

Goals for Today

- Understand the probabilistic approach to language
- 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**

Ranking Sentences

LANGUAGE MODEL

$P(S)$

- I love to models language 0.48
- I love language models 0.62
- I love to language model 0.23

MOST LIKELY TO BE OBSERVED

Language Models in Short

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

Recap Probability

Sample space & Pebbles

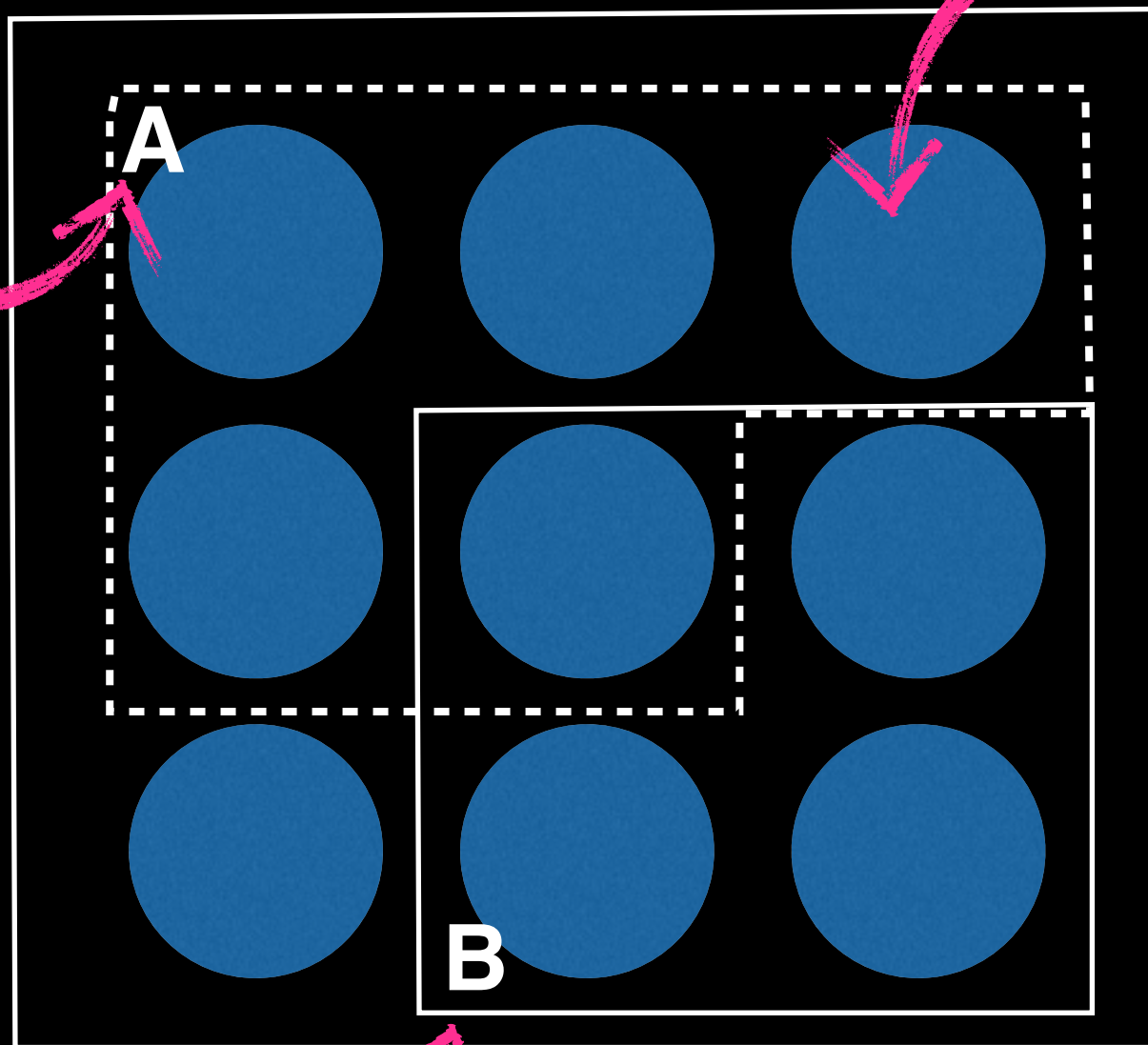
SAMPLE SPACE

OUTCOME

EVEN

B

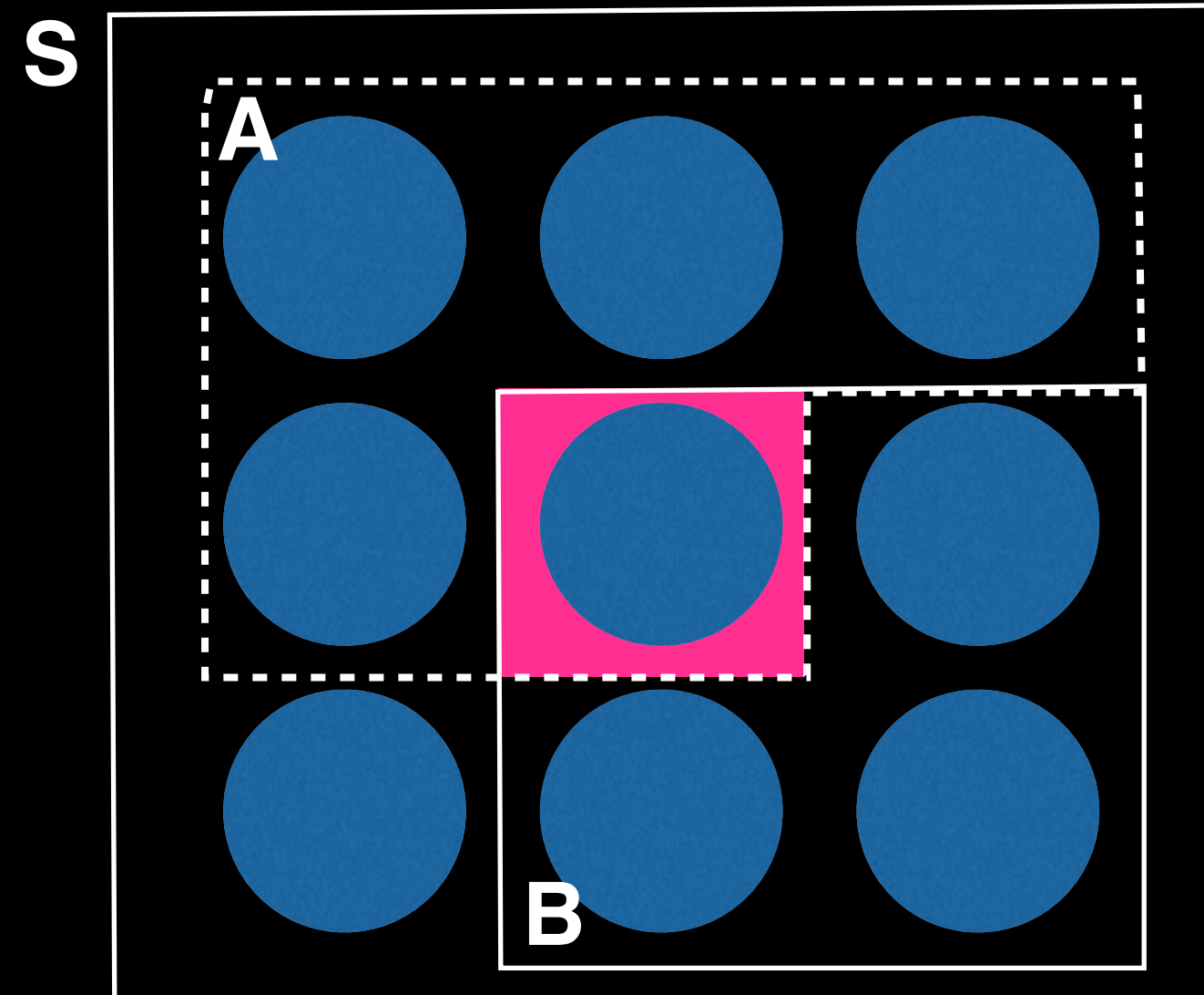
A



ANOTHER EVENT

Joint Probabilities

INTERSECTION

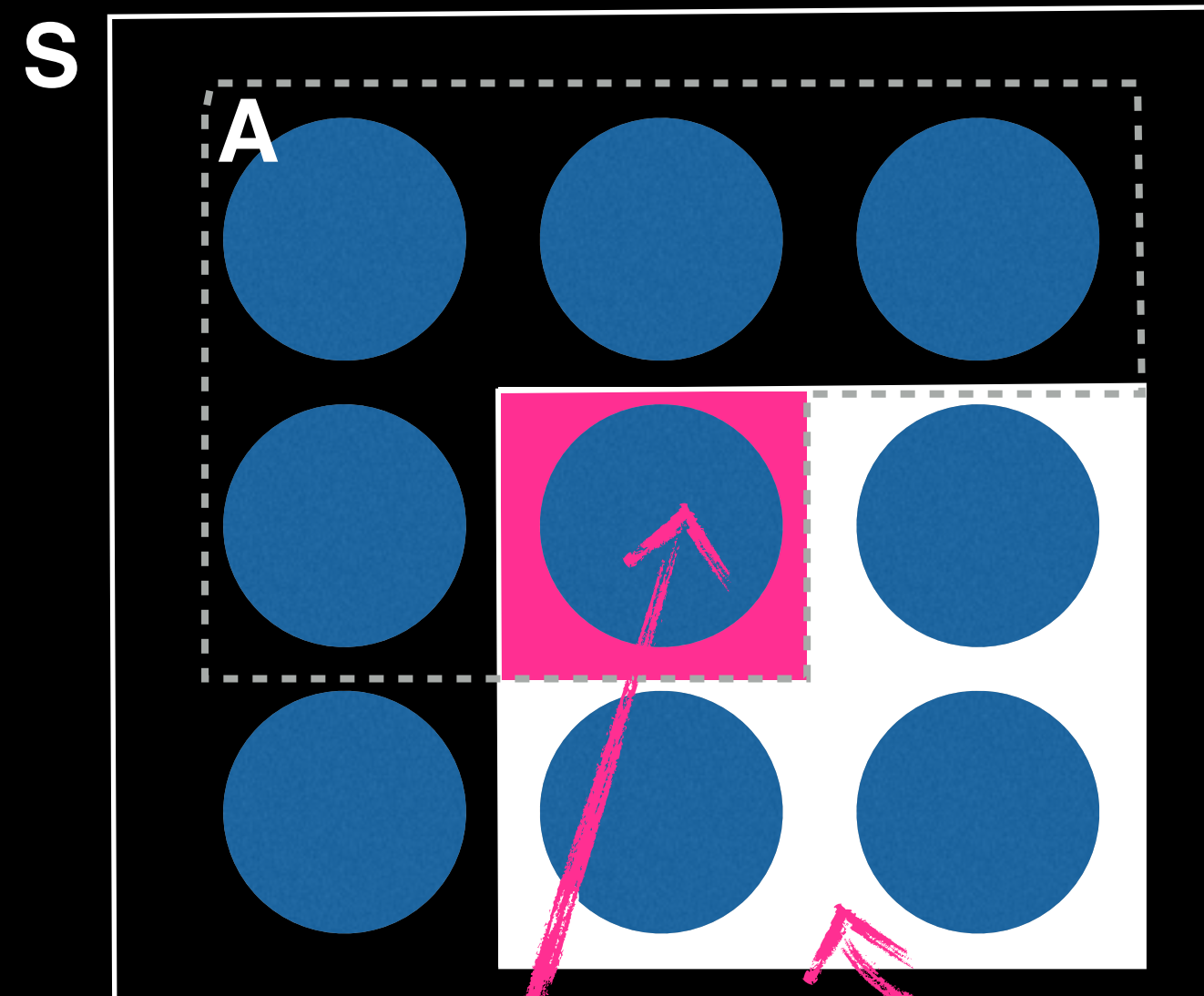


Sets: $A \cap B$

Probability: $P(A,B)$

Meaning: “AND”,
joint probability

Conditional probability

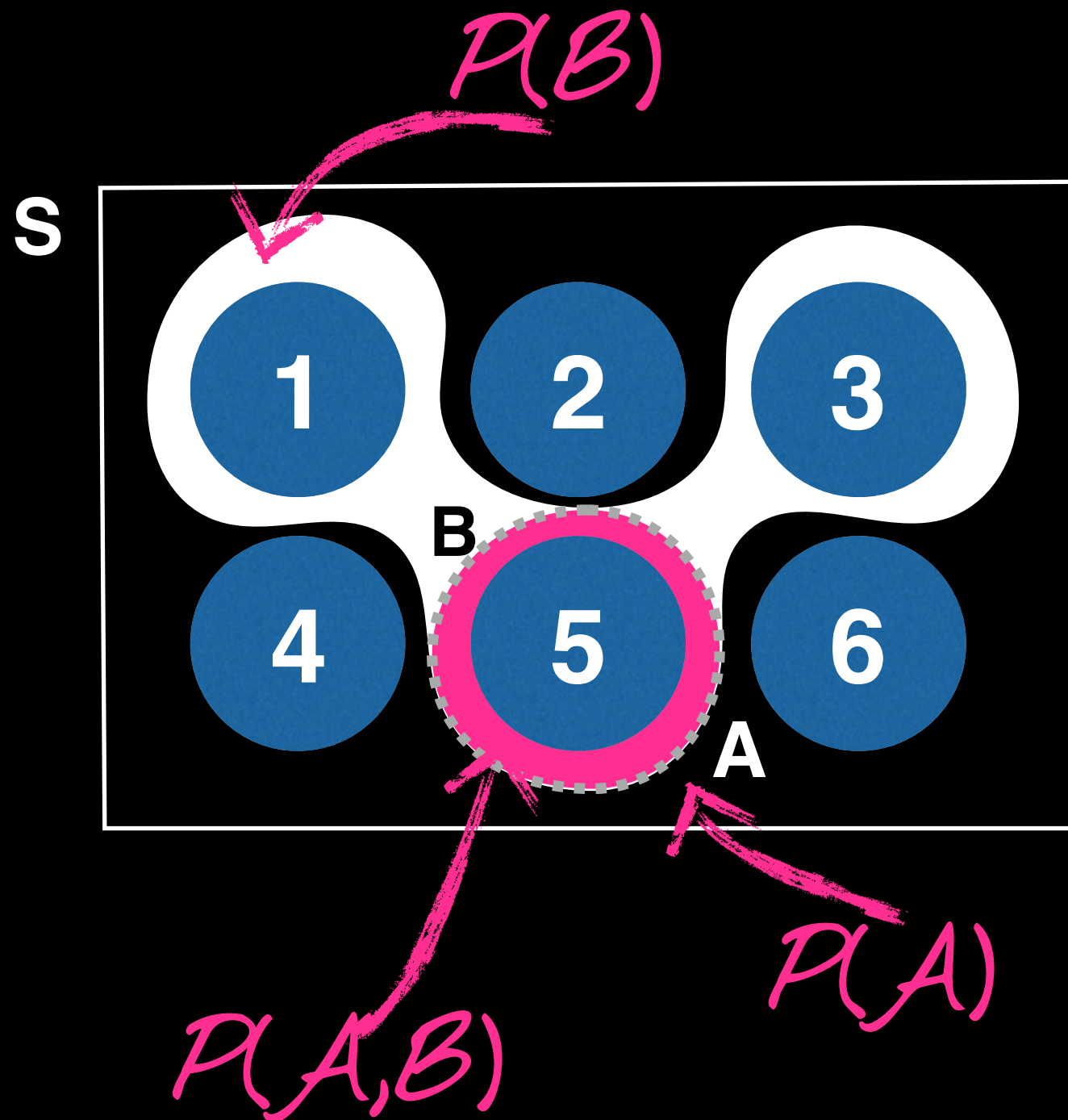


$$P(A|B) = \frac{P(A, B)}{P(B)}$$
$$= \frac{1}{4} = 0.25$$

P(A, B)
joint probability

P(B)

Conditional probability



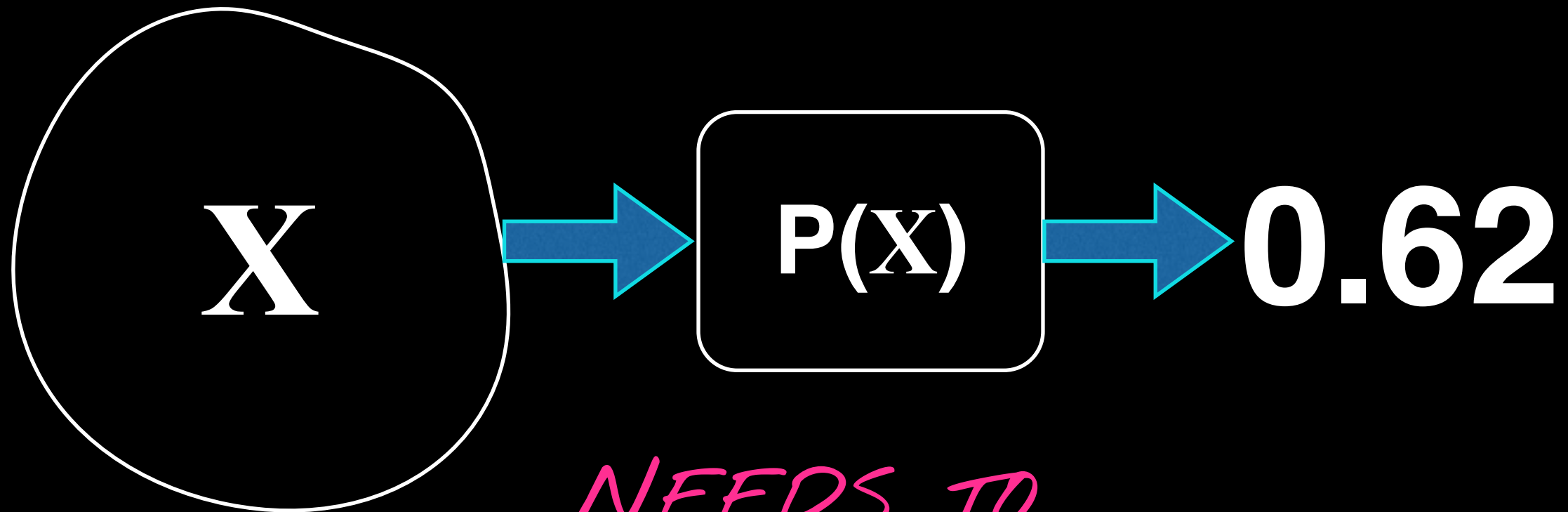
A = "get a 5" $1/6$

B = "odd number" $3/6$

A, B = "odd number and 5" $1/6$

$$P(A|B) = \frac{P(A, B)}{P(B)}$$
$$= \frac{1/6}{3/6} = 1/3$$

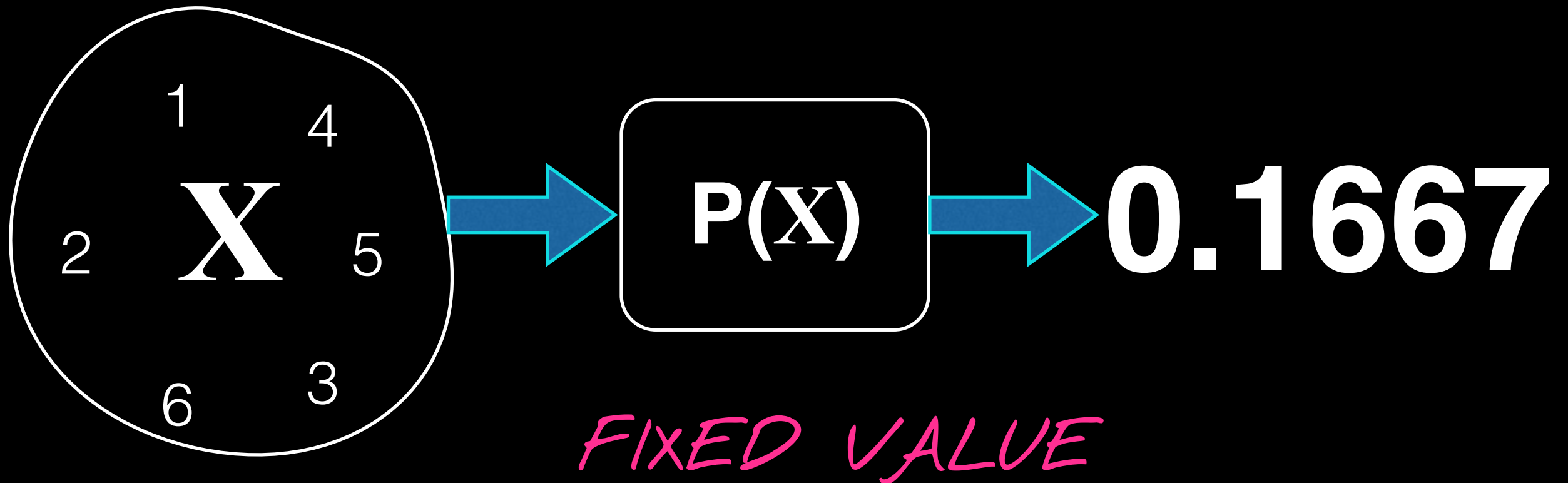
Probability functions



NEEDS TO

- COVER ALL OF X*
- SUM TO 1.0*

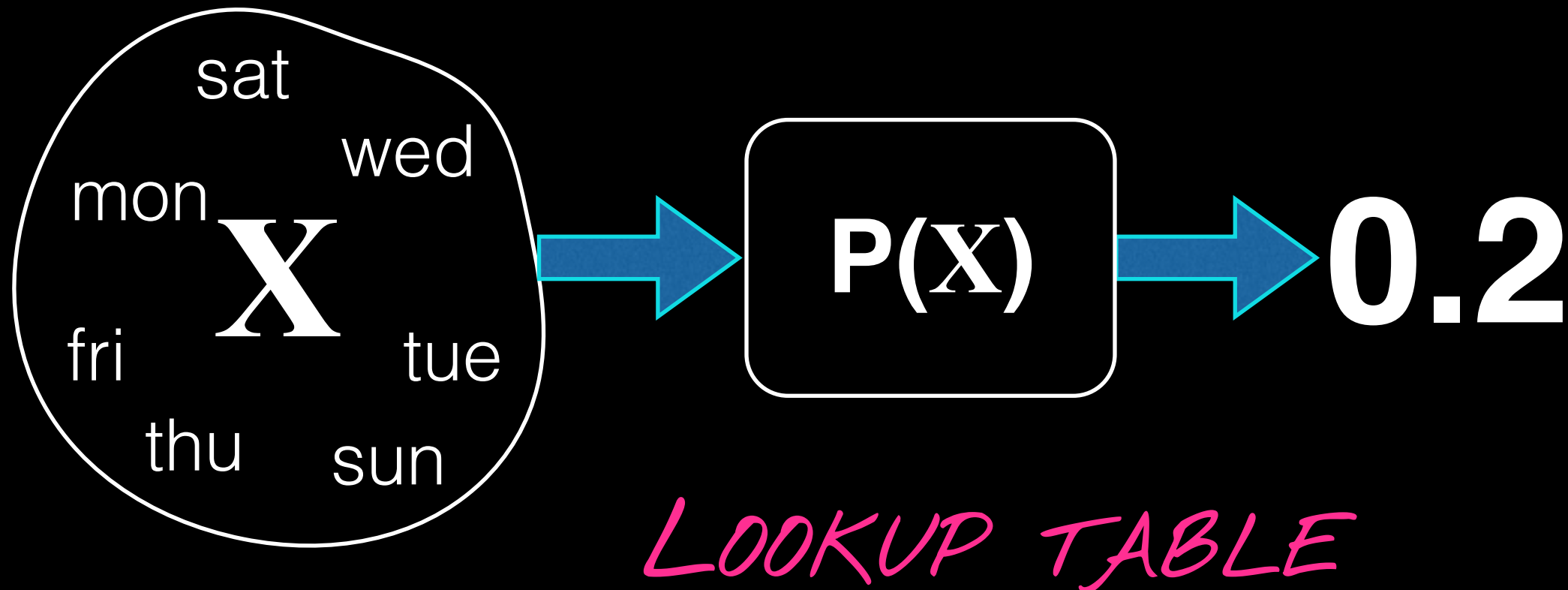
Probability functions



FIXED VALUE

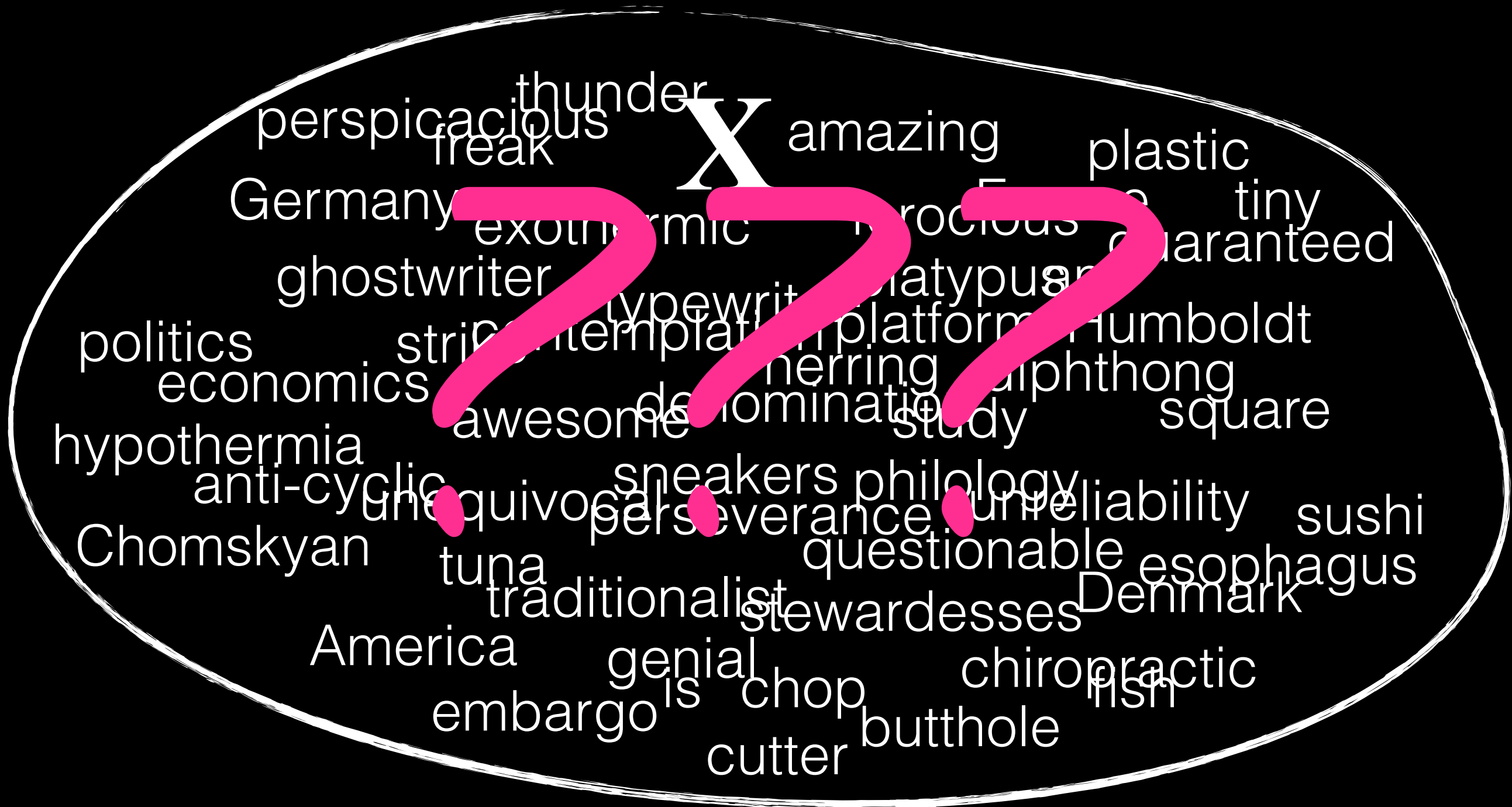
```
def p(number on_die):  
    return 0.1667
```

Probability functions



```
def p(me_at_work):  
    if me_at_work in ["mon", "tue", "wed", "thu", "fri"]:  
        return 0.2  
    else:  
        return 0.0
```

Probability functions



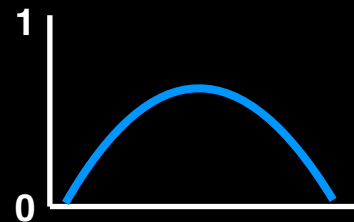
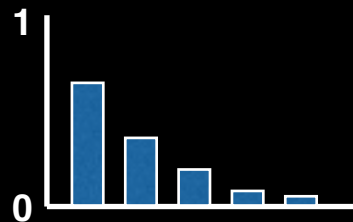
Probability distributions

- compute probability for any x
- mathematical way to describe $P(x)$

```
def p(number_on_die, die_total):  
    return number_on_die/die_total
```

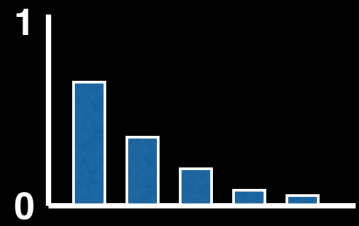
→ $P(x; N) = \frac{1}{N}$

- discrete or continuous



- define “shape” and properties with parameters

Uniform distribution

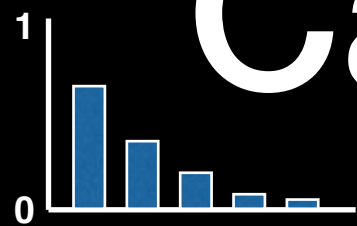


Parameters: N *NUMBER OF EVENTS*

Function: $P(x; N) = \frac{1}{N}$ *SUMS TO 1.0*



Categorical distribution



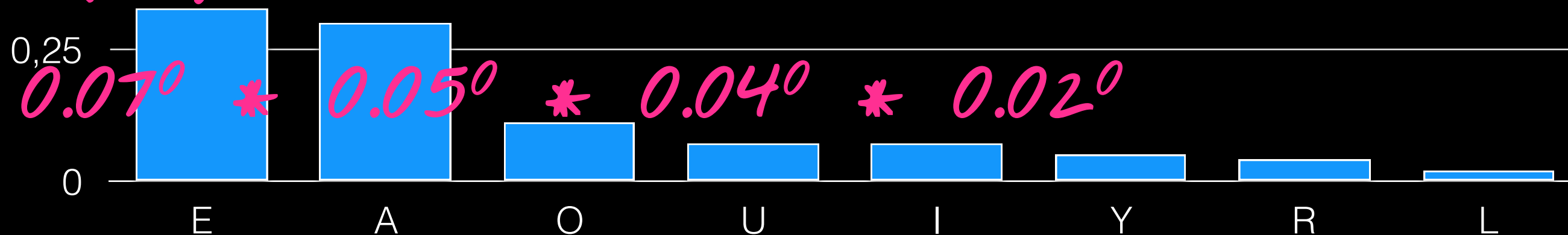
Parameters: θ *VECTOR WITH ALL PROBABILITIES*

Function: $P(x; \theta) = \prod_{j=1}^K \theta_j^{\mathbb{I}(x_j=1)}$ *USE VALUE AT VECTOR POSITION FOR x*

$\theta = [0.33, 0.30, 0.11, 0.07, 0.07, 0.05, 0.04, 0.02]$

probability $v = [0, 1, 0, 0, 0, 0, 0, 0]$

$P(v) = 0.33^0 * 0.30^1 * 0.11^0 * 0.07^0 *$



Syllable nucleus

Probability and Language

Probability of a Word



"It must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term."

Noam Chomsky

- Choose a word w
- Open a page at random and point at a word:
Is it w ?

HOW OFTEN WE
HAVE SEEN w

$$P(w) = \frac{c(w)}{\sum_{v \in V} c(v)}$$

...ALL WORDS

MAXIMUM LIKELIHOOD ESTIMATION

Probability of a Sentence?

HOW OFTEN WE

HAVE SEEN SENTENCE S

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

...ALL POSSIBLE SENTENCES

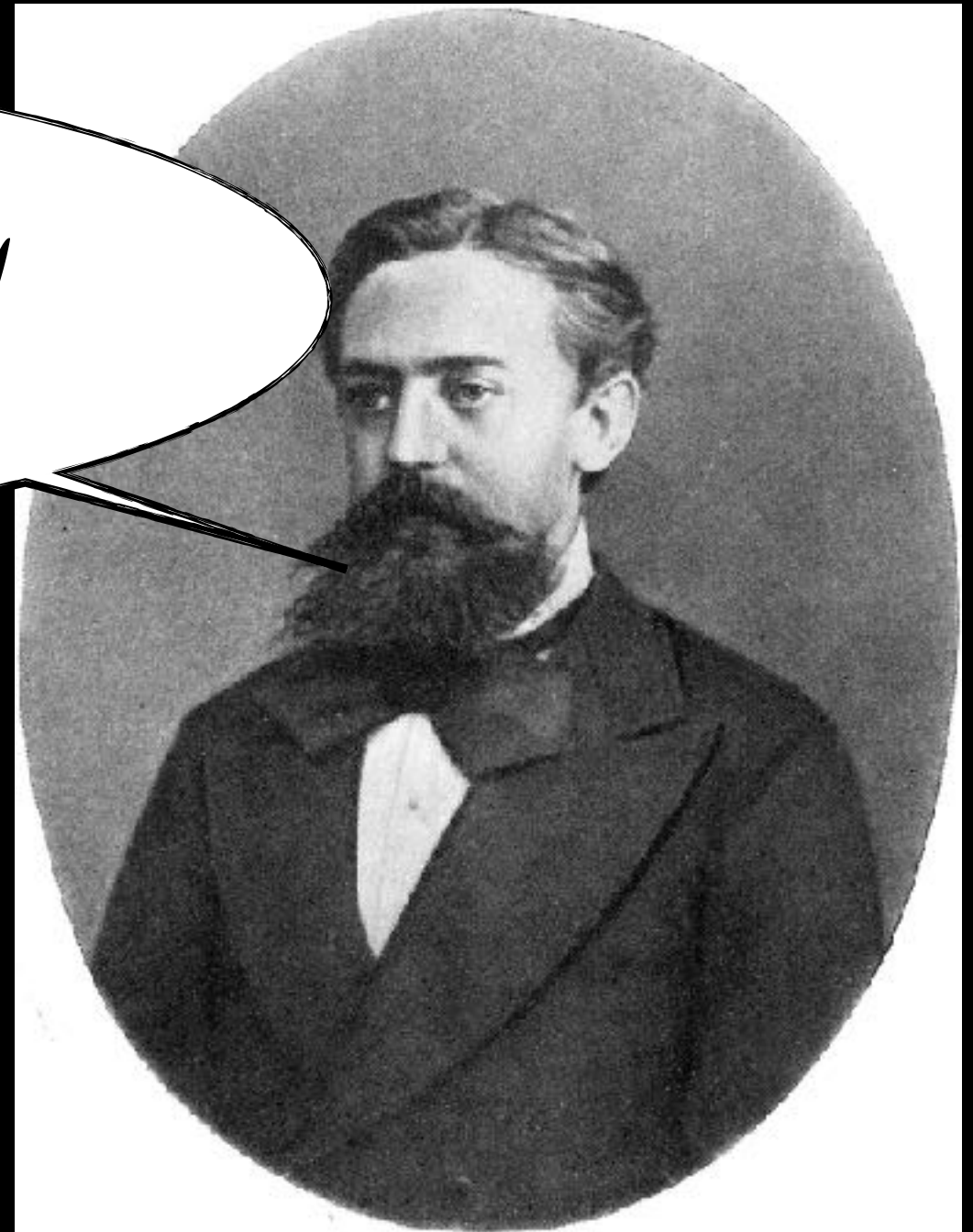
\times

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

Can We Make it Simpler?

BREAK IT DOWN!

$P(S) = P(w_1, w_2, \dots, w_n)$
*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:

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

Markov Models:

UNIGRAM MODEL ($K=0$)

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

BIGRAM MODEL ($K=1$)

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

TRIGRAM MODEL ($K=2$)

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

A Trigram Model

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

$$\begin{aligned} P(S) = P(w_1, \dots, w_n) = & P(\text{The} | * *) \\ & \times P(\text{weather} | * \text{The}) \\ & \times P(\text{today} | \text{The weather}) \\ \text{CHAIN RULE} & \times P(\text{is} | \text{weather today}) \\ & \times P(\text{fine} | \text{today is}) \\ & \times P(\text{STOP} | \text{is fine}) \end{aligned}$$

Where Probabilities Come From

The weather today is ...

*WE NEED A WAY TO ASSIGN
 $P(\text{WORD} \mid \text{"THE WEATHER TODAY IS"})$*

Count in 57m Tweets

MAXIMUM LIKELIHOOD ESTIMATION

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

(MLE)

Conditional Probabilities

y	x	$P(x y)^*$
tea with	milk	0,42
	sugar	0,35
	a	0,18
	stevia	0,05
for the	win	0,25
	majority	0,21
	birds	0,15

SUMS TO 1.0

**TOTALLY MADE UP NUMBERS*

Marginalizing

$$P(w | u, v) = \frac{c(u, v, w)}{c(u, v)}$$

TRIGRAM MLE

MARGINALIZE OUT:

$$P(w | u, v) = \frac{c(u, v, w)}{c(u, v)} = \frac{c(u, v, w)}{\sum_z^N c(u, v, z)}$$

ALL TRIGRAMS

STARTING WITH U, V

Smoothing

Many Counts are 0

* * The weather today is fine STOP

$$\begin{aligned} P(S) = P(w_1, \dots, w_n) &= P(\text{The} | * *) \\ &\times P(\text{weather} | * \text{The}) \\ &\times P(\text{today} | \text{The weather}) \\ &\times P(\text{is} | \text{weather today}) \\ c(\text{today is fine}) &= 0 \\ &\times P(\text{fine} | \text{today is}) \\ &\times P(\text{STOP} | \text{is fine}) \end{aligned}$$

Add-one (Laplace) smoothing

JUST PRETEND
YOU'VE SEEN IT!

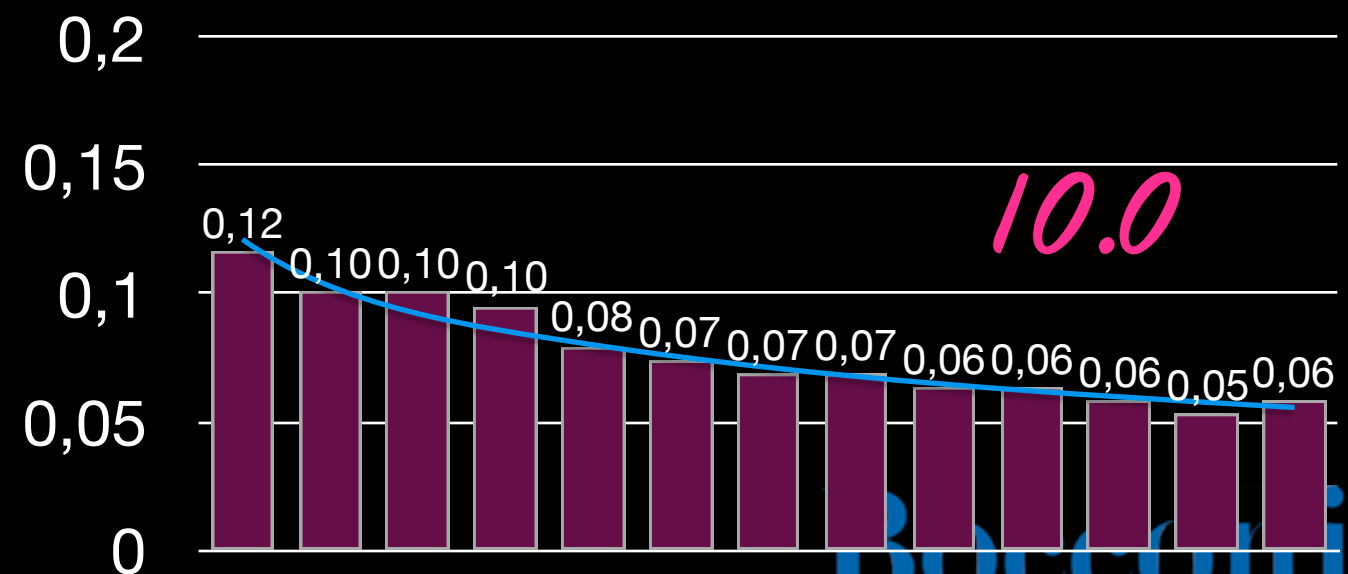
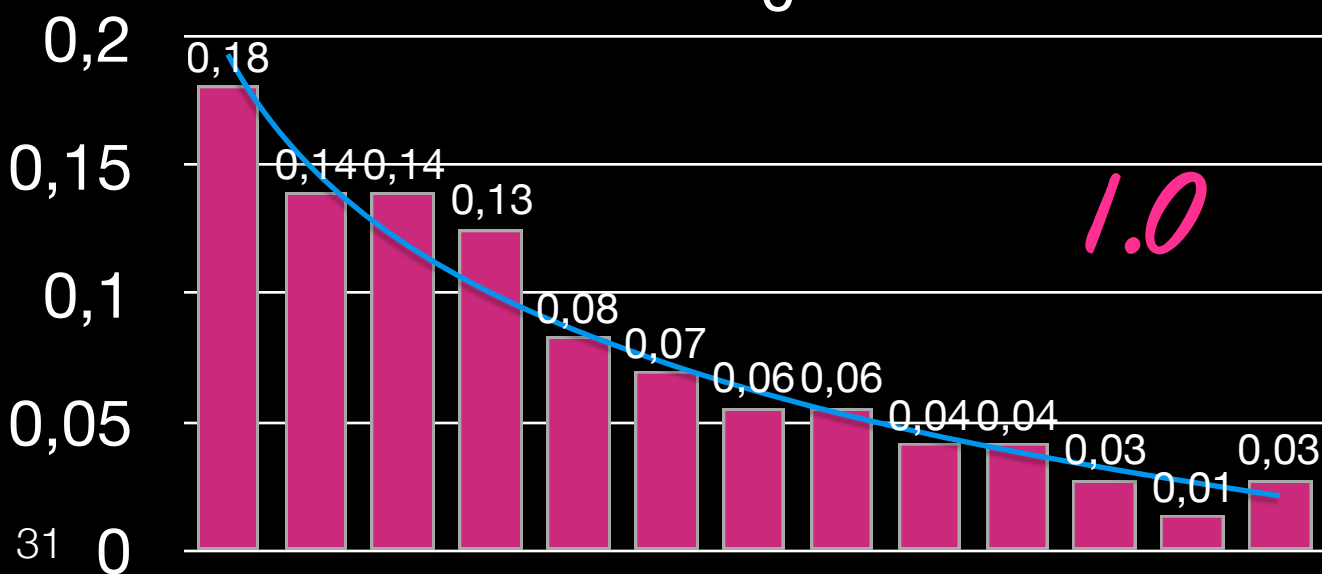
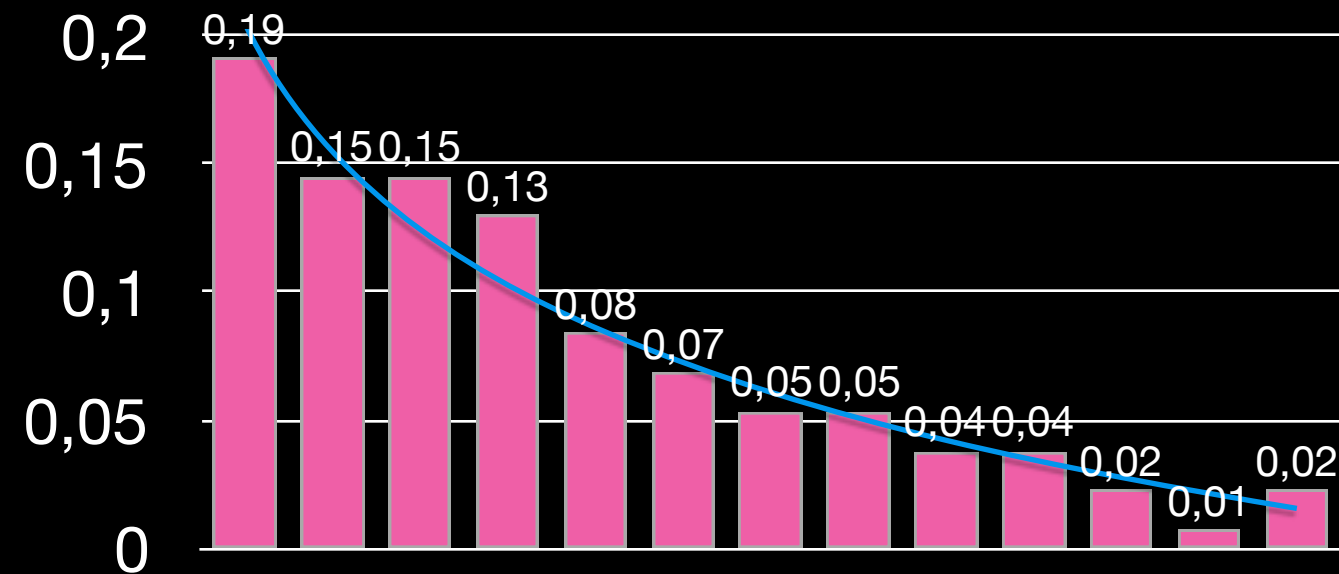
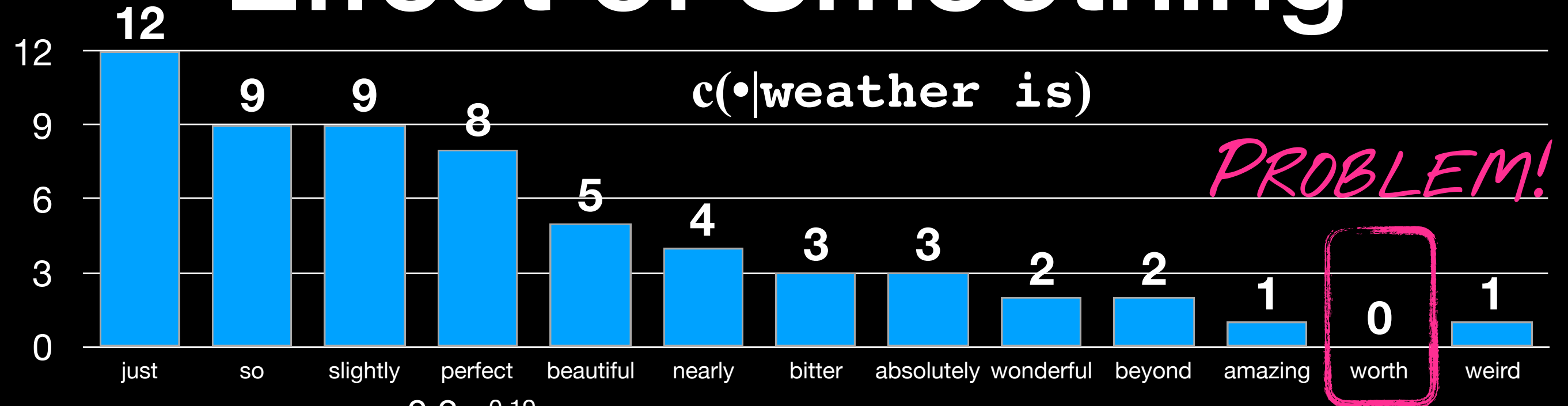
$$P(w) = \frac{c(w) + \alpha \text{ SMOOTHING FACTOR}}{N + \alpha \text{ \#WORD TYPES IN VOCAB}} |V|$$



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

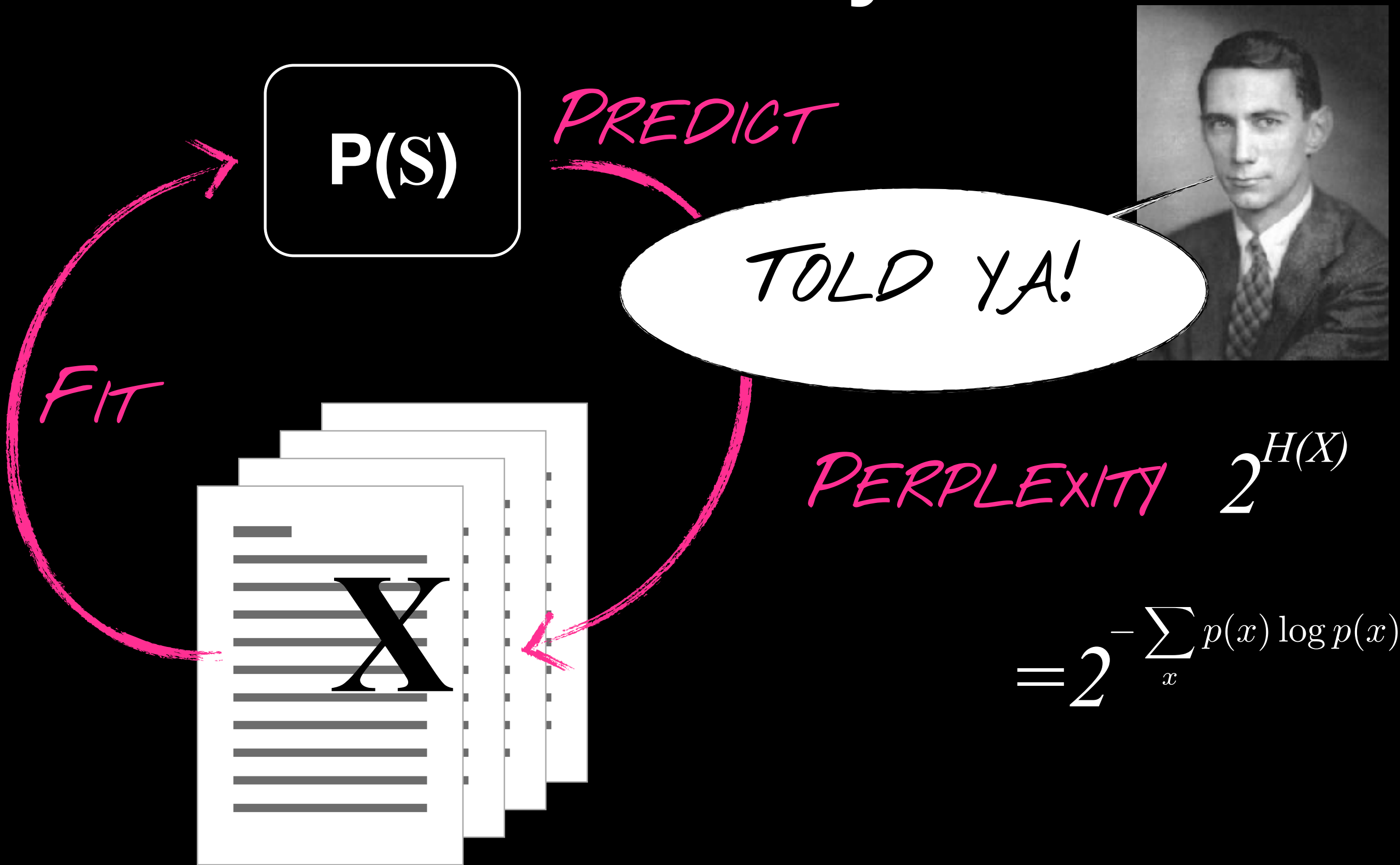
#WORD
TYPES IN VOCAB

Effect of Smoothing



Evaluating LMs

How Good is My Model?



Using LMs for Generation

From Last Year's Projects

TRAIN LMS ON PRESIDENTIAL CANDIDATE SPEECHES.

LET THE MODEL FINISH A SENTENCE STARTING WITH "AMERICA"

CANDIDATE 1: "America first — America first ..."

CANDIDATE 2: "America was actually on track to top \$ 1 trillion in spending over the coming decade -- because the freedom and dignity --"

Take a Random Walk

PROPORTIONATELY



Pick a random word w from $P(\cdot \mid * *)$

$H = [*, *, w]$

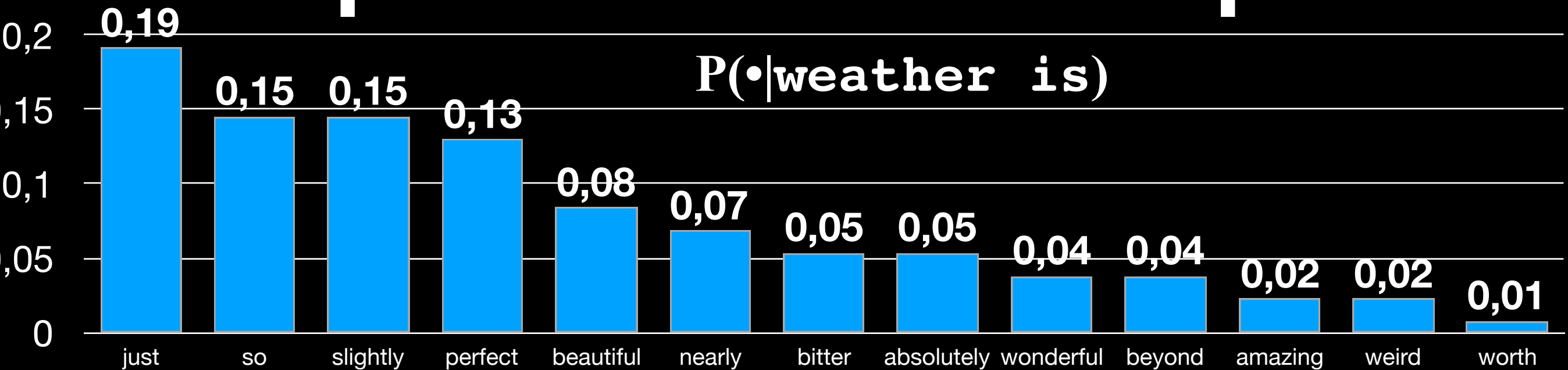
While $H[-1]$ is not STOP:

 Pick a random word w from $P(\cdot \mid H[-2:])$

$H += [w]$

return H

Proportionate Samples



20

15

10

5

0

1 SAMPLE

20

15

10

5

0

10 SAMPLES

20

15

10

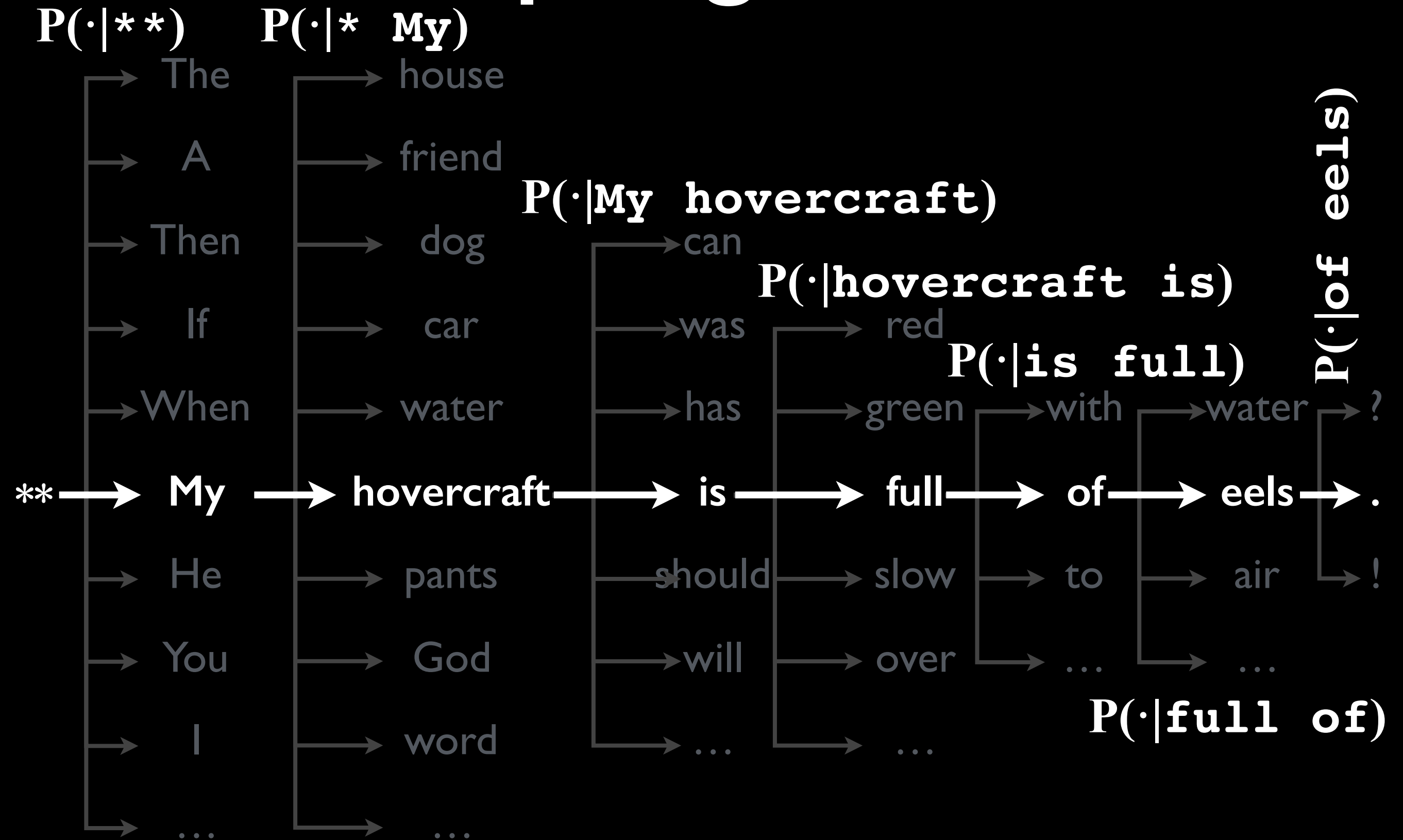
5

0

100 SAMPLES

37

Sampling Words



Wrapping Up

Language Models in Short

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

Take-Home Points

- **Language Models** assign a probability to any sentence
- The **Markov assumption** breaks sentence probability into a **chain** of word conditional probabilities
- **Markov order** determines the size of the conditional n -grams
- n -gram probabilities can be computed with **Maximum Likelihood Estimation** from a large corpus
- **Smoothing** helps address the problem of unseen words
- The same LM parameters can be used for **text generation**