CS 2731
Human Language Technologies

Session 6: N-gram language models part 2, text classification

Michael Miller Yoder

September 15, 2025

University of Pittsburgh | School of Computing and Information

Course logistics

- Homework 1 is **due next Thu Sep 25**
- Project Match Day is next class, Wed Sep 17. You will form groups of 2-4 students from the project list
    - Consider which projects you'd like to work on from the list of project options

NLP and culture talk at CMU

- David Bamman from Berkeley is giving an NLP colloquium talk at the Language Technologies Institute at CMU
- **This Fri Sep 19, 12:30-1:50pm**
- Studying movies and songs with data, NLP and computer vision techniques
- Contact Michael if you're interested! I'll be walking over from here
- Other interesting NLP speakers: https://www.lti.cs.cmu.edu/misc-pages/lti-colloquium.html

Language Technologies Institute | Carnegie Mellon University School of Computer Science

David Bamman

School of Information, UC Berkeley

David Bamman is an associate professor in the School of Information at UC Berkeley, where he works in the areas of natural language processing and cultural analytics, applying NLP and AI to empirical questions in the humanities and social sciences. His research focuses on improving the performance of computational methods for underserved domains like literature (including LitBank and BookNLP) and developing new empirical approaches for the study of literature, film and culture. Before Berkeley, he received his PhD in the Language Technologies Institute at Carnegie Mellon University and was a senior researcher at the Perseus Project of Tufts University. Bamman's work is supported by the National Endowment for the Humanities, National Science Foundation, Mellon Foundation, and an NSF CAREER award.

Opening Up the Data-Driven Measurement of Contemporary Popular Culture

In this talk, I'll discuss how computational methods (drawing from both NLP and computer vision) can shed light on two of the most influential cultural forms of the past half-century: film and popular music. How do these media sources represent who we are and the stories we tell?

First, I'll describe recent regulatory changes at the U.S. Copyright Office that allow for large-scale text and data mining of film, and chronicle our efforts to build a collection of 2,307 films representing the top 50 movies by U.S. box office over the period 1980 to 2022, along with award nominees. Building this collection allows us to carry out several large-scale computational studies of film, including documenting the changing patterns in the representation of gender and race/ethnicity over the past 43 years (where we see an increase in diversity over the past decade). Second, I'll discuss our efforts designing computational models to measure the stories told in contemporary songs, drawing on both popular songs (from the Billboard charts) and prestigious ones (nominated for Grammy awards) over the period 1960-2024. While we might expect the 1960s (with ballad-driven folk singers like Joan Baez, Bob Dylan and Simon & Garfunkel) to be a high-water mark for narrativity, we find the opposite: narrativity has been steadily increasing over this period, largely due to the rise of the strongly narrative genres of hip hop and rap. This work illustrates a new frontier of the data-driven analysis of culture at a large scale.

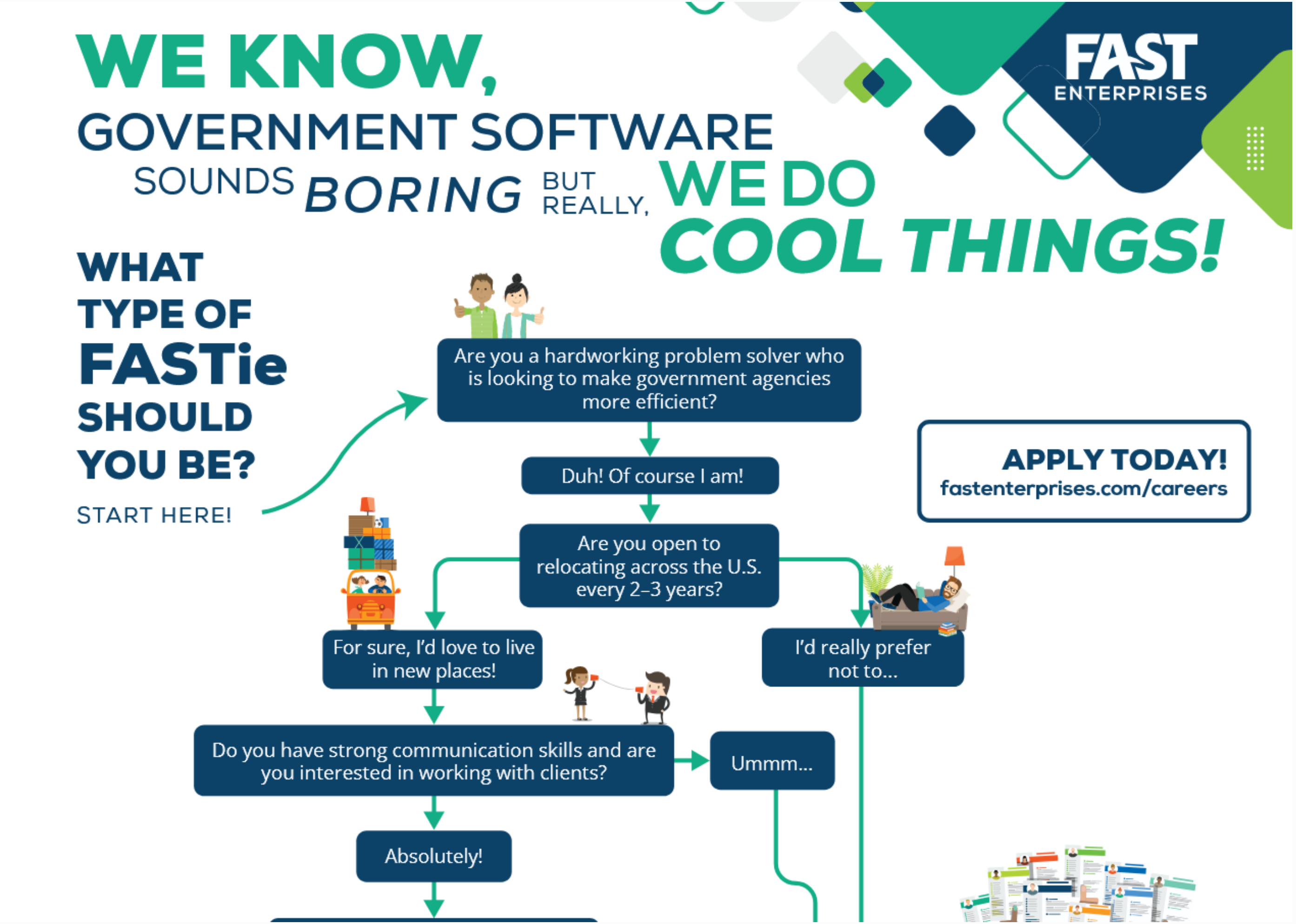
Friday, September 19th

DH A302 (Doherty Hall)

12:30PM - 1:50PM

LTI Colloquium Fall 2025

Career fair opportunity this Wed Sep 17



Lecture overview: N-gram language models part 2, text classification

- Selecting projects
- Smoothing to handle zeros in n-gram language models
- Coding activity: build your own n-gram language model!
- Text classification
- Evaluation of text classification
  - Precision, recall, f1-score
  - Train/dev/test and cross-validation sets
- Harms in classification
- Coding activity
  - Clickbait classification evaluation

Selecting projects

How to select a project

- What task/topic do I care about and/or have domain knowledge in?
- What type of contribution do I want to make? Modeling/analysis/data collection/research survey project?
- What datasets are available?
- What baseline models already exist to compare against? Is there code?
- What is the quantitative evaluation plan?
- What is the analysis/qualitative evaluation plan?

The goal: well-structured projects designed to answer clear research questions related to NLP

Slide credit: Maarten Sap, Lorraine Li

The problem of zeros in n-gram language models

The Perils of Overfitting

N-grams only work well for word prediction if the test corpus looks like the training corpus

- In real life, it often doesn't
- We need to train robust models that generalize!
    - One kind of generalization: Zeros!
    - Things that don't ever occur in the training set but occur in the test set

Slide credit: David Mortensen

N-grams in the test set that weren't in the training set

Suppose our bigram LM, trained on Twitter, reads a document by the philosopher Wittgenstein:

Whereof one cannot speak, thereof one must be silent.

This contains the bigrams: whereof one, one cannot, cannot speak, speak [comma], [comma] thereof, thereof one, one must, must be, be silent.

Suppose "whereof one" never occurs in the training corpus (`train`) but whereof occurs 20 times. According to MLE, it's probability is

$$P(\text{one}|\text{whereof}) = \frac{c(\text{whereof}, \text{one})}{c(\text{whereof})} = \frac{0}{20} = 0$$

The probability of the sentence is the **product** of the probabilities of the bigrams. What happens if one of the probabilities is zero?

Slide credit: David Mortensen

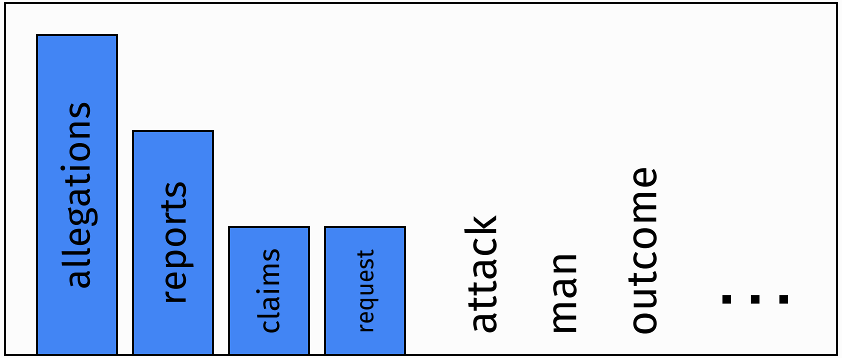
Laplace and Lidstone smoothing

The intuition of smoothing

When we have sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request

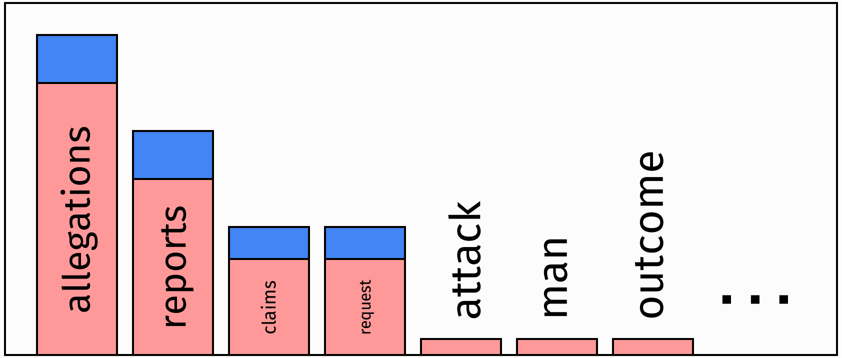
7 total



Steal probability mass to generalize better

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other

7 total



Slide adapted from Jurafsky & Martin, Dan Klein

Laplace smoothing: Pretending that we saw each word once more

MLE estimate $P_{MLE}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$

Add-1 estimate $P_{Add-1}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + |V|}$

Where V is the vocabulary of the corpus.

Slide credit: David Mortensen

Laplace smoothing is too blunt

Problem: A large dictionary makes rare words too probable.

Solution: instead of adding 1 to all counts, add $k < 0$.

How to choose k?

Slide credit: Lorraine Li

How to choose k?

Add-0.001 Smoothing

Doesn't smooth much

xya	1	1/3	1.001	0.331
xyb	0	0/3	0.001	0.0003
xyc	0	0/3	0.001	0.0003
xyd	2	2/3	2.001	0.661
xye	0	0/3	0.001	0.0003
...				
xyz	0	0/3	0.001	0.0003
Total xy	3	3/3	3.026	1

Slide credit: Lorraine Li

How to choose k?

- Hyperparameter!
  - Try many k values on dev data and choose k that gives the lowest perplexity
  - Report result on test data
- Could tune this at the same time as n in n-gram LM

Slide adapted from Lorraine Li

Coding activity: build your own n-gram LMs

N-gram language models with nltk on JupyterHub

- Click on this nbgitpuller link
- Open **session5_ngram_lm.ipynb**

Text classification

Text classification

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"

ROMANCE

Pride and Prejudice

DIALOG

Is this spam?

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

© Stanford University. All Rights Reserved.

What is the subject of this medical article?

MEDLINE Article


Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

ELSEVIER

?

MeSH Subject Category Hierarchy

Antagonists and Inhibitors

Blood Supply

Chemistry

Drug Therapy

Embryology

Epidemiology

…

Slide adapted from Jurafksy & Martin

Text Classification

We have a set of documents that we want to *classify* into a small set *classes*.

Applications:

- **Topic classification:** you have a set of news articles that you want to classify as finance, politics, or sports.
- **Sentiment detection:** you have a set of movie reviews that you want to classify as good, bad, or neutral.
- **Language Identification:** you have a set of documents that you want to classify as English, Mandarin, Arabic, or Hindi.
- **Reading level:** you have a set of articles that you want to classify as kindergarten, 1st grade, ...12th grade.
- **Author identification:** you have a set of fictional works that you want to classify as Shakespeare, James Joyce, ...
- **Genre identification:** you have a set of documents that you want to classify as report, editorial, advertisement, blog, ...

Slide credit: David Mortensen

Example: Sentiment Detection

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Slide credit: David Mortensen

How to evaluate your classifier

Gold labels and predicted labels

Document	gold label	predicted label
just plain boring	–	–
entirely predictable	–	–
no surprises and very few laughs	–	+
very powerful	+	–
the most fun film of the summer	+	+

The **gold** label is the label that a human assigned to the document.

The **predicted** or **hypothesized** label is the label that the classifier assigned to the document.

Slide credit: David Mortensen

We Can Evaluate a Classifier Using Accuracy

Accuracy is our first shot.

- Accuracy:

$$\frac{\text{how many instances your system got right}}{\text{all instances in the test set}}$$

Slide credit: David Mortensen

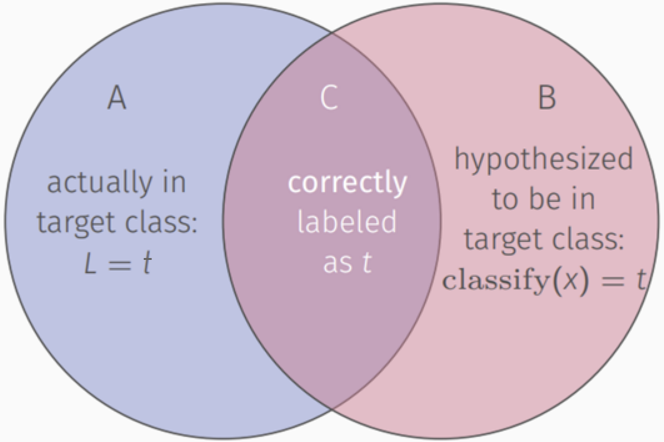
Issues with using test set accuracy

- Imagine an “important email” classifier that notifies you when you get an important email
- Suppose that 99% of the messages you receive are junk and not important (we’re being realistic here)
- An easy important email classifier: classify **nothing** as important
  - You would get lots of work done, because you wouldn’t be distracted by email
  - The email classifier would have an accuracy of ~99%
  - Everybody would be happy except for your boss
- You must take the relative importance of the classes into account, and the cost of the error types

Slide adapted from David Mortensen

Evaluation in the Two-class case

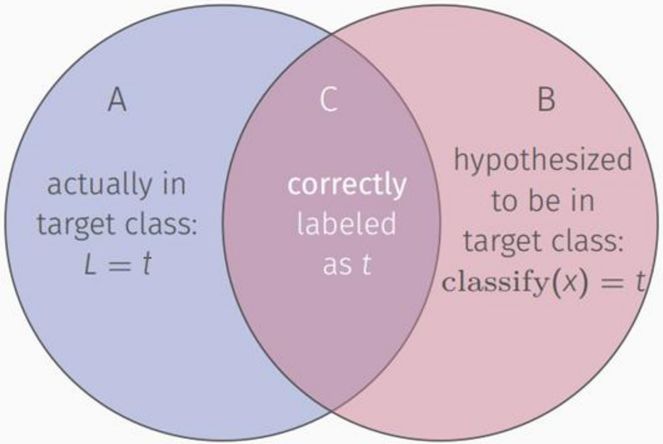
- Suppose we have one of the classes $t \in \mathcal{L}$ as the target class.
- We would like to identify documents with label t in the test data.
- We get



- Precision $\hat{P} = \frac{C}{B}$ (percentage of documents `classify` correctly labeled as t)
- Recall $\hat{R} = \frac{C}{A}$ (percentage of actual t labeled documents correctly labeled as t)
- $F_1 = 2\frac{\hat{P} \cdot \hat{R}}{\hat{P} + \hat{R}}$

Slide credit: David Mortensen

A Different View – Contingency Tables



	$L = t$	$L \neq t$	
$\texttt{classify}(X) = t$	C (true positives)	$B \setminus C$ (false positives)	B
$\texttt{classify}(X) \neq t$	$A \setminus C$ (false negatives)	(true negatives)	
	A		

precision = tp/(tp+fp)

recall = tp/(tp+fn)

Slide credit: David Mortensen

Why precision and recall

- 2-way precision and recall are specific to a target class

- Accuracy=99% on important email detection

  but

- Recall = 0 (out of all actually important emails, got none)
- Precision and recall, unlike accuracy, emphasize true positives: finding the things that we are supposed to be looking for

Slide adapted from Jurafksy & Martin

A combined measure: F1-score

We almost always use balanced F_1 (i.e., β = 1). Harmonic mean

$$F_1 = \frac{2PR}{P+R}$$

Slide adapted from Jurafksy & Martin

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**$_s = \frac{200}{3+30+200}$
		recall$_u = \frac{8}{8+5+3}$	**recall**$_n = \frac{60}{10+60+30}$	**recall**$_s = \frac{200}{1+50+200}$	

Evaluation with > 2 Classes

- **Macroaveraged precision and recall**: let each class be the target and report the average $\hat{P}$ and $\hat{R}$ across all classes.
- **Microaveraged precision and recall**: pool all one-vs.-rest decisions into a single contingency table, calculate $\hat{P}$ and $\hat{R}$ from that.

Example of more than two classes

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

$$\text{microaverage precision} = \frac{268}{268+99} = \mathbf{.73}$$

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

Figure 4.6 Separate confusion matrices for the 3 classes from the previous figure, showing the pooled confusion matrix and the microaveraged and macroaveraged precision.

Slide credit: David Mortensen, Jurafksy & Martin

Train/dev/test splits and cross-validation

Development Sets ("Devsets") and Cross-validation

Training set	Development Set	Test Set

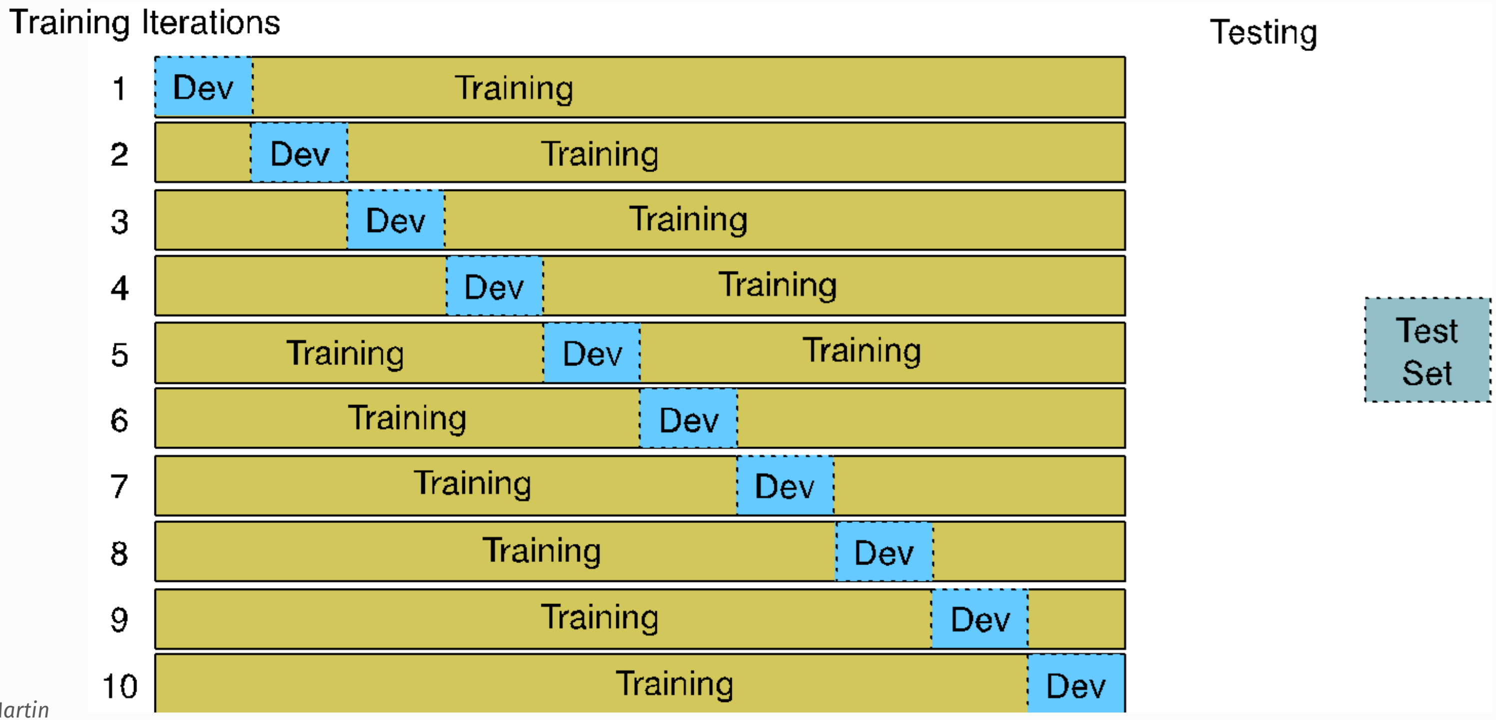
Train on training set, tune on dev set, report on test set

- **Do not look at test set**
- Using a dev set 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?

Slide adapted from Jurafksy & Martin

Cross-validation: multiple splits

- Pool results over splits, Compute pooled dev performance
- Good for when you don’t have much data (<10k instances rule of thumb)



Slide adapted from Jurafksy & Martin

Harms in classification in NLP

Harms in sentiment classifiers

Kiritchenko and Mohammad (2018) found that most sentiment classifiers assign lower sentiment and more negative emotion to sentences with African American names in them.

This perpetuates negative stereotypes that associate African Americans with negative emotions

Slide adapted from Jurafksy & Martin

Harms in toxicity classification

Toxicity detection is the task of detecting hate speech, abuse, harassment, or other kinds of toxic language

But some toxicity classifiers incorrectly flag as being toxic sentences that are non-toxic but simply mention identities like blind people, women, or gay people.

This could lead to censorship of discussion about these groups.

Slide adapted from Jurafksy & Martin

What causes these harms?

Can be caused by:

- Problems in the training data; machine learning systems are known to amplify the biases in their training data.
- Problems in the human labels
- Problems in the resources used (like lexicons)
- Problems in model architecture (like what the model is trained to optimized)

Mitigation of these harms is an open research area

Can’t fully “remove” bias because exists in societies that produced texts we use

So need to be explicit about what those biases may be through **data statements** and **model cards**

Slide adapted from Jurafksy & Martin

Data statements [Bender & Friedman 2018]

For each dataset you release, document:

- Curation rationale: why were certain texts selected
- Language variety
- Speaker demographic
- Annotator demographic
- Speech situation
  - Time and place, modality, scripted vs spontaneous, intended audience
- Text characteristics
  - Genre, topic
- Recording quality (for speech)

Slide adapted from Jurafksy & Martin

Model cards [Mitchell et al. 2019]

For each algorithm you release, document:

- training algorithms and parameters
- training data sources, motivation, and preprocessing
- evaluation data sources, motivation, and preprocessing
- intended use and users
- model performance across different demographic or other groups and environmental situations

Coding activity: clickbait classifier evaluation

Clickbait classification evaluation

- Click on this nbgitpuller link
  - Or find the link on the course website
- Open **session6_clickbait_eval.ipynb**

Conclusion

- Smoothing can handle the problem of unseen n-grams in n-gram language models
- Text classification is an NLP task learning a mapping from texts to a set of discrete labels
- Classifiers are evaluated with accuracy, precision, recall and F1-score
- Cross-validation is an alternative to train/dev/test split to estimate performance
- Text classification systems can be biased against the language or references to marginalized groups

Questions?