N-Gram Language Models Part 2

LIN 313 Language and Computers
UT Austin Fall 2025

Admin

- HW 1 Due Friday
  - two more office hours left
- Monday - "The Chatbot's True Self", Courtney Handman (under review at the *Journal of Linguistic Anthropology*)
  - Group annotation

Overview

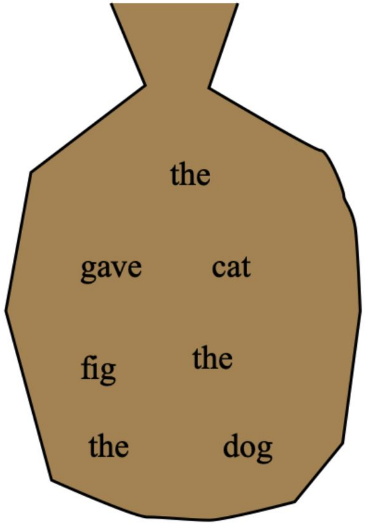
- Unigram review
- The birth of conditional probability
- Bigram Models, Trigram Models, and so on.

Friday

- model evaluation
- perplexity
- text mashups

Unigram model review: the bag of Words

Imagine you write "The cat gave the dog the fig" on a piece of paper, cut the words up and put them in a bag.

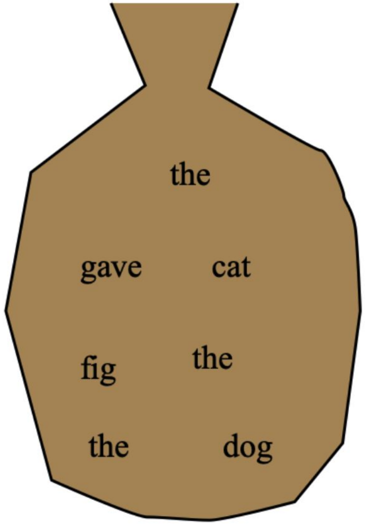


Which string is our unigram model more likely to generate?

- "the the the dog"
- "the cat the dog"

Unigram model review: the bag of Words

Imagine you write "The cat gave the dog the fig" on a piece of paper, cut the words up and put them in a bag.

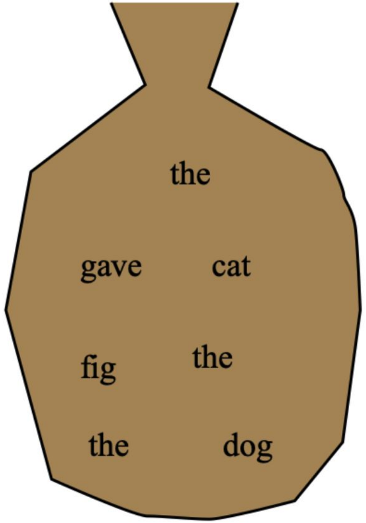


Which string is our unigram model more likely to generate?

- "the the the dog"
- "the cat the dog"

Unigram model review: the bag of Words

Imagine you write "The cat gave the dog the fig" on a piece of paper, cut the words up and put them in a bag.

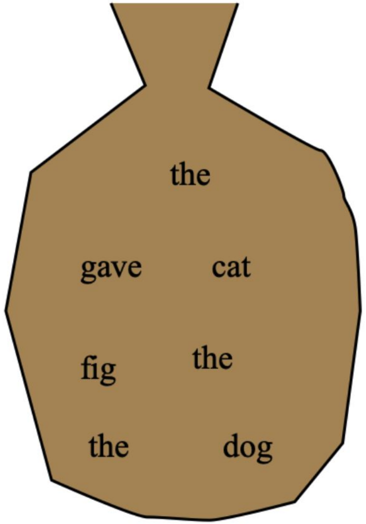


Which string is our unigram model more likely to generate?

- "the the the dog"
  - P ("the the the dog") = P("the") x P("the") x P("the") x P("dog")
  - P ("the the the dog") = 3/7 x 3/7 x 3/7 x 1/7 = ?
- "the cat the dog"
  - P ("the the the dog") = P("the") x P("the") x P("the") x P("dog")
  - P ("the cat the dog") = 3/7 x 1/7 x 3/7 x 1/7 = ?

Unigram model review: the bag of Words

Imagine you write "The cat gave the dog the fig" on a piece of paper, cut the words up and put them in a bag.

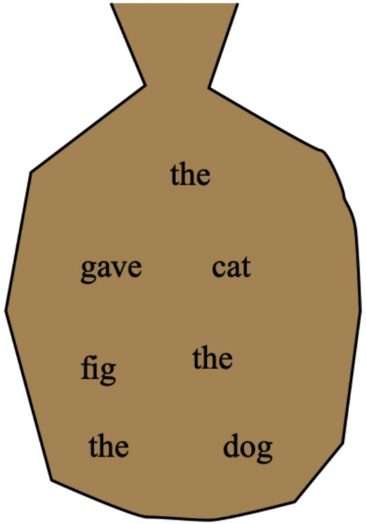


Which string is our unigram model more likely to generate?

- "the the the dog"
  - P ("the the the dog") = P("the") x P("the") x P("the") x P("dog")
  - P ("the the the dog") = 3/7 x 3/7 x 3/7 x 1/7 = $3^3 / 7^4$
- "the cat the dog"
  - P ("the the the dog") = P("the") x P("the") x P("the") x P("dog")
  - P ("the cat the dog") = 3/7 x 1/7 x 3/7 x 1/7 = $3^2 / 7^4$

Unigram model review: the bag of Words

Imagine you write "The cat gave the dog the fig" on a piece of paper, cut the words up and put them in a bag.



The unigram model assumes that words are **independent events.**

Today we are going to revisit that assumption

Pavel Nekrasov (1853–1924)

- Russian mathematician
- Tsarist and devout orthodox Christian
- In 1902 he claims to prove free will
  - Data gathered by social scientists, such as crime statistics and birth rates, conform to the law of large numbers. **The law of large numbers applies only to such independent events.** Therefore the underlying acts of individuals must be independent and voluntary.
  - That is, voluntary acts—expressions of free will—are like the independent events of probability theory, with no causal links between them.

In case you're keeping a tally

of how many mathematicians in our story sought to explain history, culture, or consciousness with math

- George Boole
  - "No general method for the solution of questions in the theory of probabilities can be established which does not explicitly recognise, not only the special numerical bases of the science, but also **those universal laws of thought which are the basis of all reasoning, and which, whatever they may be as to their essence, are at least mathematical as to their form**"
- George Zipf
  - Originally intended the inverse power law for linguistics. Later observed that the rank vs. frequency distribution law at work in individual incomes in a unified nation. His 1941 book *National Unity and Disunity,* **theorized that breaks in this "normal curve of income distribution" portend social pressure for change or revolution**.
- Pavel Nekrasov
  - Argued for the existence of free will on the basis of the Law of Large Numbers

Andrei Markov (1856 - 1922)

- Soviet Russian mathematician
- Absolutely hated Nekrasov
- Set out to prove that the law of large numbers applied to **dependent events** too
- In the process gives us the whole idea of **conditional probability**

Markov's *Eugene Onegin* Demonstration (1912)

He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*
- counted up how many consonants C and how many vowels V
  - P (V) = 0.389
  - P (C) = 0.611
- If C and V are **independent**, we should be able to easily predict the probability that one occurs after the other **(the transition probabilities)**:
  - P (C | V) = ?
  - P (C | C) = ?
  - P (V | V) = ?
  - P (V | C) = ?

Markov's *Eugene Onegin* Demonstration (1906)

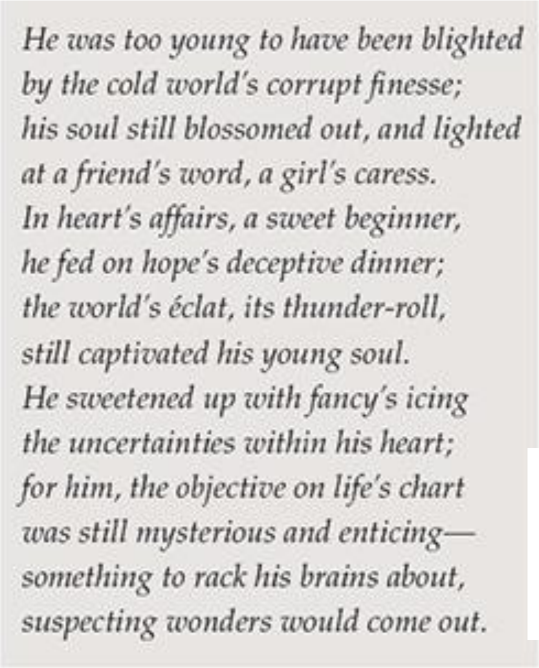
- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*

He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*


He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

If the C and V are independent, what are the **transition probabilities**?

	V	C
after V		
after C		

https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*

He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

If the C and V are independent, what are the **transition probabilities**?

	V	C
after V	0.389	0.611
after C	0.389	0.611

https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*

He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

If the C and V are independent, what are the **transition probabilities**?

	V	C
after V	0.389	0.611
after C	0.389	0.611

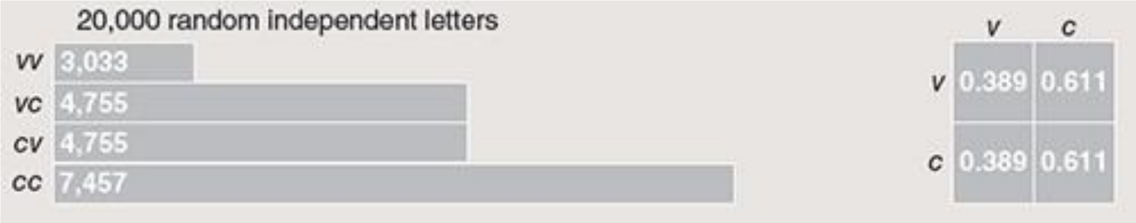
transition probability **matrix**

https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*

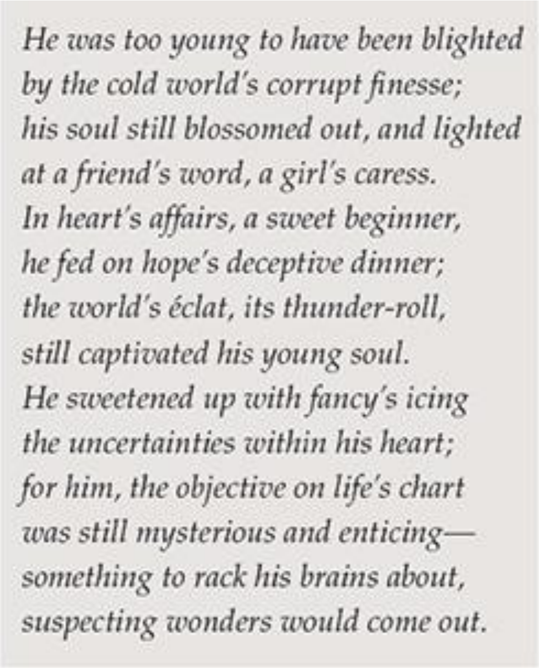
He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.



https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*



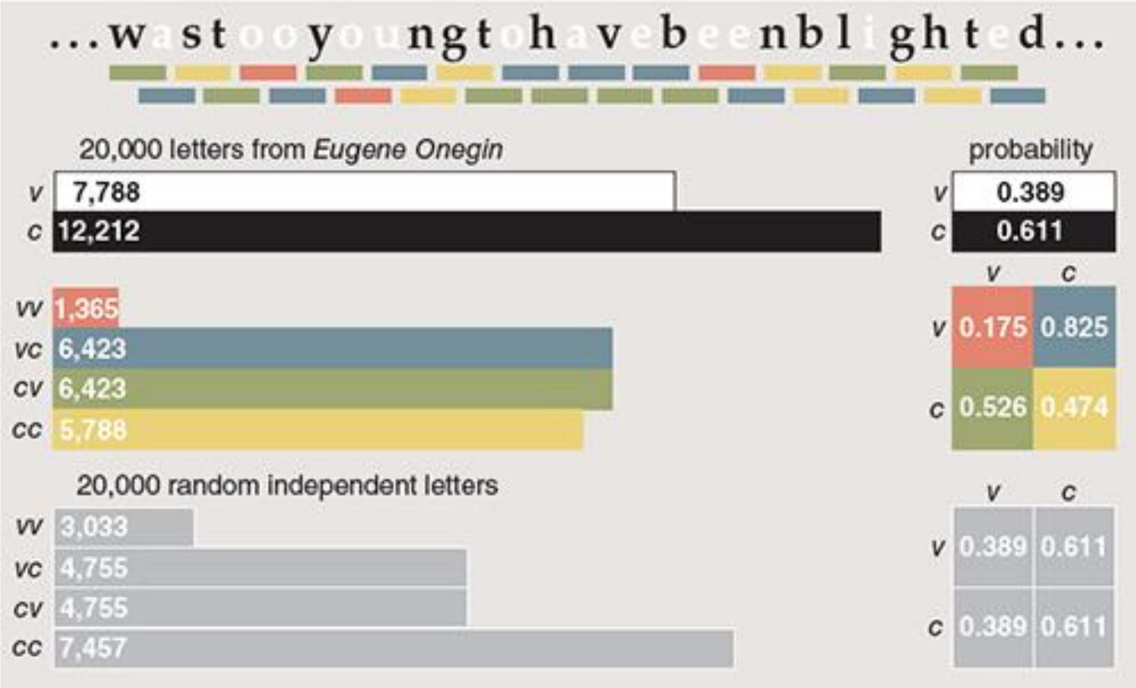
He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

https://www.americanscienti g/article/first-link the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*

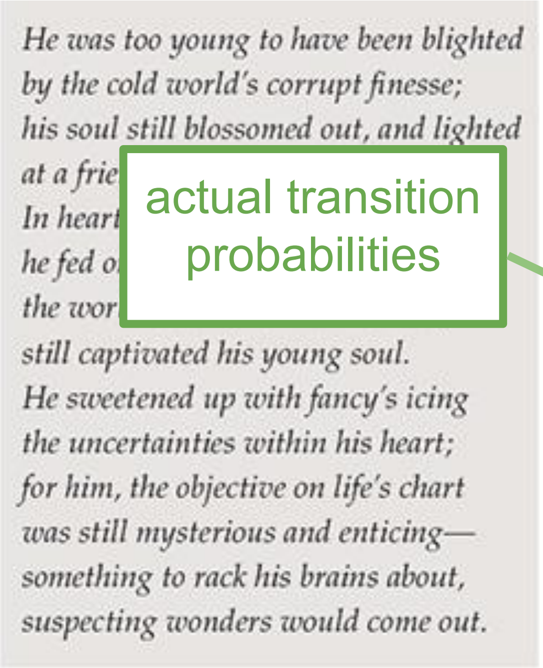
He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.



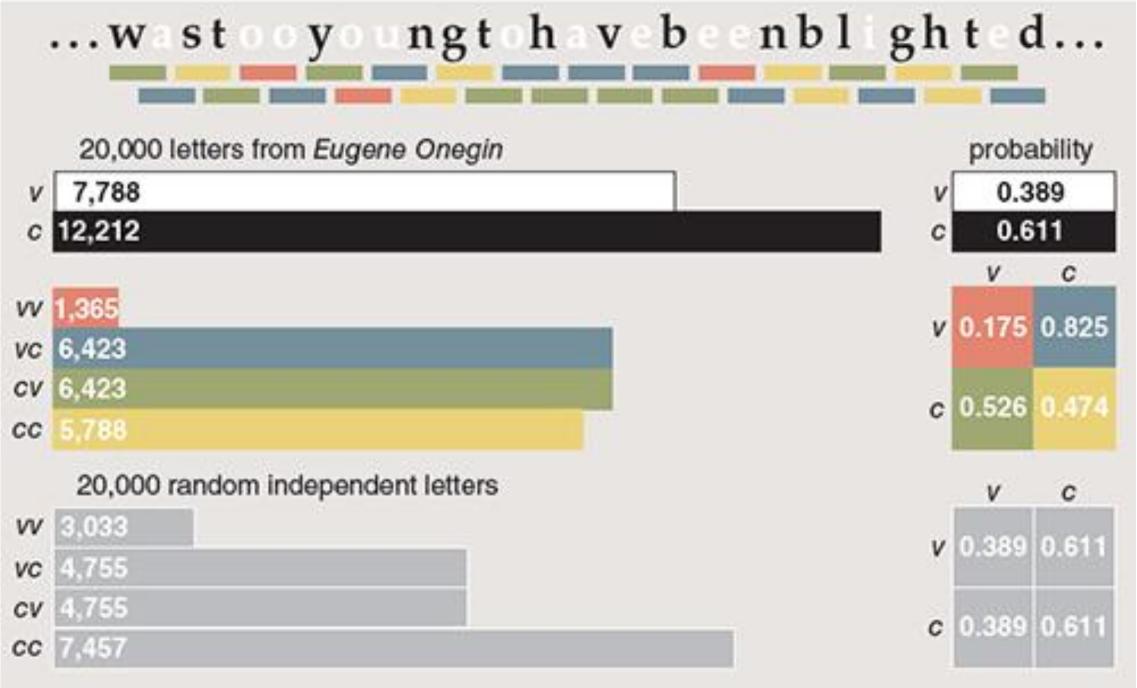
https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov's *Eugene Onegin* Demonstration (1906)

- First 20,000 letters of Alexander Pushkin's poem *Eugene Onegin*


He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a frie
In heart
he fed o
the wor
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.

actual transition probabilities



https://www.americanscientist.org/article/first-links-in-the-markov-chain

What does this have to do with free will?

- Markov chopped the letters of *Eugene Onegin* into blocks of 36 letter (shown to the right) and treated them as **trials**
- The observed distribution of bigram probabilities might be skewed inside any given block
- But as the system evolves over time, the frequency of each **state** converges to a fixed average value.
- 2 observations
  - The event (C or V) is dependent on the outcome of the previous event (C or V)
  - The law of large numbers applies to the event (C or V)
- QED, the law of large numbers applies to dependent events as well!

6	8	11	11	13	49	16	11	9	8	7	51	14	12	7	3	6	42	5	11	10	6	10	42	10	6	6	6	7	35
12	11	7	7	5	42	4	8	9	11	10	42	5	5	11	9	11	41	12	8	8	11	7	46	9	12	15	6	9	51
6	6	6	7	13	38	9	9	9	7	10	44	8	10	6	10	7	41	7	7	12	10	9	45	9	3	6	10	9	37
8	10	11	9	4	42	12	9	6	10	7	44	11	11	8	3	10	43	8	12	7	9	9	45	9	11	8	5	6	39
10	11	5	10	8	44	3	8	10	8	9	38	4	4	11	14	8	41	12	8	10	9	8	47	9	10	10	10	9	48
42	46	40	44	43	15	44	45	43	44	43	19	42	42	43	39	42	8	44	46	47	45	43	25	46	42	45	37	40	10
8	7	8	7	10	40	11	11	8	7	7	44	11	10	10	12	6	49	12	9	8	10	10	49	8	9	9	5	8	39
10	9	9	8	8	44	9	6	10	11	11	47	4	4	9	7	9	33	3	10	12	9	10	44	7	9	9	11	7	43
8	9	8	8	8	41	12	9	9	5	6	41	11	13	6	9	10	49	11	11	6	11	10	49	10	6	6	9	9	40
10	6	13	6	12	47	10	8	6	11	11	46	6	7	11	8	6	38	10	8	11	6	7	42	7	8	15	6	9	45
8	12	5	13	6	44	7	6	8	9	8	38	8	6	10	7	12	43	6	8	7	9	6	36	11	7	6	11	10	45
44	43	43	42	44	16	49	40	41	43	43	16	40	40	46	43	43	12	42	46	44	45	43	20	43	39	45	42	43	12
7	7	7	7	9	37	12	7	7	6	8	40	7	4	11	5	7	34	5	5	7	5	9	31	8	6	5	14	11	44
9	13	6	8	4	40	6	8	7	10	8	39	11	14	9	11	9	54	12	6	10	10	8	46	8	12	10	7	4	41
9	7	11	12	14	53	9	10	10	8	7	44	7	6	9	8	9	39	8	14	11	11	10	54	8	10	9	8	14	49
7	11	8	9	7	42	9	5	6	7	7	34	10	9	8	10	5	42	4	3	9	5	9	30	9	5	9	9	6	38
8	10	10	11	9	48	7	11	9	13	7	47	11	10	8	9	11	49	13	14	9	11	7	54	8	13	11	5	10	47
40	48	42	47	43	20	43	41	39	44	37	4	46	43	45	43	41	18	42	42	46	42	43	15	41	46	44	43	45	19
10	9	13	6	12	50	4	11	10	12	5	42	5	11	10	6	5	37	4	4	10	11	5	34	13	11	13	10	10	57
8	8	8	9	5	38	14	9	8	7	14	52	8	9	8	10	10	45	6	12	9	8	10	45	7	10	9	6	2	34
10	10	8	9	10	47	4	8	9	8	4	33	8	8	6	9	9	40	13	4	10	8	6	41	8	8	7	8	12	43
7	9	10	7	10	43	8	14	11	12	6	51	10	6	9	7	6	38	7	10	7	12	11	47	9	11	9	10	6	45
9	8	3	11	7	38	11	6	7	4	14	42	11	9	8	10	12	50	9	13	8	1	8	39	6	3	7	9	9	34
44	44	42	42	44	16	41	48	45	43	43	20	42	43	41	42	42	10	39	43	44	40	40	6	43	43	45	43	39	13
11	6	8	9	5	39	10	10	4	7	9	40	10	8	7	8	8	41	10	3	11	13	5	42	8	8	13	5	8	42
6	10	6	8	13	43	11	10	13	13	9	56	6	9	9	8	7	39	7	11	9	7	10	44	9	10	7	14	9	49
10	5	11	11	6	43	10	7	5	9	6	37	15	9	11	13	9	57	10	10	4	7	7	38	9	11	6	8	7	41
9	12	6	8	10	45	10	5	8	10	10	43	5	10	5	4	7	31	7	7	14	13	7	48	7	9	12	6	9	43
7	11	9	10	10	47	6	13	10	5	6	40	8	9	10	12	9	48	11	9	9	6	15	50	10	9	9	12	9	49
43	44	40	46	44	17	47	45	40	44	40	16	44	45	42	45	40	16	45	40	47	46	44	22	43	47	47	45	42	24
12	7	12	5	12	48	10	14	7	6	6	43	9	6	7	10	5	37	12	13	5	9	11	50	5	11	8	12	10	46
10	8	5	13	4	40	4	6	8	10	14	42	11	10	7	8	9	45	7	7	10	5	8	37	12	8	9	8	6	43
10	13	8	7	9	47	13	6	12	8	5	44	10	10	9	9	10	48	7	7	9	14	7	44	8	11	9	8	7	43
9	4	12	6	9	40	7	13	5	8	10	43	8	6	12	10	10	46	12	13	7	8	10	50	8	5	7	11	8	39
4	12	9	9	8	42	8	5	15	10	9	47	9	11	8	5	11	44	4	4	12	11	9	40	11	11	10	6	8	46
45	44	46	40	42	17	42	44	47	42	44	19	47	43	43	42	45	20	42	44	43	47	45	21	44	46	43	45	39	17
9	11	10	6	13	49	5	9	7	10	6	37	8	6	8	7	14	43	7	9	8	6	7	37	9	11	11	8	8	47
9	8	6	8	6	37	10	9	11	7	7	44	8	14	13	8	4	47	9	8	6	10	11	44	10	8	5	9	10	42
7	7	12	10	9	45	11	11	11	10	8	51	12	4	6	9	11	42	10	9	10	8	10	47	6	8	16	12	11	53
12	12	6	8	8	46	7	7	5	10	10	39	6	8	9	10	8	41	8	7	4	9	4	32	12	11	5	7	8	43
5	7	9	11	4	36	13	8	9	8	10	48	6	8	11	8	6	39	11	8	10	8	9	46	6	5	9	10	8	38
42	45	43	43	40	13	46	44	43	45	41	19	40	40	47	42	43	12	45	41	38	41	41	6	43	43	46	46	45	23
5	7	4	3	7	26	4	7	9	11	10	41	10	8	7	8	7	40	12	10	11	4	5	42	12	13	6	6	10	47
14	10	13	9	5	51	10	7	9	4	9	39	10	8	11	10	7	46	5	9	10	11	11	46	6	3	10	10	4	33
7	8	6	8	9	38	8	13	9	12	10	52	6	11	11	10	10	48	10	8	10	7	13	48	11	11	9	7	14	52
7	10	9	5	9	40	7	5	7	7	12	38	12	8	7	6	5	38	11	8	8	11	5	43	5	8	8	9	9	39
9	10	11	16	7	53	13	10	10	9	5	47	5	9	11	12	11	48	4	8	8	9	11	40	11	6	11	12	7	47
42	45	43	41	37	8	42	42	44	43	46	17	43	44	47	46	40	20	42	43	47	42	45	19	45	41	44	44	44	18

Markov Model

In probability theory, a Markov model is a *stochastic* model used to model pseudo-randomly changing systems.

- They allow us to predict future states based on observations of the past.
- The **Markov assumption**: It is assumed that future states depend only on the current state (a slice of the recent past), not on the events that occurred before it (that is, it assumes the Markov property).
- Generally, this assumption enables reasoning and computation with the model that would otherwise be intractable.
  - it's easier to simulate the problem and record an average of the results
  - the Law of Large Numbers means that if you run enough simulations you'll know your outcome.

Monte Carlo Method

Markov Models are the basis of the Monte Carlo Method used to study complex dynamical (time-based) systems.

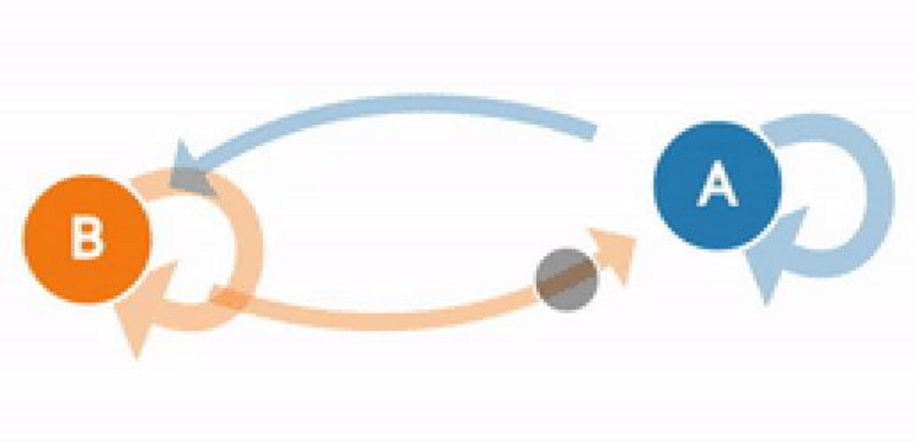
- weather patterns
- emotions
- microbiology
- google's ranking algorithm

Instrumental in building the hydrogen bombs that destroyed Hiroshima and Nagasaki in WWII.

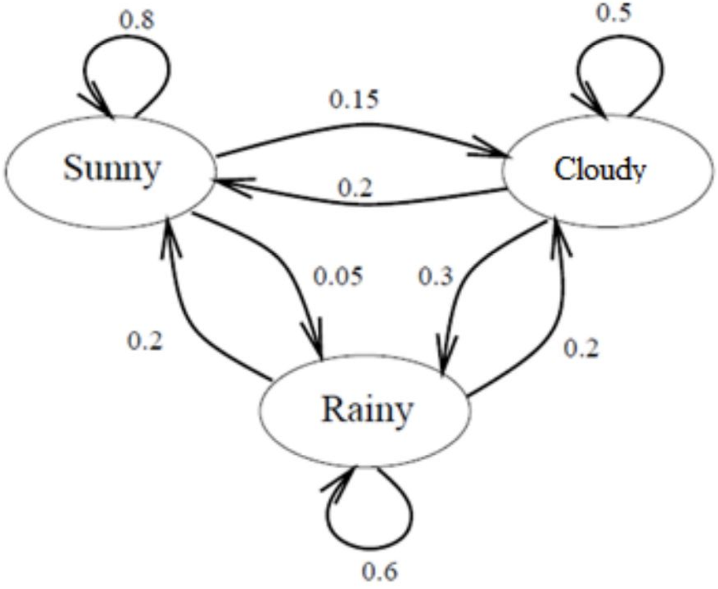
Hidden Markov Model

Defined by three things

- **States**
- Transition probabilities (often represented as a **matrix** [a.k.a. a table])
- A probability distribution over **start states**



https://www.blopig.com/blog/2021/09/chained-or-unchained-markov-nekrasov-and-free-will/



$$\left.\begin{aligned} P(Sunny|Sunny) &= 0.8 \\ P(Rainy|Sunny) &= 0.05 \\ P(Cloudy|Sunny) &= 0.15 \end{aligned}\right\} 1$$

$$\left.\begin{aligned} P(Sunny|Rainy) &= 0.2 \\ P(Rainy|Rainy) &= 0.6 \\ P(Cloudyy|Rainy) &= 0.2 \end{aligned}\right\} 1$$

$$\left.\begin{aligned} P(Sunny|Cloudy) &= 0.2 \\ P(Rainy|Cloudy) &= 0.3 \\ P(Cloudy|Cloudy) &= 0.5 \end{aligned}\right\} 1$$

LANGUAGE MODELS

- A way of representing and predicting the most likely segment to follow a sequence of words, characters, sounds, etc.

It was the best of times, it was the worst of ___________

N-GRAM LANGUAGE MODELS

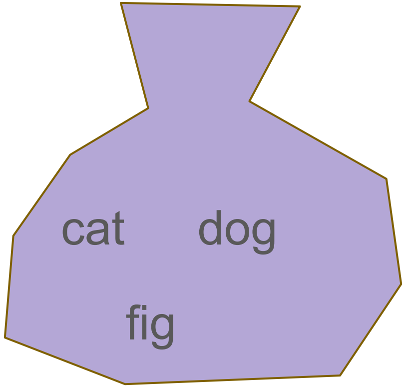
- When dealing with words, n-gram models are a way of predicting the next word based on the n-1 preceding words
- This is done by counting words and recording probabilities
  - Unigram model: $P(word_i \mid context) = P(word_i)$
  - Bigram model: $P(word_i \mid context) = P(word_i \mid word_{i-1})$
  - Trigram model: $P(word_i \mid context) = P(word_i \mid word_{i-1}, word_{i-2})$

Bigram Language Model

- generation probabilities **depend** on what state we are in.
- For a bigram model, we represent the current state as a single word.
- now, we have a bag of words for "state": each **word type** in our vocabulary

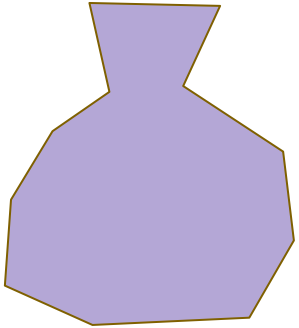
"The cat gave the dog the fig"

"the"
bag

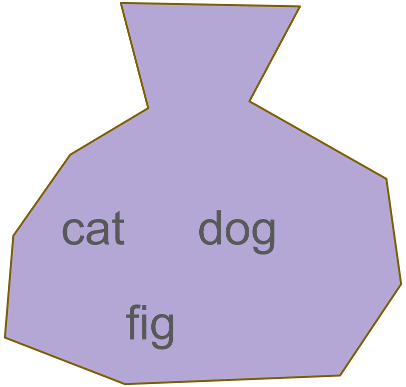


Bigram Language Model

"cat"
bag



"the"
bag



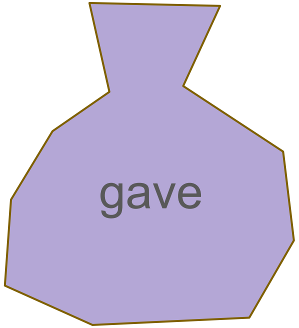
- generation probabilities **depend** on what state we are in.
- For a bigram model, we represent the current state as a single word.
- now, we have a bag of words for "state": each **word type** in our vocabulary

"The cat gave the dog the fig"

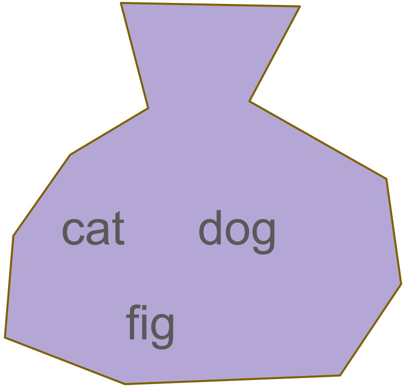
Q: What goes in the "cat" bag?

Bigram Language Model

"cat"
bag



"the"
bag



- generation probabilities **depend** on what state we are in.
- For a bigram model, we represent the current state as a single word.
- now, we have a bag of words for "state": each **word type** in our vocabulary

"The cat gave the dog the fig"

Q: What goes in the "cat" bag?

Today's Corpus

One fish two fish red fish blue fish

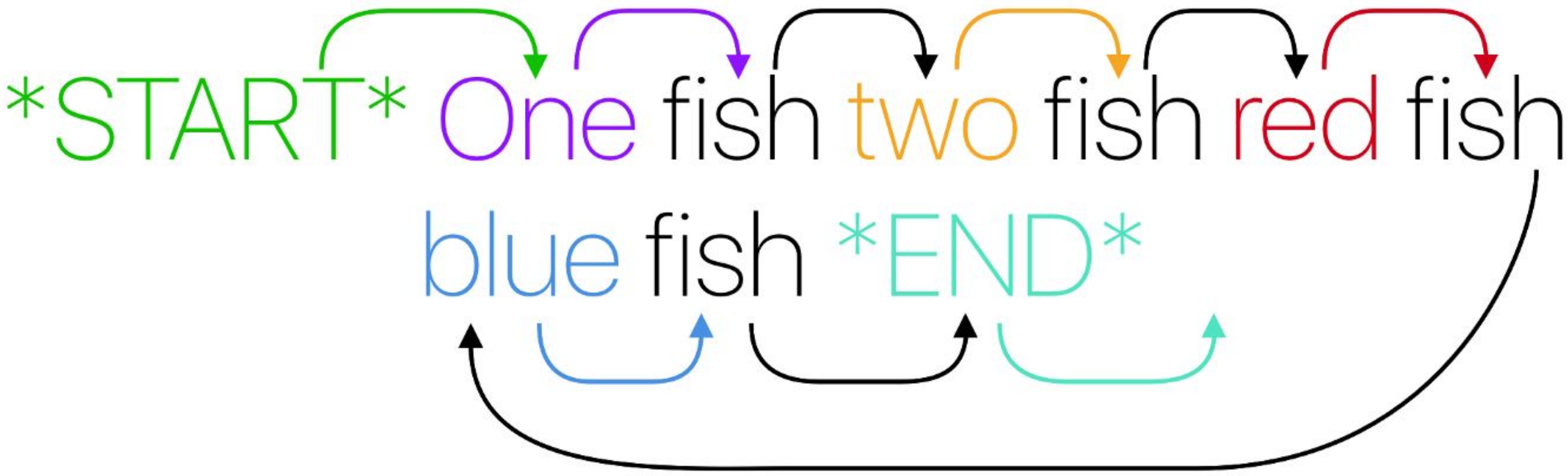
Tokenize the text

One fish two fish red fish blue fish

Add <start> and <end> tokens

START One fish two fish red fish blue fish *END*

Identify state transitions



List all (state, transition) pairs

(*Start*, One)
(One, fish)
(fish, two)
(two, fish)
(fish, red)
(red, fish)
(fish, blue)
(blue, fish)
(fish, *END*)
(*END*, none)

We have generated **state/transition pairs** which we formed by using a “window” to look at what the next token is in a pair. Pairs are in the form

(current state, next word)

We then organize these pairs by the current state. We match every state to the number of possible ways to transition out of that state.

Create a "bag" of transitions for each state

(*Start*, One)

(One, fish)

(fish, two) (fish, red) (fish, blue) (fish, *END*)

(two, fish)

(red, fish)

(blue, fish)

(*END*, none)

Create a "bag" of transitions for each state

(*Start*, One)

(One, fish)

the "fish" bag

(fish, two) (fish, red) (fish, blue) (fish, *END*)

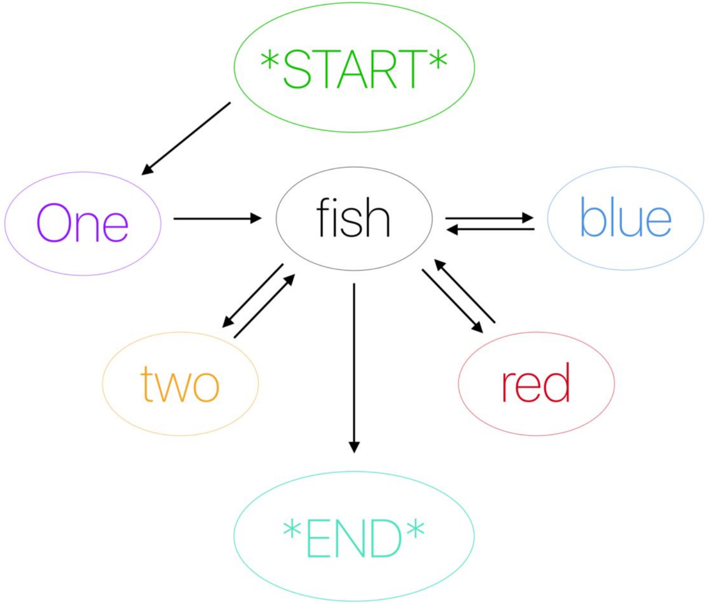
(two, fish)

(red, fish)

(blue, fish)

(*END*, none)

Draw the state transitions



One fish two fish red fish blue fish

Calculate conditional bigram probabilities

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{3.11}$$

w_{n-1} = "two"

w_n = "fish"

P("fish" | "two") = count of "two fish" / count of "two"

= ???

Calculate conditional bigram probabilities

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{3.11}$$

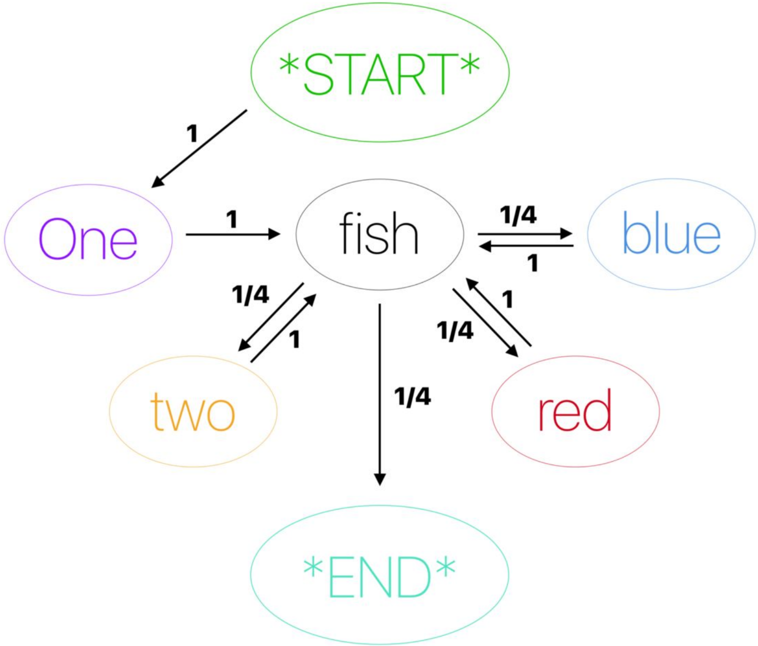
w_{n-1} = "two"

w_n = "fish"

P("fish" | "two") = count of "two fish" / count of "two"

= 1

Add transition weights



One fish two fish red fish blue fish

How to building a Bigram Model in a nutshell

1. tokenize the corpus
2. count the state-transition pairs (bigrams)
3. for each **type** of state, generate the conditional probability distribution
    a. Put all the bigram **tokens** that start with the same word type (e.g. "fish") into a bag.
    b. Label this bag with the word type ("fish"). This bag represents the possible futures if you are in the state of just having seen the word fish
    c. Count up transition pairs inside that bag
    d. **normalize** the transition pair counts: divide count of one transition by count of total transitions
    e. this yields the probability distribution for transitioning to different states out of the current state

Markov Chain Generation

We can use these n-gram language models to generate text. This process is called **Markov chain generation**, which is a phrase people prefer because it sounds cool and sci-fi.

1. Train a **Markov model** on a particular corpus
2. Pick a place to start
3. Follow the arrows.

Comparing Unigrams and Bigrams

Our old friend: "The cat gave the dog the fig"

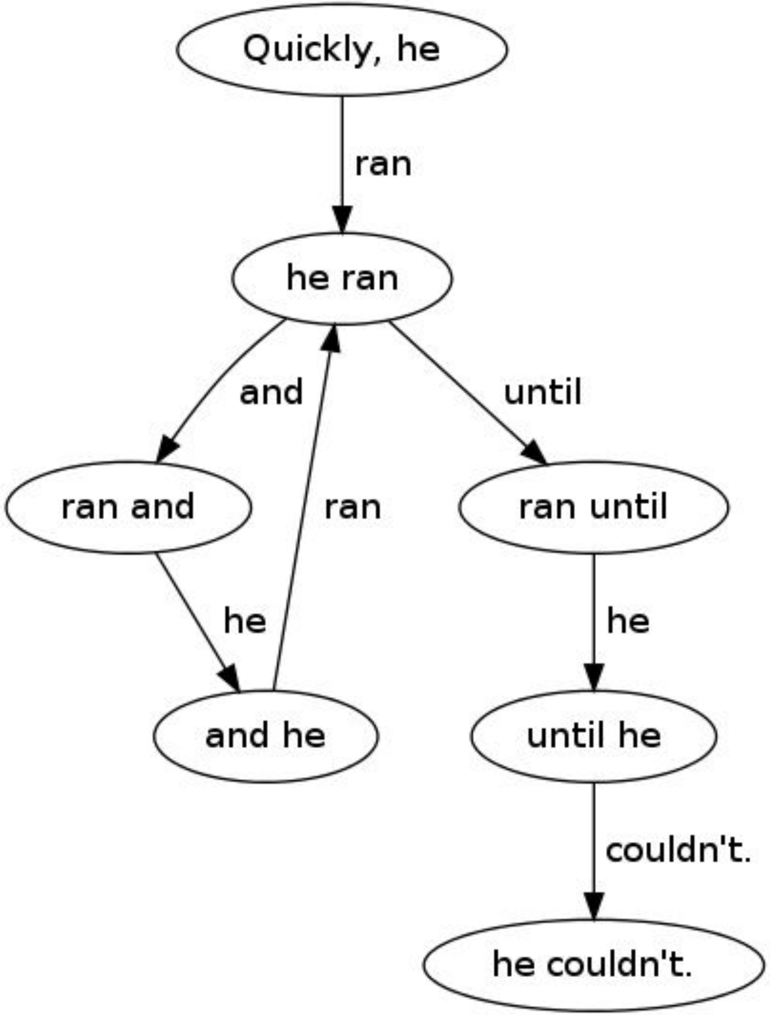
1. Use the conditional probability dice to create a bigram model for this corpus (https://gchronis.github.io/conditional-dice.html)
2. Generate some strings using this model
3. Compare with the unigram model. Which one is better? What is different about the strings it produces?

If you finish that: try making an n-gram model from a corpus of your choosing

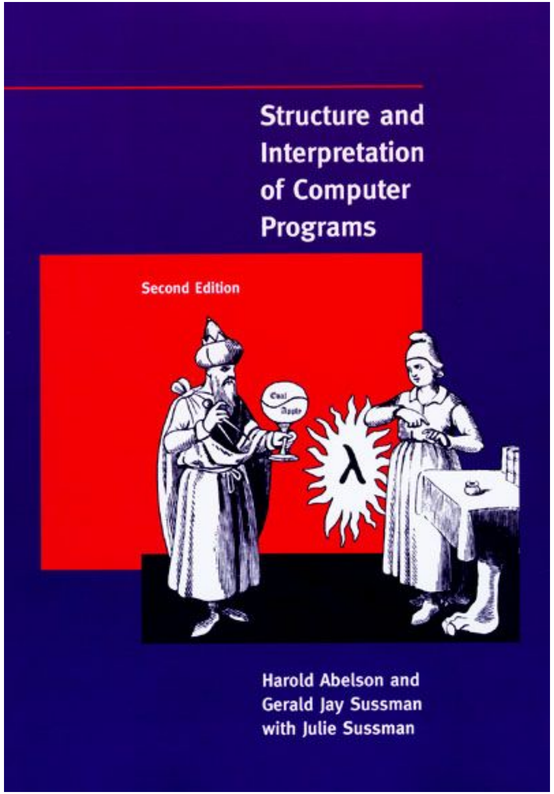
Order

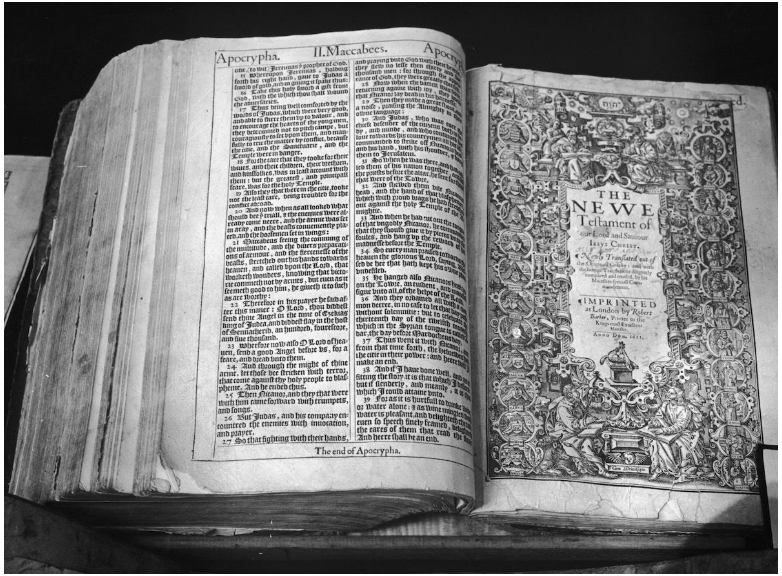
The **order** of a Markov model is the number of previous states that the model takes into account. This model is **second order**, because we care about the previous two words when determining the next word.

If we learn our transition probabilities from a corpus, this is just a bi-gram language model!



King James Programming





King James Programming

Investigate the shell's here documents and Python's triple-quote construct to find out the Almighty unto perfection

5 years ago *59 notes* *#kjv* *#bible* *#sicp* *#esr* *#markov chains*

51:11 Cast me not off in the time of execution of a sequential instruction stream

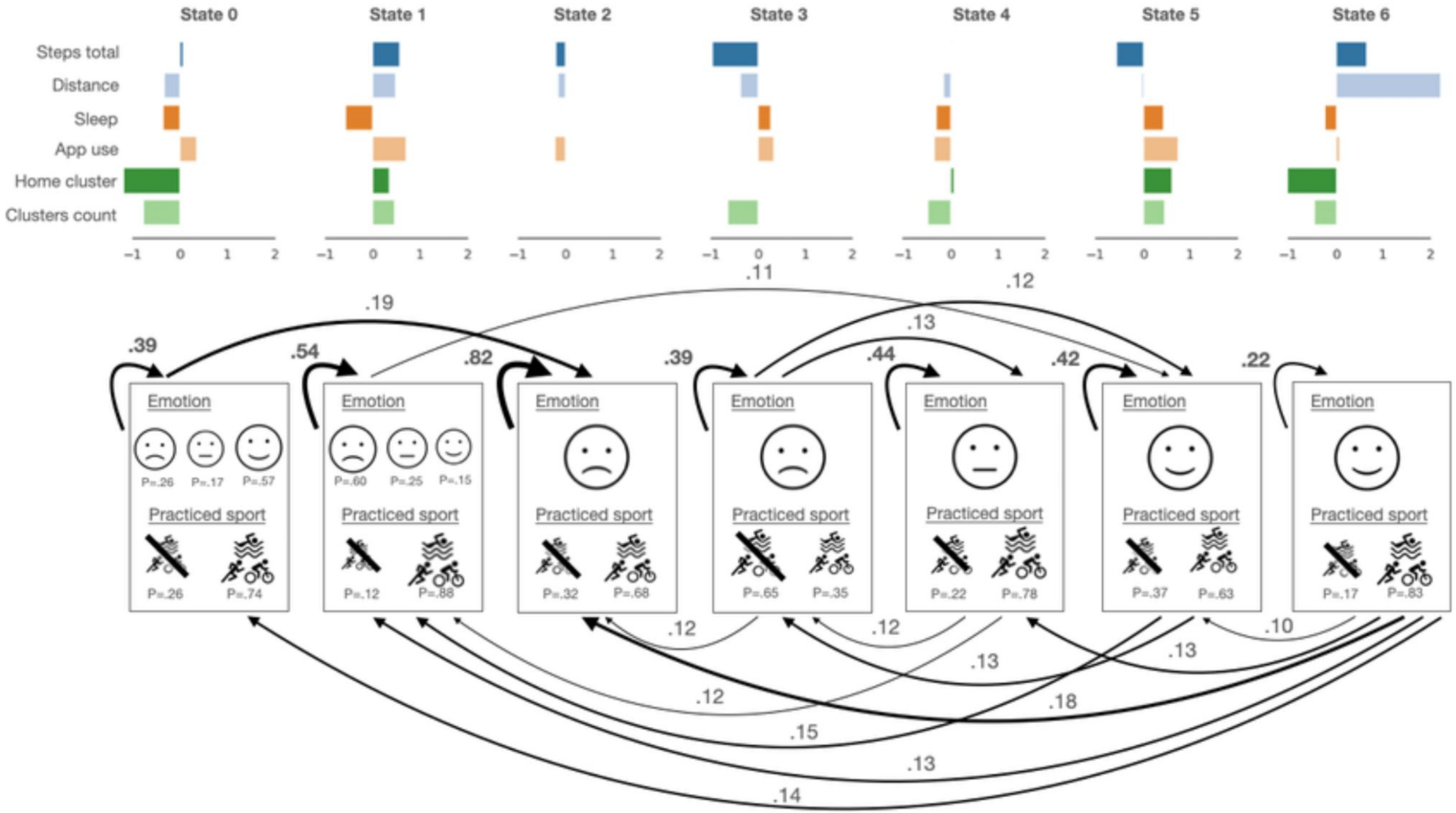
5 years ago *59 notes*

25:12 And thou shalt put into the heart of today's IBM mainframe operating systems.

5 years ago *84 notes* *#kjv* *#bible* *#sicp* *#esr* *#markov chains*

37:29 The righteous shall inherit the land, and leave it for an inheritance unto the children of Gad according to the number of steps that is linear in b.

4 years ago *87 notes* *#kjv* *#bible* *#sicp* *#poignant guide* *#markov chains*


State 0
State 1
State 2
State 3
State 4
State 5
State 6
Steps total
Distance
Sleep
App use
Home cluster
Clusters count
–1 0 1 2
Emotion
Practiced sport
P=.26 P=.17 P=.57
P=.26 P=.74
P=.60 P=.25 P=.15
P=.12 P=.88
P=.32 P=.68
P=.65 P=.35
P=.22 P=.78
P=.37 P=.63
P=.17 P=.83
.39
.54
.82
.39
.44
.42
.22
.19
.11
.12
.13
.12
.12
.10
.13
.13
.12
.18
.15
.13
.14