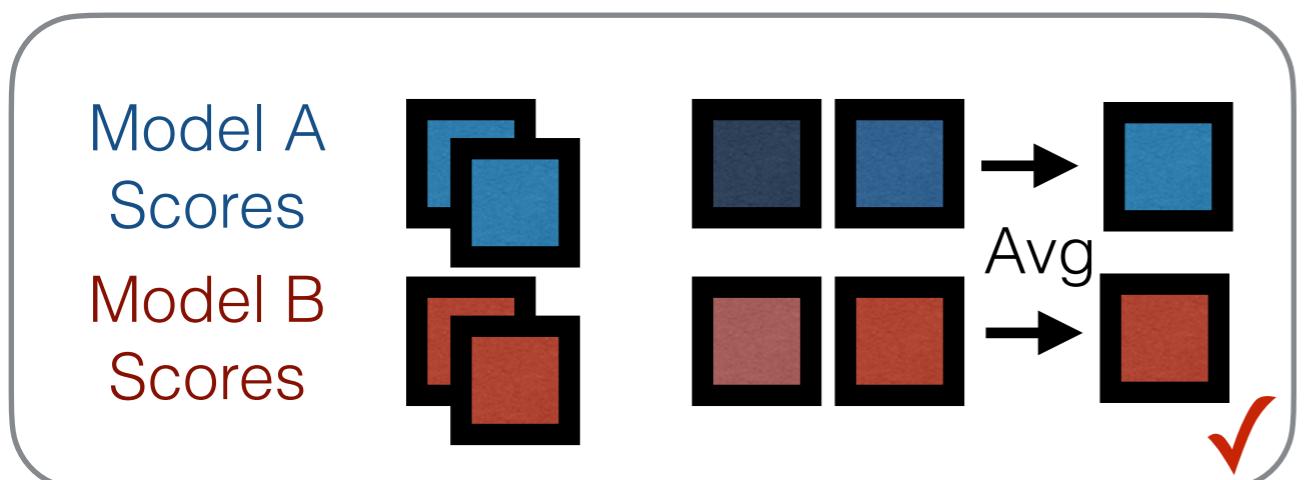
# Bootstrap Tests

- A method that can measure p-values, confidence intervals, etc. by **re-sampling data with replacement**.
- Sample many (e.g. 10,000) **subsets** from your dev/test set with replacement, and measure the statistic you're interested in.

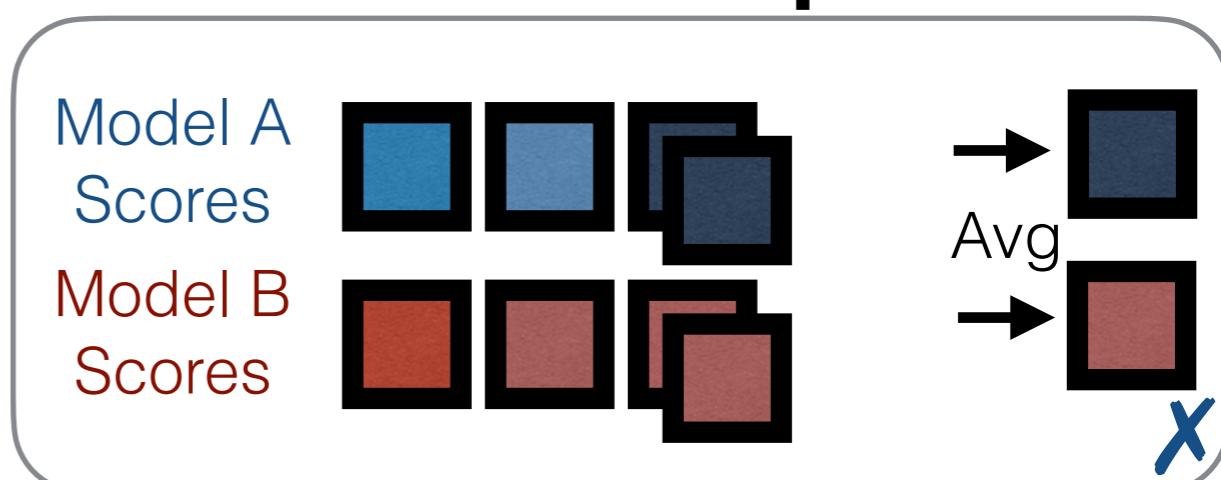
## Original Dataset



## Random Sample 1



## Random Sample K



# Bootstrap Tests

## p-values:

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

## Confidence Intervals:

Sort the K values (one-per-sample). The middle percentile range (e.g. 2.5-97.5) forms a CI.

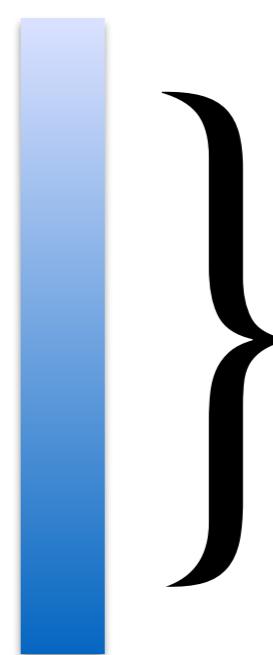
Model 1 Accs



Model 2 Accs



Model 1 Accs



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

# Code Demo

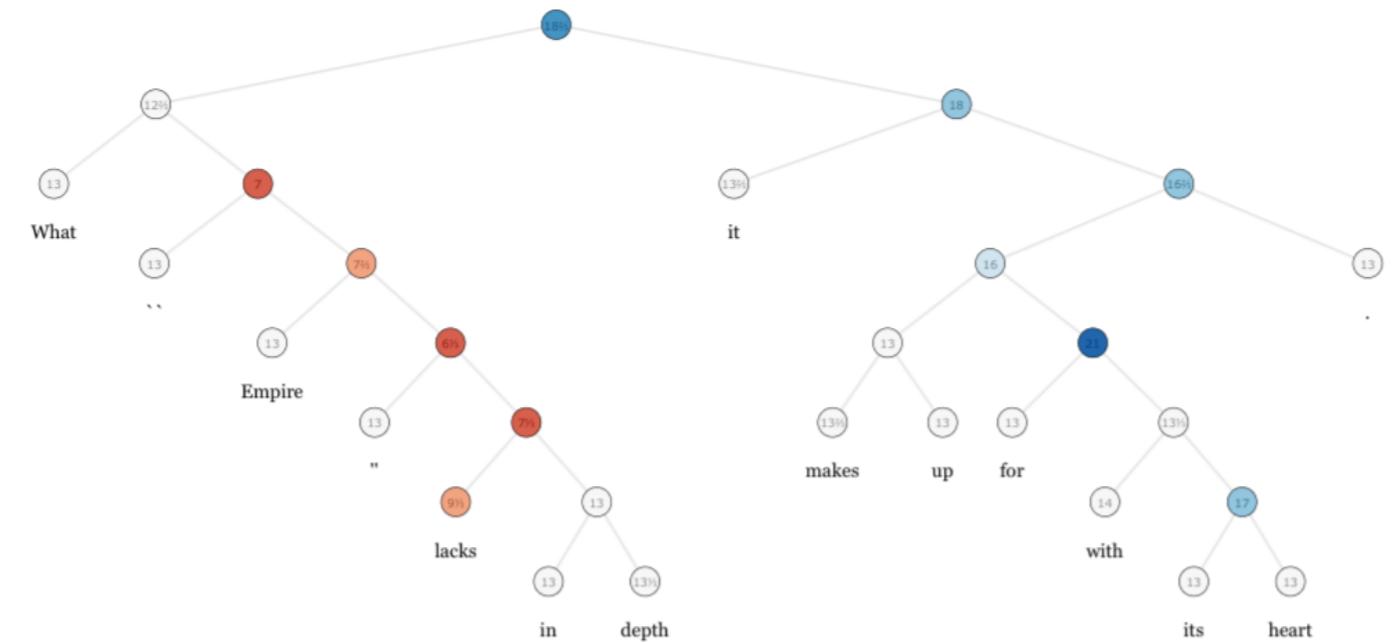
<https://github.com/cmu-anlp/anlp-code/blob/main/02-bowclassifier/bowclassifier.ipynb>

# Text Classification Datasets

# Stanford Sentiment Treebank

(Socher et al. 2013)

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



*What Empire lacks in depth, it makes up for with its heart.*

- Data:** reviews from [rottentomatoes.com](http://rottentomatoes.com) collected by Pang and Lee (2005)
- Annotator details:** People from MTurk

# DBpedia

<b>text (string)</b>	<b>11 (string)</b>	<b>12 (string)</b>	<b>13 (string)</b>
"William Alexander Massey (October 7, ...	"Agent"	"Politician"	"Senator"
"Lions is the sixth studio album by...	"Work"	"MusicalWork"	"Album"

- 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

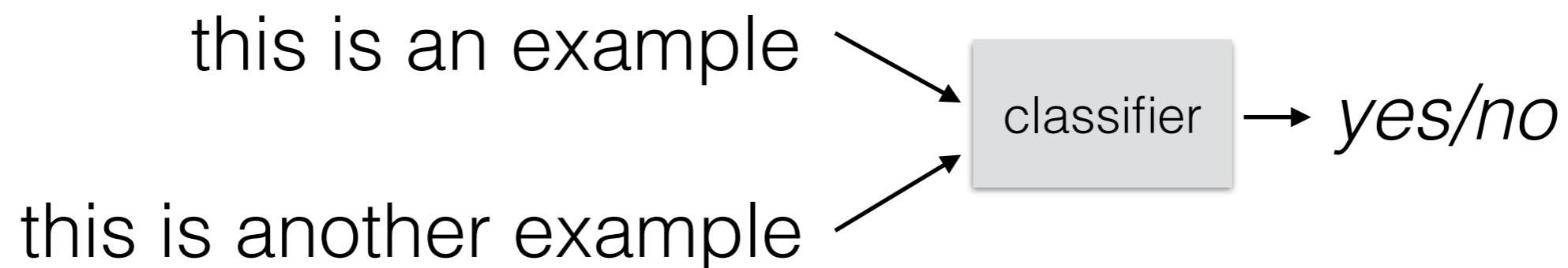
<https://www.kaggle.com/danofer/dbpedia-classes>

[https://huggingface.co/datasets/DeveloperOats/DBpedia\\_Classes](https://huggingface.co/datasets/DeveloperOats/DBpedia_Classes)

# Text Pair Classification Tasks

# Text Pair Classification

- Classify over multiple texts



# Paraphrase Identification

(Dolan and Brockett 2005)

- Identify whether A and B mean the same thing

Charles O. Prince, 53, was named as Mr. Weill's successor.



Mr. Weill's longtime confidant, Charles O. Prince, 53, was  
named as his successor.

- **Note:** *exactly* the same thing is too restrictive, so use a loose sense of similarity

# Semantic Similarity/Relatedness

(Marelli et al. 2014)

- Do two sentences mean something similar?

Relatedness score	Example
1.6	A: “A man is jumping into an empty pool” B: “There is no biker jumping in the air”
2.9	A: “Two children are lying in the snow and are making snow angels” B: “Two angels are making snow on the lying children”
3.6	A: “The young boys are playing outdoors and the man is smiling nearby” B: “There is no boy playing outdoors and there is no man smiling”
4.9	A: “A person in a black jacket is doing tricks on a motorbike” B: “A man in a black jacket is doing tricks on a motorbike”

- Like paraphrase identification, but with shades of gray (regression, not classification)

# Textual Entailment / Natural Language Inference

(Dagan et al. 2006, Marelli et al. 2014)

- **Entailment:** if A is true, then B is true (c.f. paraphrase, where opposite is also true)
  - The woman bought a sandwich for lunch  
→ The woman bought lunch
- **Contradiction:** if A is true, then B is not true
  - The woman bought a sandwich for lunch  
→ The woman did not buy a sandwich
- **Neutral:** cannot say either of the above
  - The woman bought a sandwich for lunch  
→ The woman bought a sandwich for dinner

# Data Example:

## Stanford Natural Language Inference Dataset

(Bowman et al. 2015)

- Data created from **Flickr captions**
- **Crowdsource** creation of one entailed, neutral, and contradicted caption for each caption
- **Verify** the captions with 5 judgements, 89% agreement between annotator and “gold” label
- Also, **expansion to multiple genres: MultiNLI**

Generative Classifiers

Discriminative Classifiers

Classification Eval

Data Creation

Example Datasets

# Questions?