Natural Language Processing And Text Analysis

Python Course

Dirk Hovy

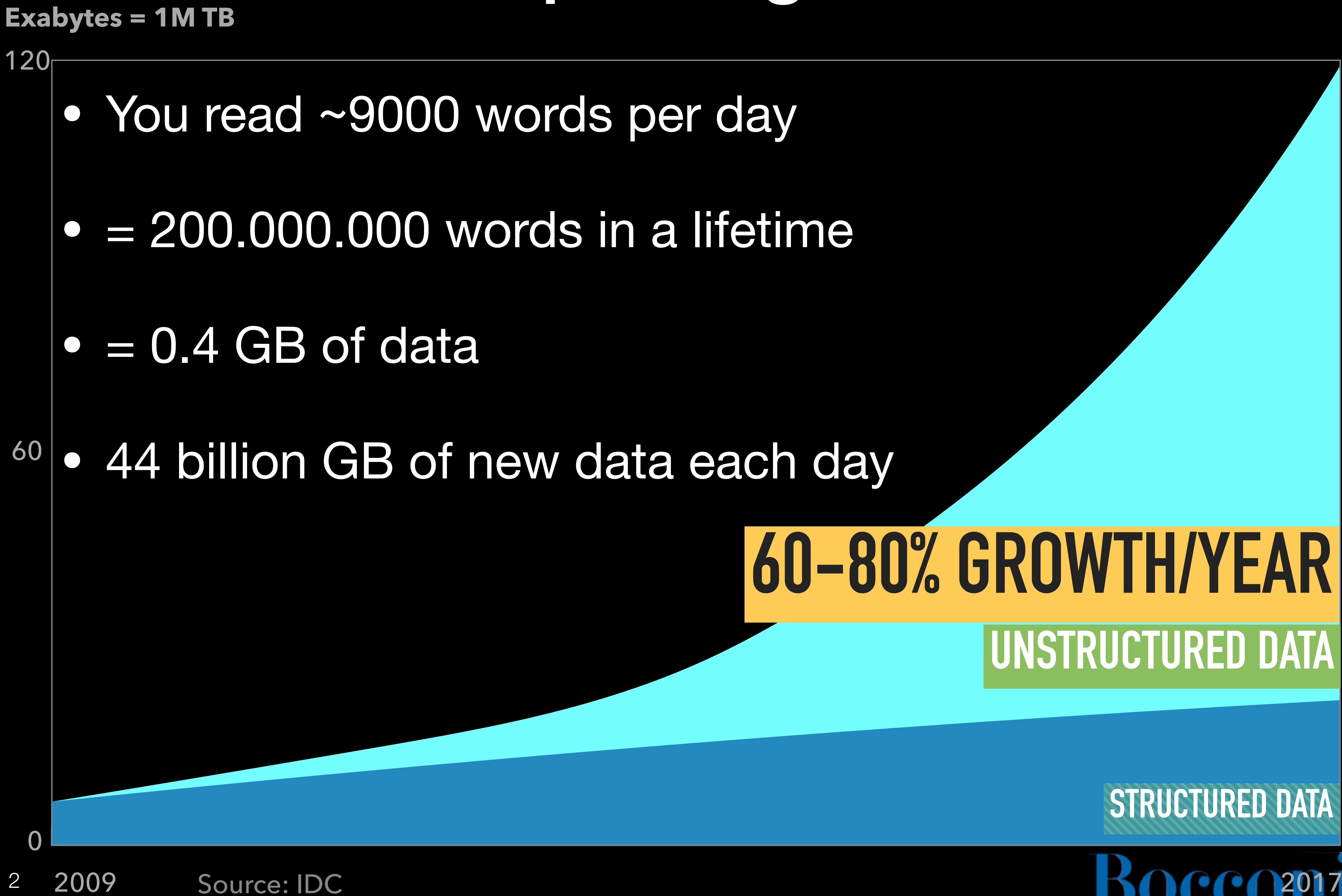
dirk.hovy@unibocconi.it

@dirk_hovy

Bocconi

Text is an exploding data source

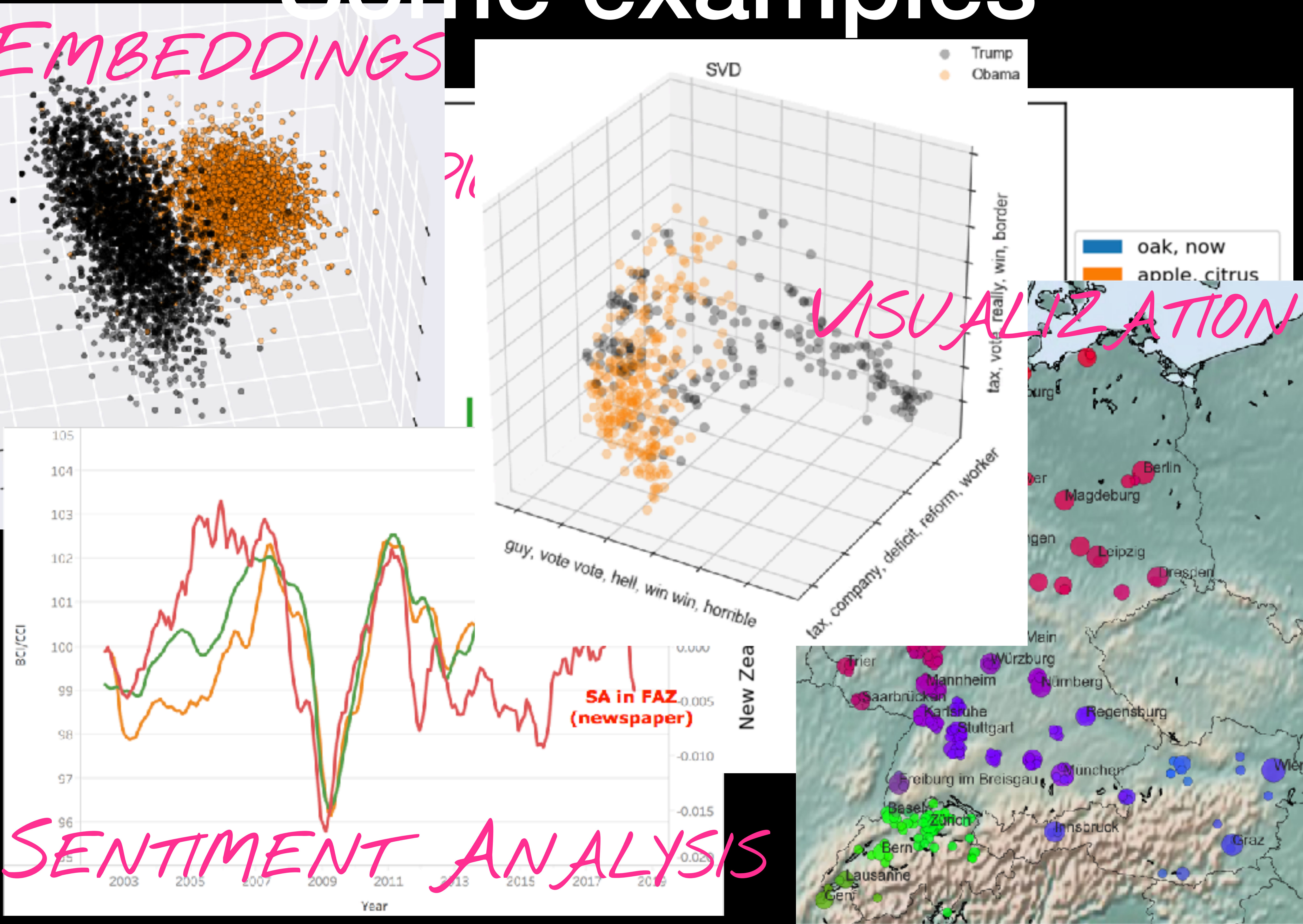
- You read ~9000 words per day
- = 200.000.000 words in a lifetime
- = 0.4 GB of data
- 44 billion GB of new data each day



NLP is booming

Source: Tractica

Some examples



Today's Goals

- Learn to apply **text classification**
- Understand **bag of words** (BOW) representations and **TFIDF**
- Learn about **preprocessing**
- Understand **evaluation metrics**
- Understand **regularization**

Ham or Spam?

From: offr4u@rsph.com
Subject: Unique wealth offerings
To: dirk.hovy@unibocconi.it

Greetings dear friend

We have an amazing offer 4U: Click here to get access to a free consultation for serious wealth benefits! Urgent: offer expires soon.
Works guaranteed! Triple your income.

Spam terms:

- 4U
- click
- amazing
- free
- guarantee
- offer
- urgent
- dear friend
- income
- serious

Pre-processing

Pre-processing steps

```
<div id="text">I've been in New York
in 2011, but didn’t like it. I
preferred Los Angeles.</div>
```

GOAL: MINIMIZE VARIATION

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
I've been in New York in
2011, but didn’t like
it. I preferred Los
Angeles.
```


Bocconi

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
I've been in New York in
2011, but didn’t like
it.

I preferred Los Angeles.
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
I ’ve been in New York
in 2011 , but did n’t
like it .

I preferred Los
Angeles .
```


Bocconi

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
i ‘ve been in new york
in 0000 , but did n’t
like it .

i preferred los
angeles .
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
i have be in new york in
0000 , but do not like
it .

i prefer los angeles .
```


Bocconi

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
i new york 0000 , like .

i prefer los angeles .
```


Bocconi

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
new york 0000 like

prefer los angeles
```

CONTENT = (NOUN, VERB, NUM)

Bocconi

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
new_york 0000 like

prefer los_angeles
```

Pre-processing steps

<div id="text">I've been in New York in 2011, but didn’t like it. I preferred Los Angeles.</div>

MINIMIMAL VARIATION

"BAG OF WORDS"

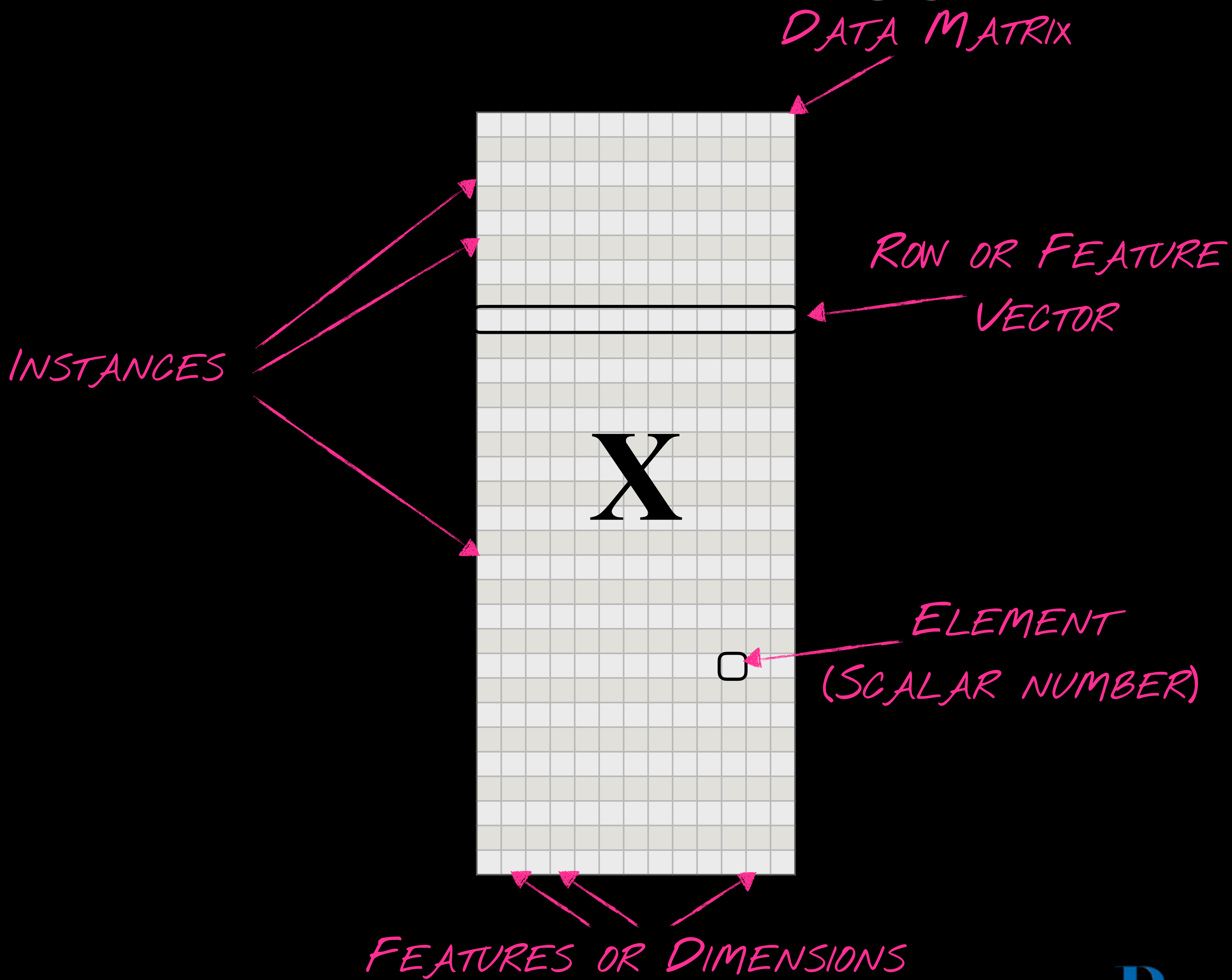
new_york 0000 like

prefer los_angeles

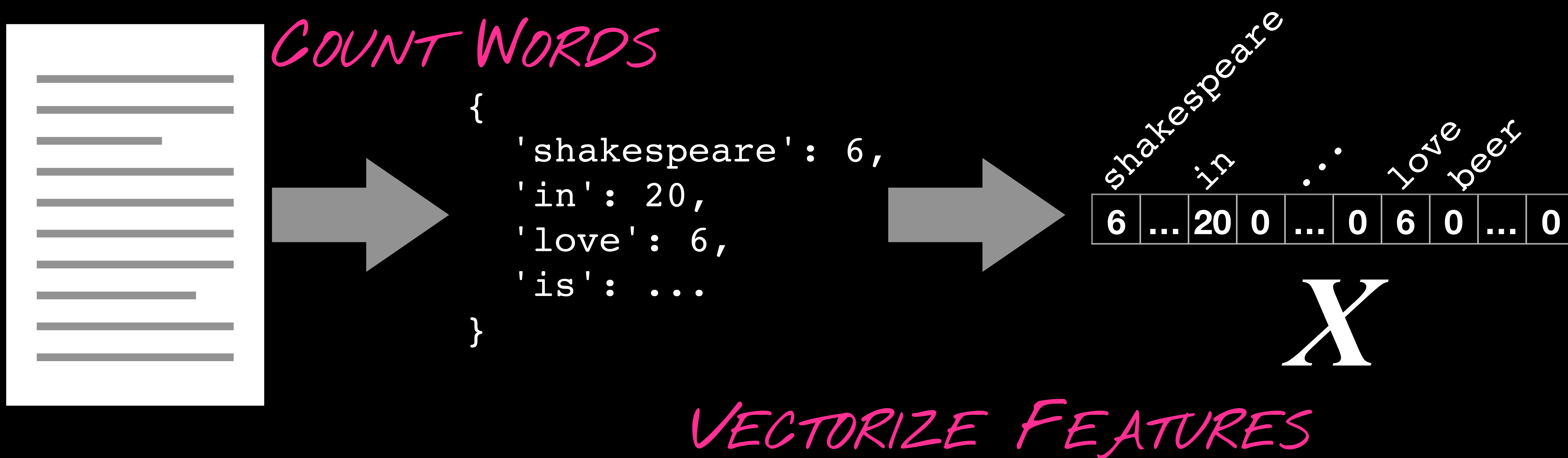
Bocconi

Discrete Representations

Terminology



Bags of words (BOW)

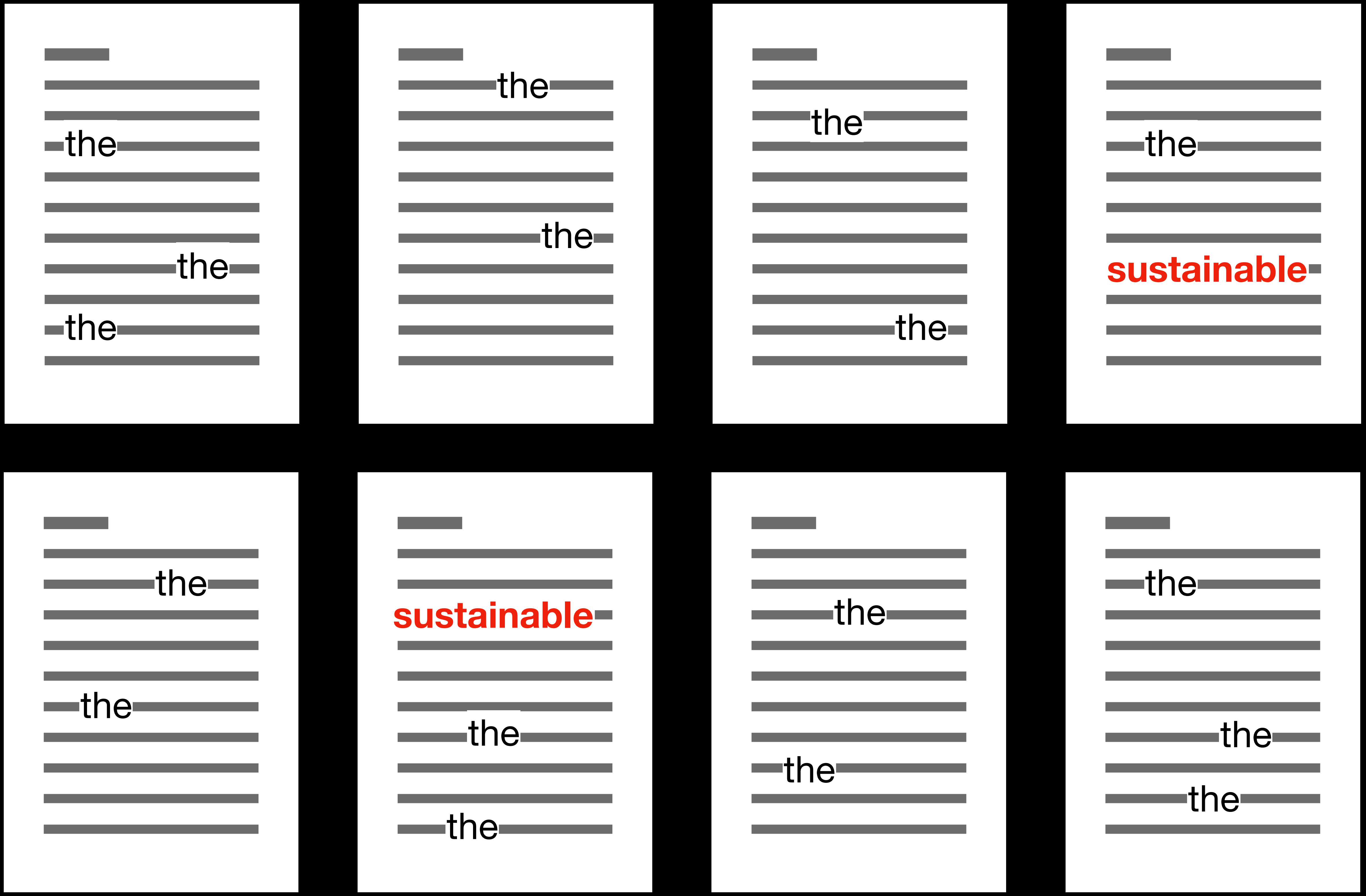


N-grams

"As Gregor Samsa awoke one morning from uneasy dreams, he found himself transformed in his bed into a gigantic insect-like creature."

Unigrams	`As, Gregor, Samsa, awoke, one, morning, from, uneasy, dreams, ...`
Bigrams	`As_Gregor, Gregor_Samsa, Samsa_awoke, awoke_one, one_morning, ...`
Trigrams	`As_Gregor_Samsa, Gregor_Samsa_awoke, Samsa_awoke_one, awoke_one_morning, ...`
4-grams	`As_Gregor_Samsa_awoke, Gregor_Samsa_awoke_one, Samsa_awoke_one_morning, ...`

Some Words are Just More Interesting…

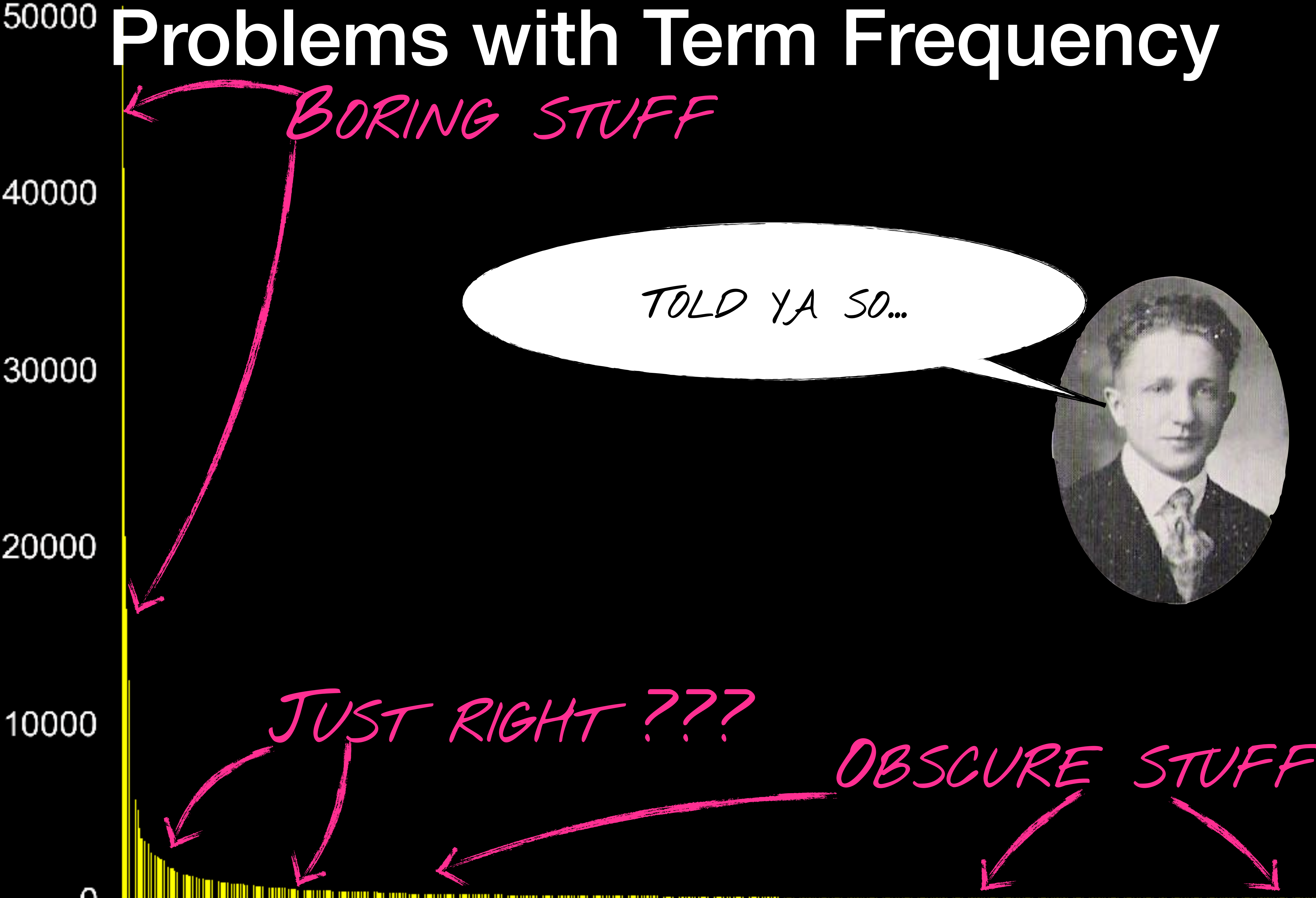


Karen Spärck Jones

1935–2007

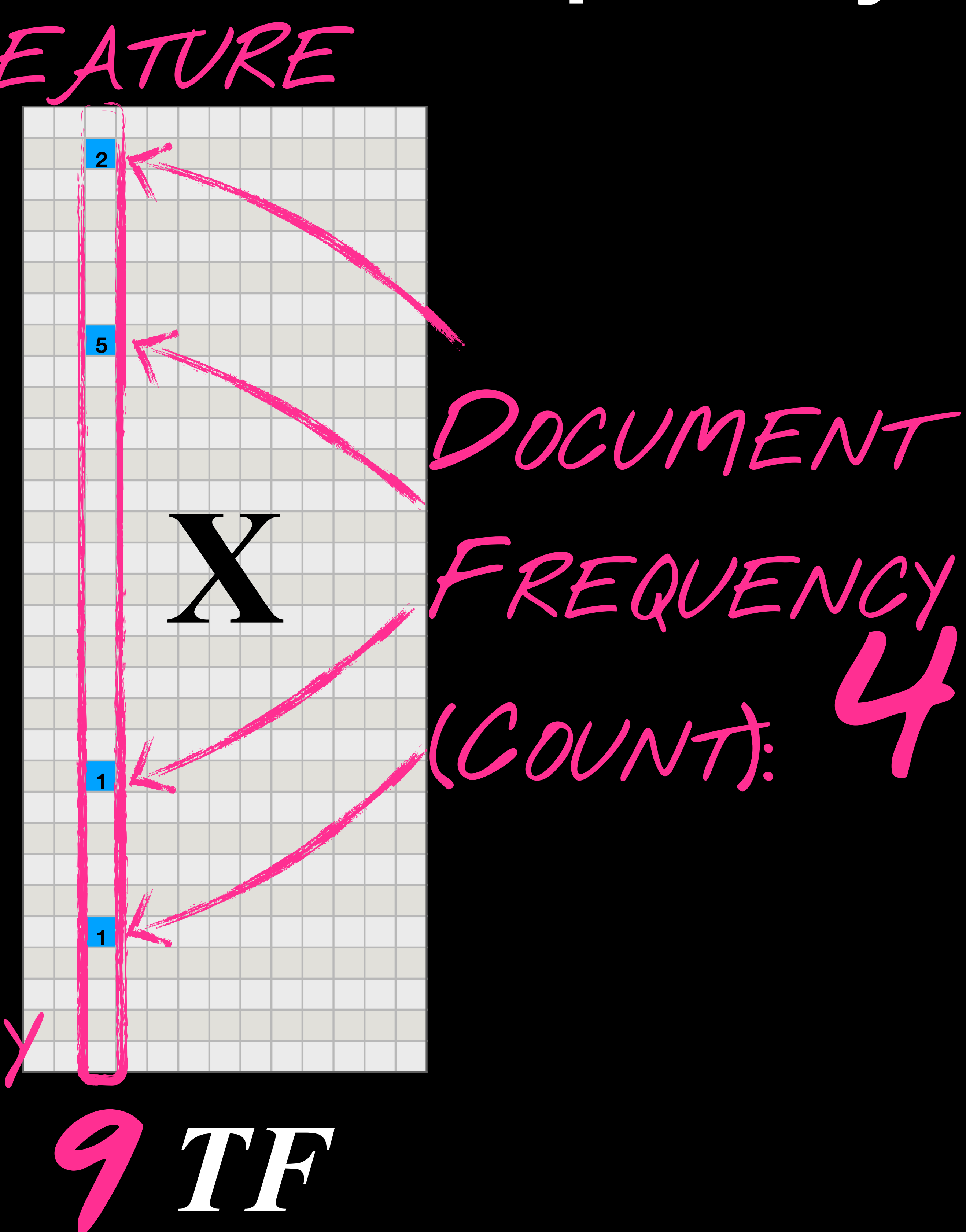
- Became a teacher before starting CS career at Cambridge
- Laid the foundation for modern NLP, Google Search, text classification
- Campaigned for more women in CS
- Namesake of prestigious CS prize

Problems with Term Frequency

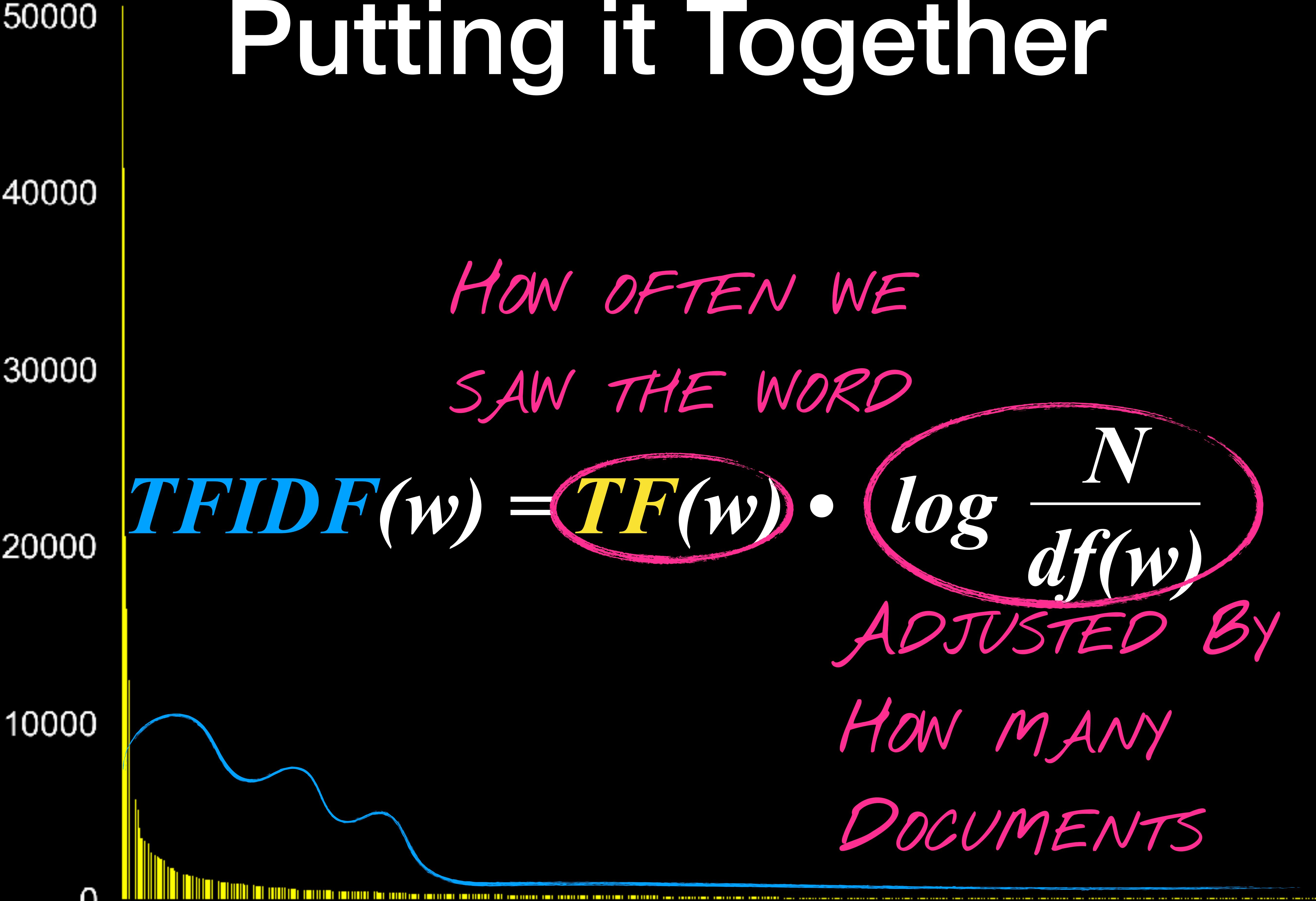


Document and Term Frequency

$$IDF = \log \frac{N}{df(w)}$$



Putting it Together

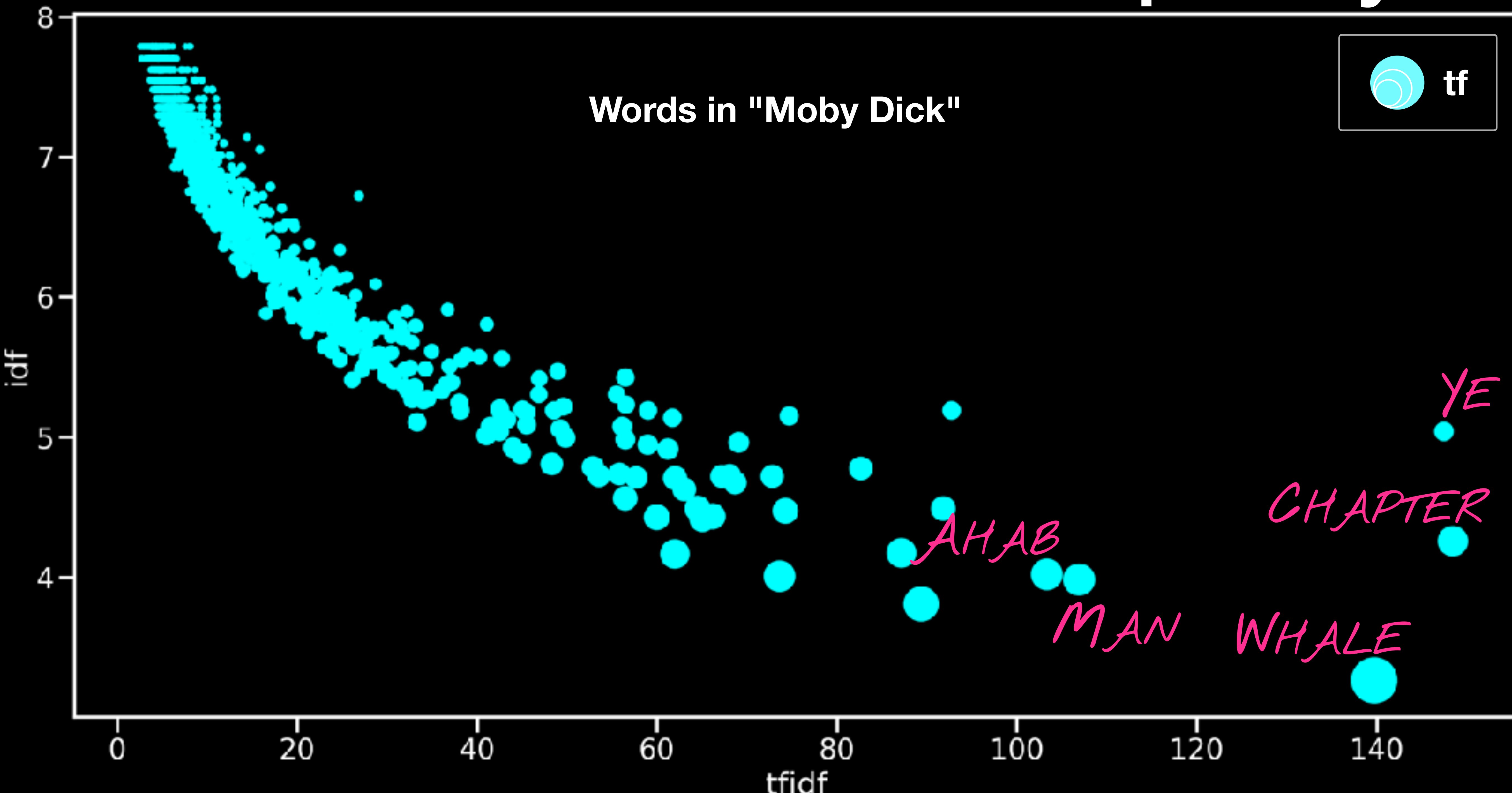


HOW OFTEN WE SAW THE WORD

$$TFIDF(w) = TF(w) \cdot \log \frac{N}{df(w)}$$

ADJUSTED BY HOW MANY DOCUMENTS

Document and Term Frequency



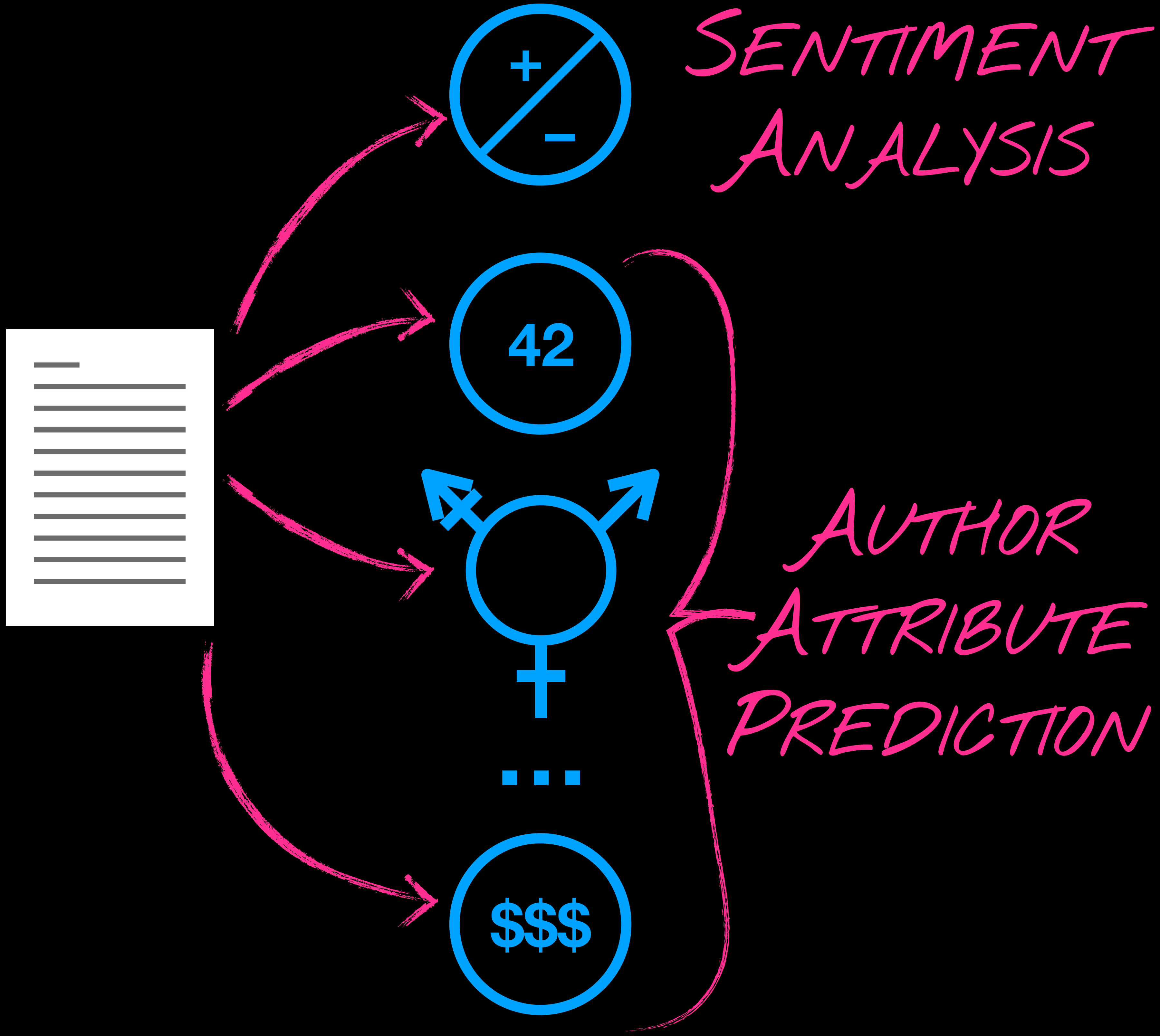
word	tf	idf	tfidf
ye	467	4.257380	148.497079
chapter	171	5.039475	147.504638
whale	1150	3.262357	139.755743
man	525	3.982412	106.932953
ahab	511	4.019453	103.357774

Text Classification

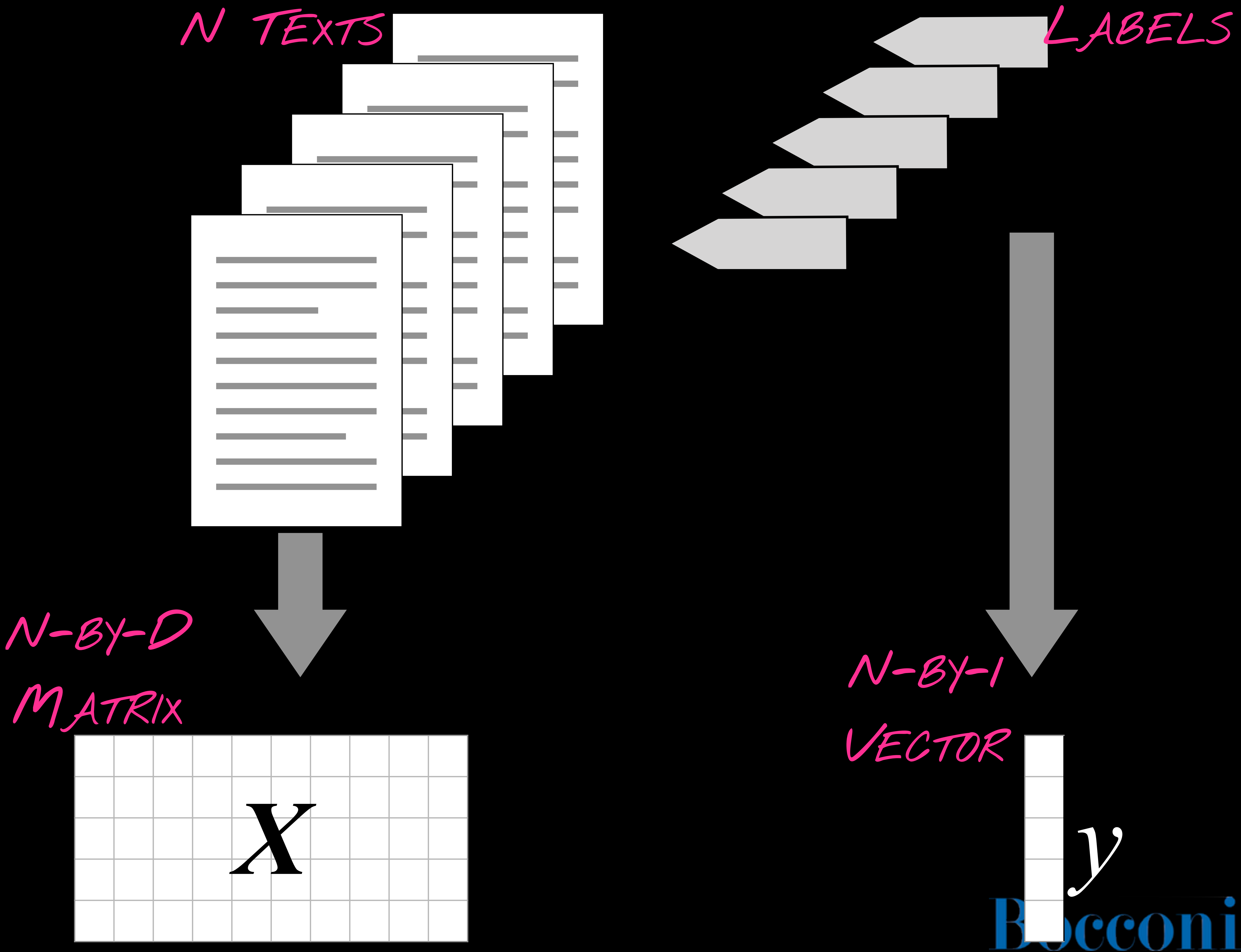
Bocconi

Text Classification

Examples



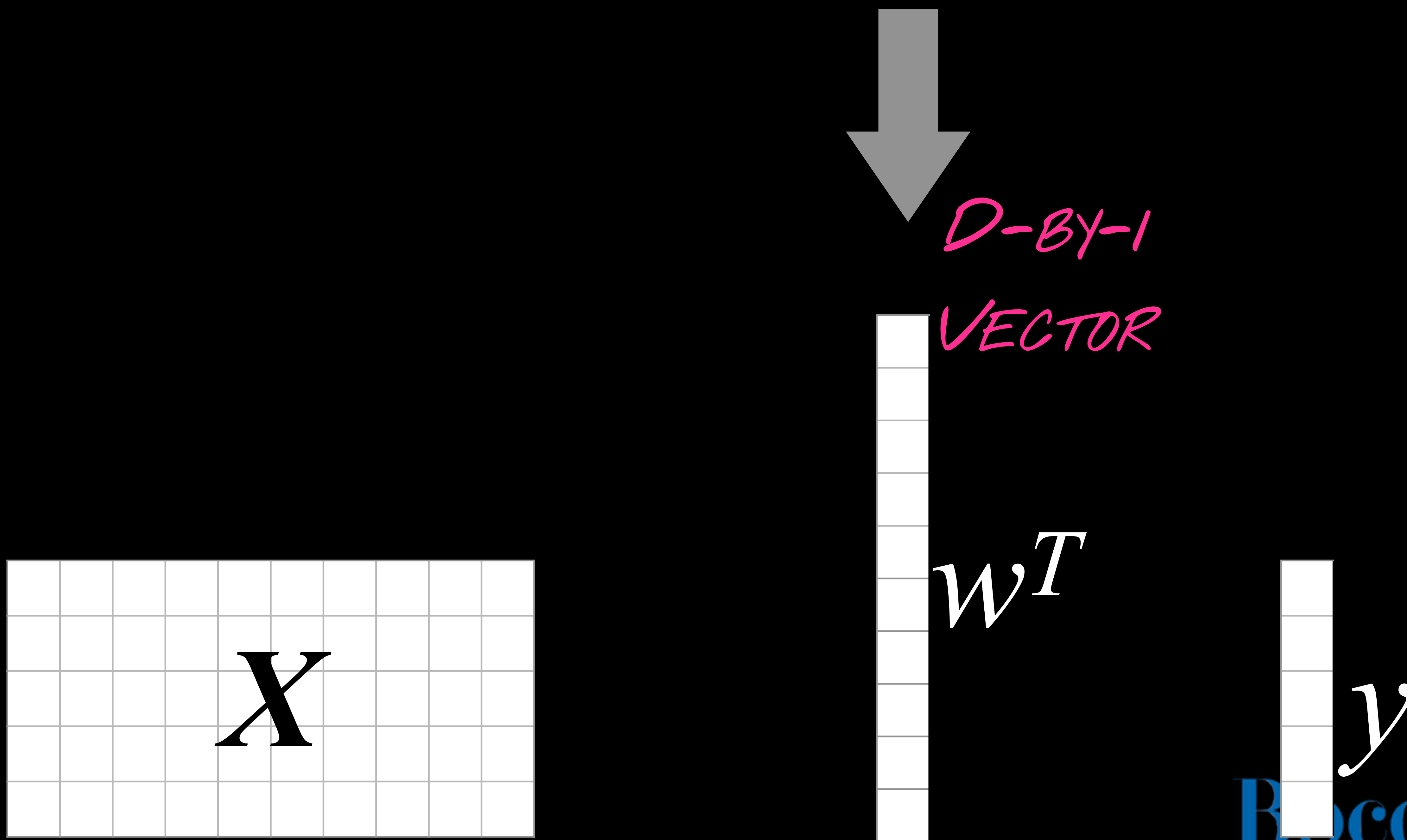
Bocconi



N TEXTS
LABELS
N-BY-D
MATRIX
X
N-BY-1
VECTOR
y

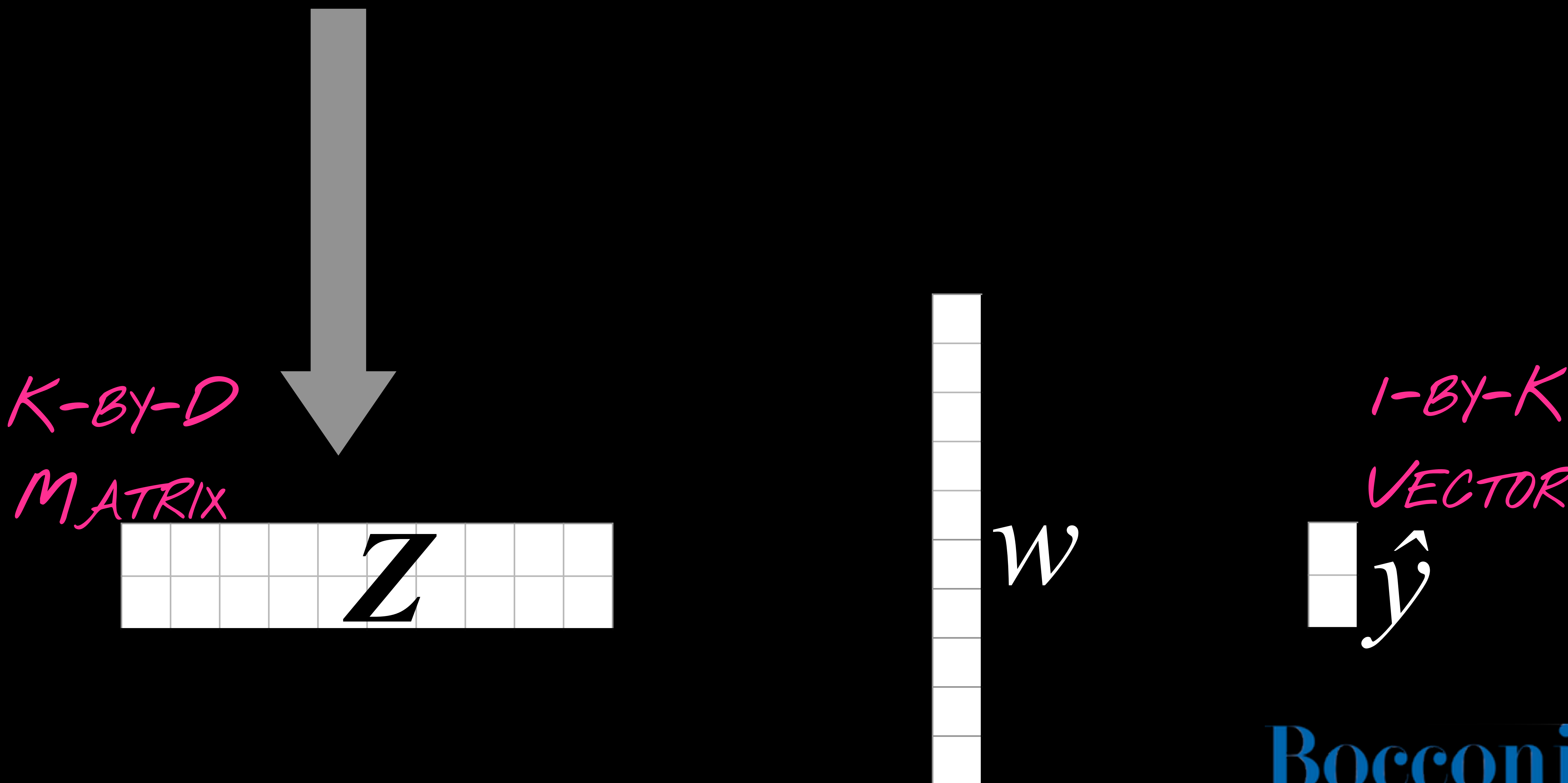
Fitting

$$f(\boldsymbol{X}) = y$$



Predicting

$$f(\mathbf{Z}) = \mathbf{Z}\, w^T = \hat{y}$$

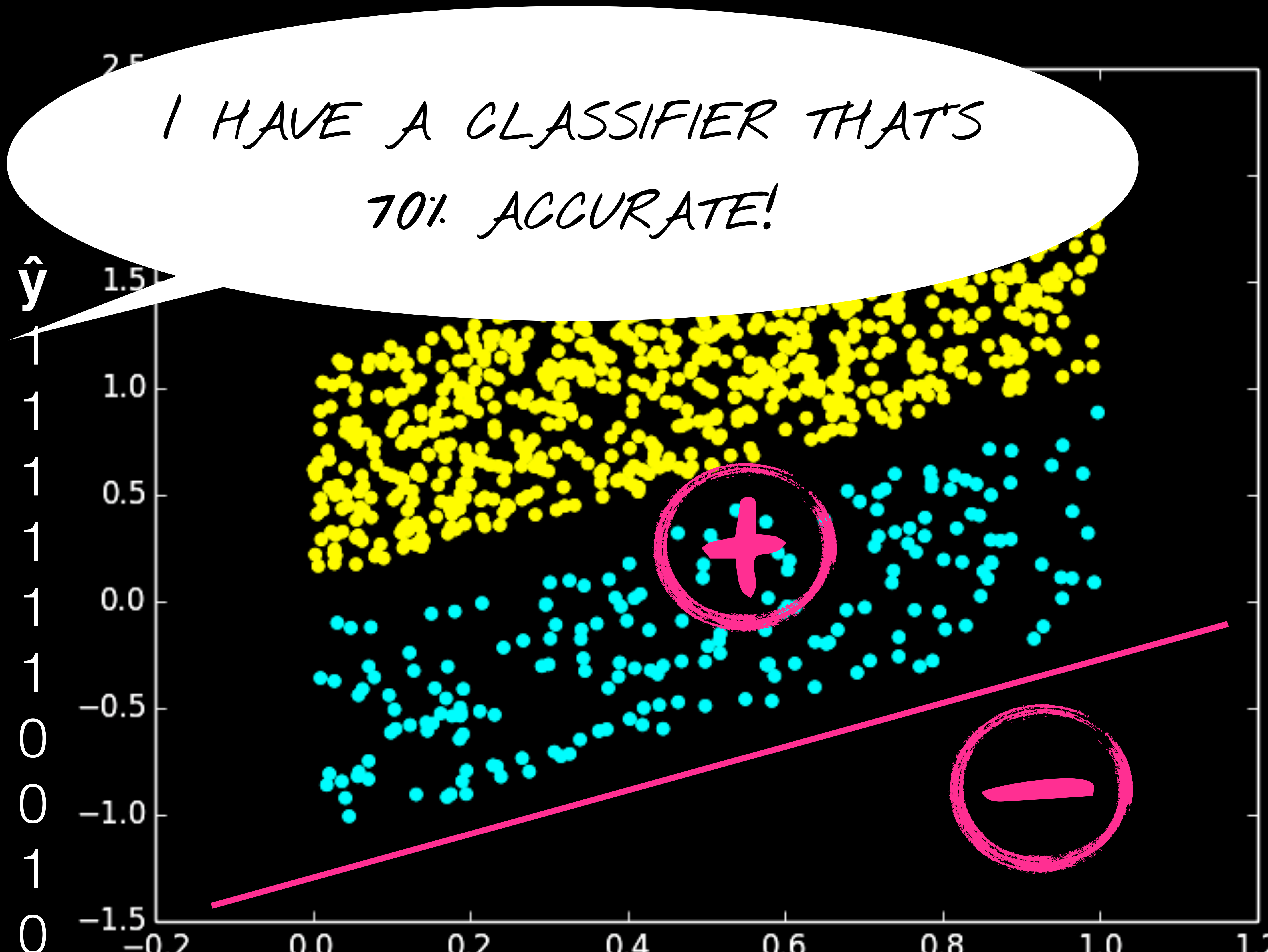


Evaluating Performance

Bocconi

Performance Problems

x	y	$\hat{y}$
frog	1	1
deer	1	1
wolf	1	1
dog	1	1
bear	1	1
fish	1	1
bird	1	0
cat	1	0
stone	0	1
tree	0	0



A 70% ACCURATE CLASSIFIER

		predicted	
gold		1	0
	1	TP	FN
	0	FP	TN

True and False

accuracy = (TP+TN) / (P + N)

precision = TP / (TP + FP)

recall = TP / (TP + FN)

F1 = 2 (prec x rec) / (prec + rec)

TARGET = ANIMAL

x	**y**	**ŷ**	
frog	1	1	
deer	1	1	
wolf	1	1	true positive
dog	1	1	
bear	1	1	
fish	1	1	
bird	1	0	false negative
cat	1	0	
stone	0	1	false positive
tree	0	0	true negative

ACCURACY = 7/10 = 0.7

PRECISION = 6/7 = 0.86

RECALL = 6/8 = 0.75

F1 = 0.81

Changing Target Class

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

TARGET = THING

x	y	ŷ	
frog	0	0	true negative
deer	0	0	
wolf	0	0	
dog	0	0	
bear	0	0	
fish	0	0	
bird	0	1	false positive
cat	0	1	
stone	1	0	false negative
tree	1	1	true positive

accuracy = (TP+TN) / (P + N)

precision = TP / (TP + FP)

recall = TP / (TP + FN)

F1 = 2 (prec x rec) / (prec + rec)

ACCURACY = 7/10 = 0.7

PRECISION = 1/3 = 0.33

RECALL = 1/2 = 0.5

F1 = 0.4

Bocconi

MICRO Averaging

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

WEIGH BY CLASS SIZE

accuracy = (TP+TN) / (P + N)

precision = TP / (TP + FP)

recall = TP / (TP + FN)

F1 = 2 (prec x rec) / (prec + rec)

ANIMAL

x	y	ŷ
frog	1	1
deer	1	1
wolf	1	1
dog	1	1
bear	1	1
fish	1	1
bird	1	1
cat	1	0
stone	0	1
tree	0	0

THING

x	y	ŷ
frog	0	0
deer	0	0
wolf	0	0
dog	0	0
bear	0	0
fish	0	0
bird	0	0
cat	0	1
stone	1	0
tree	1	1

ACC = (7+7)/(10+10) = 14/20 = 0.7

PREC = (6+1)/(7+3) = 7/10 = 0.7

REC = (6+1)/(8+2) = 7/10 = 0.7

F1 = 0.7

Bocconi

MACRO Averaging

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

WEIGH ALL CLASSES EQUALLY

accuracy = (TP+TN) / (P + N)

precision = TP / (TP + FP)

recall = TP / (TP + FN)

F1 = 2 (prec x rec) / (prec + rec)

ANIMAL

x	y	ŷ
frog	1	1
deer	1	1
wolf	1	1
dog	1	1
bear	1	1
fish	1	1
bird	1	1
cat	1	0
stone	0	1
tree	0	0

THING

x	y	ŷ
frog	0	0
deer	0	0
wolf	0	0
dog	0	0
bear	0	0
fish	0	0
bird	0	0
cat	0	1
stone	1	0
tree	1	1

ACC = (0.7 + 0.7) / 2 = 0.7

PREC = (0.86 + 0.33) / 2 = 0.6

REC = (0.5 + 0.75) / 2 = 0.63

F1 = 0.61

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

Baseline: Total Recall

PREDICT MAJORITY CLASS FOR ALL

accuracy = (TP+TN) / (P + N)

precision = TP / (TP + FP)

recall = TP / (TP + FN)

F1 = 2 (prec x rec) / (prec + rec)

TARGET = ANIMAL

x	**y**	**ŷ**	
frog	1	1	
deer	1	1	
wolf	1	1	
dog	1	1	true positive
bear	1	1	
fish	1	1	
bird	1	1	
cat	1	1	
stone	0	1	
tree	0	1	false positive

ACCURACY = 8/10 = 0.8

PRECISION = 8/10 = 0.8

RECALL = 8/8 = 1.0

F1 = 0.9

Metrics Overview

- **accuracy** can be too general
- **precision** and **recall** are per-class measures
- **precision** = how many of instances labeled as target class are actually *in* target class?
- **recall** = how many of *all* target class instances in data identified correctly?
- **F1** = symmetric mean of precision and recall

Beware: Overgeneralization

False Positives

June 6 2019

Dear Ms Hovy,

Congratulations on reaching retirement age!

Also, you're on a no-fly list because of your political views and religious beliefs.

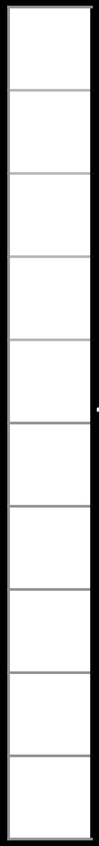
Regularization

Regularization

$$y = \boldsymbol{X}\, w^T + e$$

D-BY-1

VECTOR



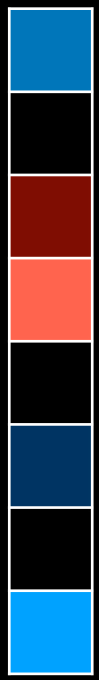
w^T

$$||w||$$

Regularization Norms

L1 NORM

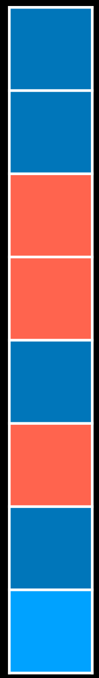
$$||W||_1 = \sum_{i=1}^{N} |w_i|$$



SPARSE

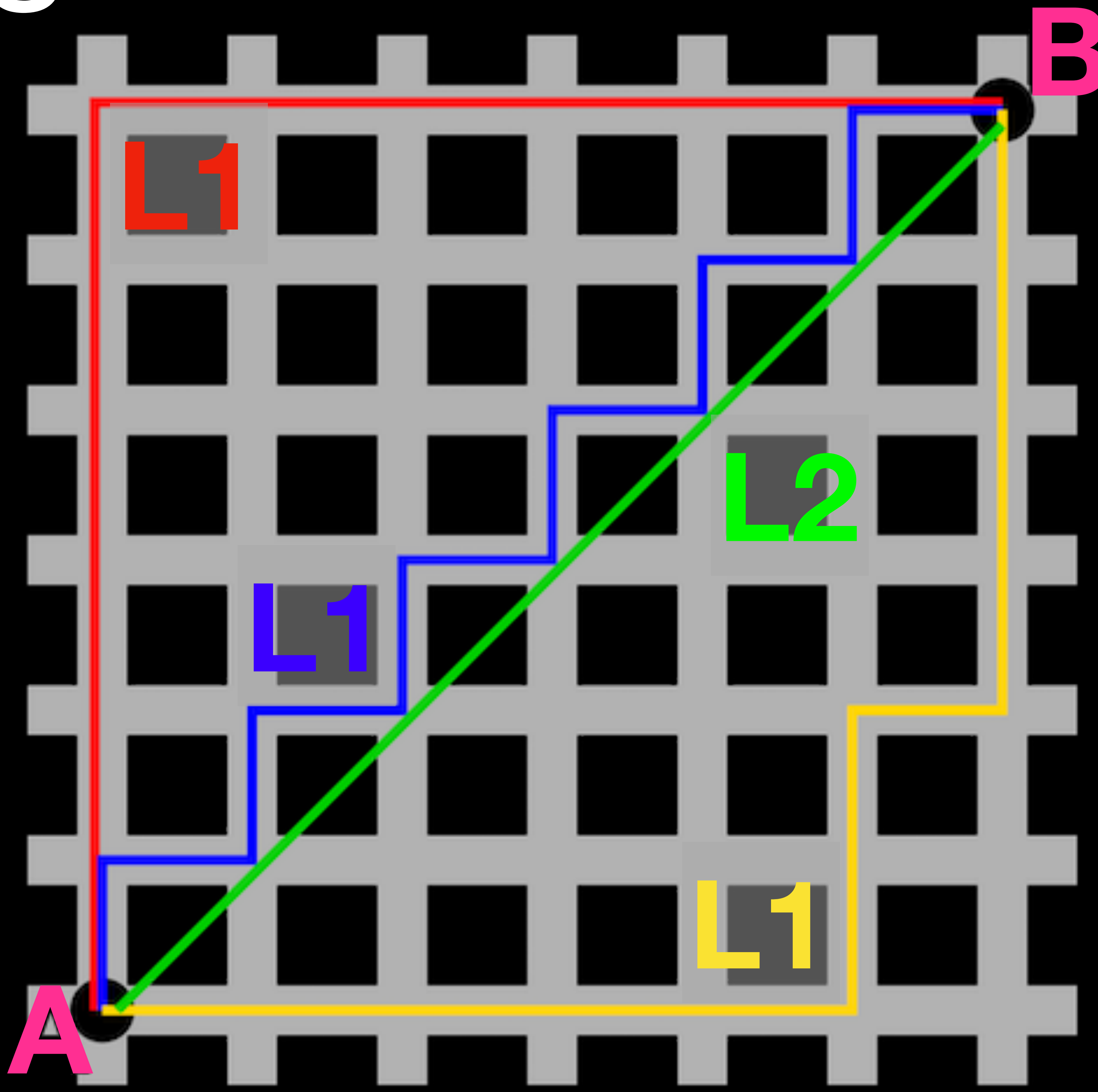
L2 NORM

$$||W||_2 = \sqrt{\sum_{i=1}^{N} w_i^{\mathbf{2}}}$$



EVENLY DISTRIBUTED

Regularization Norms



Bocconi

Wrapping Up

Take home points

- Texts can be represented as **sparse, discrete** feature vectors over TFIDF counts
- Choose the **appropriate performance metric**
- Choose an **informative baseline**
- **Regularize, regularize, regularize**
- **Feature selection** can improve performance and provide insights
- Ask yourself: "*Am I comfortable having my system classify myself?*"

Bocconi