Natural Language Processing

Text Classification

Dirk Hovy

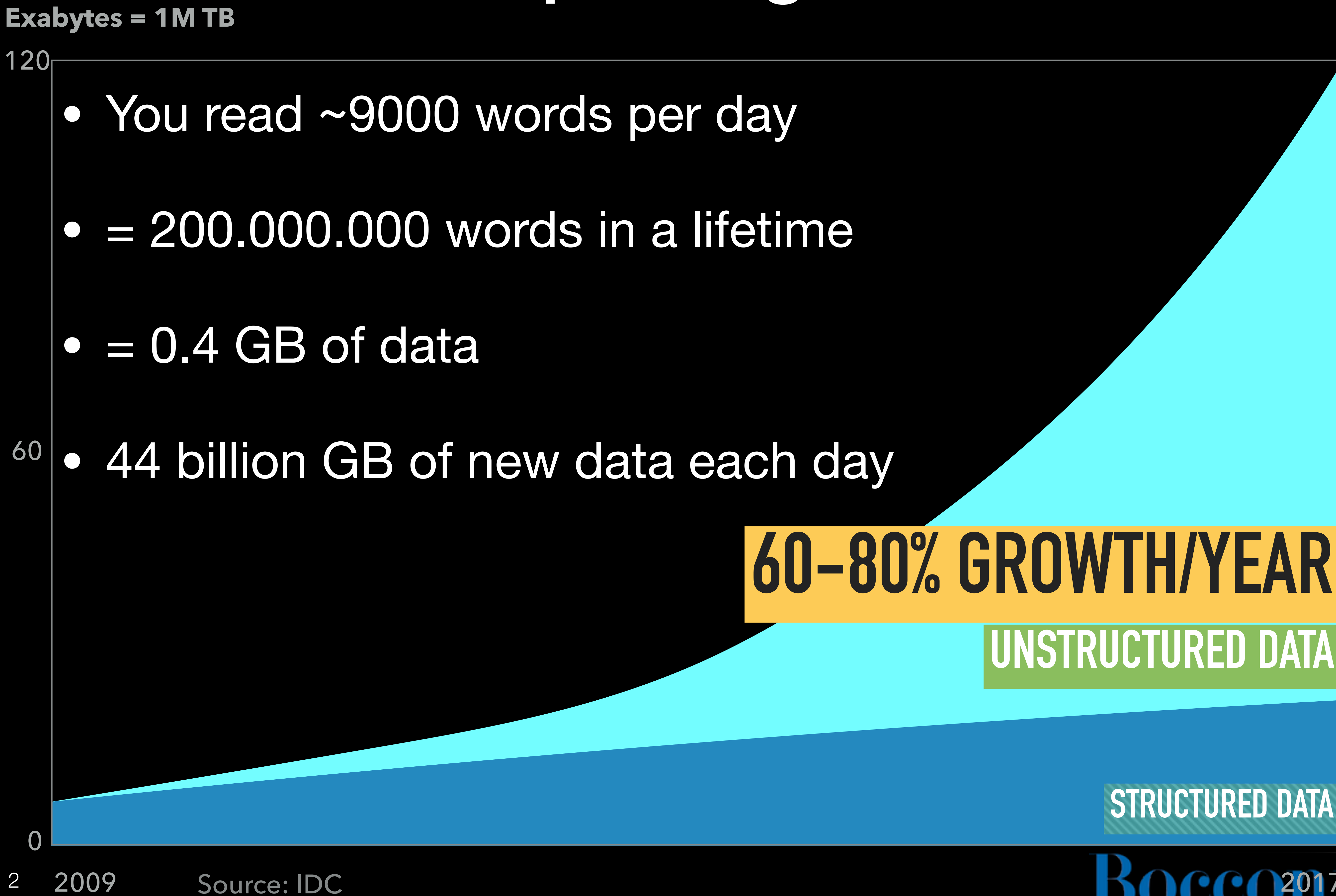
dirk.hovy@unibocconi.it

@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



Bocconi

NLP is booming

Source: Tractica

Examples: Sentiment

- Input: reviews
- Output: positive, negative, neutral ★★★★☆
- Use: business intelligence, market analysis

Examples: Hate Speech

- Input: tweets, blog posts, etc.
- Output: presence (binary) or type of hate speech
- Use: platform health, public policy

Examples: Mental Health

- Input: social media
- Output: presence of risk for mental health condition
- Use: psychologist support, risk screening

Examples: Geolocation

Author Attribute Prediction

- Input: tweet history
- Output: coordinates or predefined region
- Use: social media analysis, targeting

Sentiment Analysis

Bocconi

Classification Steps

- **preprocess** the data
- choose **text representation** (discrete or continuous)
- **select** a **model** (CV, metrics, regularization)
- **fit** the final **model**

Let's start!

Today's Goals

- Understand where NLP comes from
- Learn about the different steps of **preprocessing**
- Learn about **bag of words** (BOW) representations
- Learn about forms of **TF-IDF** and its possibilities
- Understand the difference between sparse and dense representations
- Learn about word2vec and doc2vec

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

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 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 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 '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)

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

Telling Neighbors: Pointwise Mutual Information

Some are not like the Others

FREQUENT BIGRAMS

frying pan

social media

and thought

New York

of the

COLLOCATIONS

Bocconi

Mutual Informativity

HOW WELL CAN WE GUESS THE BLANK?

social ______

and ______

______ media

______ the

Bocconi

Pointwise Mutual Information

CHANCE OF SEEING THEM TOGETHER

$$PMI(x, y) = \log \frac{P(x, y)}{P(x)P(y)}$$

...SEEING EITHER

x	y	c(x)	c(y)	c(xy)	P(x)	P(y)	P(x, y)	PMI(x; y)
moby	dick	83	83	82	0.0003	0.0003	0.0003	**3.48**
captain	ahab	327	511	61	0.0013	0.0020	0.0002	**1.97**
white	whale	280	1150	106	0.0011	0.0045	0.0004	**1.93**
under	the	119	14175	45	0.0005	0.0553	0.0002	**0.83**
is	a	1690	4636	110	0.0066	0.0181	0.0004	**0.56**

c(X) = 256,149
c(XY) = 256,148

Bocconi

Representing Text

Bocconi

Ham or Spam?

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

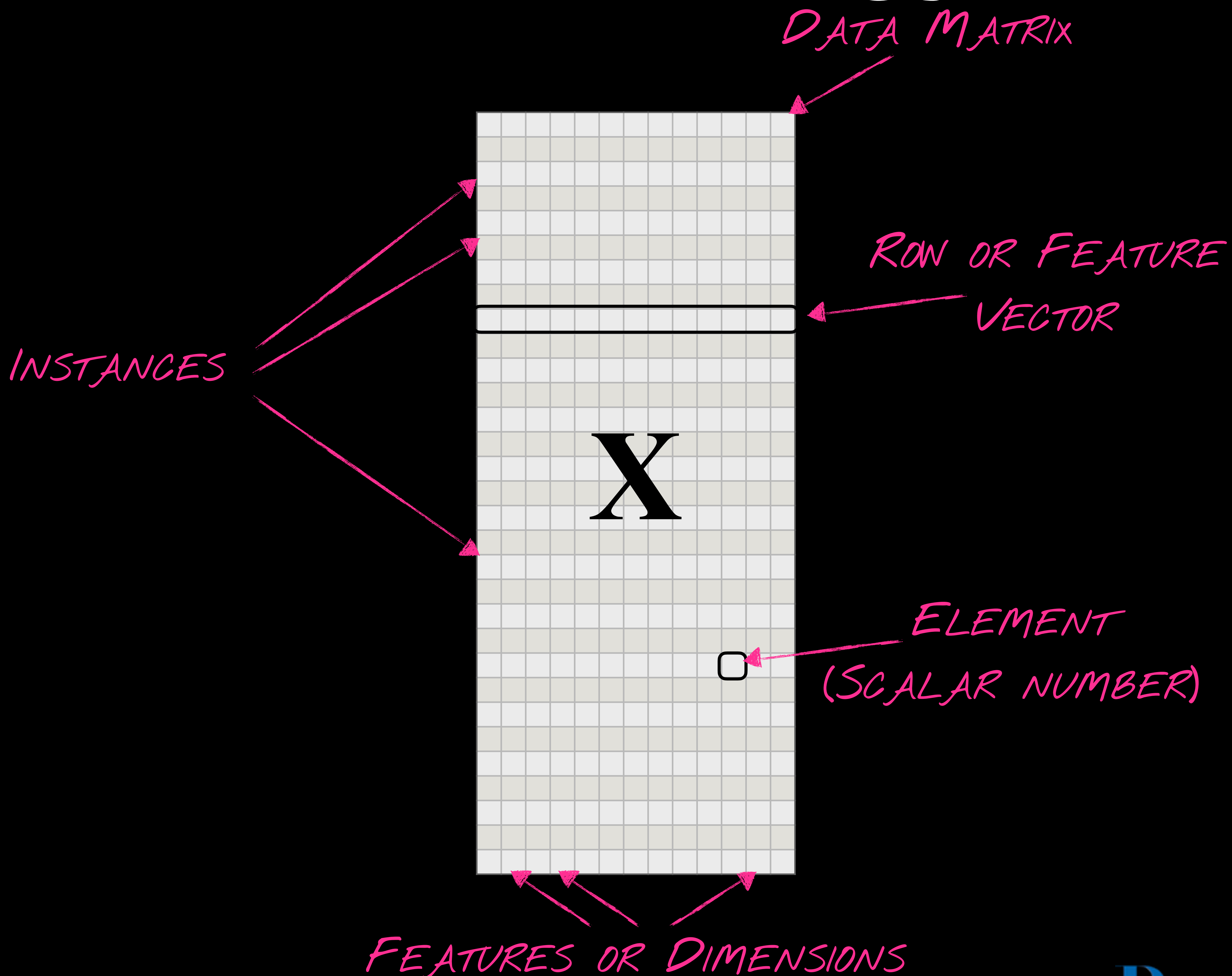
Greetings dear friend

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:

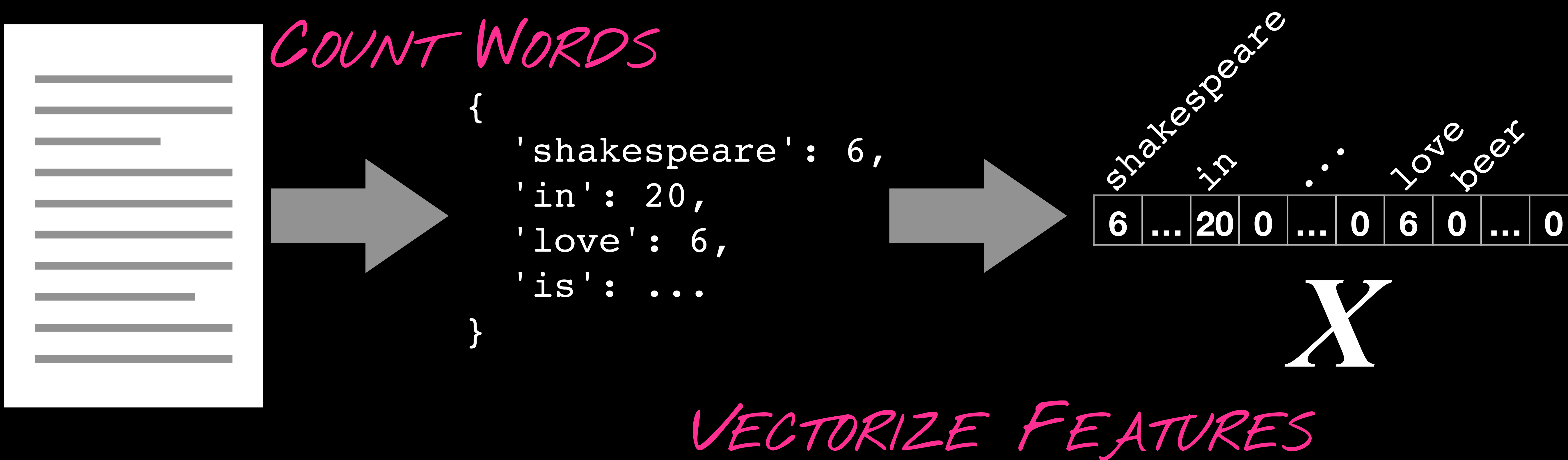
- 4U
- click
- amazing
- free
- guarantee
- offer
- urgent
- dear friend
- income
- serious

Terminology



Discrete Representations

Bags of words (BOW)



Quiz!

What happens if we allow ***every*** *possible word* to constitute a feature?

Expensive computation, and vectors have too many zeros.

Limit to most frequent/informative words!

Counting Trouble

...And a Man named Zipf

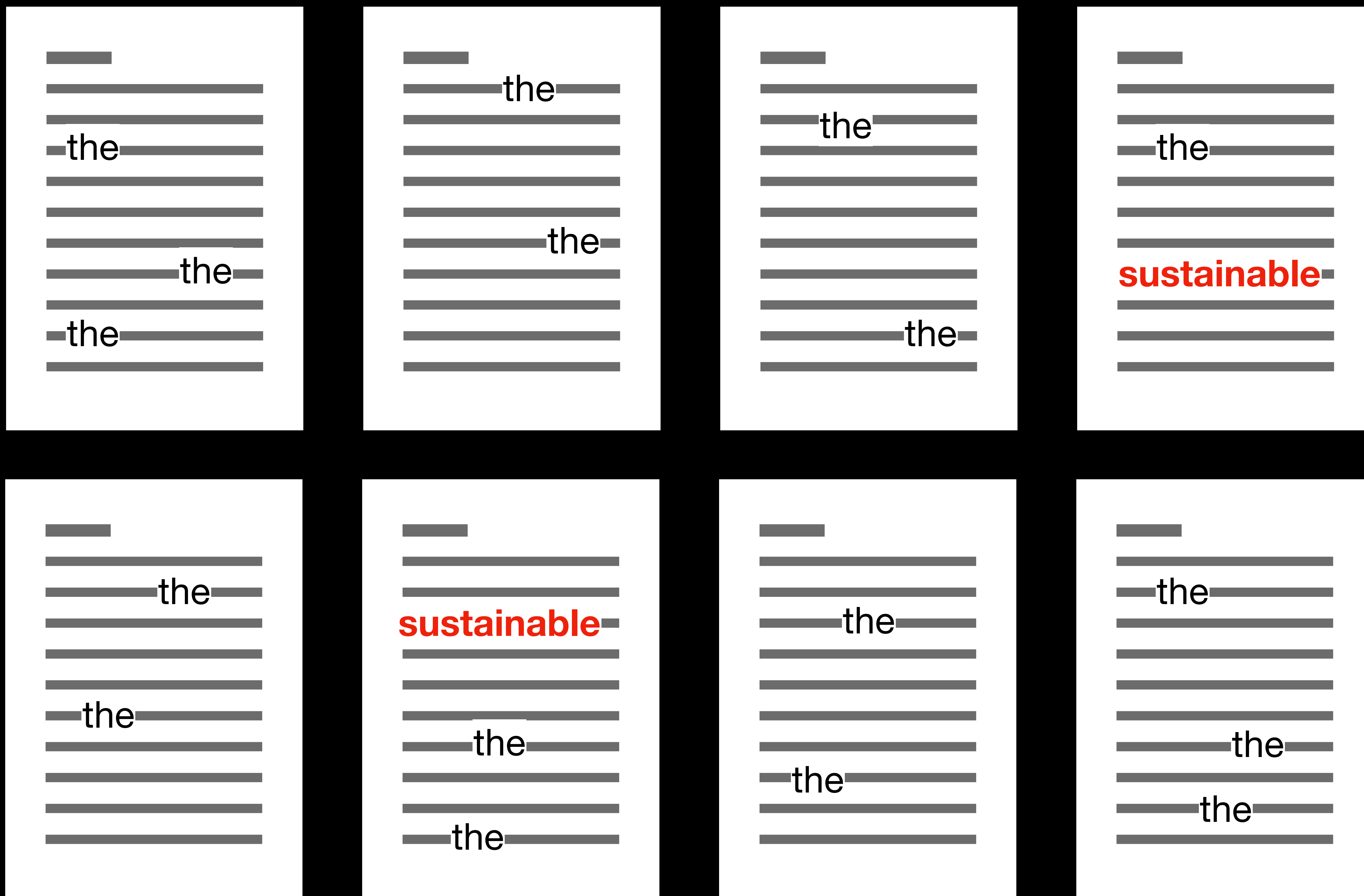
N-grams

"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, ...`**

Finding Important Words: TF-IDF

Some Words are Just More Interesting…

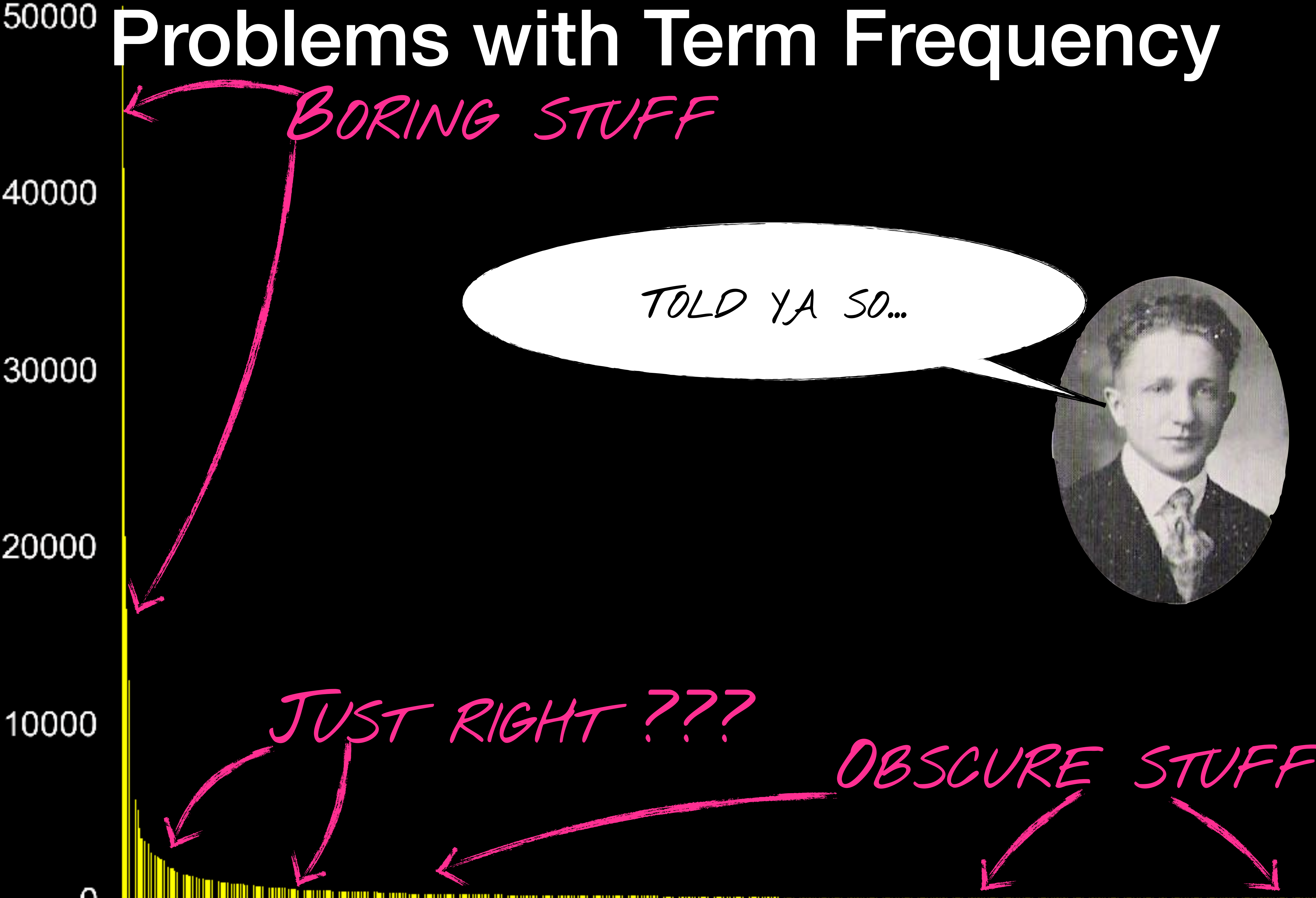


Karen Spärck Jones

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

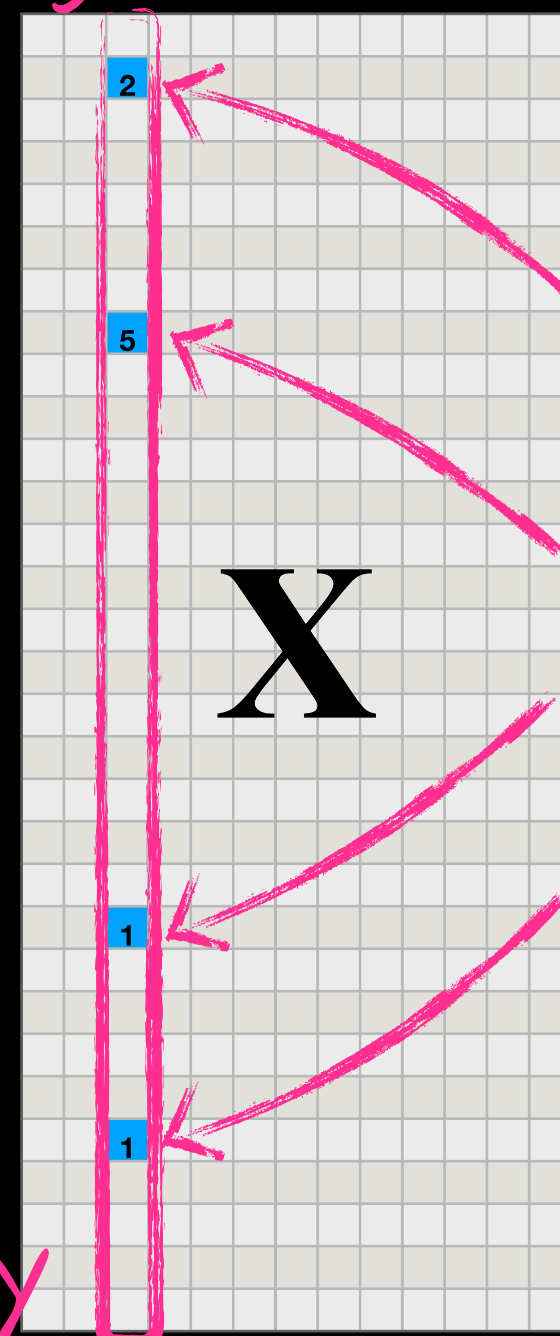
Problems with Term Frequency



Document and Term Frequency

FEATURE

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

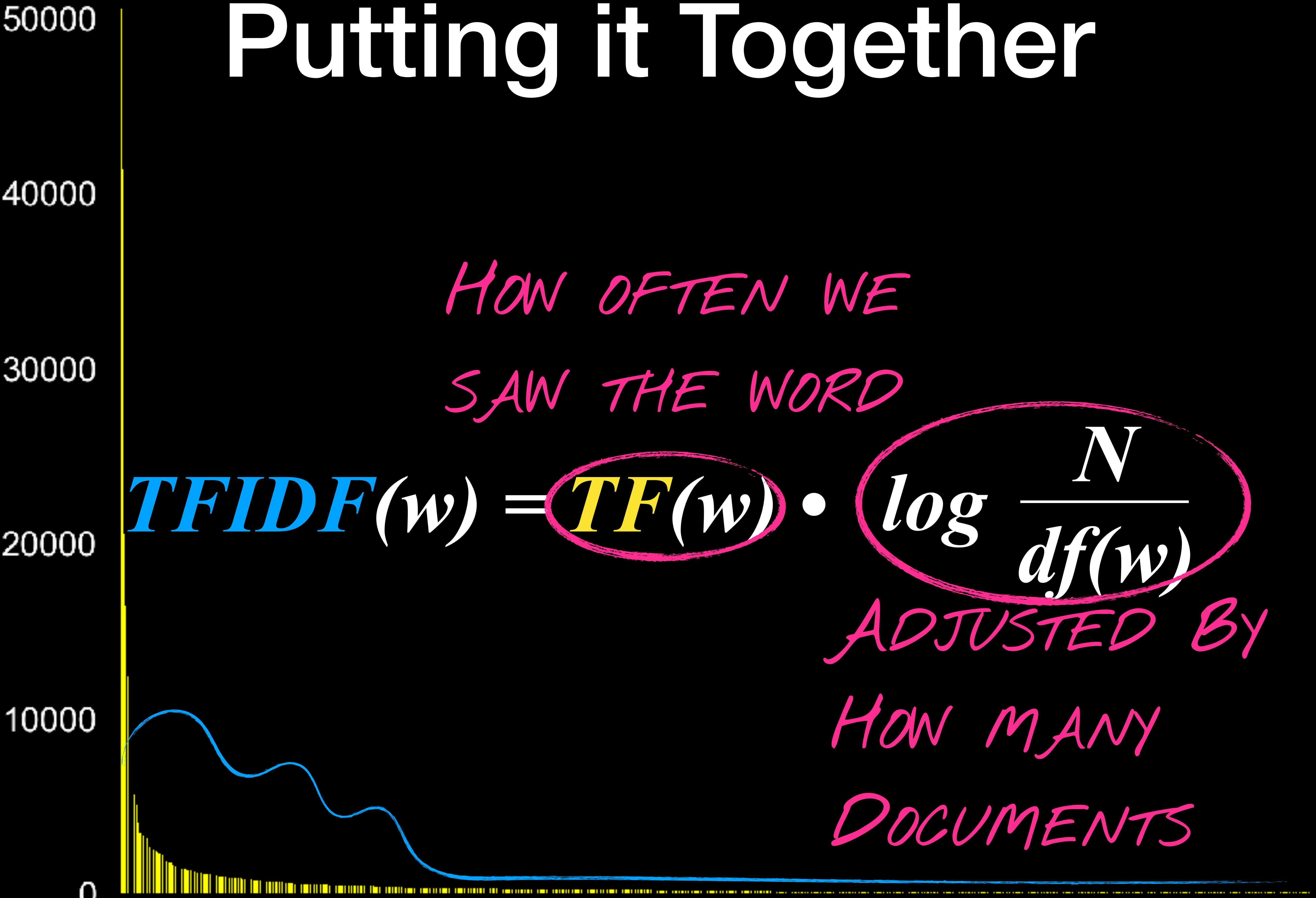


DOCUMENT FREQUENCY (COUNT): 4

TERM FREQUENCY (SUM): 9

TF

Putting it Together

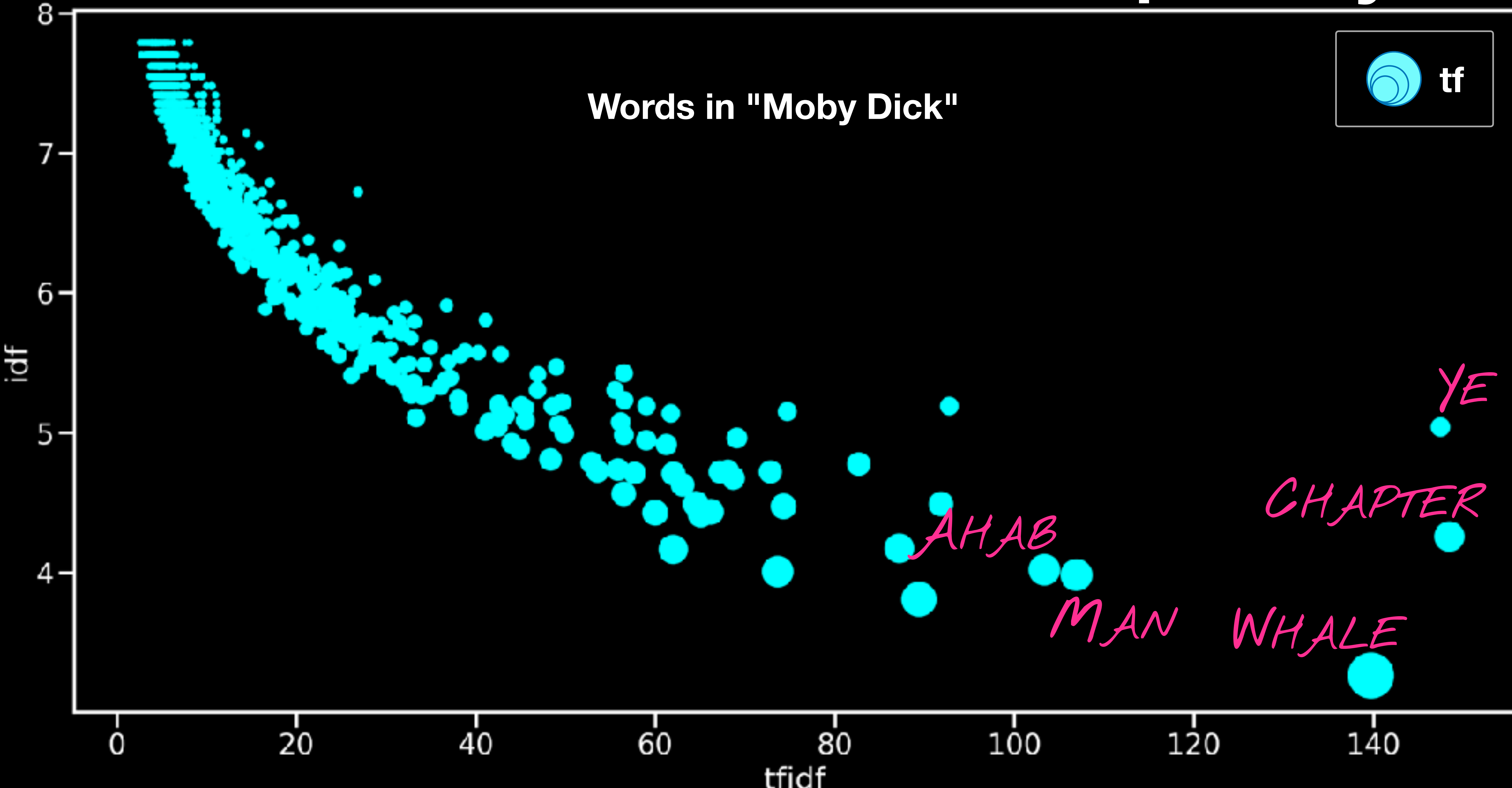


HOW OFTEN WE SAW THE WORD

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

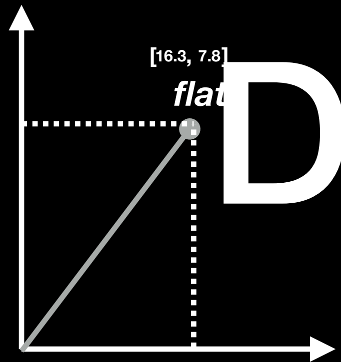
ADJUSTED BY HOW MANY DOCUMENTS

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

Dense Distributed Representations



Distributional Hypothesis

“You shall know the meaning of a word by the company it keeps”

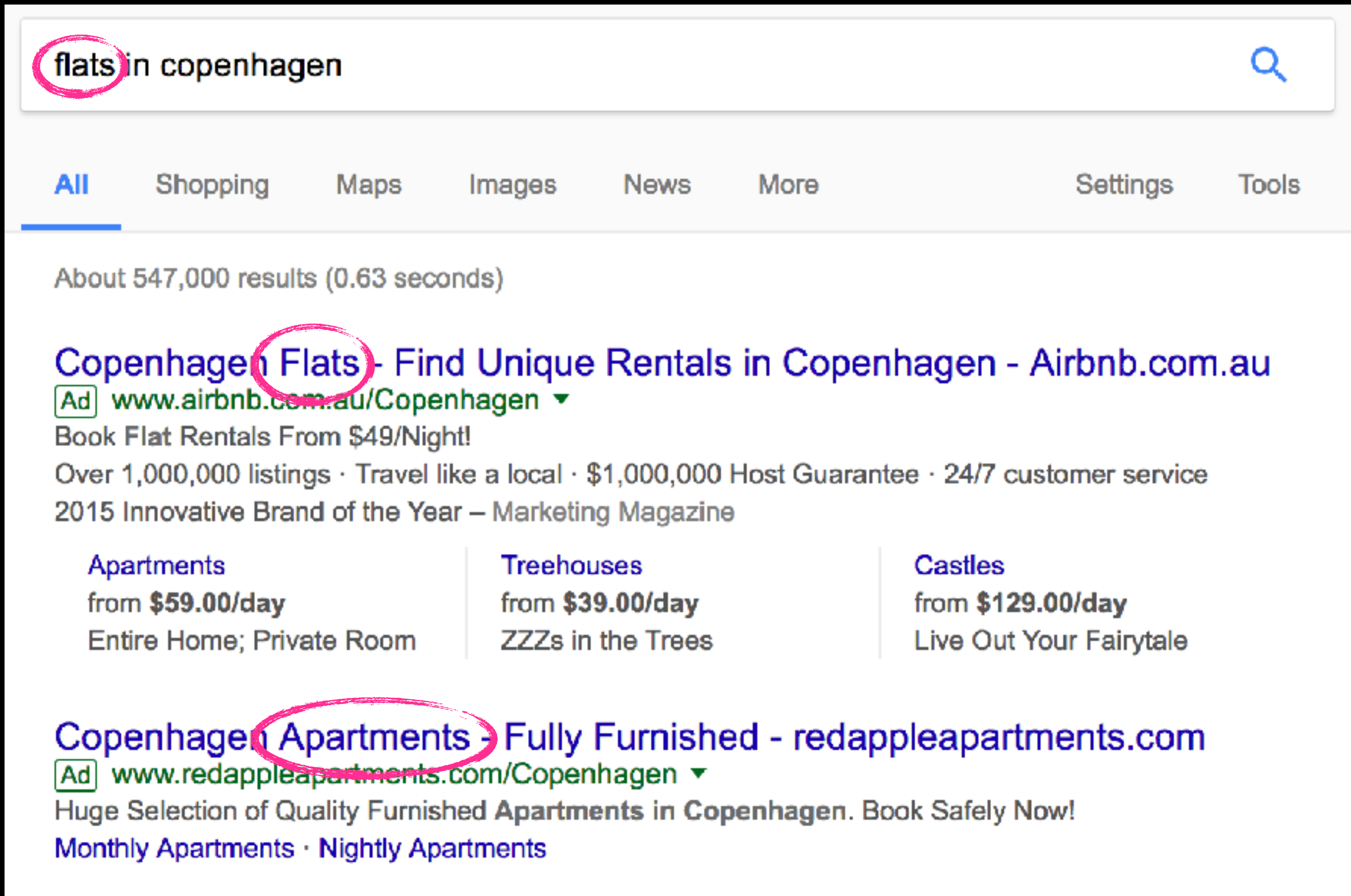
Firth (1957)

Similar words have similar **contexts**

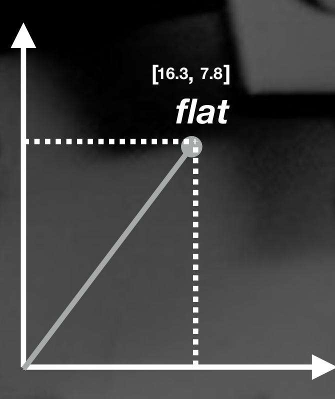
Represent **words** as **vectors**/points in space

Similar words have similar vectors

An Example

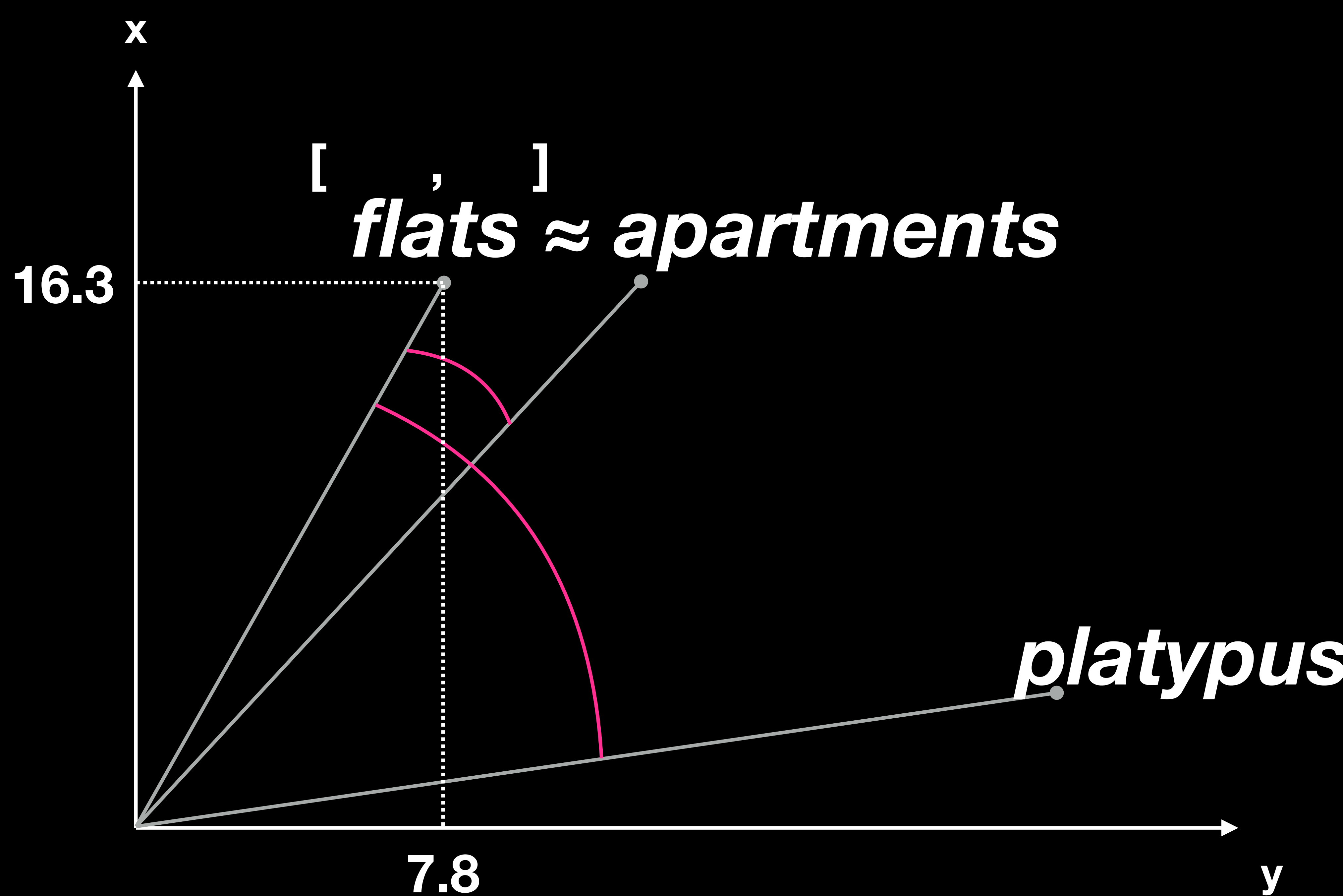

flats in copenhagen
All
Shopping
Maps
Images
News
More
Settings
Tools
About 547,000 results (0.63 seconds)
Copenhagen Flats - Find Unique Rentals in Copenhagen - Airbnb.com.au
Ad www.airbnb.com.au/Copenhagen
Book Flat Rentals From $49/Night!
Over 1,000,000 listings · Travel like a local · $1,000,000 Host Guarantee · 24/7 customer service
2015 Innovative Brand of the Year – Marketing Magazine
Apartments
from $59.00/day
Entire Home; Private Room
Treehouses
from $39.00/day
ZZZs in the Trees
Castles
from $129.00/day
Live Out Your Fairytale
Copenhagen Apartments - Fully Furnished - redappleapartments.com
Ad www.redappleapartments.com/Copenhagen
Huge Selection of Quality Furnished Apartments in Copenhagen. Book Safely Now!
Monthly Apartments · Nightly Apartments

Part 1
Representing Words as Vectors



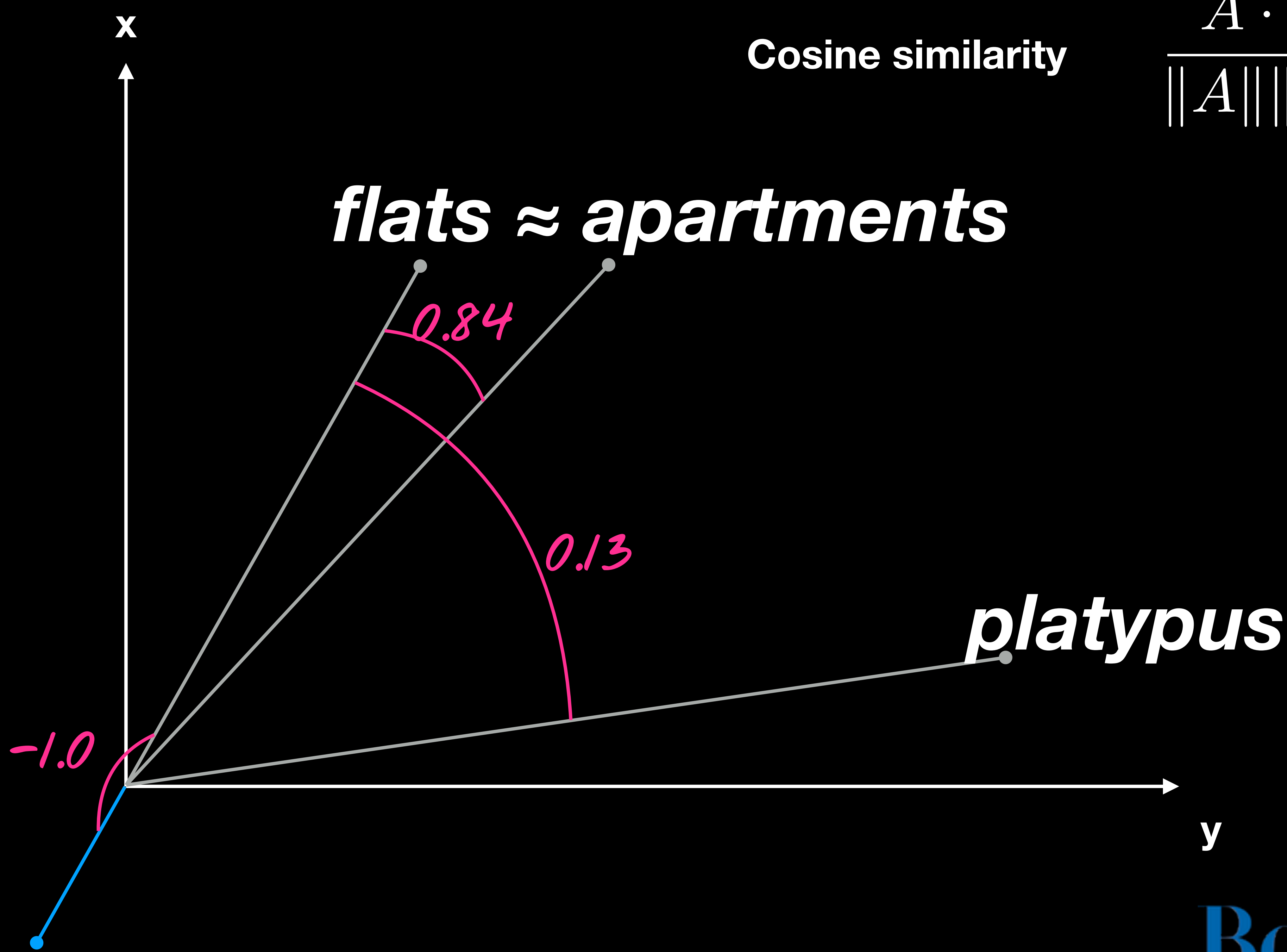
Bocconi

Semantic Similarity



Similarity Measures

Cosine similarity $\dfrac{A \cdot B}{\|A\|\|B\|}$



Bocconi

Dot Product

- “combine” vectors to a scalar

SUM

$$x \cdot y = \sum_{i=1}^{D} x_i y_i$$

MULTIPLY

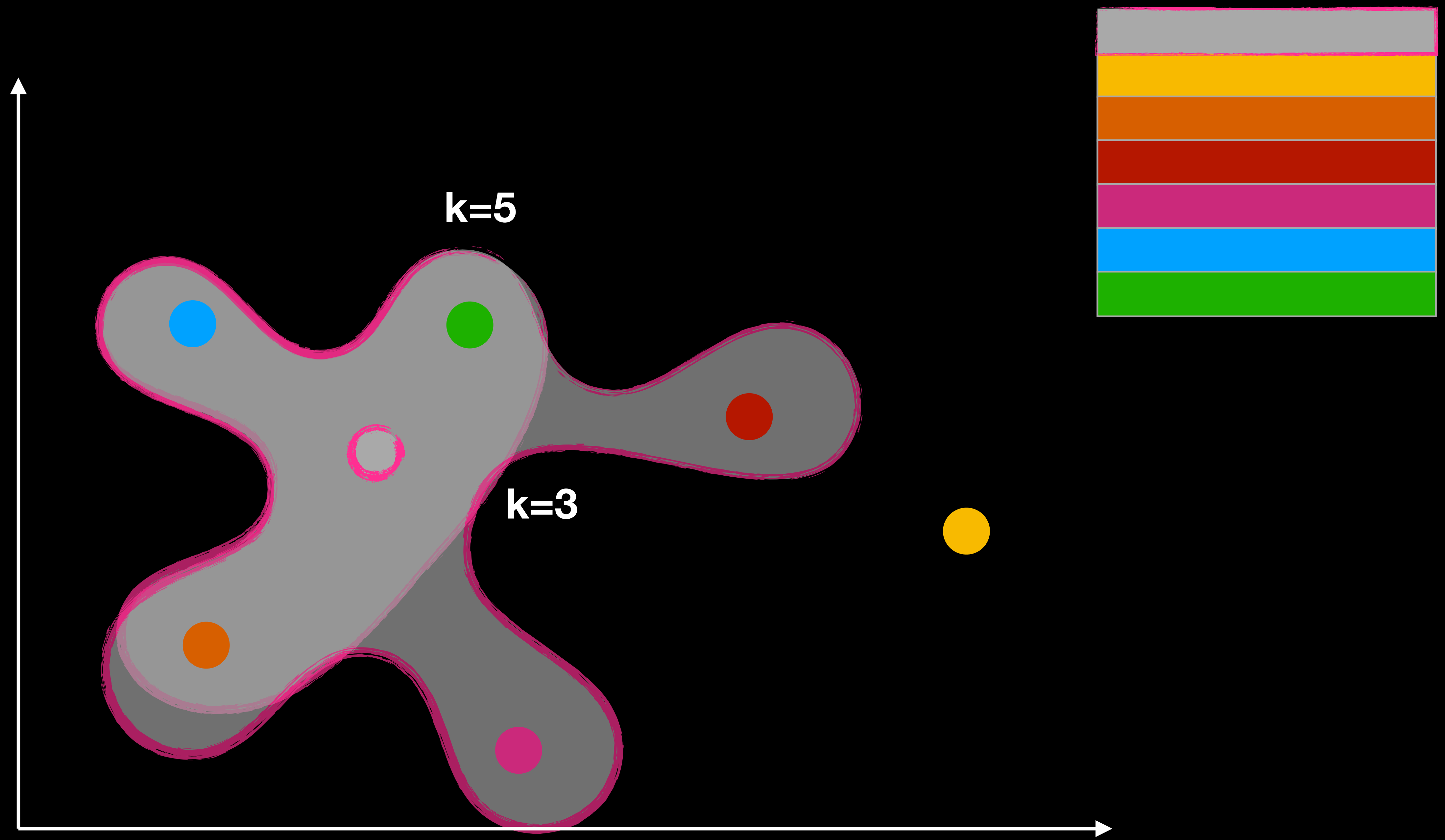
$$\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 6 \end{bmatrix} = \begin{matrix} 1 \\ 4 \\ 3 \end{matrix}$$

Vector Norm

- add up square of each element, take $\sqrt{}$

$$\begin{bmatrix} 2 \\ 6 \end{bmatrix} = \sqrt{2^2 + 6^2} = 6.324$$

Nearest neighbors

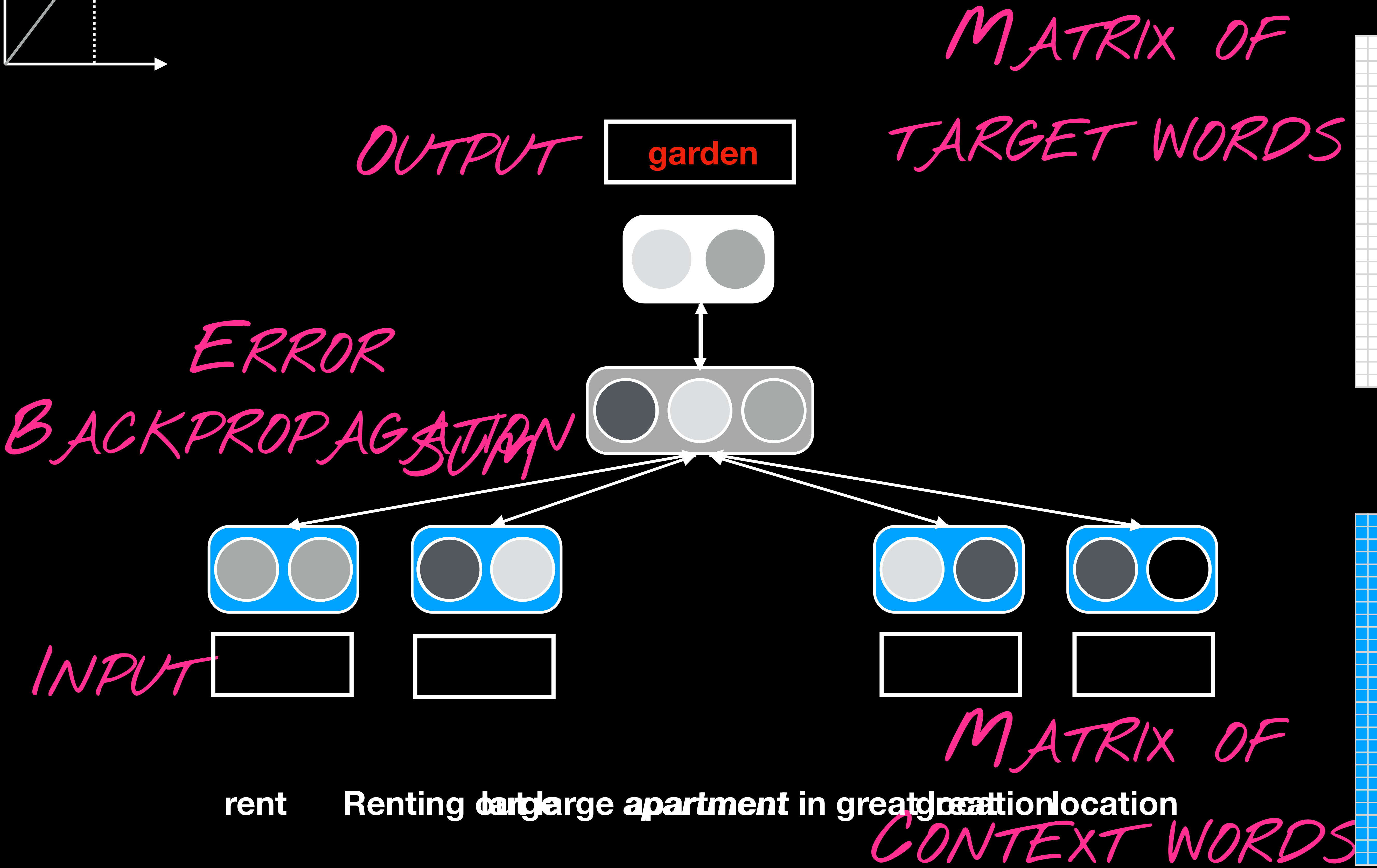


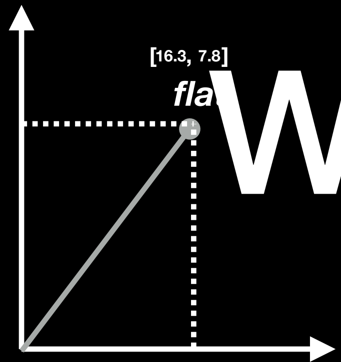
Bocconi

Word2Vec – Intuitively

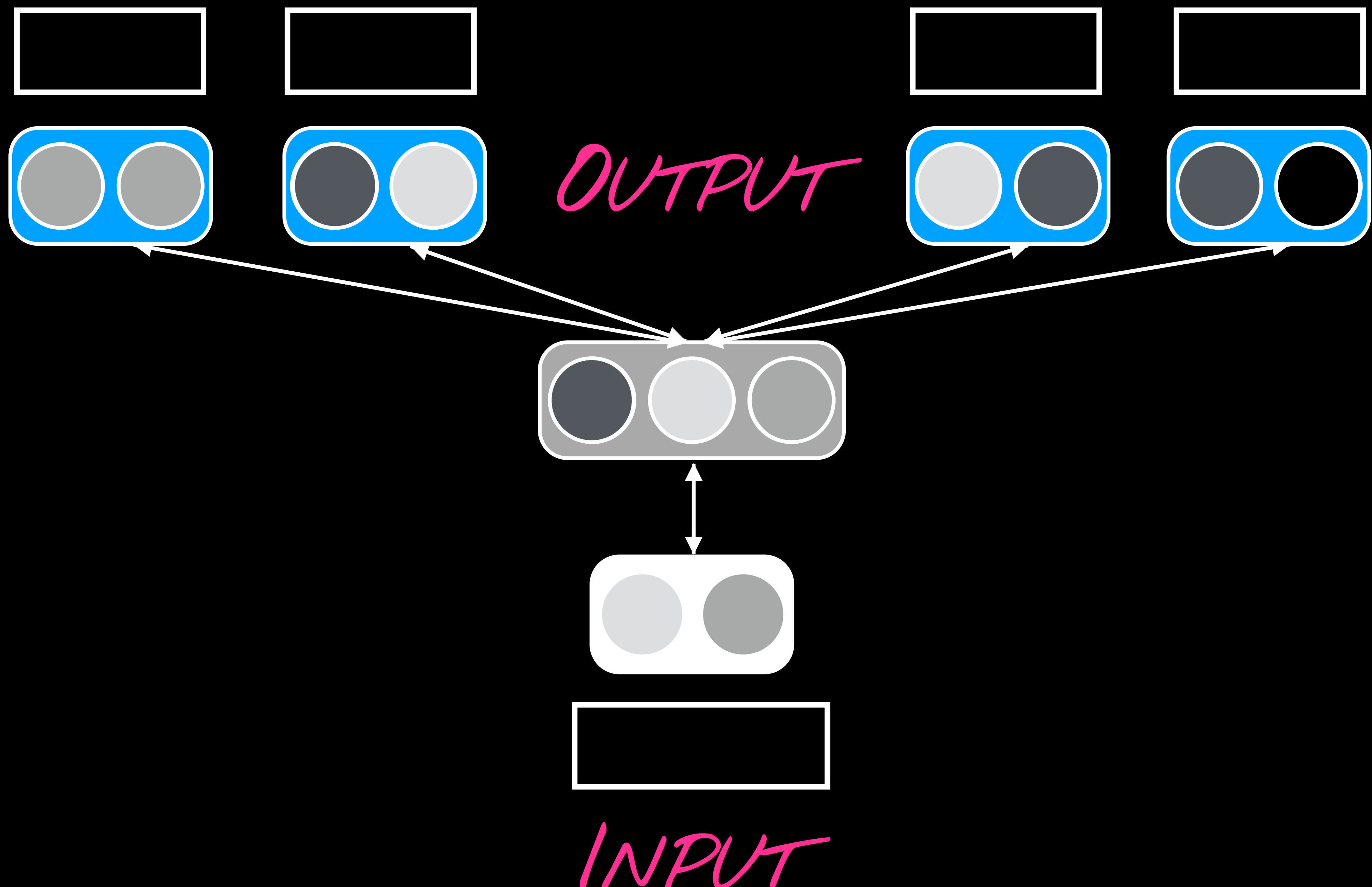
```
place all words randomly on fridge
for each pair of words:
    if in same sentence:
        move closer together
    else:
        move further apart
```

Word2Vec – CBOW Model





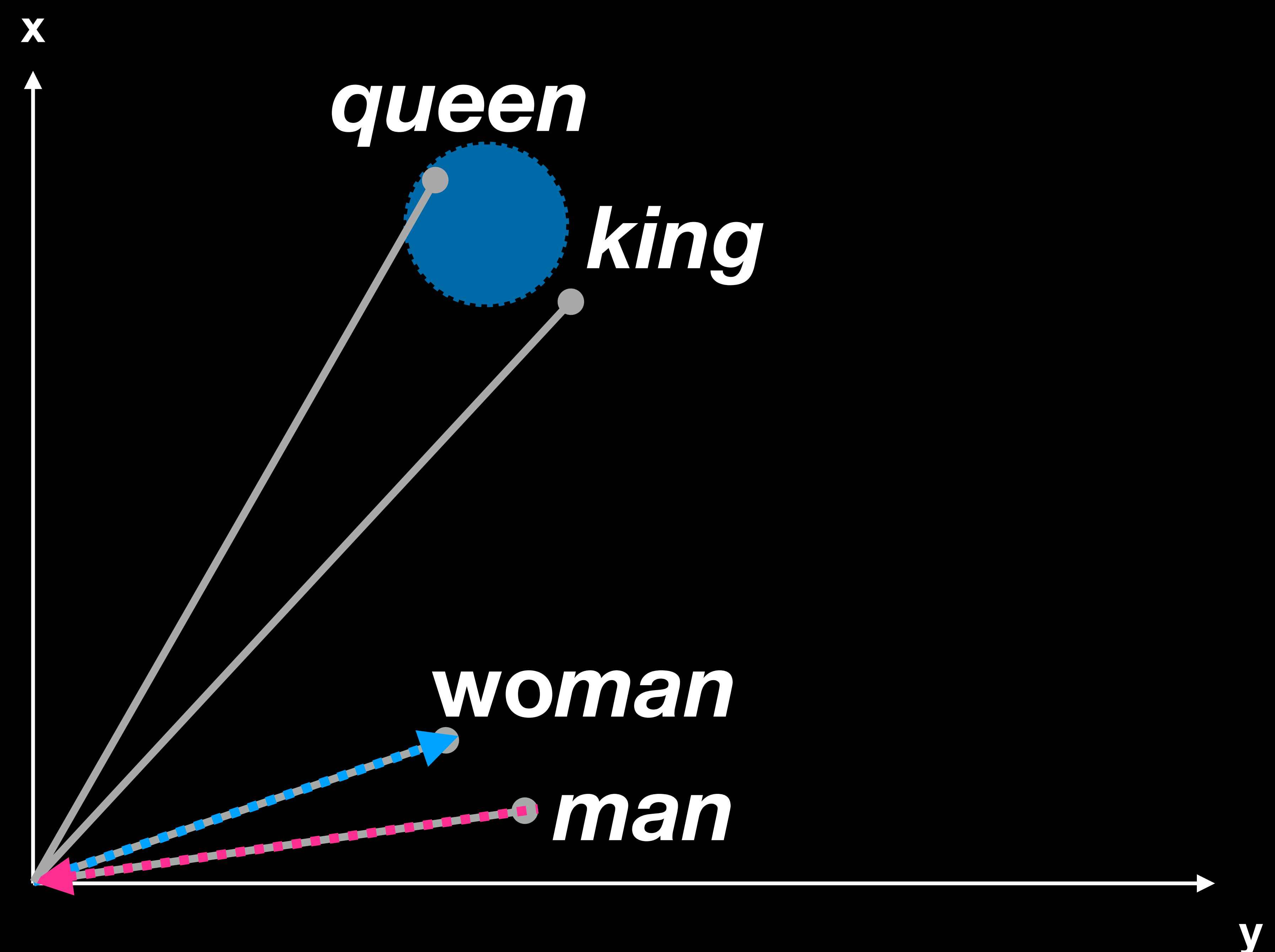
Word2Vec – Skipgram Model



rent Renting out a large *apartment* in great location location

Vector Space Semantics

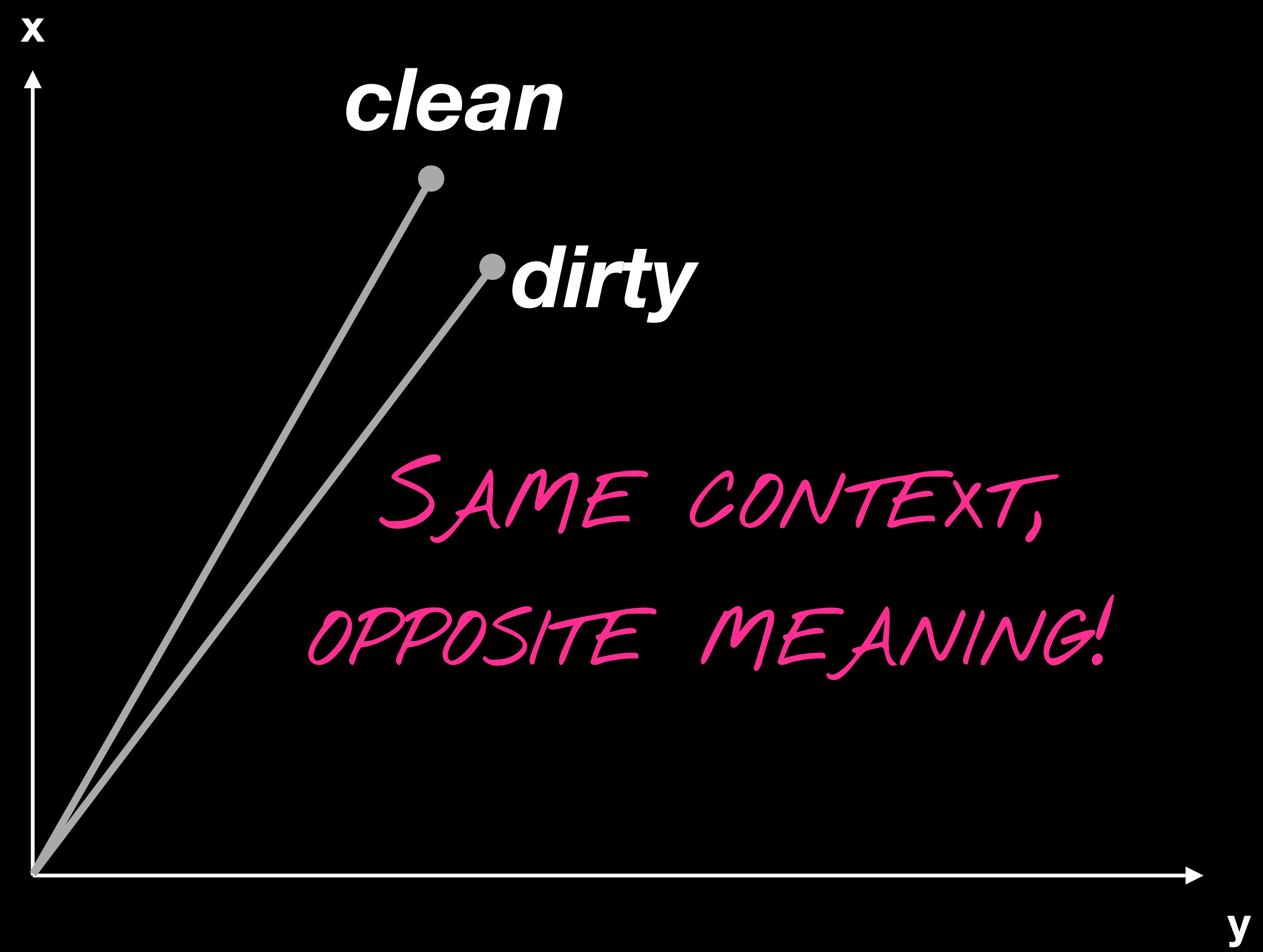
king – man + woman ≈ queen



Bocconi

Caveat: Antonyms

His kitchen was always very __________



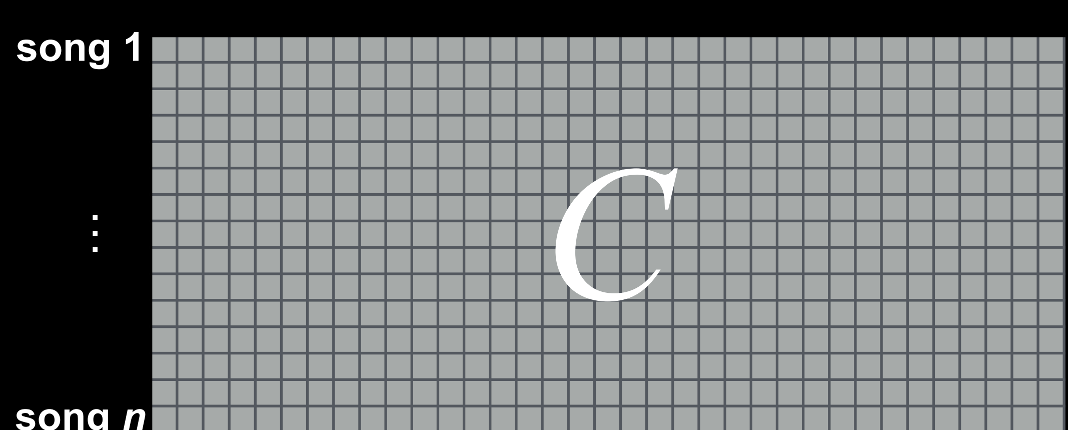
Bocconi

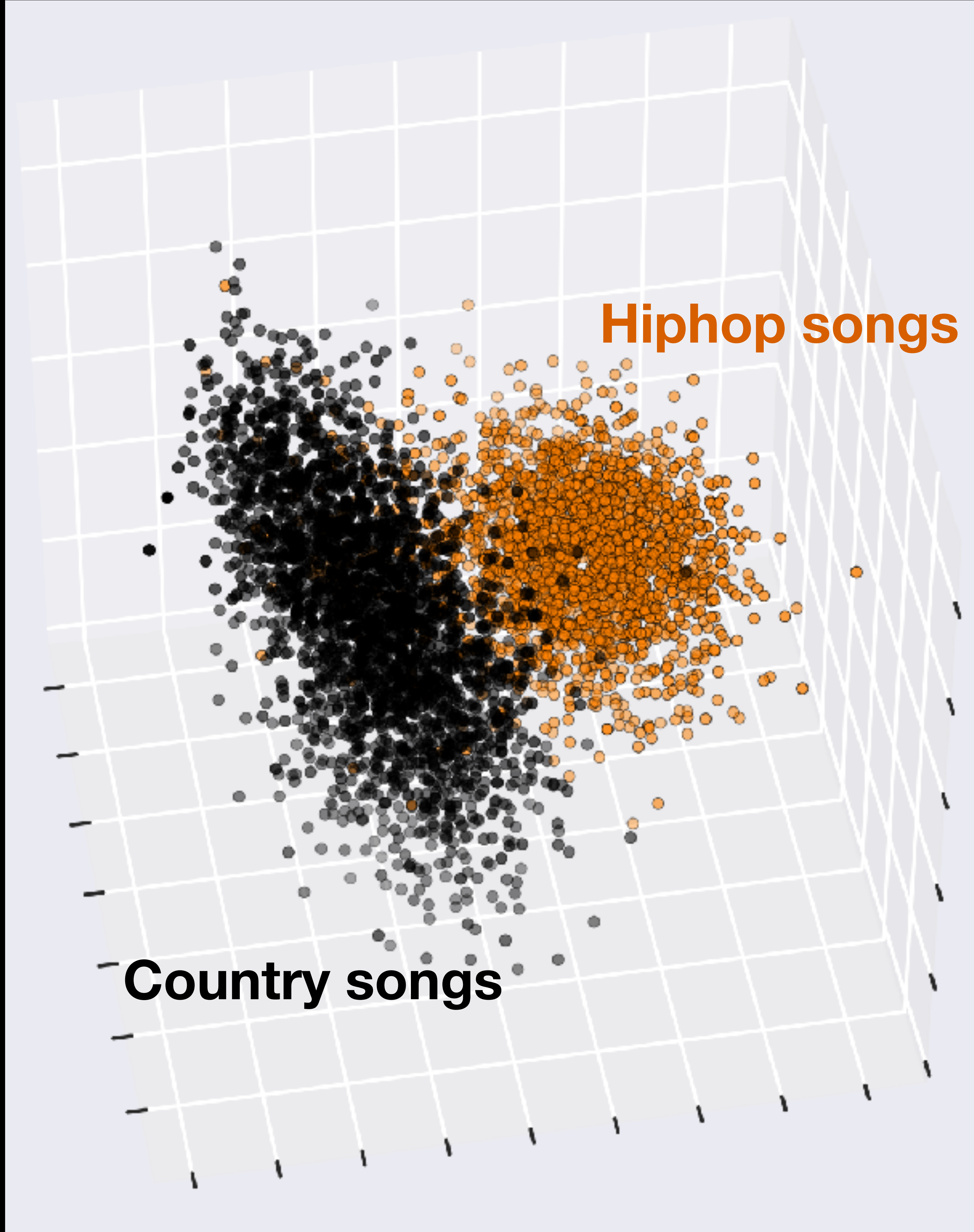
Part 2
Representing Documents as Vectors

Bocconi

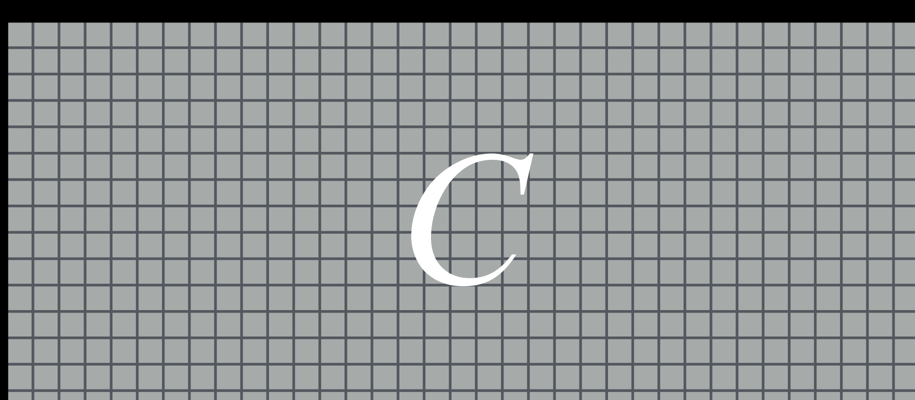
Example 1: Songs

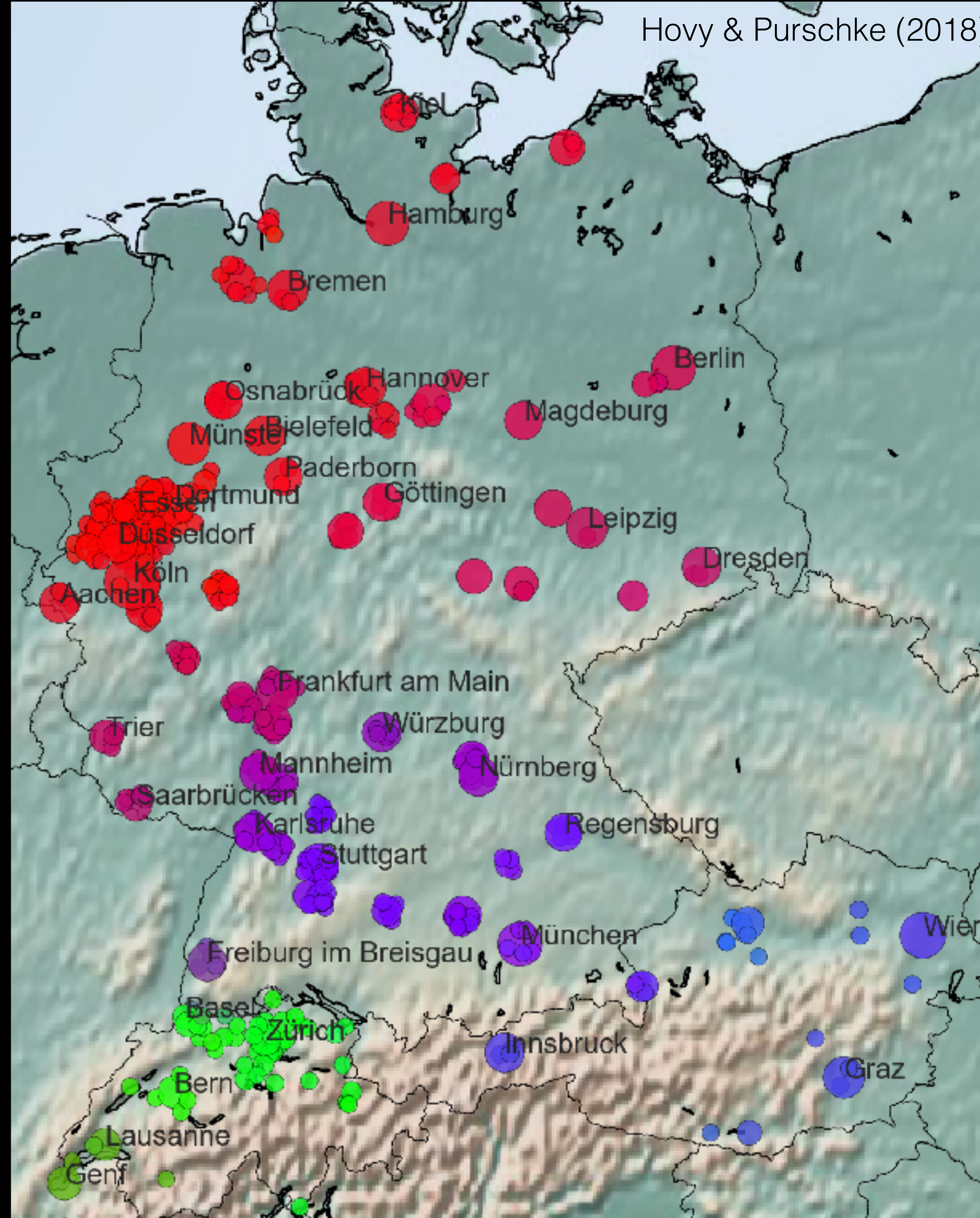
Billboard HOT 100





Example 2: Cities





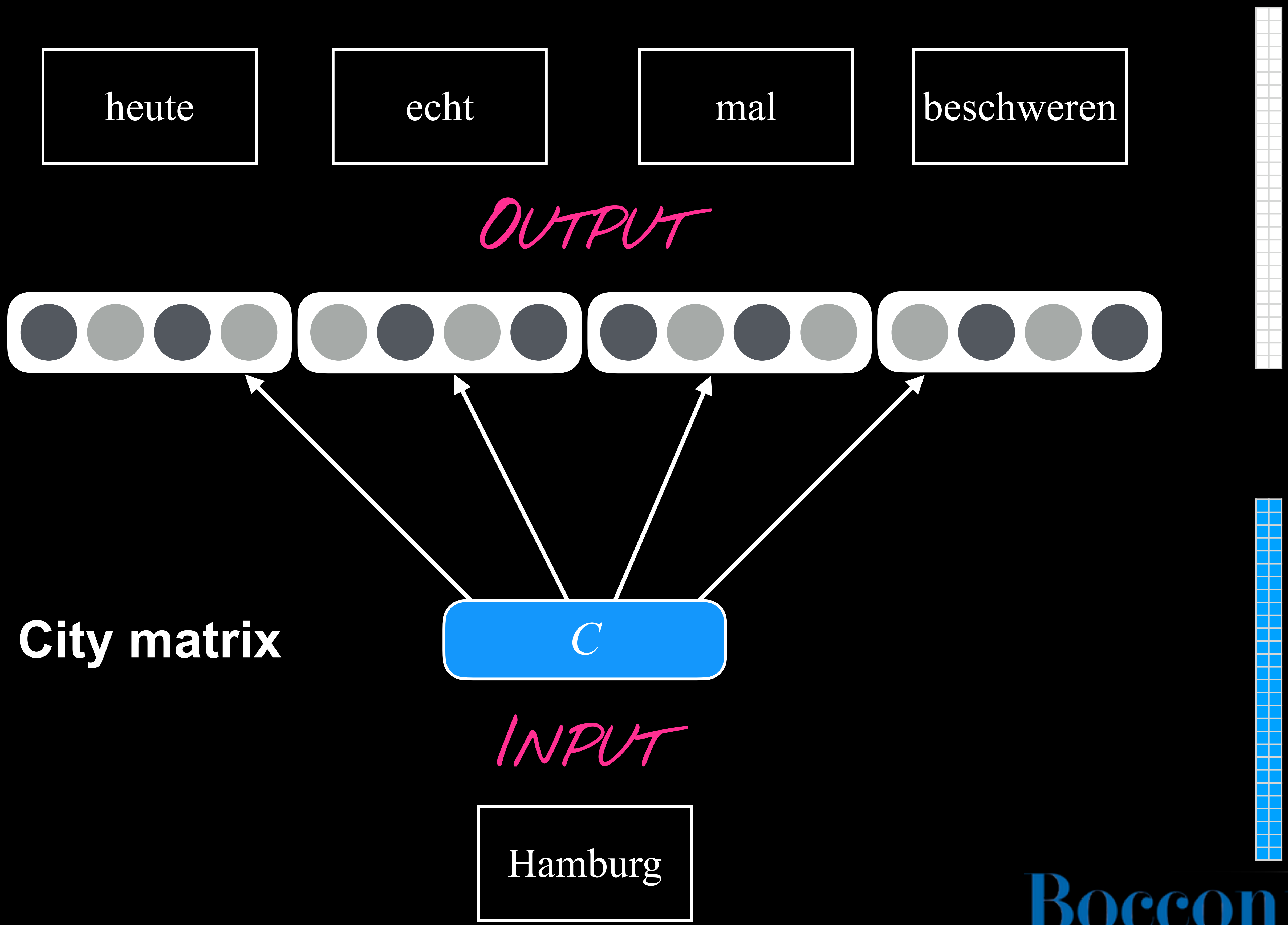
Hovy & Purschke (2018)

Le & Mikolov (2014)

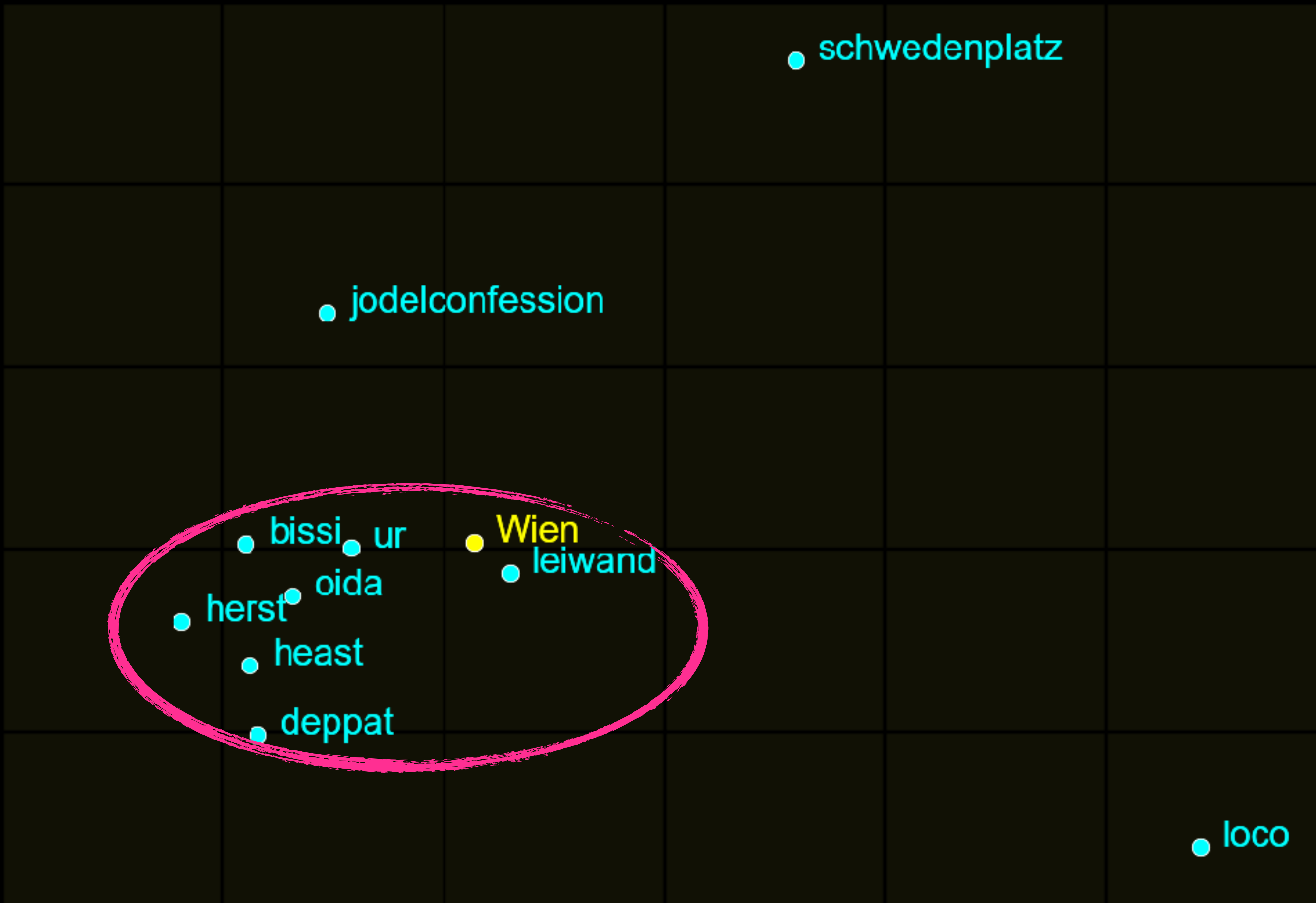
Doc2Vec – Intuitively

```
place words & cities randomly on fridge
for each pair of (word, city):
    if word seen in city:
        move closer together
    else:
        move further apart
```

Doc2Vec – Model



Words and Documents

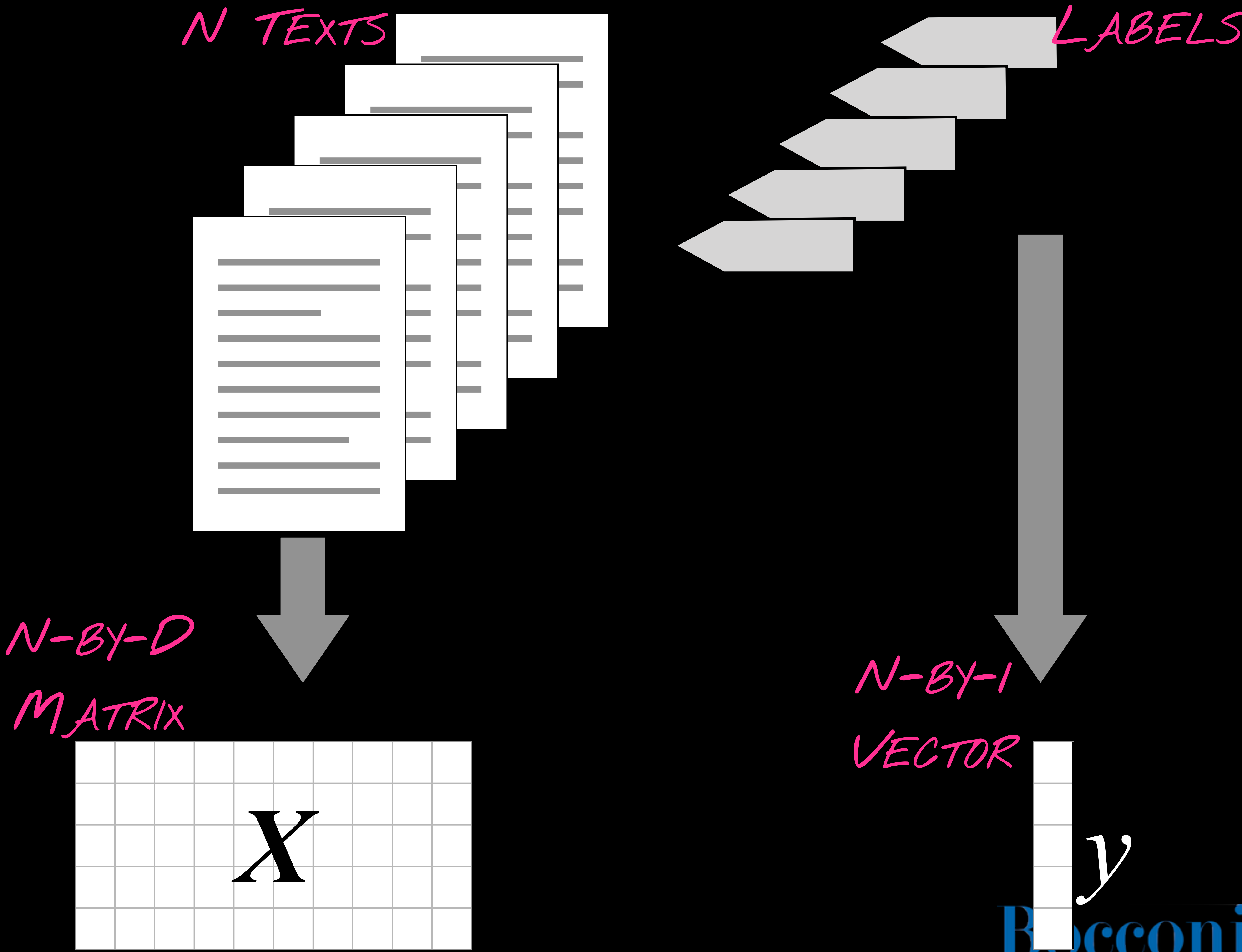


Wrapping up

Comparison

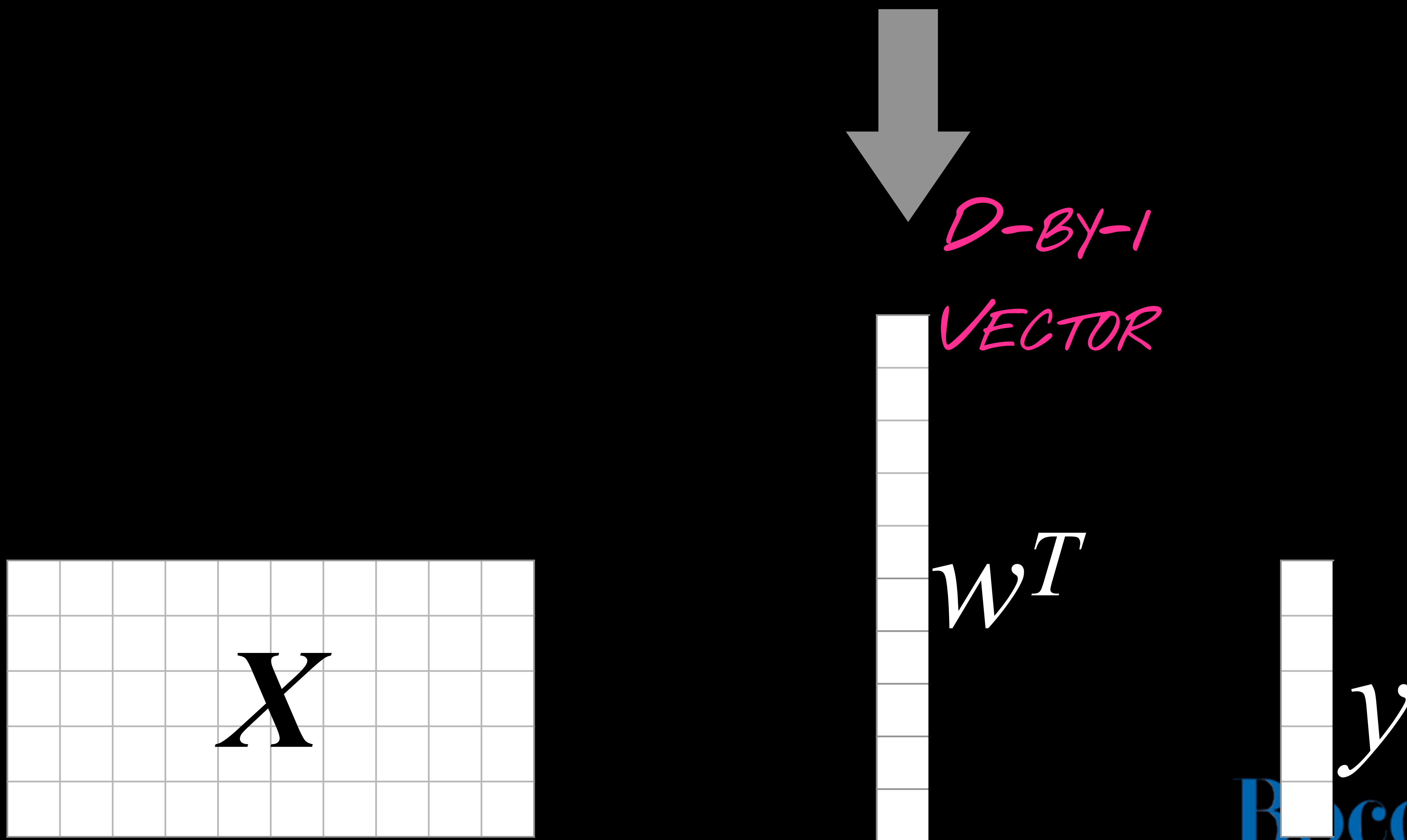
	Discrete	Distributed
#Dimensions	Data-dependent	Pre-defined
Content	Count-based	Coefficients
Density	Sparse	Dense
Strength	Interpretability	Similarity
Application	Understanding	Performance
School of thought	Rationalism	Empiricism

Text Classification


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

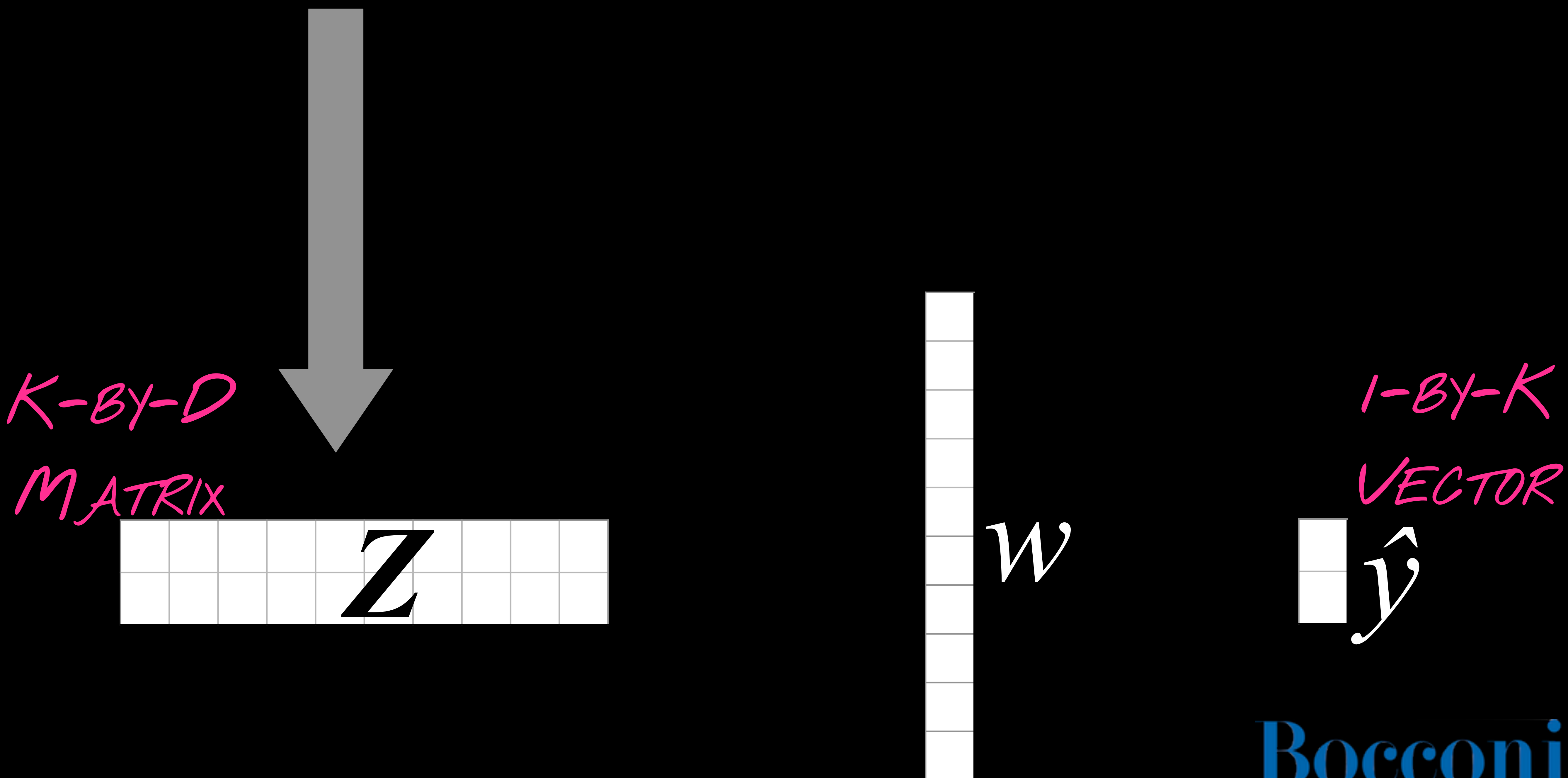
Fitting

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



Predicting

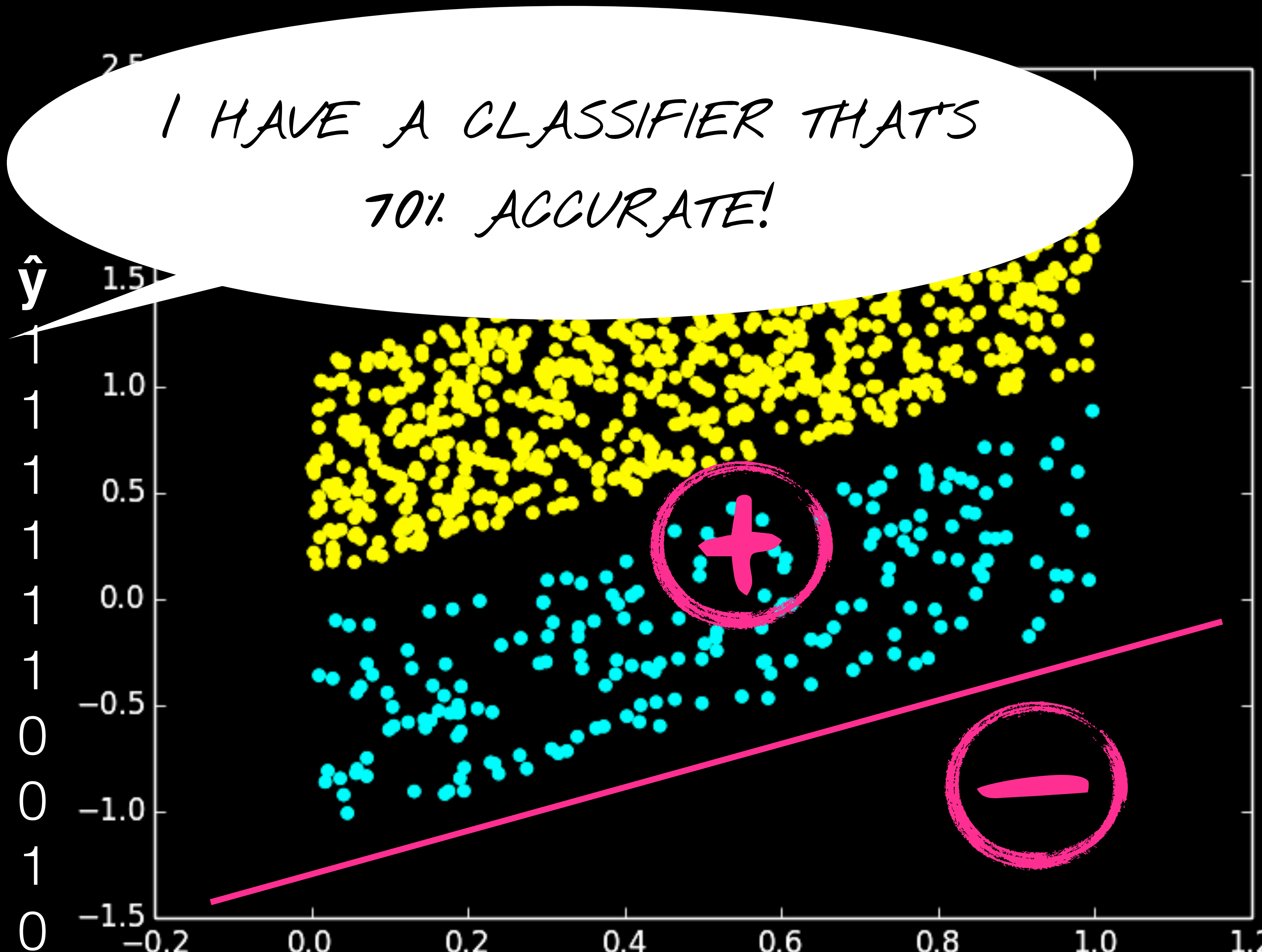
$$f(\mathbf{Z}) = \mathbf{Z}\, w^T = \hat{y}$$



Evaluating Performance

Performance Problems

x	y	ŷ
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

True and False

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

TARGET = ANIMAL

x	**y**	**ŷ**	
frog	1	1	true positive
deer	1	1	
wolf	1	1	
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 = (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 = 6/7 = 0.86

RECALL = 6/8 = 0.75

F1 = 0.81

Bocconi

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

Changing Target Class

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

precision = TP / (TP + FP)

recall = TP / (TP + FN)

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

TARGET = THING

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

ACCURACY = 7/10 = 0.7

PRECISION = 1/3 = 0.33

RECALL = 1/2 = 0.5

F1 = 0.4

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

MICRO Averaging

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

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

MACRO Averaging

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

Baseline: Total Recall

PREDICT MAJORITY CLASS FOR ALL

		predicted	
gold		1	0
	1	TP	FN
	0	FP	TN

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
bear	1	1
fish	1	1
bird	1	1
cat	1	1
stone	0	1
tree	0	1

true positive

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

Bocconi

Beware: Overgeneralization

FALSE POSITIVES

June 6 2019

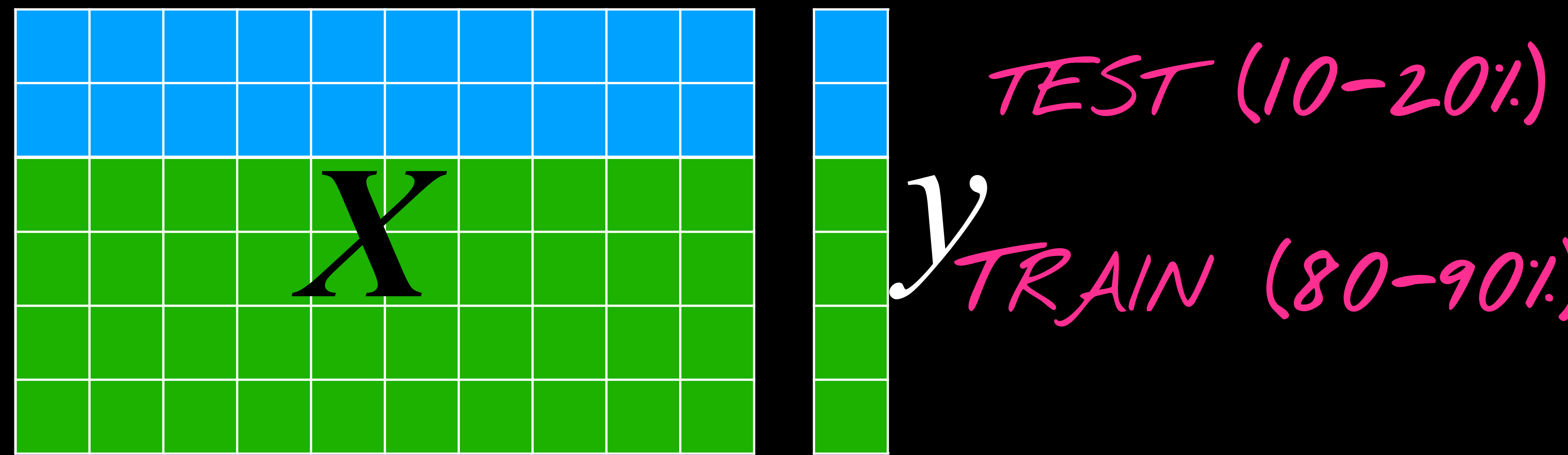
Dear Ms Hovy,

Congratulations on reaching retirement age!

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

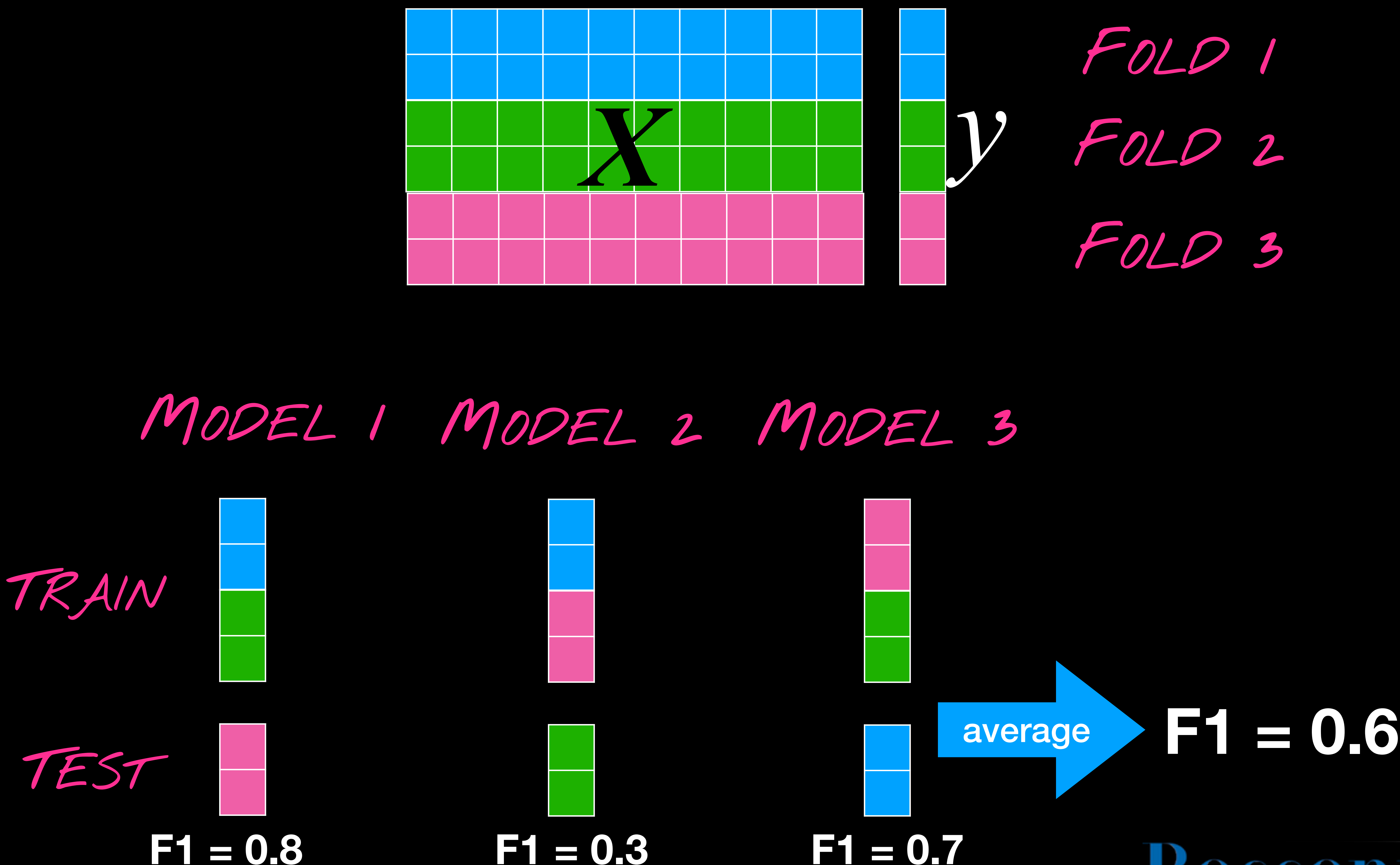
Cross Validation

Prediction Data



SPLIT DATA INTO TRAINING AND HELD-OUT TEST DATA

k-fold Cross-Validation



Baselines

Baseline: Total Recall

	predicted	
gold	1	0
1	TP	FN
0	FP	TN

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

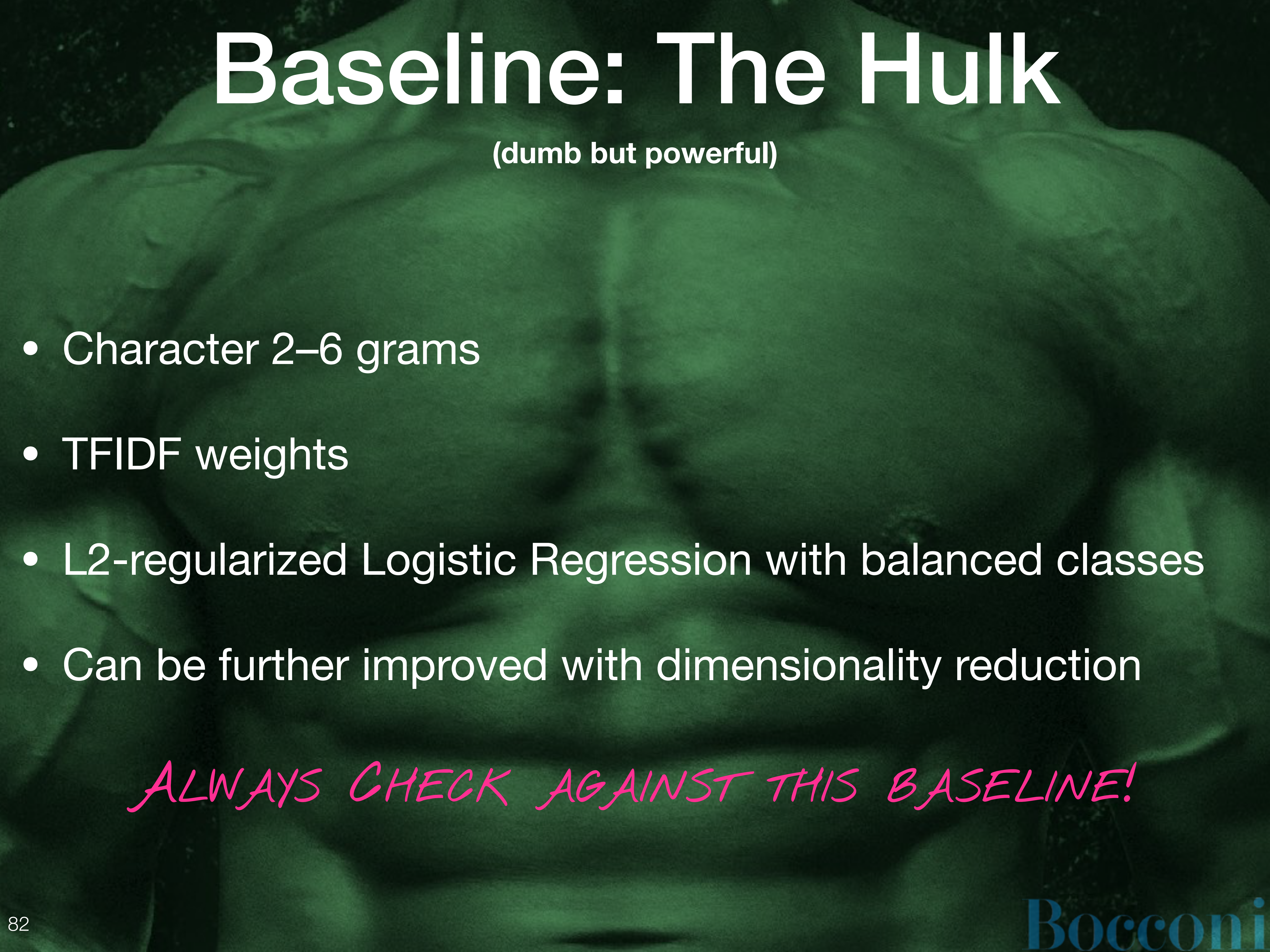
x	**y**	**ŷ**	
frog	1	1	
deer	1	1	
wolf	1	1	
dog	1	1	true positive
bear	1	1	
fish	1	1	
bird	1	1	
cat	1	1	
stone	0	1	
tree	0	1	false positive

ACCURACY = 8/10 = 0.8

PRECISION = 8/10 = 0.8

RECALL = 8/8 = 1.0

F1 = 0.9



Baseline: The Hulk

(dumb but powerful)

- Character 2–6 grams
- TFIDF weights
- L2-regularized Logistic Regression with balanced classes
- Can be further improved with dimensionality reduction

ALWAYS CHECK AGAINST THIS BASELINE!

Regularization

Bocconi

Regularization

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

D-BY-1

VECTOR

$$||w||$$

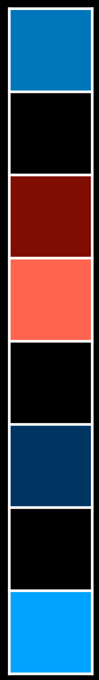
w^T

Regularization Norms

L1 NORM

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

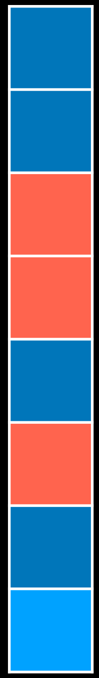
SPARSE



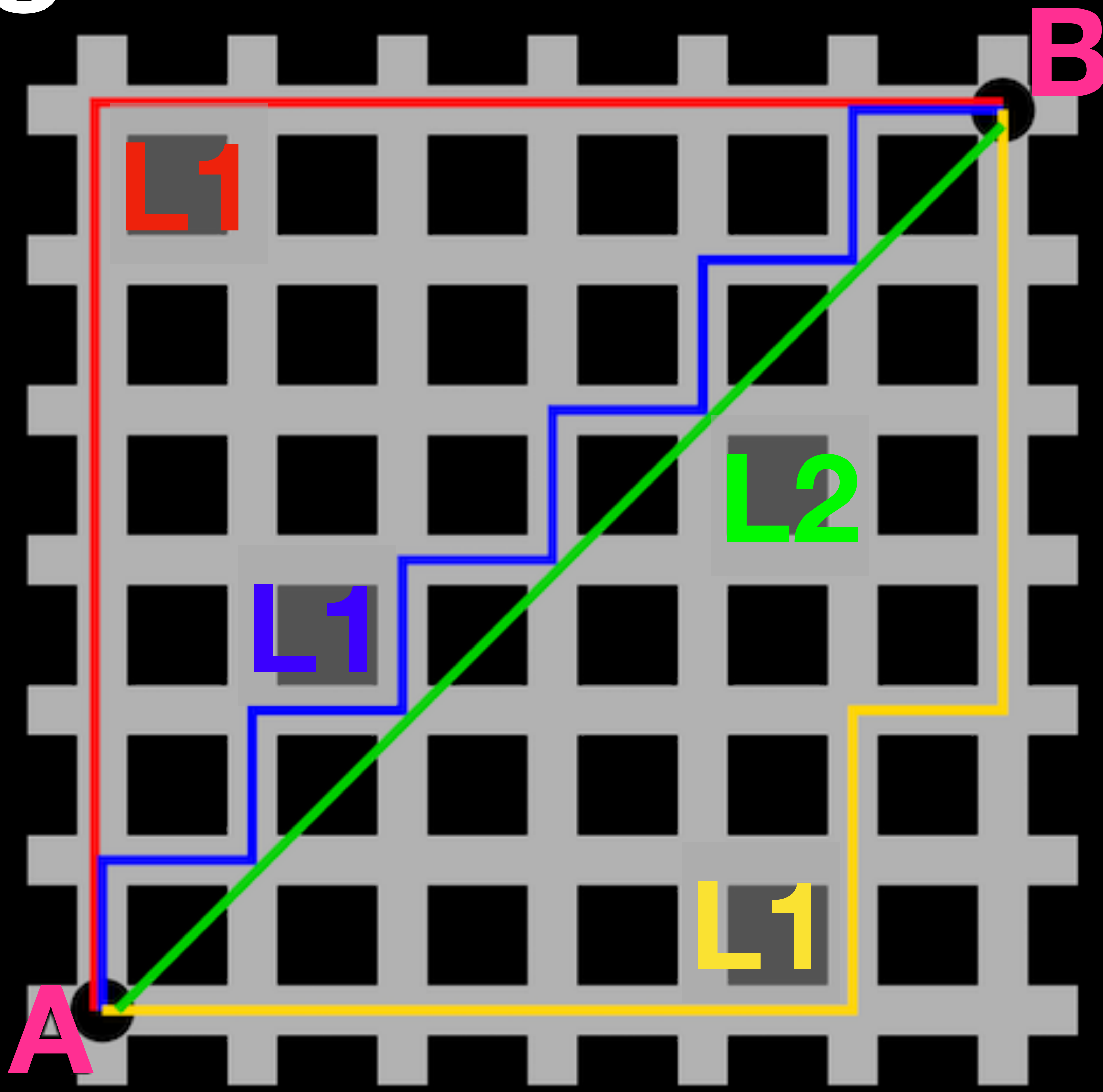
L2 NORM

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

EVENLY DISTRIBUTED



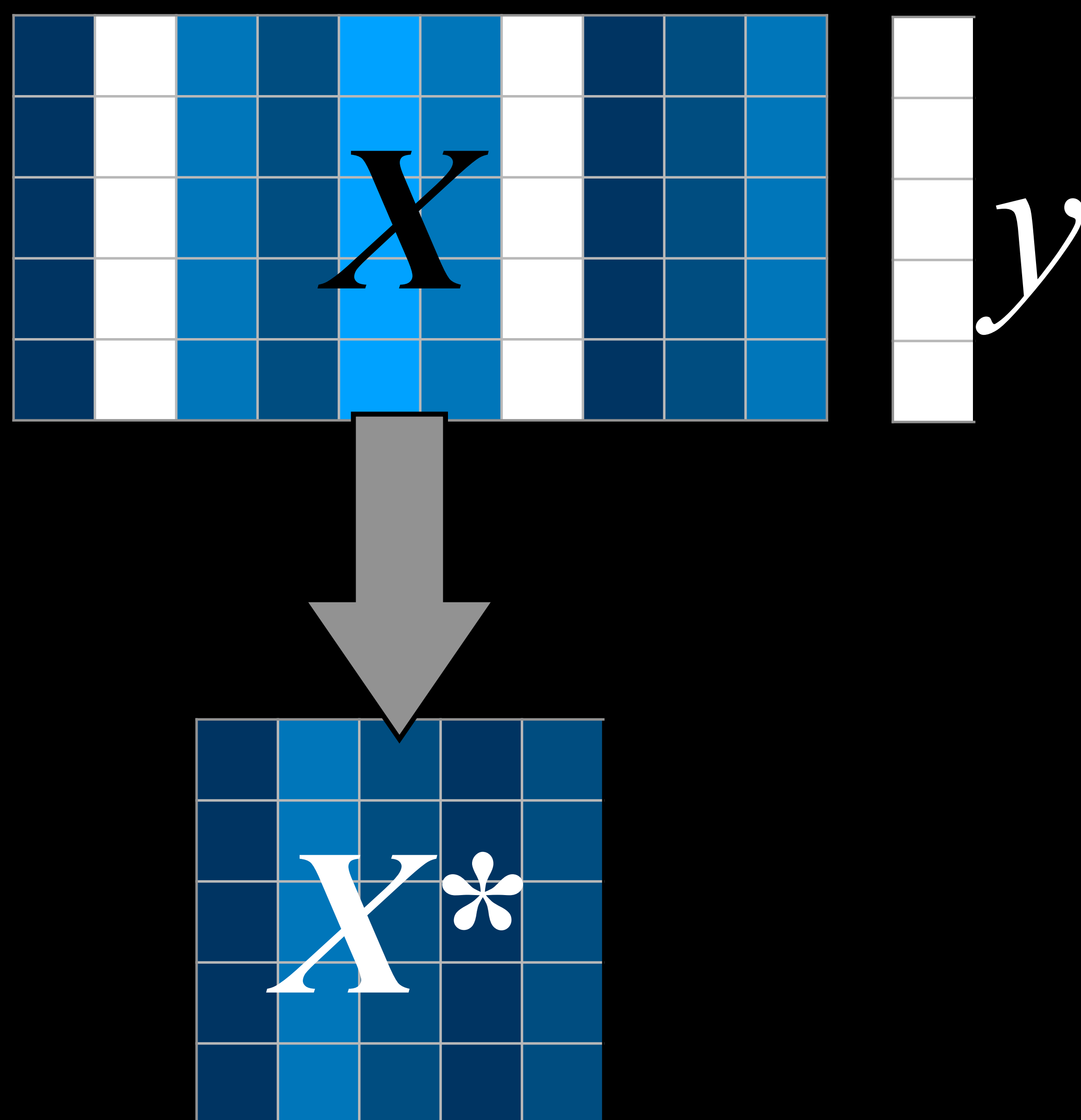
Regularization Norms



Bocconi

Feature Selection

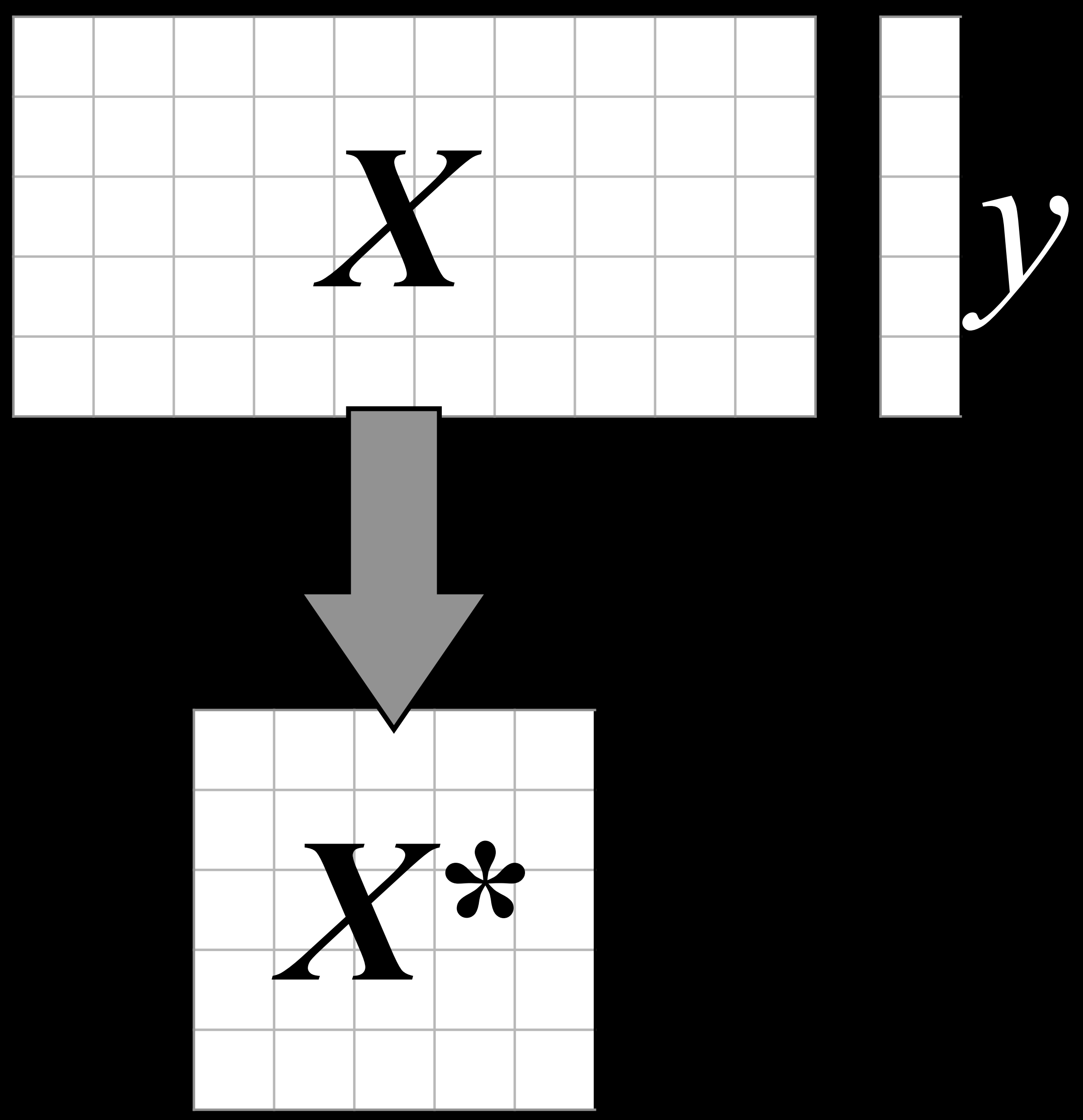
Chi-Squared Selection



MEASURE CHI2 VALUE (CORRELATION) FOR EACH FEATURE WITH TARGET, SELECT TOP K BY CUTOFF

Bocconi

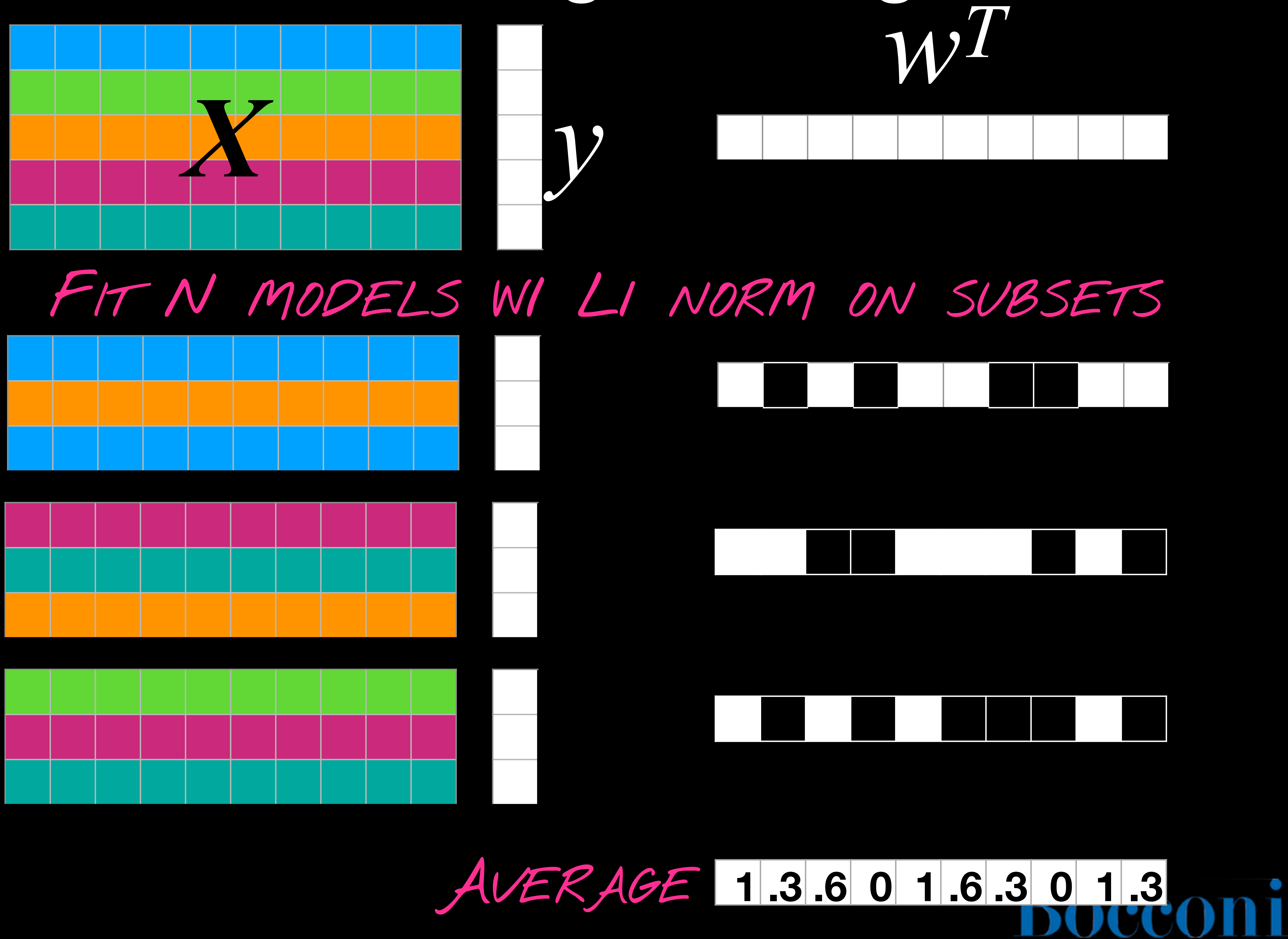
Dimensionality Reduction



REDUCE DIMENSIONALITY TO PREVENT SPURIOUS CORRELATIONS WITH TARGET, BRING OUT LATENT DIMENSIONS

Bocconi

Randomized Logistic Regression



Wrapping Up

Take Home Points

- **Preprocessing** removes noise and unwanted variation
- Words and texts can be represented as:
  - **Sparse, discrete** feature vectors (counts/TFIDF)
  - **Dense, continuous embedding** vectors
- Choose the appropriate performance **metric**
- Choose an informative **baseline**
- **Regularize, regularize, regularize**
- **Feature selection** can improve performance and provide insights