Natural Language Processing

Lecture 05

Dirk Hovy

dirk.hovy@unibocconi.it

@dirk_hovy

Bocconi

Today's Goals

- Inroduce ***n*-grams**
- Learn how to compute *n*-gram probabilities with **Maximum Likelihood Estimation**
- Understand the **Markov assumption**
- Understand the effect of **smoothing**
- Learn how to use probabilistic language models for **inference** and **generation**
- Learn about **bag of words** (BOW) representations
- Learn about forms of **TF-IDF** and its possibilities

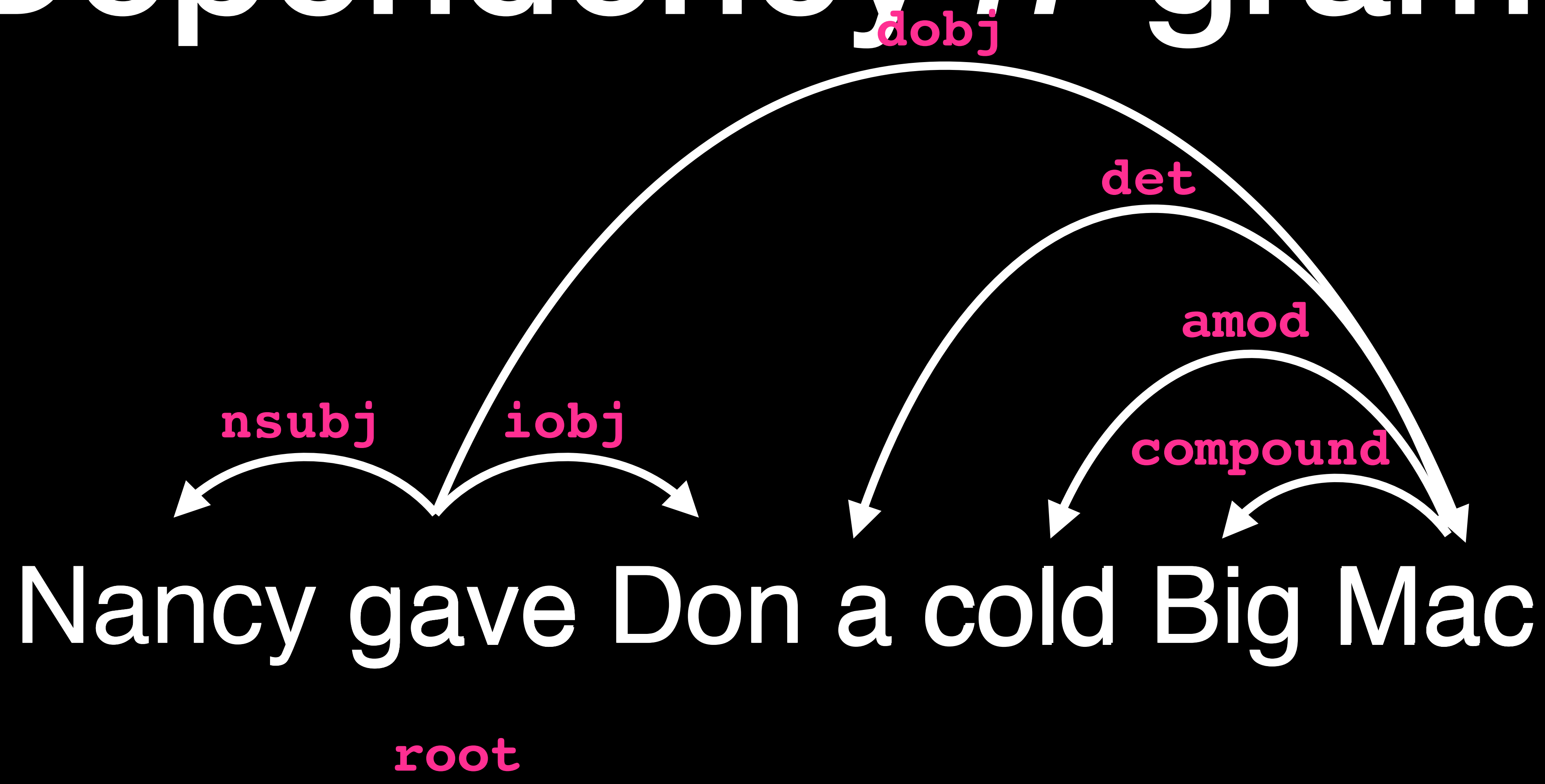
Bocconi

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, ...`

Dependency *n*-grams



Using n-grams in Language Models

Bocconi

What are LMs? Ranking Sentences

LANGUAGE MODEL

P(S)

- I love to models language **0.03**
- I love language models **0.62**
- I love to language model **0.48**

MOST LIKELY TO BE OBSERVED

Bocconi

Machine Translation

Text Generation

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-3, WILL COME BACK LATER

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Language Models in Short

1. Break sentence into *n*-grams
2. Increase their counts
3. Compute probabilities
4. Multiply them together

Probability of a Sentence?

HOW OFTEN WE

HAVE SEEN SENTENCE S

$$P(S) = \frac{c(S)}{\sum_{z \in Z} c(Z)}$$

...ALL POSSIBLE SENTENCES

$$= \frac{x}{\text{INFINITY}} = 0$$

Can We Make it Simpler?

BREAK IT DOWN!

$$P(S) = P(w_1, w_2, \ldots, w_n)$$

JOINT PROBABILITY OF ALL THE WORDS

Andrey Andreyevich Markov (1856 – 1922)

Markov Assumption

BREAKING IT DOWN:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^{N} P(w_i | w_1, \dots, w_{i-1})$$

HISTORY

LIMITING THE HISTORY:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^{N} P(w_i | w_{\underline{i-k}}, \dots, w_{i-1})$$

Bocconi

Markov Models:

UNIGRAM MODEL (K=0)

$$P(w_1, w_2, ..., w_n) \approx \prod_{i=1}^{N} P(w_i)$$

BIGRAM MODEL (K=1)

$$P(w_1, w_2, ..., w_n) \approx \prod_{i=1}^{N} P(w_i|w_{i-1})$$

TRIGRAM MODEL (K=2)

$$P(w_1, w_2, ..., w_n) \approx \prod_{i=1}^{N} P(w_i|w_{i-2},w_{i-1})$$

A Trigram Model

*** * The weather today is fine STOP**

$$P(S) = P(w_1, \ldots, w_n) = P(\texttt{The} \mid \texttt{* *})$$
$$\times\, P(\texttt{weather} \mid \texttt{* The})$$
$$\times\, P(\texttt{today} \mid \texttt{The weather})$$
$$\times\, P(\texttt{is} \mid \texttt{weather today})$$
$$\times\, P(\texttt{fine} \mid \texttt{today is})$$
$$\times\, P(\texttt{STOP} \mid \texttt{is fine})$$

CHAIN RULE

Count in 57m Tweets

MAXIMUM LIKELIHOOD ESTIMATION (MLE)

```
12 The weather today is just
 9 The weather today is so
 9 the weather today is slightly
 8 The weather today is perfect
 5 The weather today is beautiful
 4 The weather today is slightly
 3 the weather today is so
 3 the weather today is perfect
 3 The weather today is nearly
 3 the weather today is bitter
 3 The weather today is absolutely
 2 The weather today is wonderful
 2 The weather today is beyond
 2 The weather today is amazing
 2 The weather today is a
 1 the weather today is worth
 1 the weather today is weird
 1 The weather today is too
 1 the weather today is the
 1 The weather today is that
 1 the weather today is that
 1 The weather today is splendid
 1 THE WEATHER TODAY IS SO
 1 the weather today is simply
 1 The weather today is sickening
 1 The weather today is seriously
 1 The weather today is pretty
 1 the weather today is pretty
 1 The weather today is Perrfff
 1 the weather today is PERFECT
```

Dealing with the Unknown: Smoothing

Many Counts are Still 0

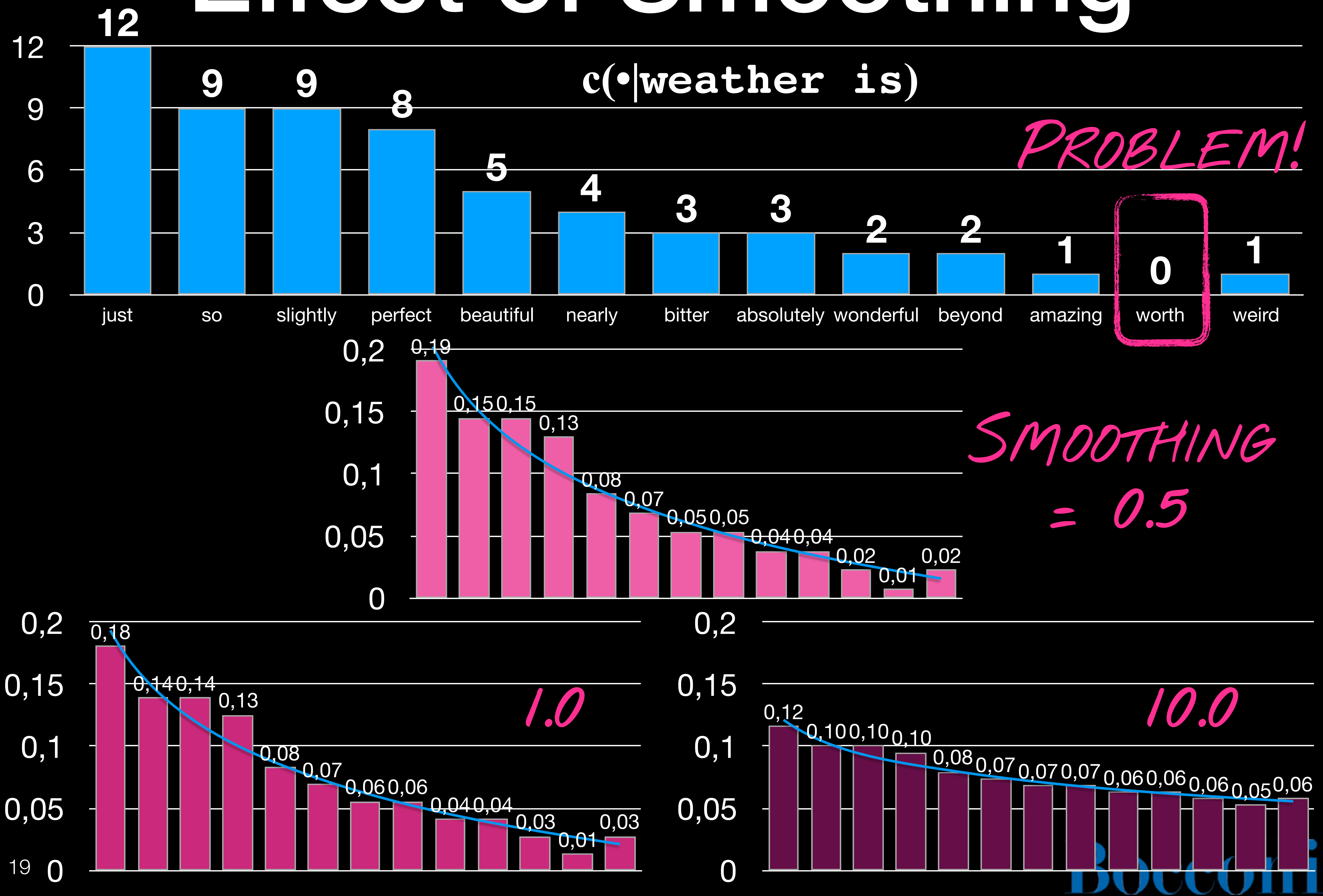
`* * The weather today is fine STOP`

$$P(S) = P(w_1, \ldots, w_n) = P(\texttt{The} \mid \texttt{* *})$$
$$\times P(\texttt{weather} \mid \texttt{* The})$$
$$\times P(\texttt{today} \mid \texttt{The weather})$$
$$\times P(\texttt{is} \mid \texttt{weather today})$$
$$c(\texttt{fine}) = 0 \qquad \times P(\texttt{fine} \mid \texttt{today is})$$
$$\times P(\texttt{STOP} \mid \texttt{is fine})$$

Add-one (Laplace) smoothing

Pierre-Simon, marquis de Laplace (1749 – 1827)

Effect of Smoothing



Evaluating LMs

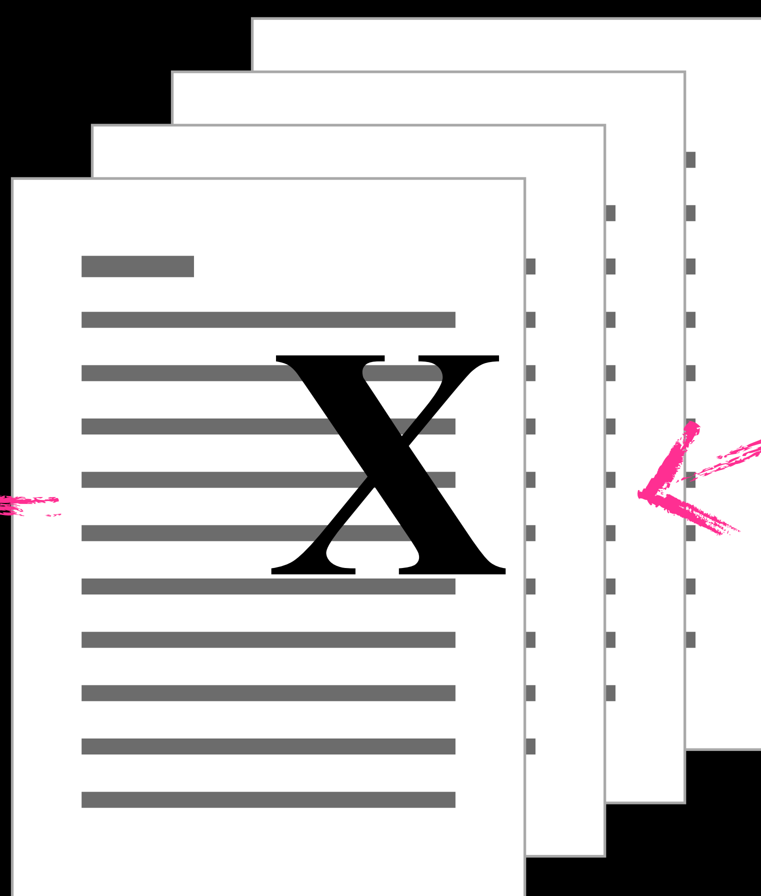
How Good is My Model?

P(S)

PREDICT

TOLD YA!

FIT



ENTROPY

PERPLEXITY $= 2^{H(X)}$

$$= 2^{-\sum_{x \in X} P(x) \cdot \log P(x)}$$

Bocconi

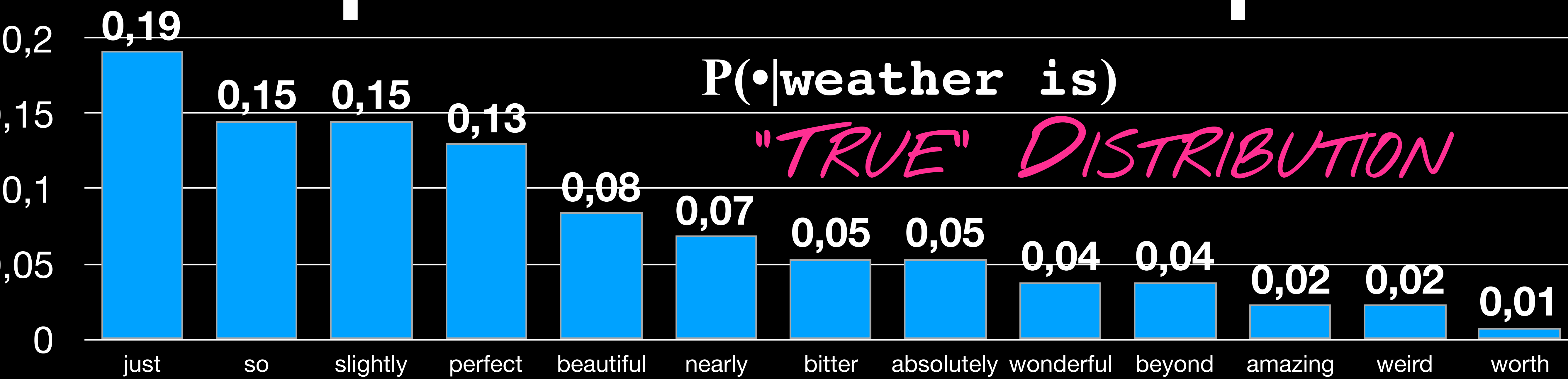
Using LMs for Generation

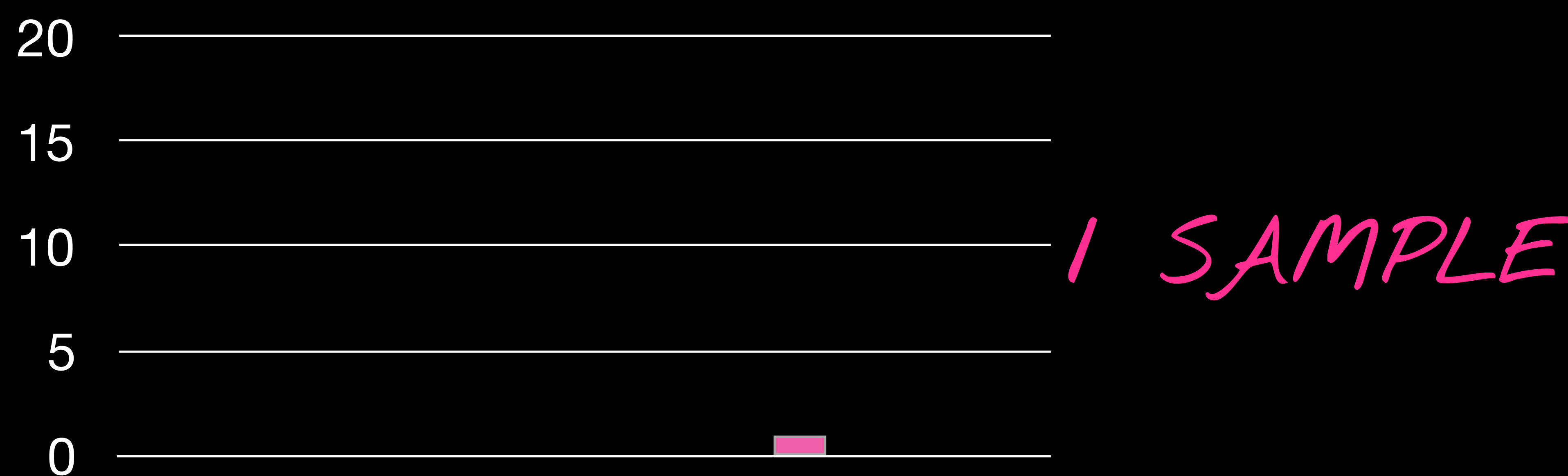
Take a Random Walk

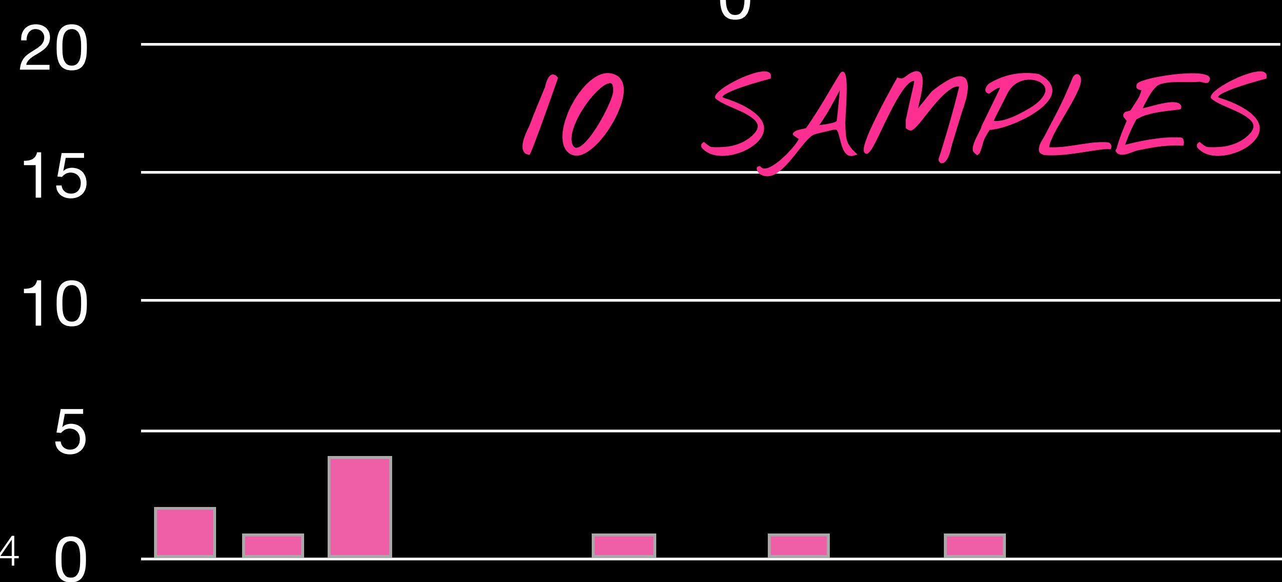
PROPORTIONATELY

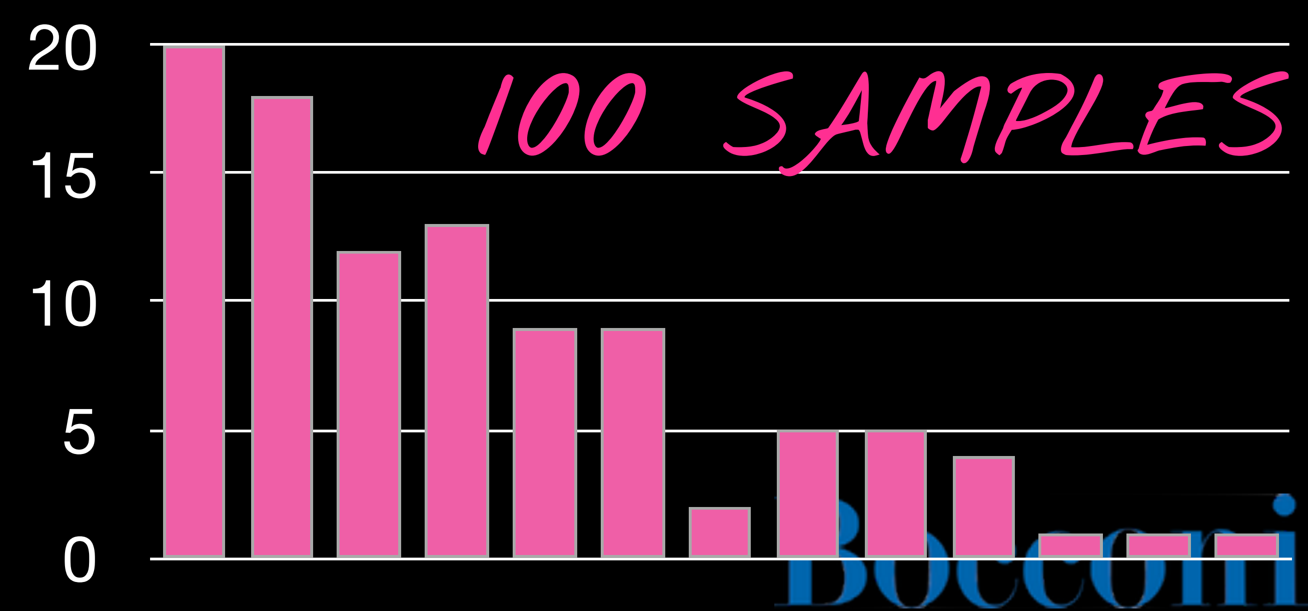
```
Pick a random word w from P(·| * * )

H = [*, *, w]

While H[-1] is not STOP:

   Pick a random word w from P(·|H[-2:])

   H += [w]

return H
```

Proportionate Samples

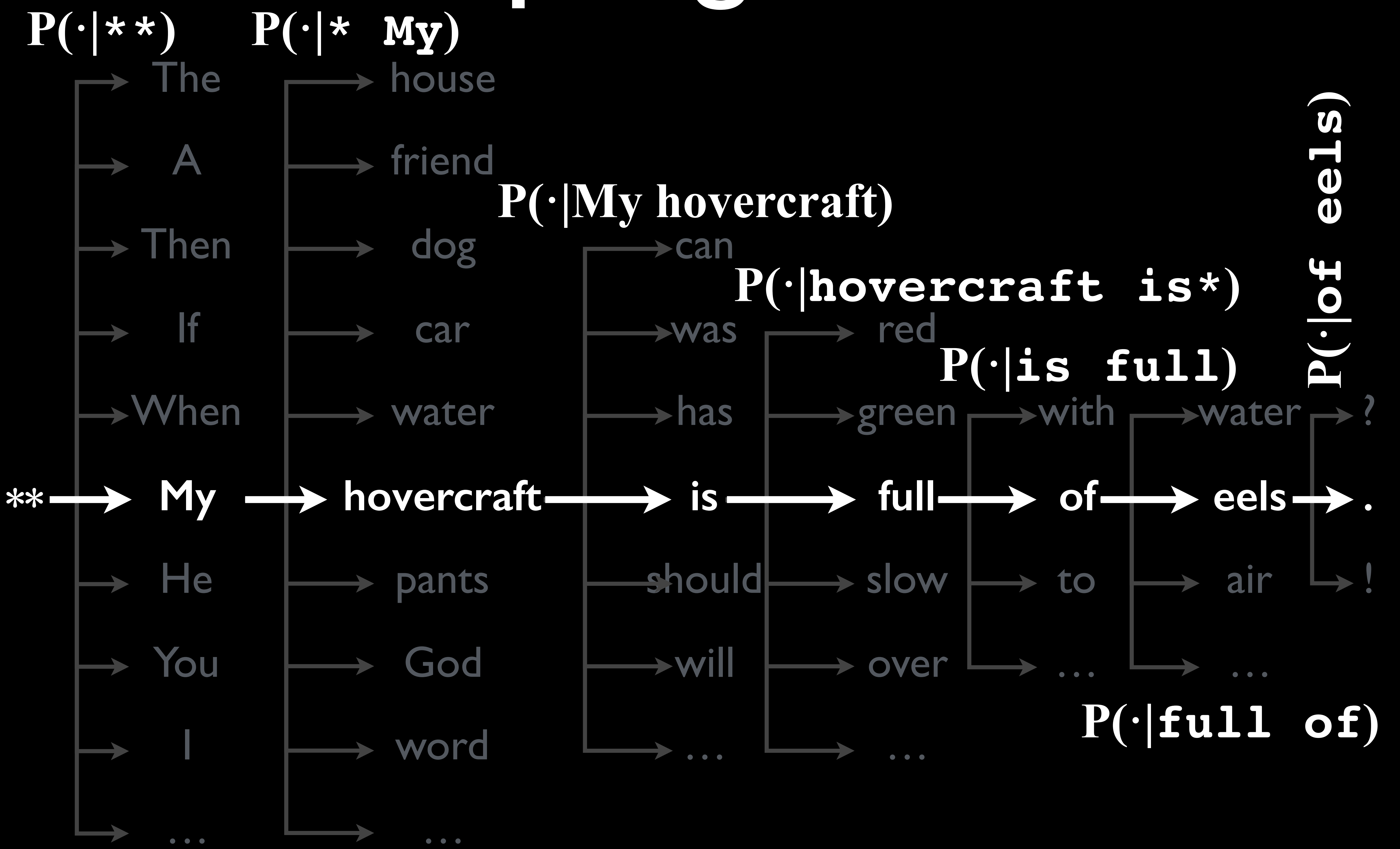








Sampling Words



Language Model in Short

1. Break sentence into *n*-grams MARKOV HORIZON
2. Increase their counts SMOOTHING
3. Compute probabilities MLE
4. Multiply them together MARKOV ASSUMPTION

Bocconi

Using n-grams in Text Analysis

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.

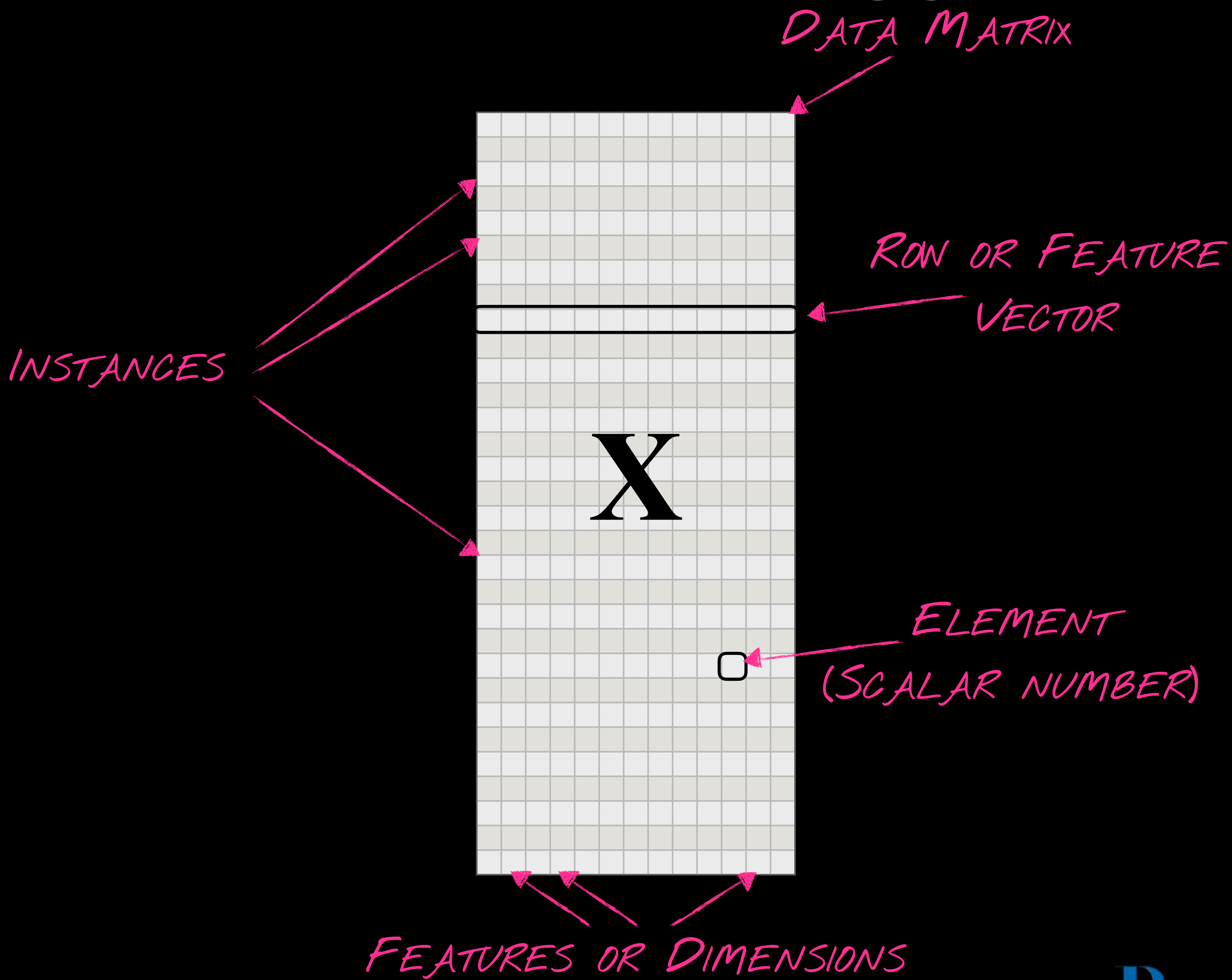
Spam terms:

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

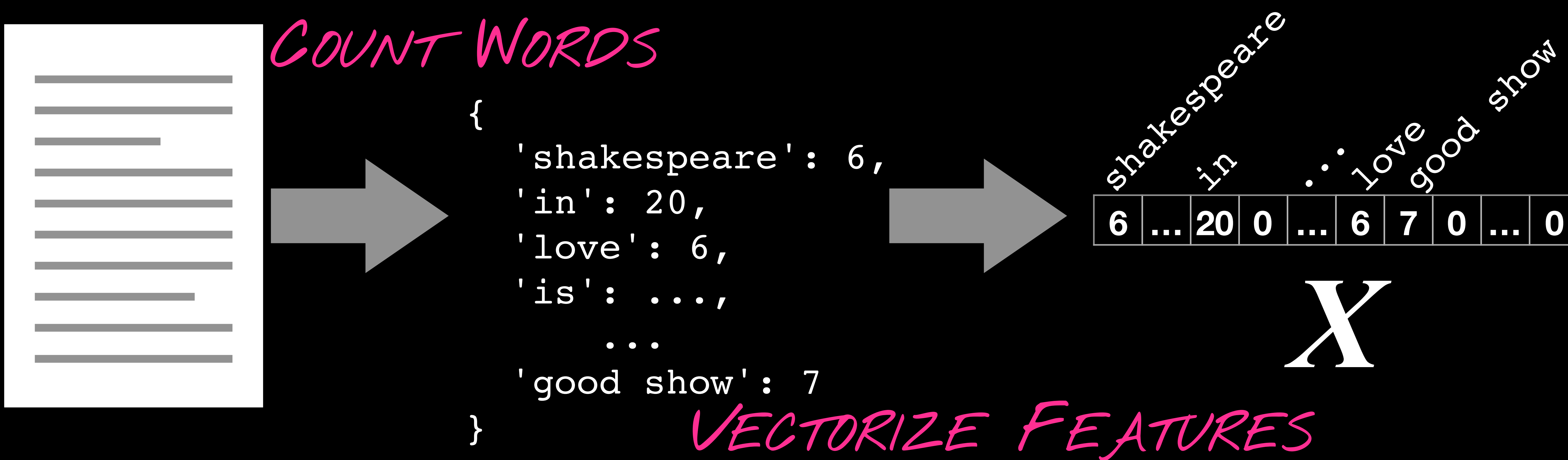
Bocconi

Discrete Representations

Terminology



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.

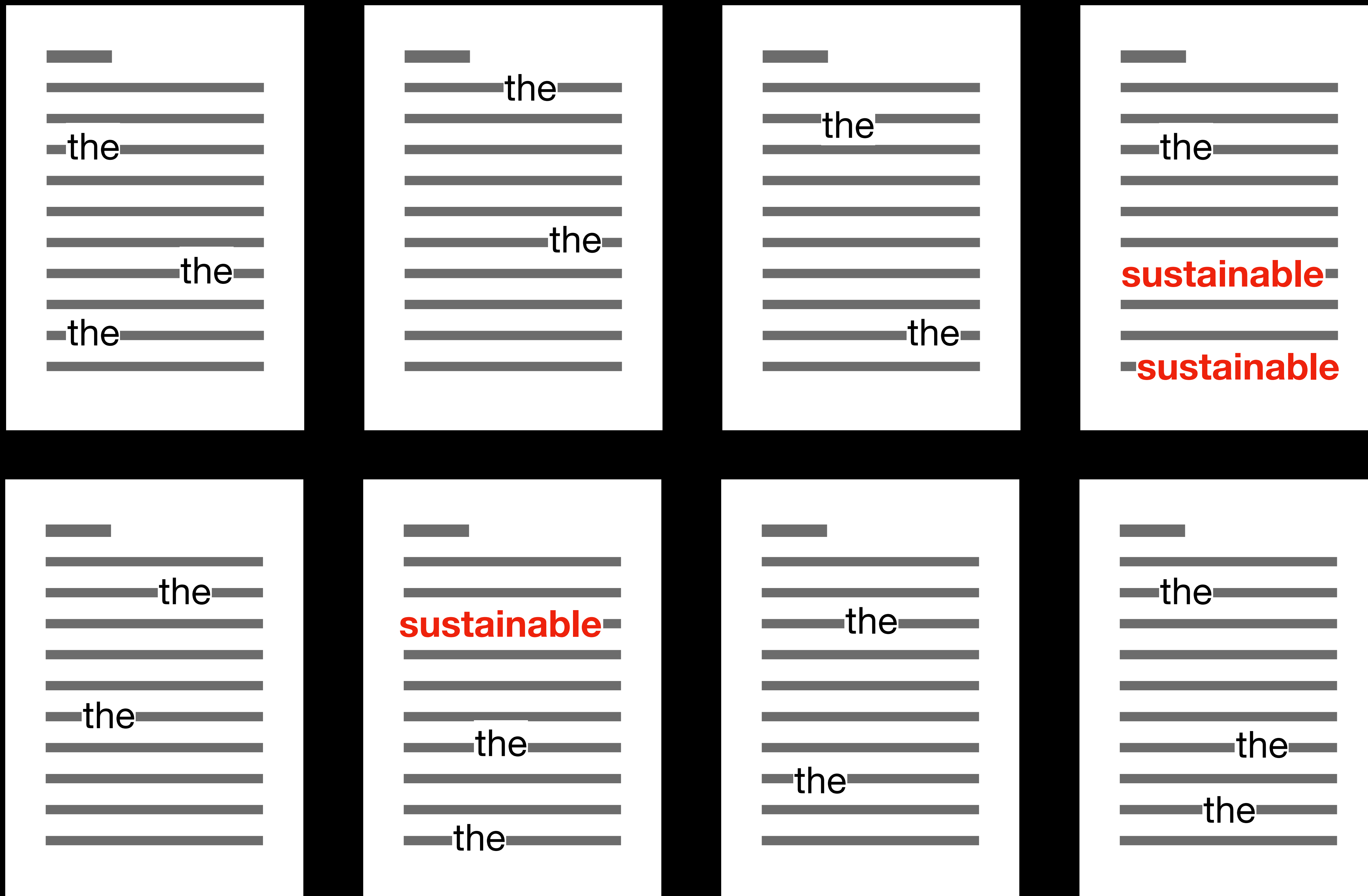
Limit to most frequent/informative words!

Counting Trouble

...AND A MAN NAMED ZIPF

Finding Important Words: TF-IDF

Some Words are Just More Interesting…



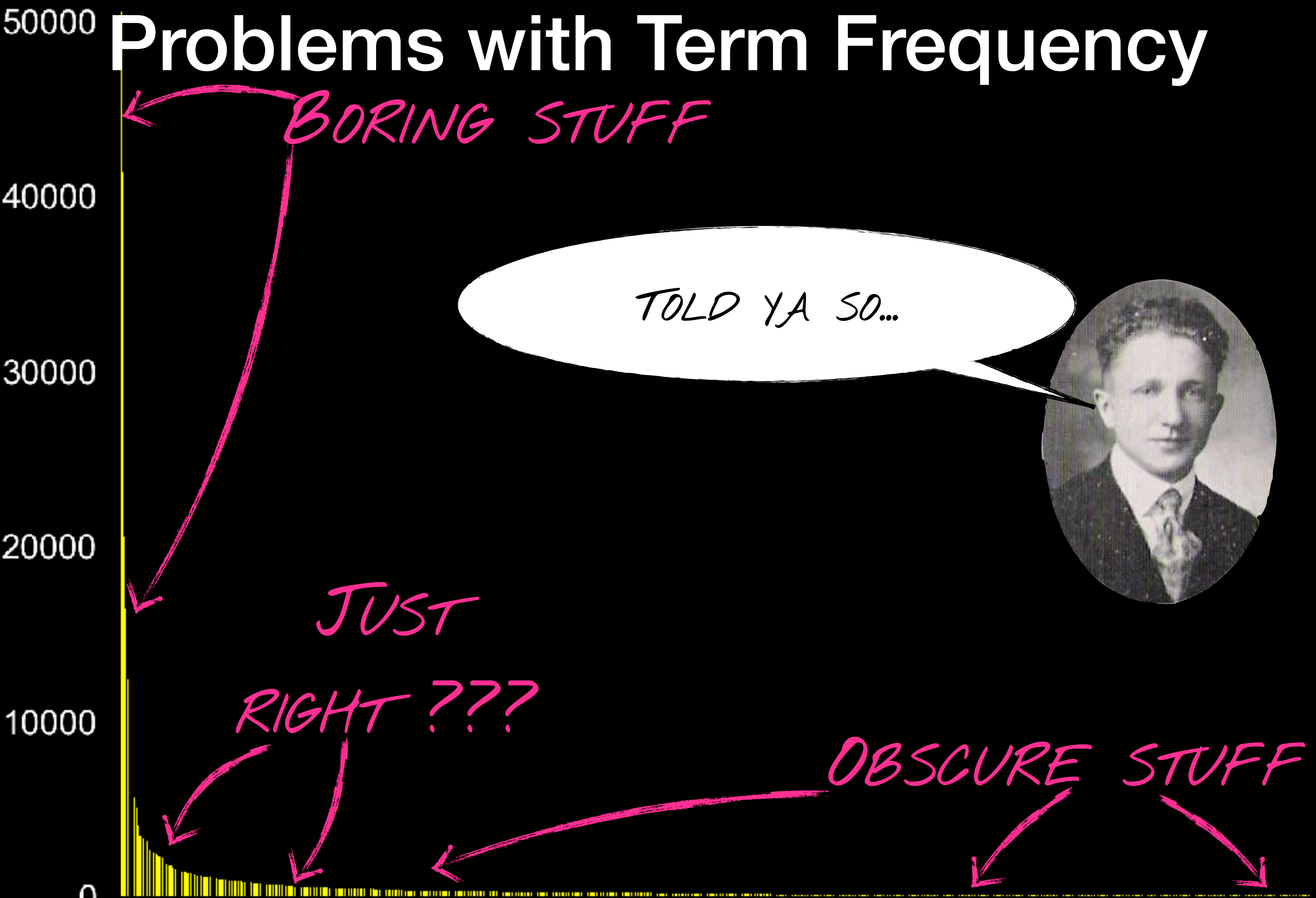
Karen Spärck Jones

THANKS, KAREN!

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

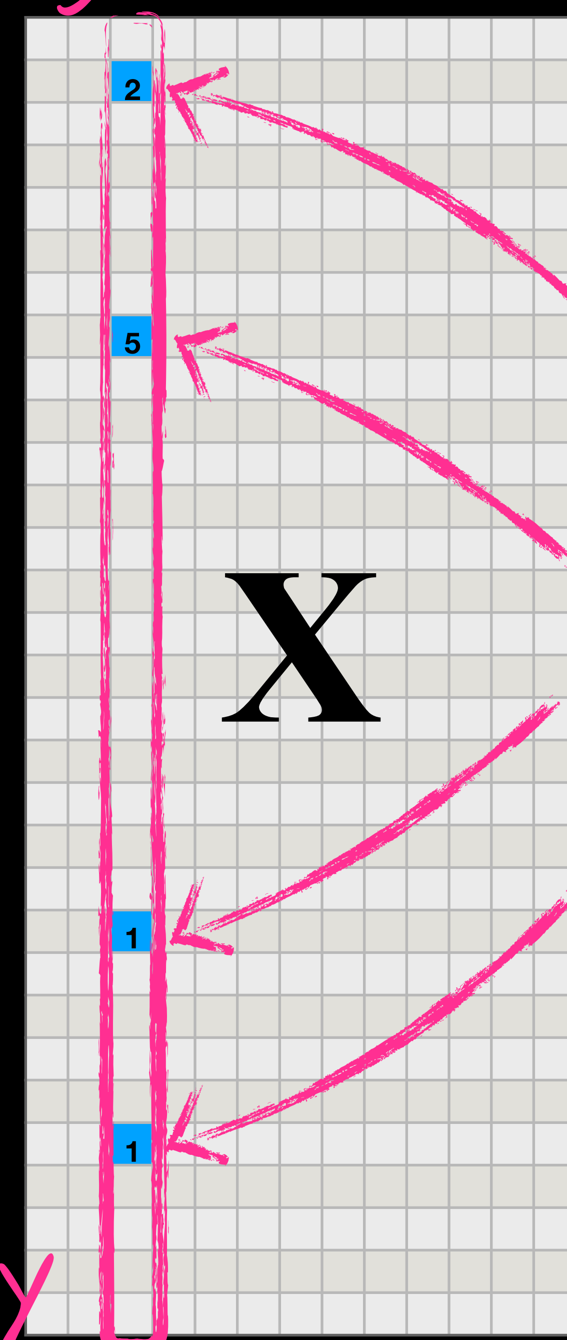
Problems with Term Frequency



Document and Term Frequency

FEATURE

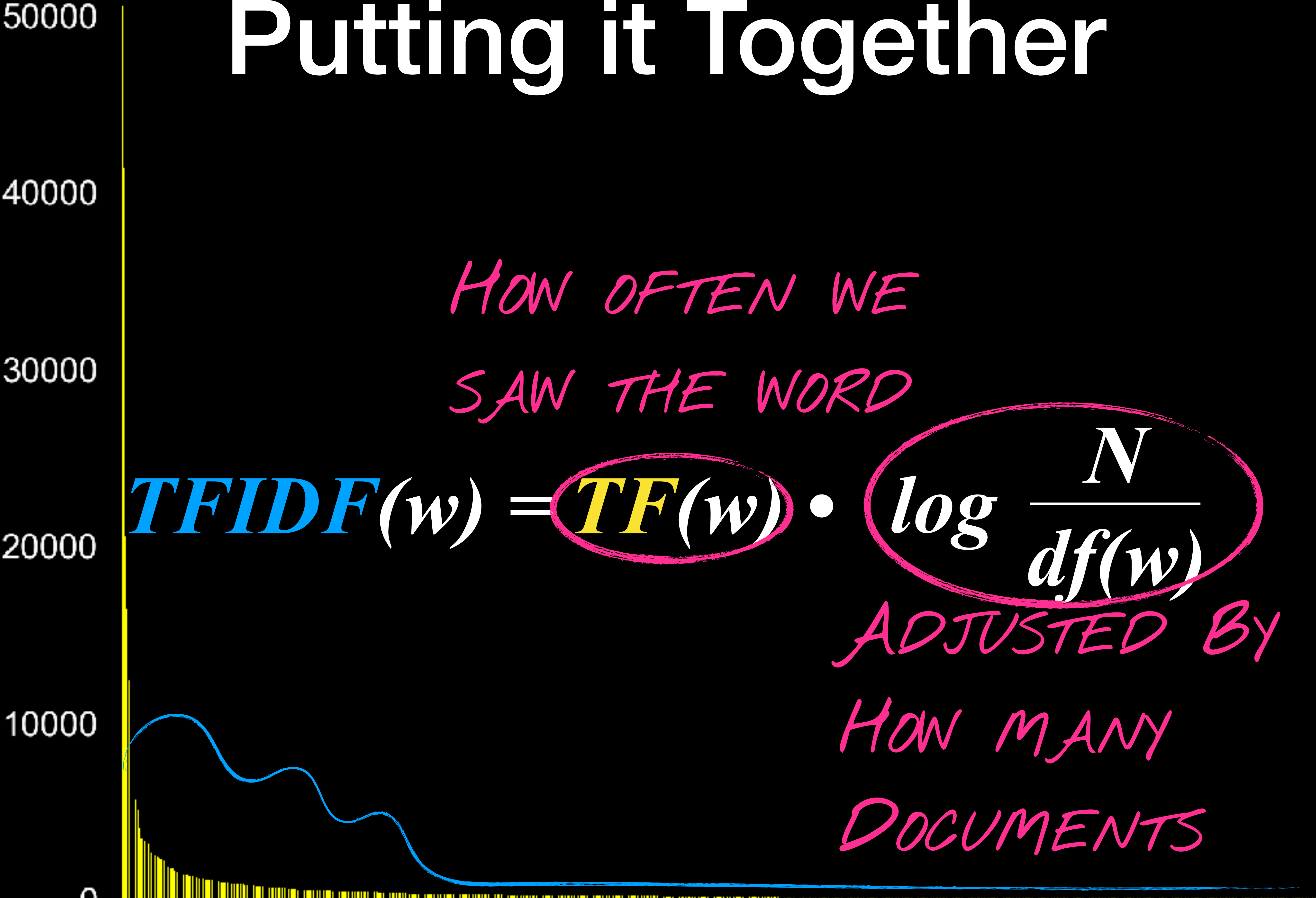
$$IDF = \log \frac{N}{df(w)}$$



DOCUMENT FREQUENCY (COUNT): 4

TERM FREQUENCY (SUM): 9 TF

Putting it Together

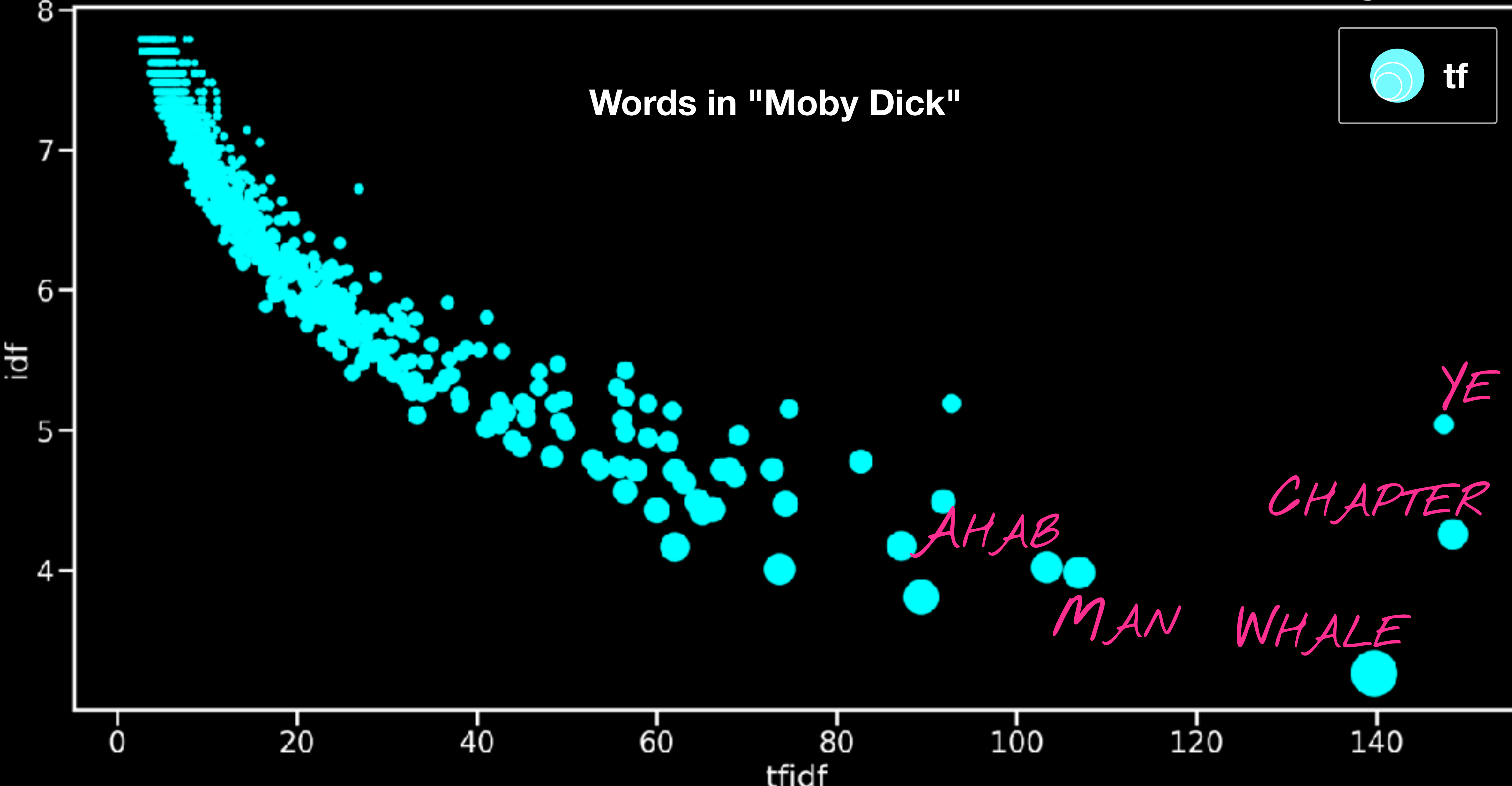


HOW OFTEN WE SAW THE WORD

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

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

Variants

	TF
binary	*1 if word in D, else 0*
raw	*c(word, D)*
relative	*c(word, D) / len(D)*
smooth	*log(c(word, D) + 1)*

	IDF
regular	$\log \frac{N}{df(word)}$
smooth	$\log \frac{N}{df(word) + 1} + 1$

Wrapping up

Take home points

- **Language Models** assign a probability to any sentence, can be used for **text generation**
- The **Markov assumption** breaks sentence probability into a **chain** of word conditional probabilities
- **Markov order** determines the size of the conditional *n*-grams
- **Smoothing** helps address the problem of unseen words
- Words and texts can be represented as **sparse, discrete** feature vectors over counts
- **TF-IDF** finds "bursty" words: medium frequency overall, but concentrated in few documents