Doing more with less: how to make good text classifiers without a lot of data

Toby Sachs-Quintana

Application: keeping track of trending issues

- Low data situation
- The number of documents for a set of issues is small
- 10 - 100 training examples, not 1000s

It can be hard to find trending issues

Detecting trending issues is important for customer service

1. Stopping problems before they become really big problems

2. Helps the product development cycle

If only Ironman could've spot this trending issue earlier!

Machine learning and automation to the rescue!

Steps to fix the problem of detecting trending issues

1. Tools to test for over-fitting
2. Simple models
3. Multi-dimensional word vectors
4. Neural nets
5. Bonus topics

Models trained on 95 data points

Model	Train time	Test Score
CNN with 2D word embedding	1.5h on a CPU	0.70
Two layer neural net	1.5h on a CPU	0.70
SVM	< 1sec	0.71+/- 0.05
Naïve Bayes	< 1sec	0.63 +/- 0.14

20 newsgroups public dataset

- 1800 newsgroups posts on 20 topics
- Rec.motorcycles vs. rec.autos
- Filter headers, footers, and quotes
- Tokenize and lemmetize ‘autos’ = ‘auto’

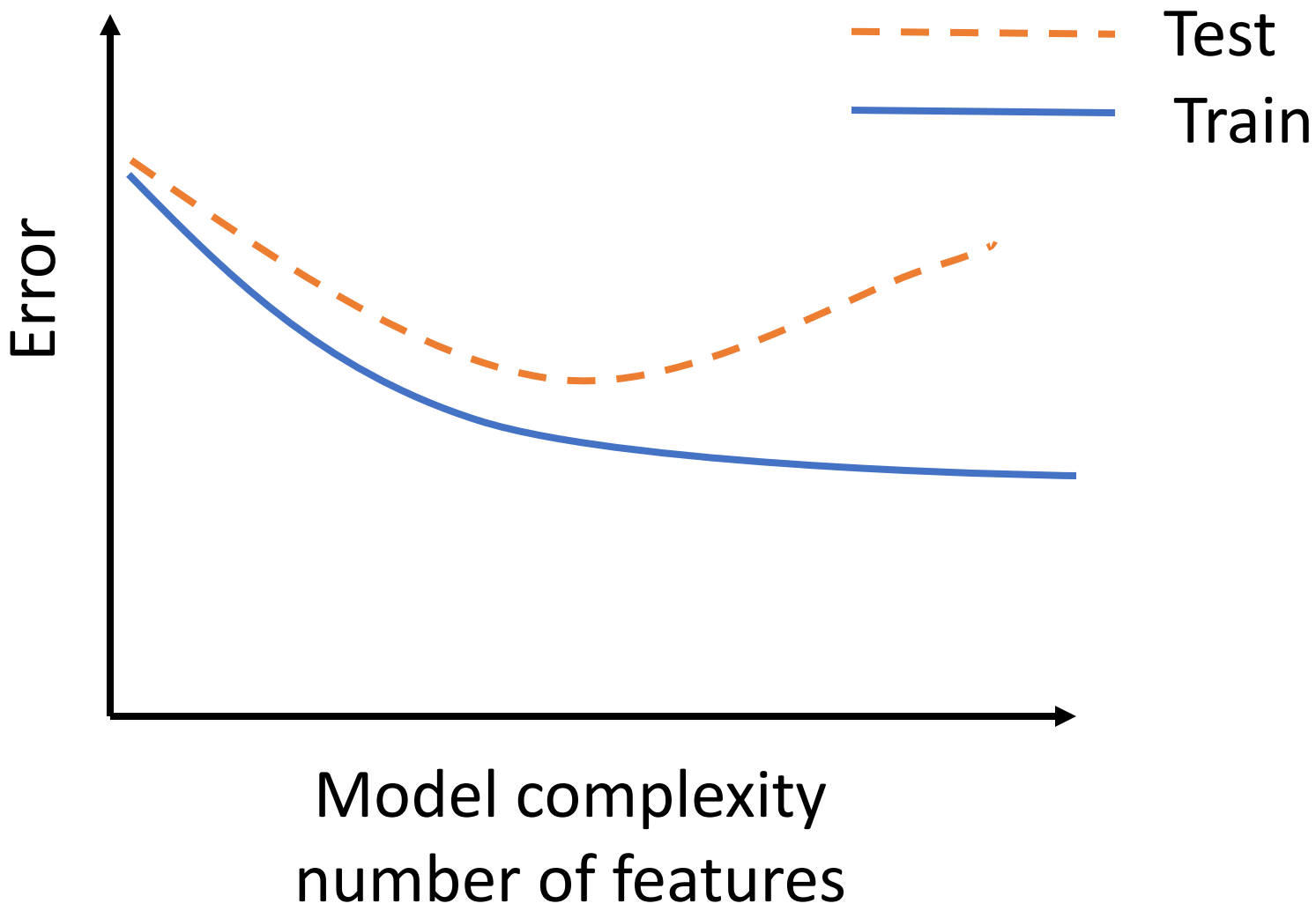
Training data

Newsgroups	Number of posts
Autos	594
Motorcycles	598

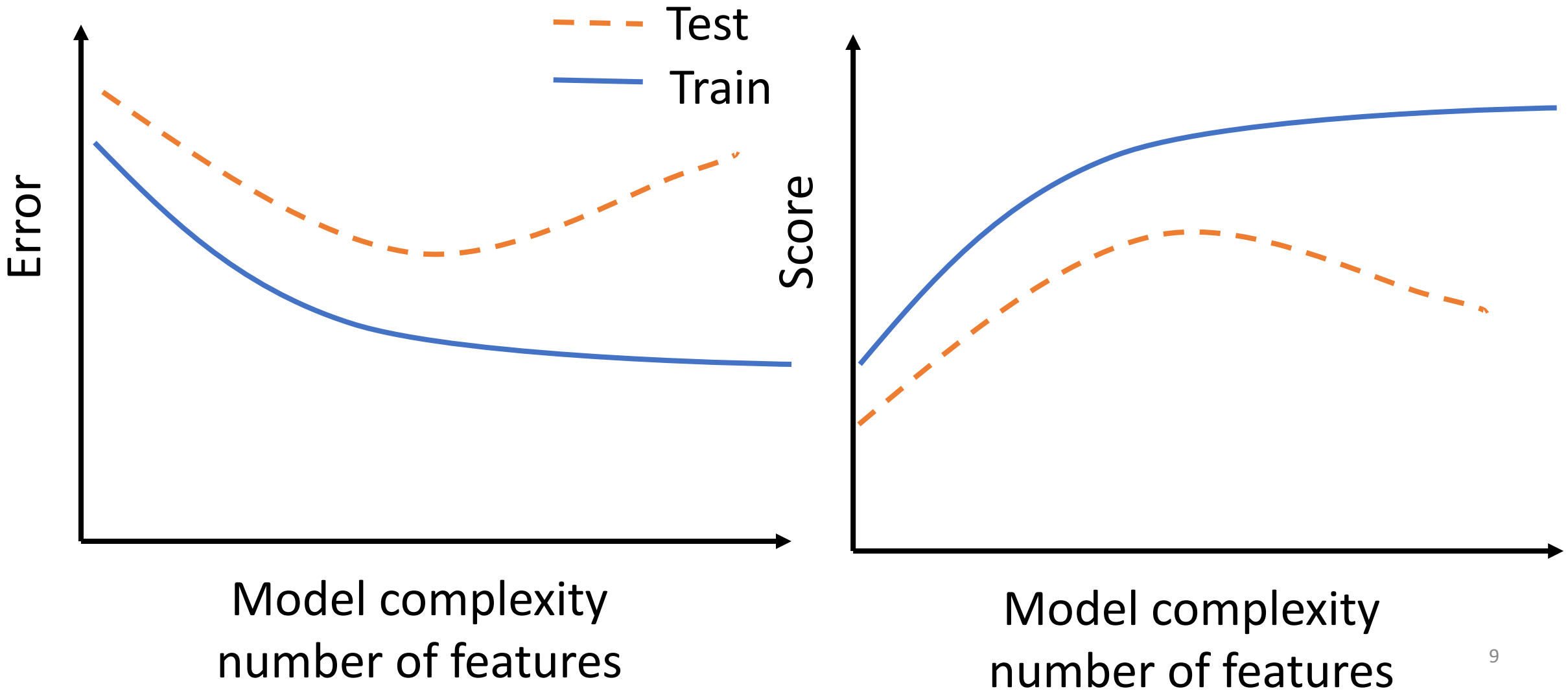
Test data

Newsgroups	Number of posts
Autos	396
Motorcycles	398

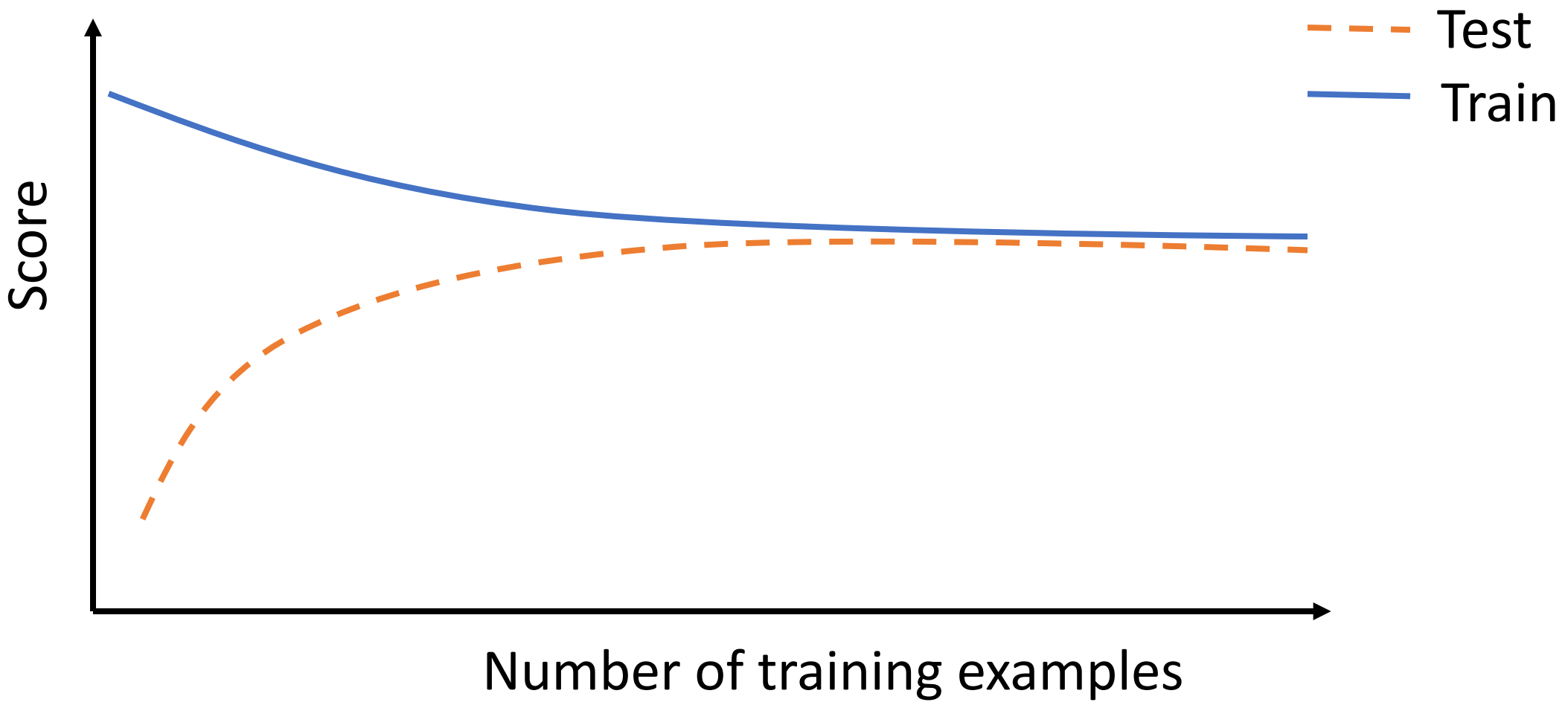
Cross-validation curves are a great tool to measure over-fitting.



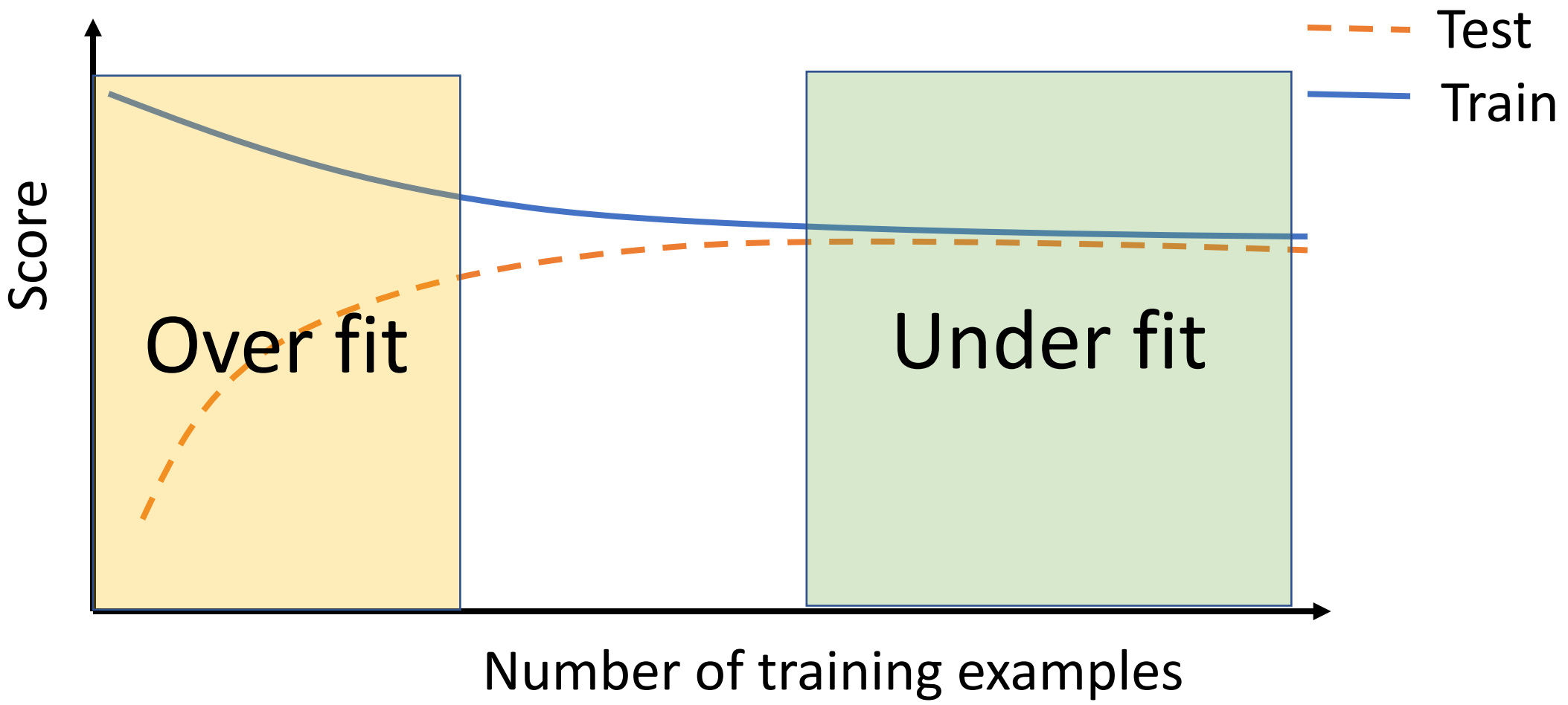
Cross-validation curves are a great tool to measure over-fitting.



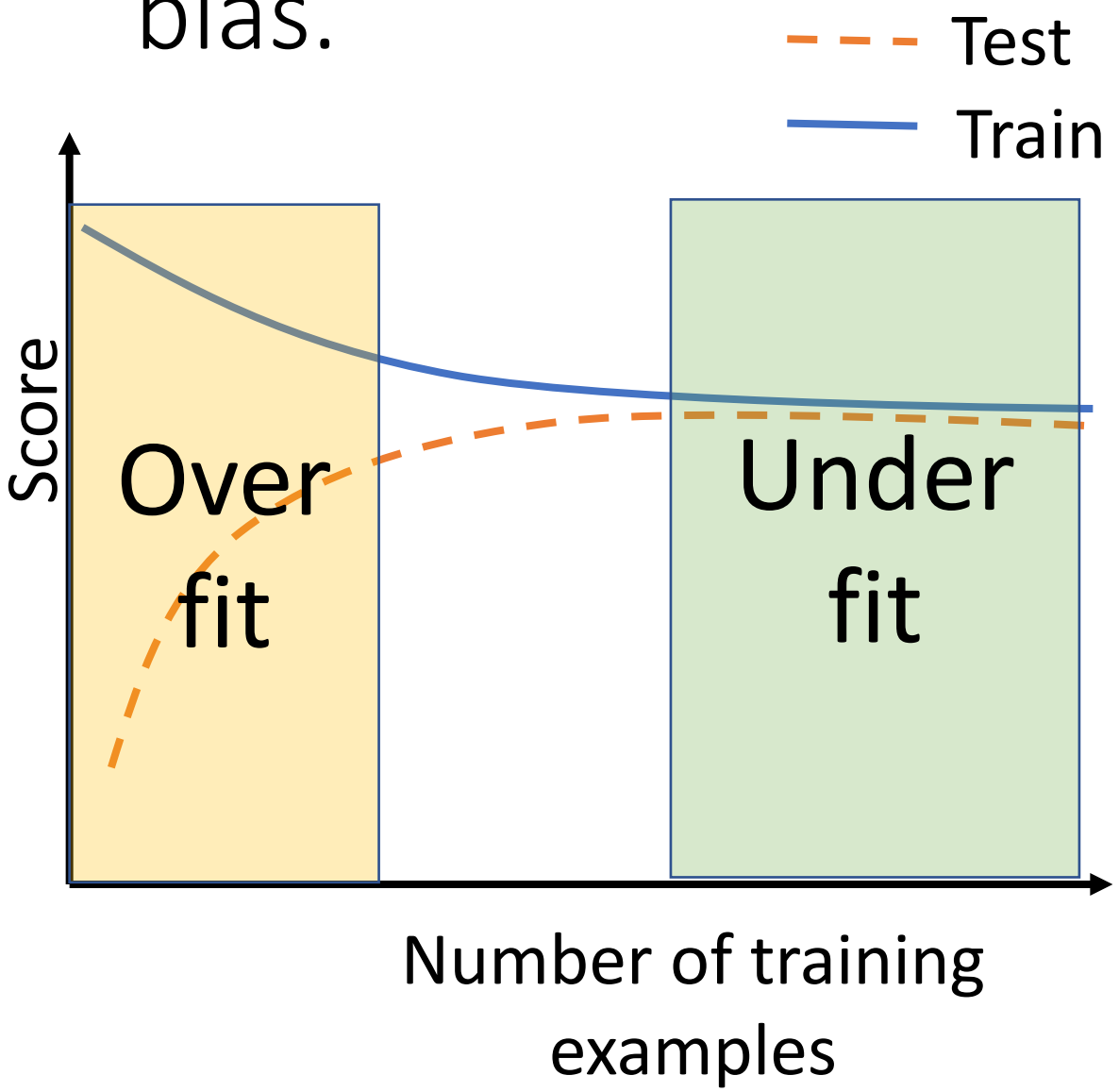
Learning curves are a great tool to measure bias.



Learning curves are a great tool to measure over fitting.



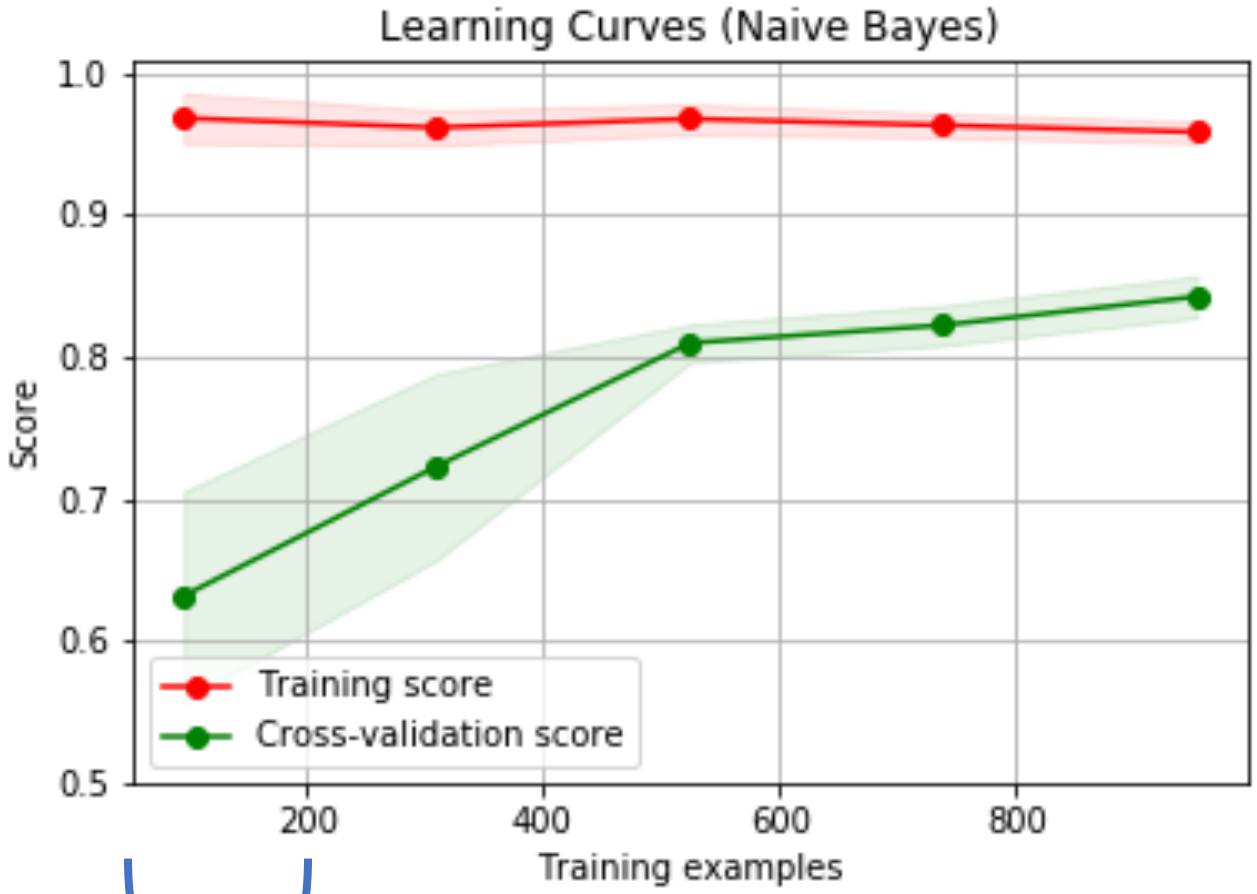
Learning curves are a great tool to measure bias.



- How to fix over fitting
  - Less features
  - More data
  - More regularization
- How to fix under fitting
  - More features
  - Less regularization

Naïve Bayes

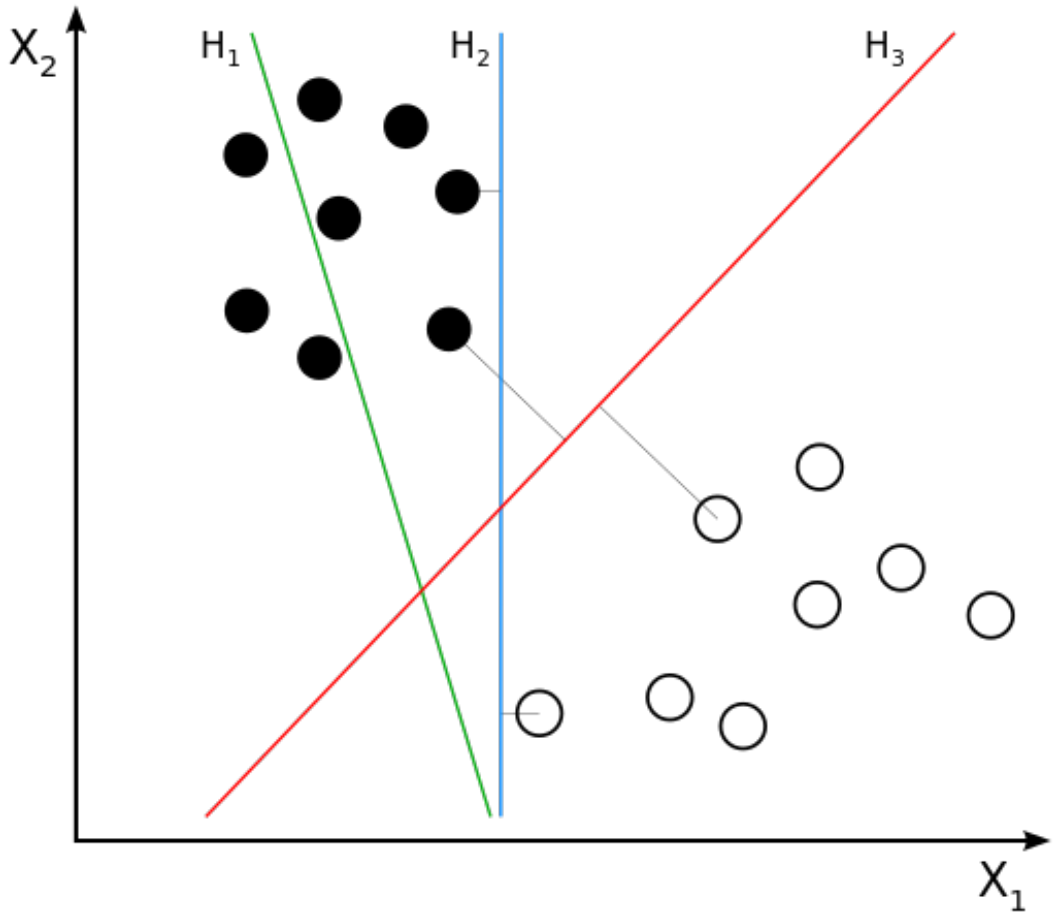
- Probability of document being a class is joint probability of the words indicating the class
- Accuracy = 83%
- 95 training examples 63% +/- 14%
- TF-IDF representation of words



The area we care about

Support Vector Machines

- Maximize separation
- Non linear separation is fast
- Regularization penalty for more features

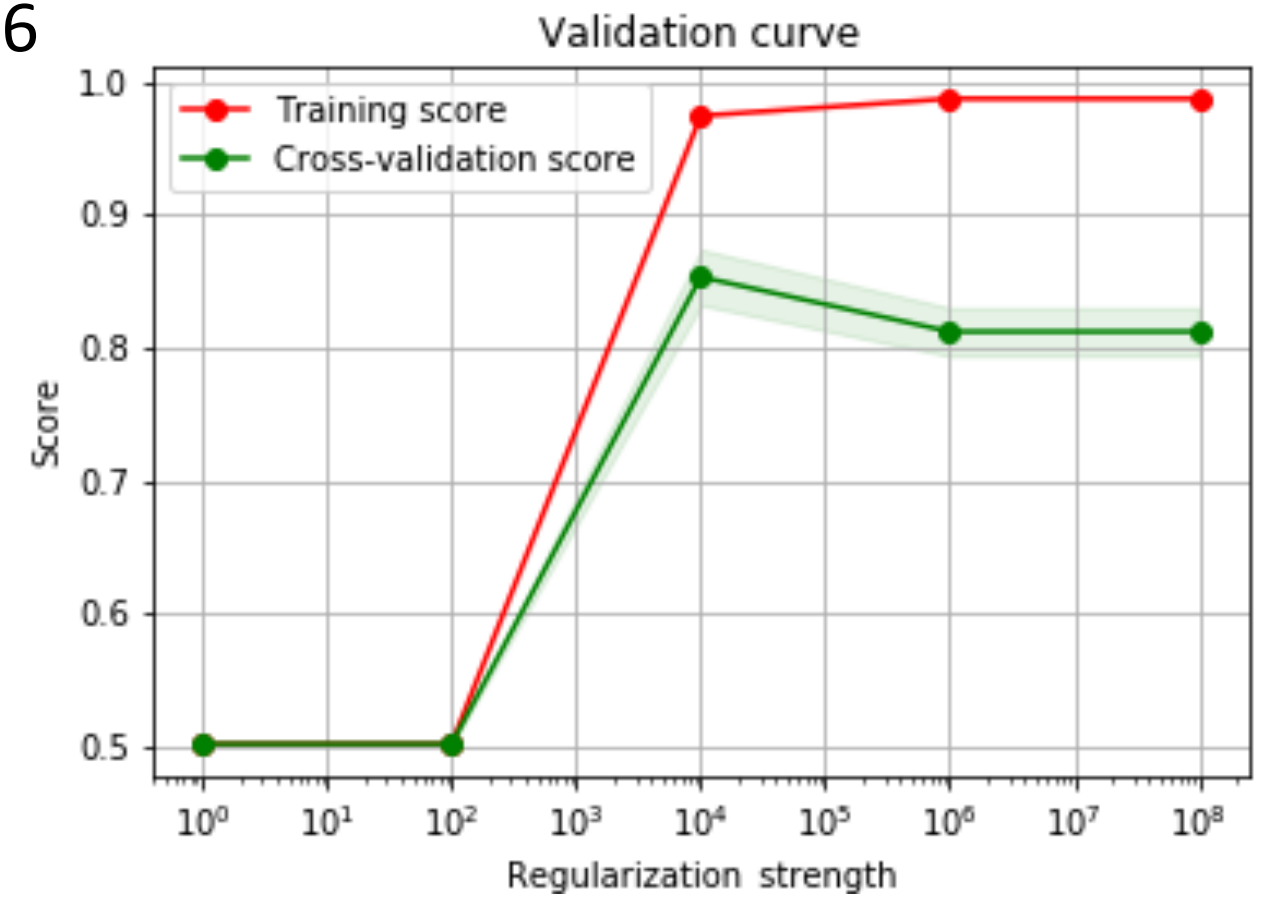


Support Vector Machines

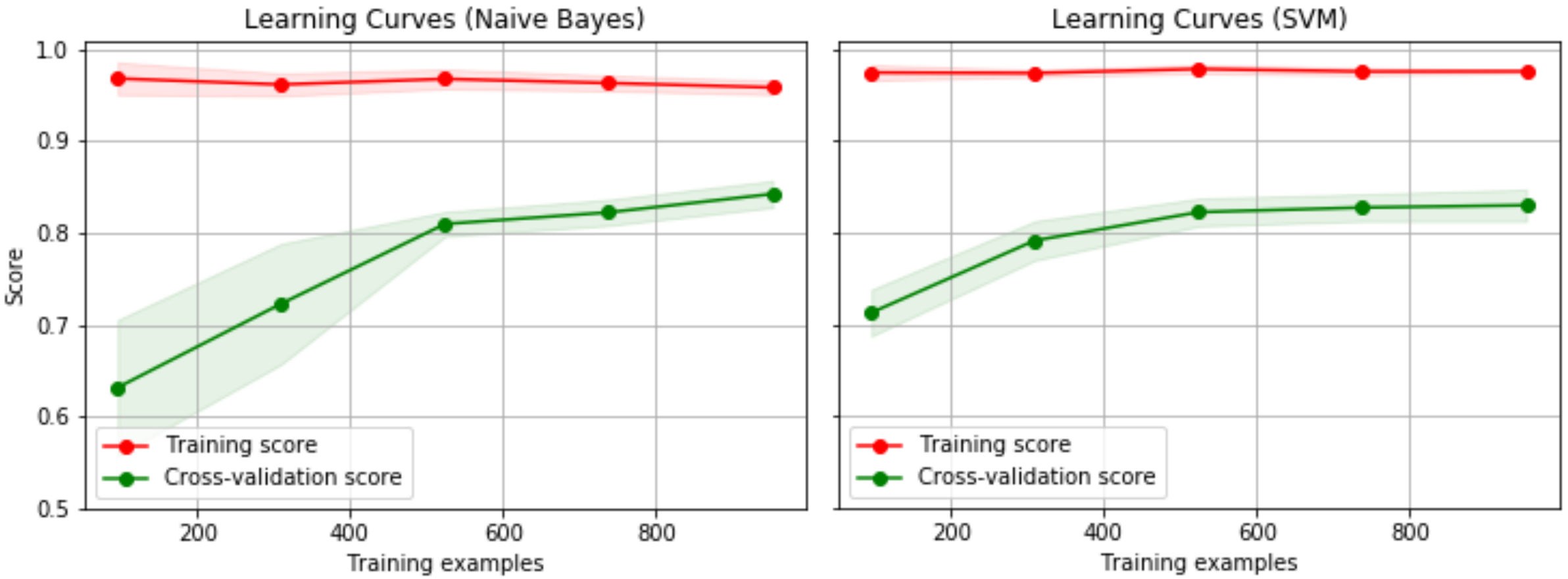
- Maximize separation
- Non linear separation is fast
- Regularization penalty for more features

Support Vector Machines

- Regularization coefficient of 1e6 determined by regularization curve
- Rbf kernel

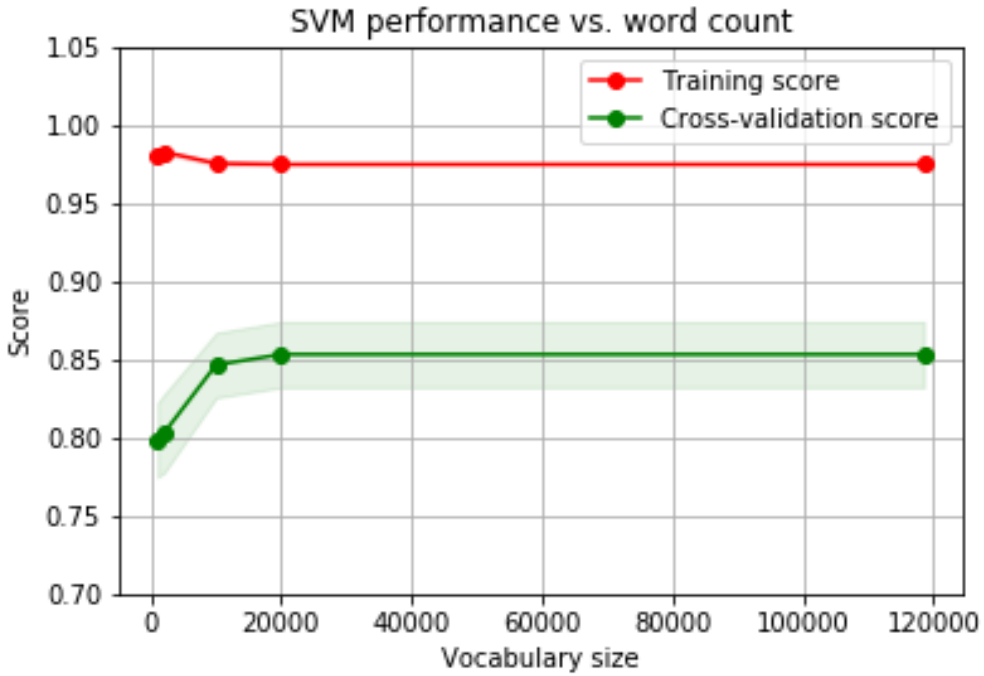


SVM learns faster than naïve bayes



Vocabulary needed for SVM

- Increasing the regularization strength allows for more features (words)
- TF-IDF representation of words



TF-IDF

Number of times the word is in the post (TF)	5
Number of times in all posts (DF)	100
TF/DF	0.05

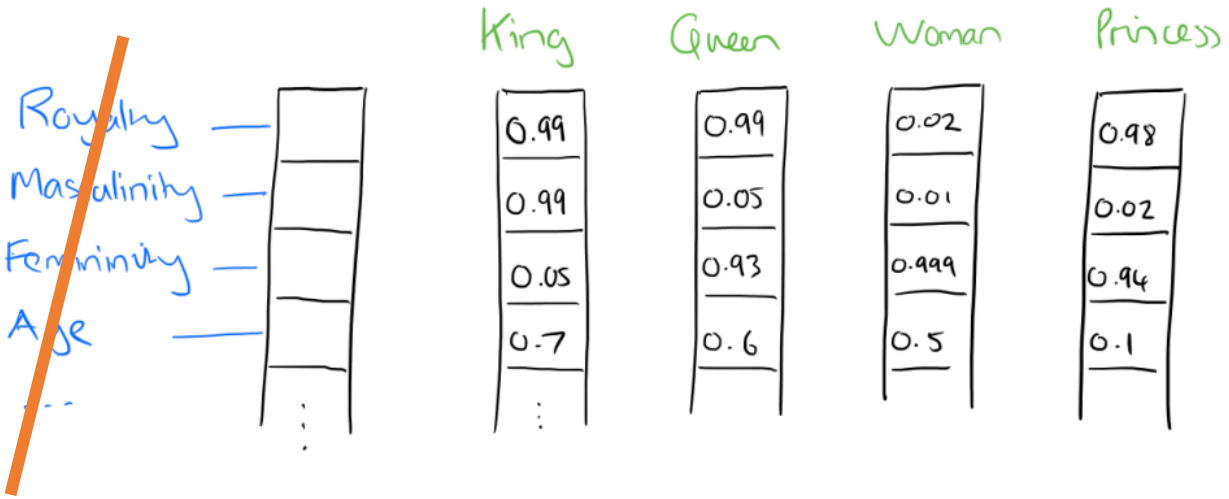
Word Vectors

Adrian Colyer
https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

TF-IDF

Number of times the word is in the post (TF)	5
Number of times in all posts (DF)	100
TF/DF	0.05

Word Vectors



Adrian Colyer
https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

Word vectors

- Previously words were represented as a single number
- Now a word is represented as a vector.
- Predict surrounding words based on center words
- Use gensim word2vec

Predict center hero
based on context heroes

Word vectors

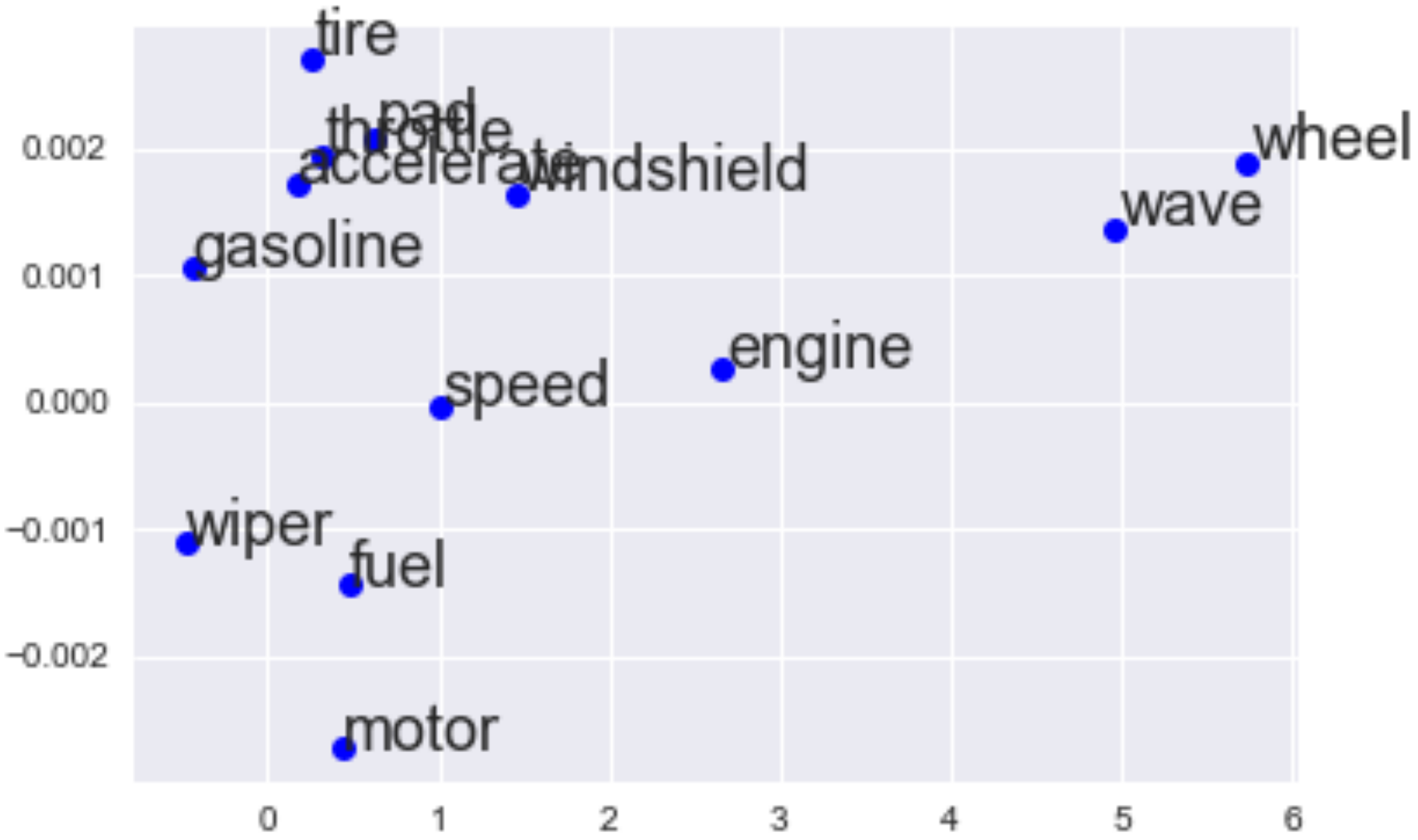
- Previously words were represented as a single number
- Now a word is represented as a vector.
- Predict surrounding words based on center words
- Use gensim word2vec

Check the vector similarity

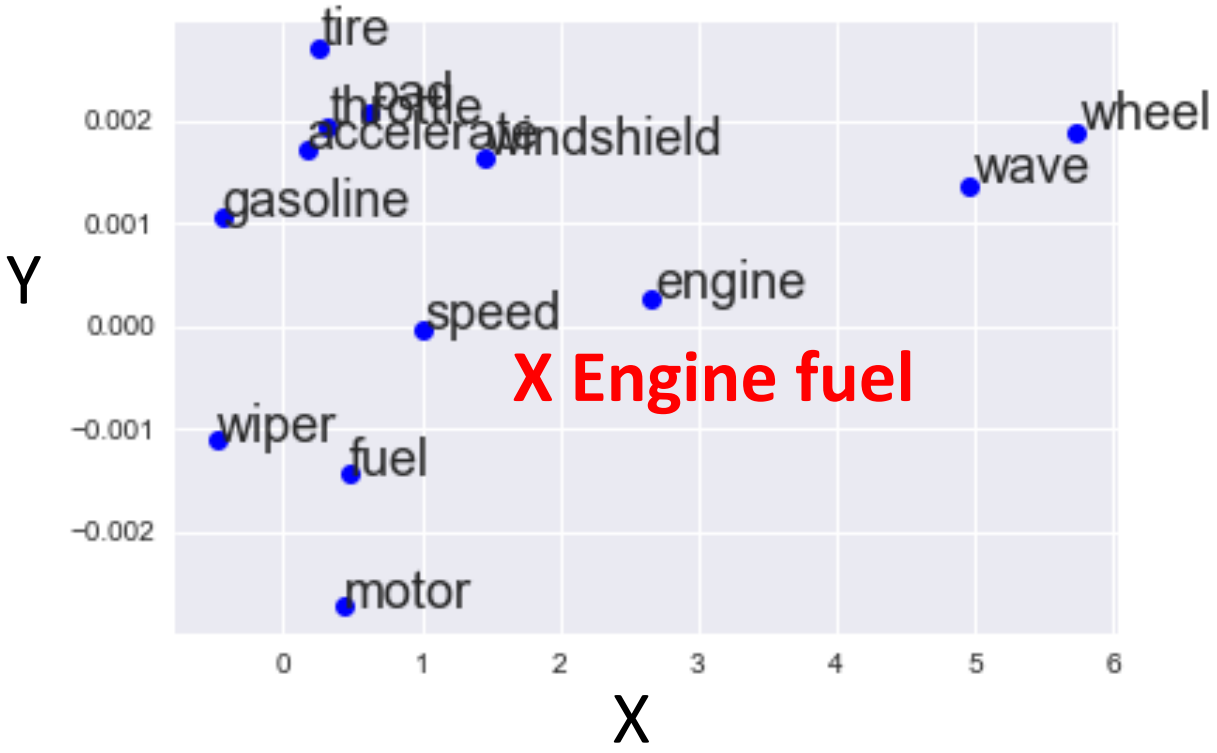
- `model_100.wv.doesnt_match(['wheel', 'car', 'wave'])`
- Helpful for debugging before applying a model

Word Dimensionality	Corpus used for training	Word that does not belong
2	Twenty newsgroups	car
100	Twenty newsgroups	wave
300	Google newsgroup	wave

Averaging word vectors in a document



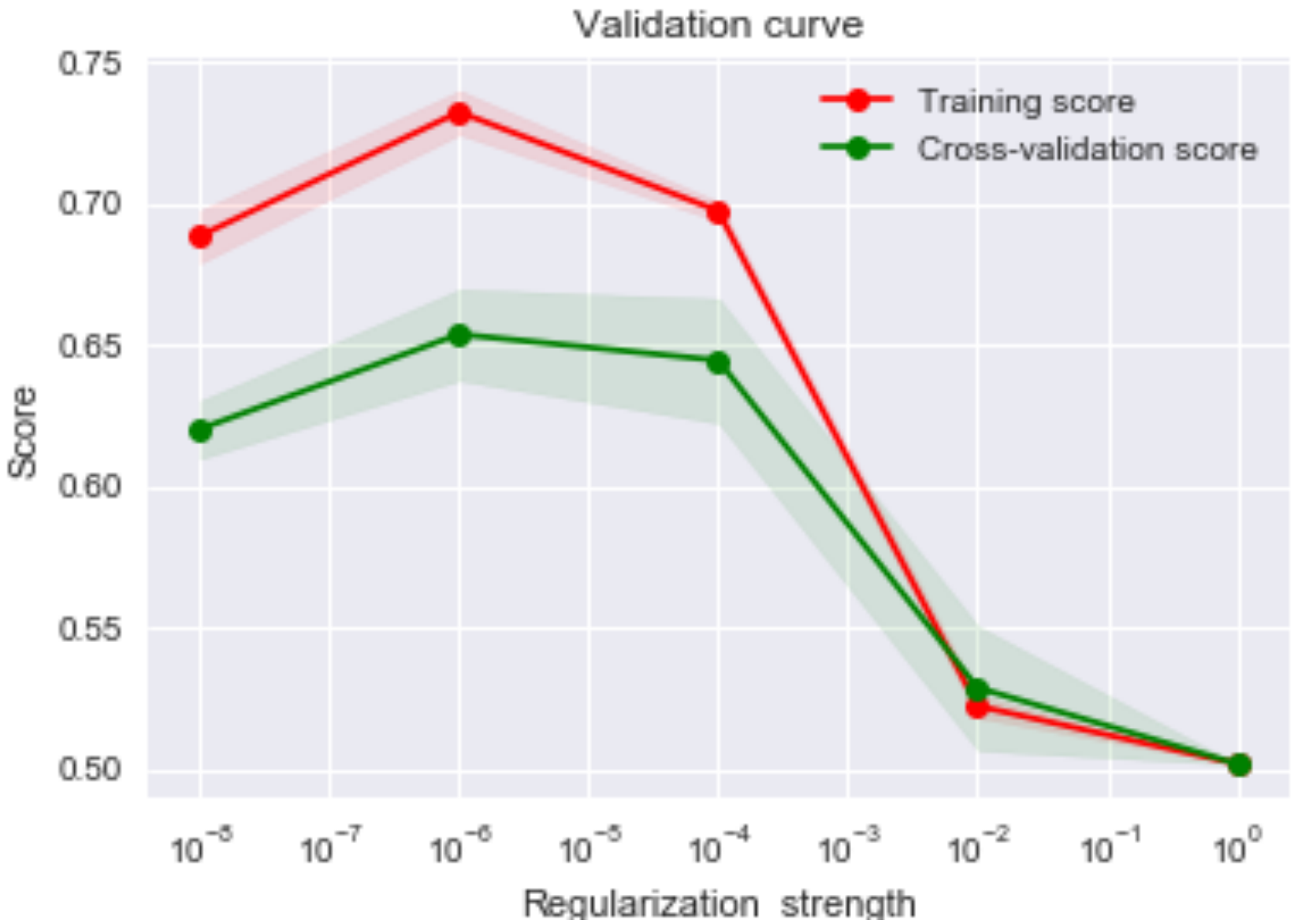
Documents can be built by averaging the vectors or stacking them



Example phrase “engine fuel”

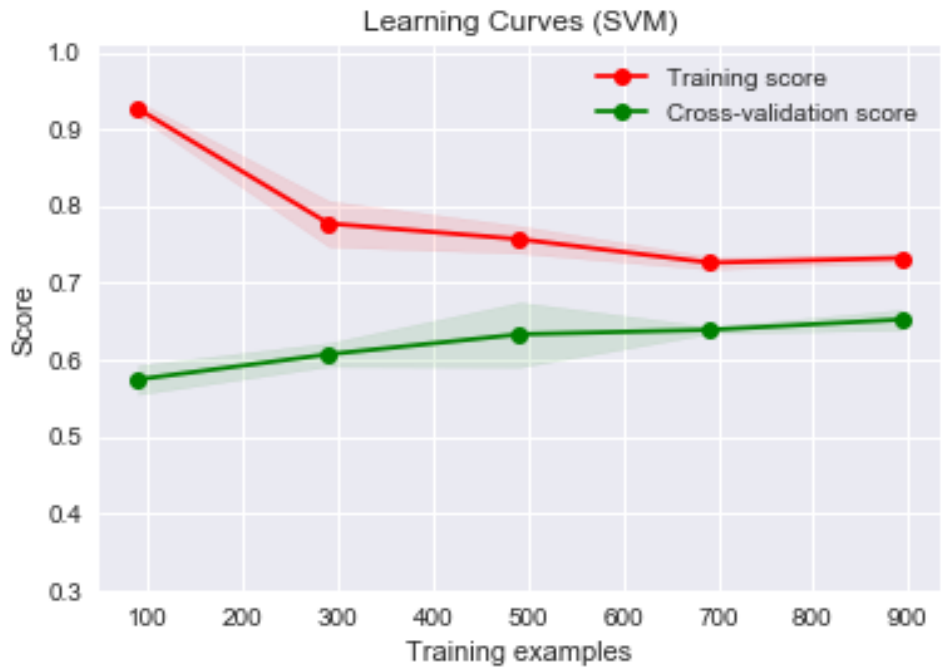
	X1	X2
engine	2.8	0.0002
fuel	0.2	-0.0013

Performance of SVM averaging word vectors

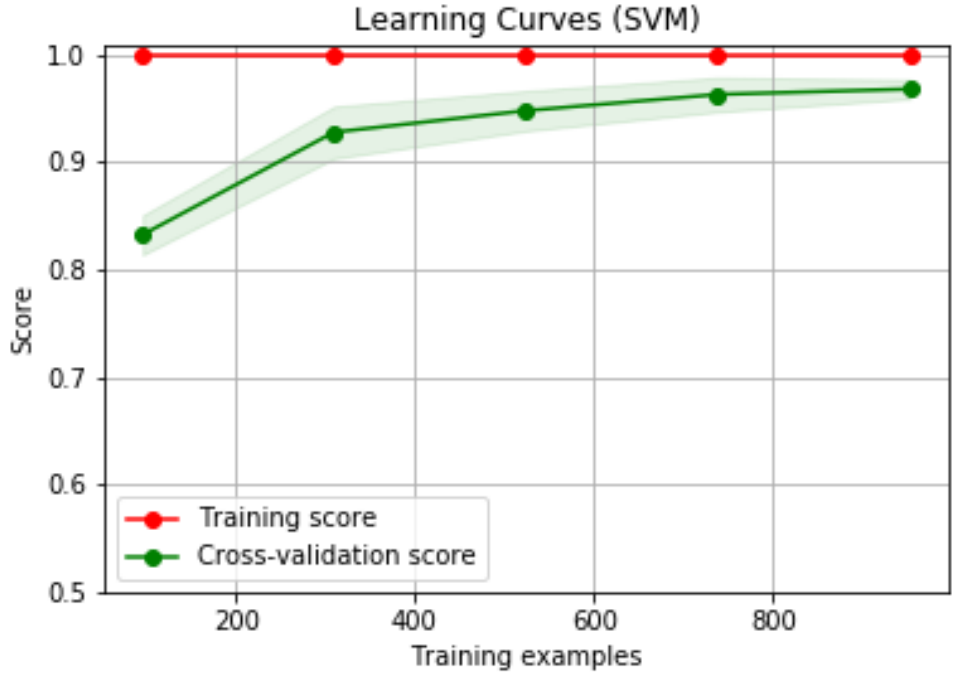


TF-IDF performs better than word embeddings

SVM using average word embeddings



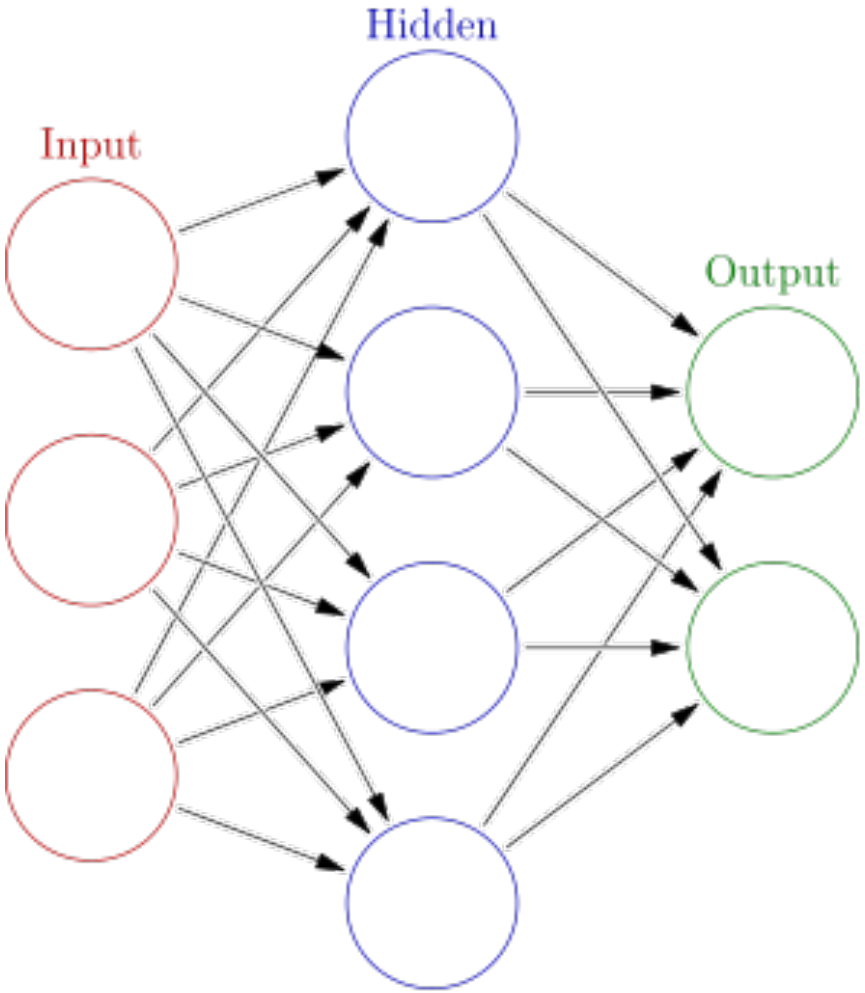
SVM using TF-IDF vectorization



Simple neural network

300 hidden units

Input word vector length = 300



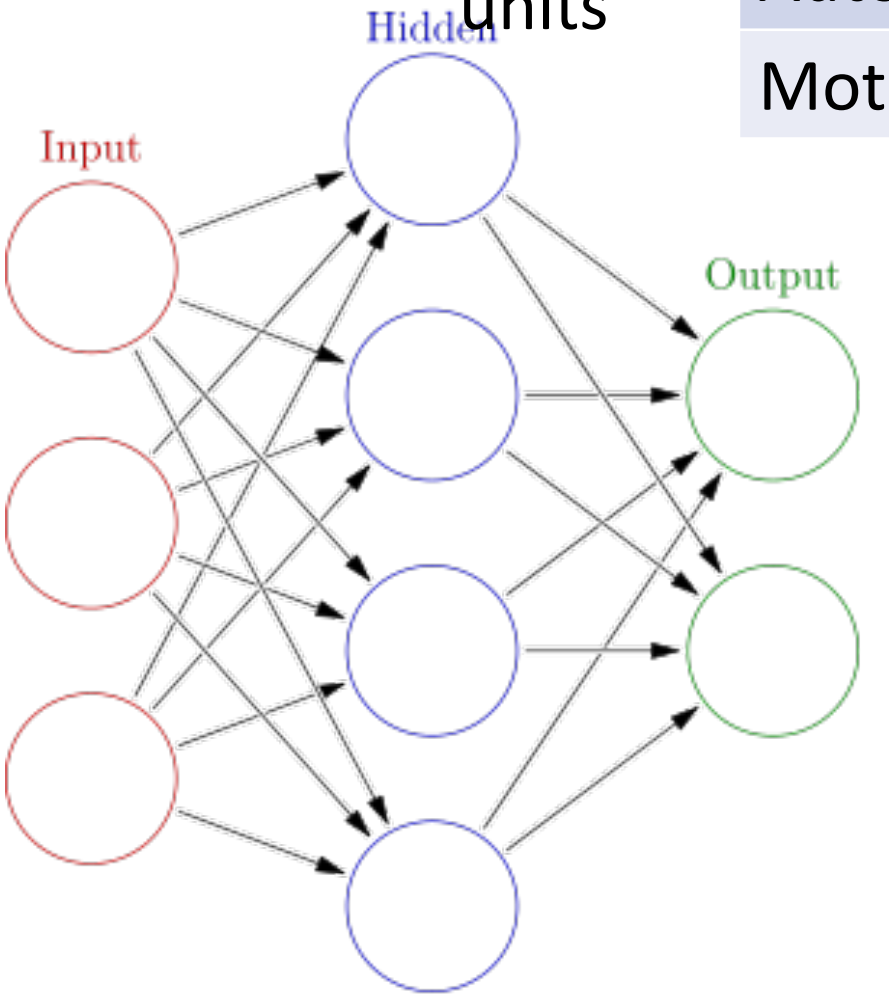
2 outputs

Simple neural network

Newsgroups	Number of posts
Autos	594
Motorcycles	598

300 hidden units

300 input units

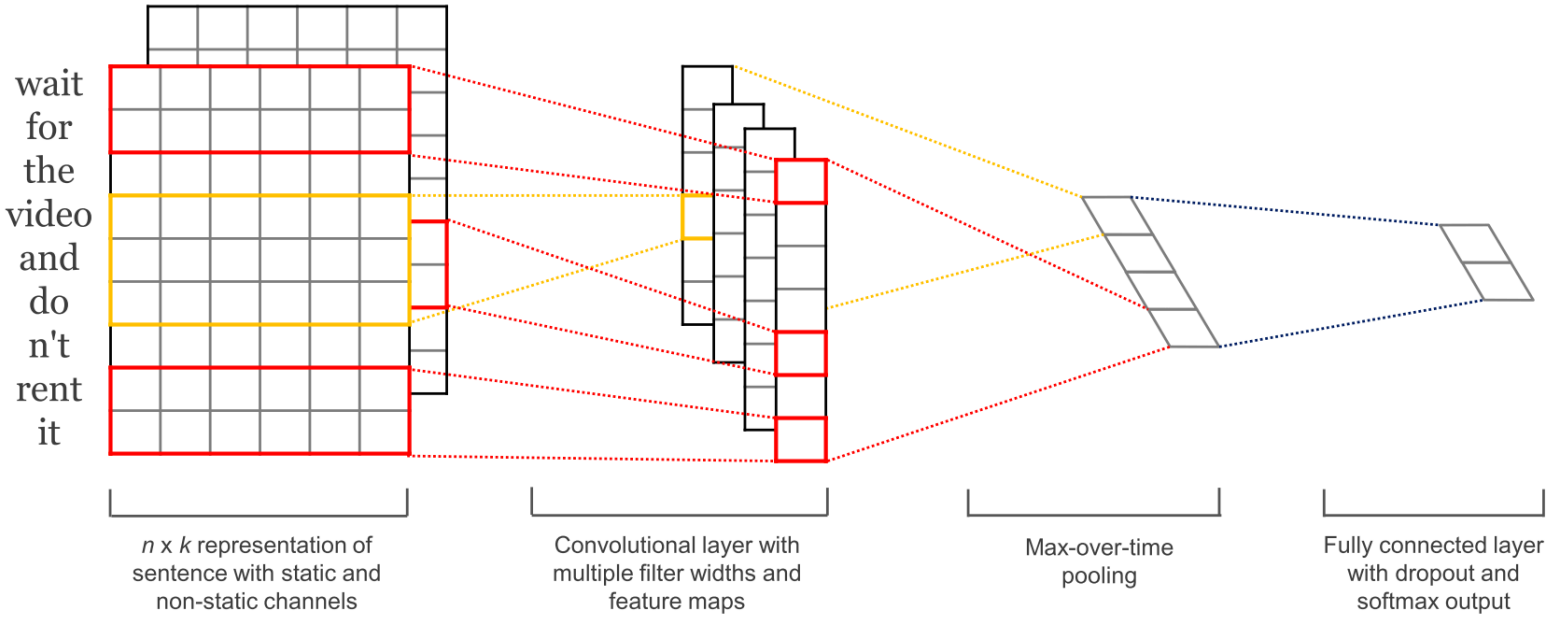


2 outputs

	Features
Input to hidden	~90K
Hidden to output	~600

Convolutional Neural Networks

- Documents are transformed to two dimensional arrays.
- Apply the same algorithm that is used for images



Convolutional Neural Networks trained on the full data set

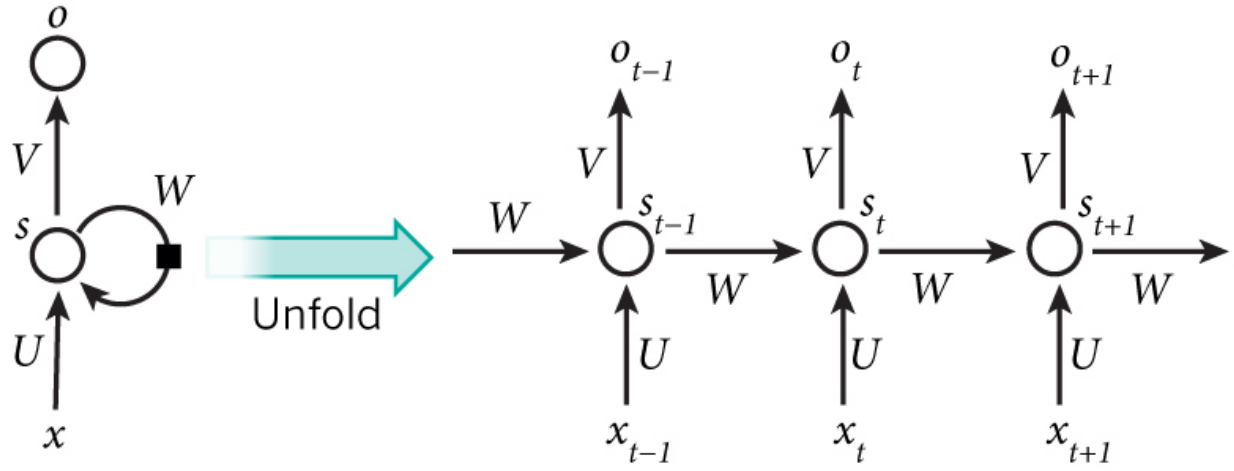
Model	Train time	Test Score
CNN with 2D word embedding	1.5h on a CPU	0.79
Two layer neural net	1.5h on a CPU	0.65
SVM	< 1sec	0.83 +/- 0.03
Naïve Bayes	< 1sec	0.84 +/- 0.03

Convolutional Neural Networks trained on 95 data points

Model	Train time	Test Score
CNN with 2D word embedding	1.5h on a CPU	0.70
Two layer neural net	1.5h on a CPU	0.70
SVM	< 1sec	0.71+/- 0.05
Naïve Bayes	< 1sec	0.63 +/- 0.14

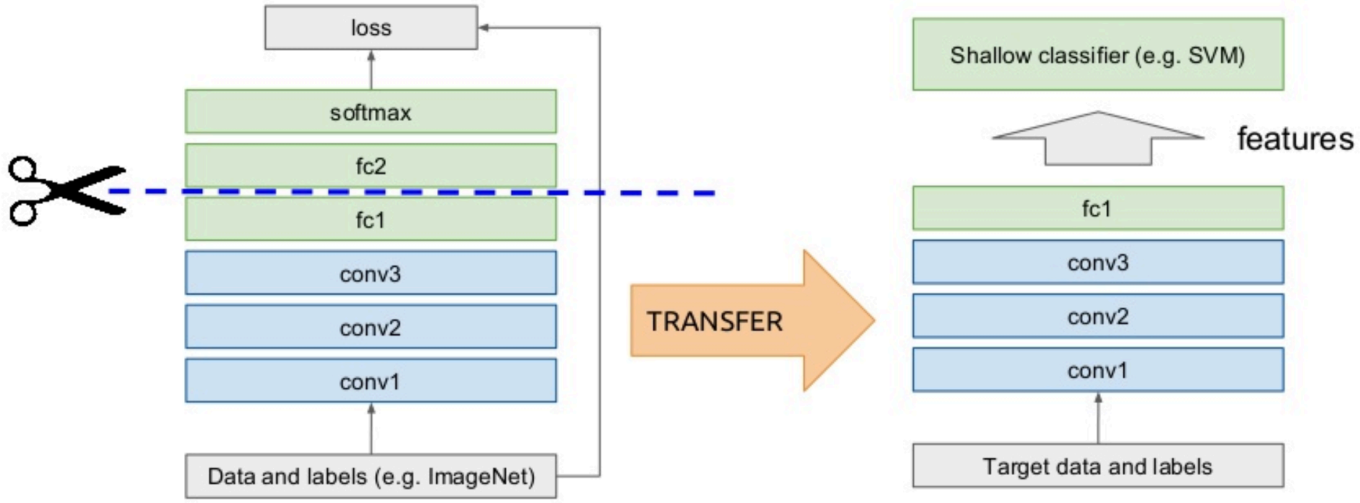
Recurrent Neural Networks

- Have memory of the history of inputs
- $s_t = f(Ux_t + Ws_{t-1})$
- f is a function to squish output (ReLU, tanh) that now depends on *previous* data



Transfer learning

1. Train one model on a larger dataset to get the initial weights.
2. Save the weights
3. Feed in the new data and adjust the weights



Kevin McGuinness, UPC 2016

Good for images, limited success for documents

Building more data for images

Arden Dertat *"Applied Deep Learning"*

Building more data through finite state grammar

“[How much time] [is it going to take for] [my uncle’s] [Salitop] to [arrive] [at [his] [house]]?”
[How much time] : How much time, How long
[is it going to take for] : is it going to take for, is it gonna take for, will it take for, does it take for,
[my uncle’s] : my uncle’s, my aunt’s, our son’s, my partner’s, our friend’s, etc.
[Saltitop] : Saltitop, Acne cream, Penicillin, etc.
[arrive] : arrive, get here, come
[at ___ ___] : at __ __, to __ __, here
[his] : his, their, her
[house] : apartment, house, office, mailbox

Alex Wahl https://blog.insightdatascience.com/automating-customer-support-when-you-lack-data-e9975fd8a053

How to get good results with not a lot of data

- Simple models (SVM)
- Transfer learning if a larger store of similar documents
- Finite state grammar if the text has similar structure

Waiting for deep learning model to train

Questions?

- Simple models (SVM)
- Transfer learning if a larger store of similar documents
- Finite state grammar if the text has similar structure

Waiting for deep learning model to train

Toby Sachs-Quintana

tobys@alumni.stanford.edu