Data Science of Text Generation
2. Markov Chains

Ernst C. Wit, Francisco Richter

wite@usi.ch, richterf@usi.ch
Università della Svizzera italiana

Bachelor in Data Science
https://www.usi.ch/en/education/bachelor/data-science

Let's make a haiku together...

Task 1. Make a haiku by each choosing one word after hearing what previous person has chosen.

You can select from following 9 words:

- **4 nouns**: prof, kid, maths, sky
- **2 verbs:** is, flies
- **3 others:** down, weird, cool

NOTE: Haiku should have 17 words (or syllables)

Independence?

It is unlikely a sequence of words are *independent*:

$$P(W_1 \cap \ldots \cap W_{11}) \neq P(W_1) \cdots P(W_{11})$$

Independence?

It is unlikely a sequence of words are *independent*:

$$P(W_1 \cap \ldots \cap W_{11}) \neq P(W_1) \cdots P(W_{11})$$

Instead, word 2 depends on 1st, word 3 on 1st and 2nd, etc:

$$P(W_1 \cap \ldots \cap W_{11}) = P(W_1)P(W_2|W_1) \cdots P(W_{11}|W_1 \cap \ldots \cap W_{10})$$

Independence?

It is unlikely a sequence of words are *independent*:

$$P(W_1 \cap \ldots \cap W_{11}) \neq P(W_1) \cdots P(W_{11})$$

Instead, word 2 depends on 1st, word 3 on 1st and 2nd, etc:

$$P(W_1 \cap \ldots \cap W_{11}) = P(W_1)P(W_2|W_1) \cdots P(W_{11}|W_1 \cap \ldots \cap W_{10})$$

Very difficult to learn all these probabilities! Can we simplify?

Independence?

It is unlikely a sequence of words are *independent*:

$$P(W_1 \cap \ldots \cap W_{11}) \neq P(W_1) \cdots P(W_{11})$$

Instead, word 2 depends on 1st, word 3 on 1st and 2nd, etc:

$$P(W_1 \cap \ldots \cap W_{11}) = P(W_1)P(W_2|W_1) \cdots P(W_{11}|W_1 \cap \ldots \cap W_{10})$$

Very difficult to learn all these probabilities! Can we simplify?

Perhaps words just **depend on last word**:

$$P(W_1 \cap \ldots \cap W_{11}) = P(W_1)P(W_2|W_1)P(W_3|W_2) \cdots P(W_{11}|W_{10})$$

Markov Chains

Markov Chain

stochastic process in (discrete) time

$$W_0, W_1, W_2, \ldots, W_t, W_{t+1}, \ldots$$

that always only looks at last state, e.g.

$$W_t$$

to decide where to go next, i.e.,

$$W_{t+1}$$

Chess is a Markov Chain

- Future W_{t+1} depends only on present W_t.
- Past states $(W_0, W_1, \ldots, W_{t-1})$ have no added influence.

In simple terms, system "forgets" its history at each step; only present matters for predicting future.

Transition Probabilities

Let's define:

$$W_t = t\text{-th word of the haiku.}$$

which takes as value one of 9 possible words.

Transition matrix

For a Markov chain $\{W_t\}$ with 9 possible words,

- the 9×9 matrix P, where

$$P_{ij} = P(W_{t+1} = j | W_t = i)$$

is called the probability *transition matrix*.

Example

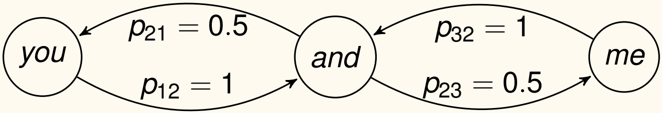


Figure: Markov Chain with 3 states and 4 transition probabilities.

For above Markov Chain, transition matrix is given as:

Example

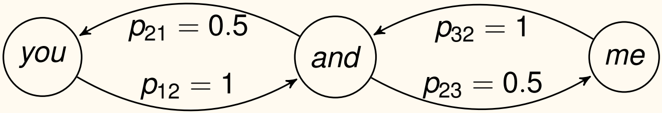


Figure: Markov Chain with 3 states and 4 transition probabilities.

For above Markov Chain, transition matrix is given as:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{bmatrix}$$

NOTE: rows of transition matrix must sum to 1.

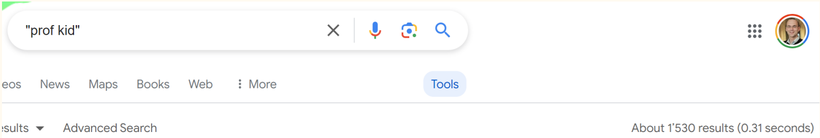
Training the Haiku Markov Chain

Data Science uses available data to train model.

We *estimate* or *train* P_{Haiku} from data:

$$\hat{P}_{ij} = \frac{\text{\# instances of word } i \text{ followed by word } j}{\text{\# instances of word } i \text{ follow by any other 8 words}}$$

By scanning through ... the internet.



We find 1,530 instances.

What follows after "prof"?

We saw: $|\text{prof} \rightarrow \text{kid}| = 1{,}530 = 1.5\text{K}$

Now, we do the same thing for all 7 other words:

prof	kid	maths	sky	is	flies	down	weird	cool
count	1.5							

What follows after "prof"?

We saw: $|\text{prof} \rightarrow \text{kid}| = 1{,}530 = 1.5\text{K}$

Now, we do the same thing for all 7 other words:

prof	kid	maths	sky	is	flies	down	weird	cool
count	1.5	170						

What follows after "prof"?

We saw: $|\text{prof} \to \text{kid}|$ = 1,530 = 1.5K

Now, we do the same thing for all 7 other words:

prof	kid	maths	sky	is	flies	down	weird	cool
count	1.5	170	5.4	5.5	0.3	3.4	0.9	17.4

Total count = 204,400 documents

$$P_{\text{prof}\to\text{kid}} =$$

What follows after "prof"?

We saw: $|\text{prof} \rightarrow \text{kid}|$ = 1,530 = 1.5K

Now, we do the same thing for all 7 other words:

prof	kid	maths	sky	is	flies	down	weird	cool
count	1.5	170	5.4	5.5	0.3	3.4	0.9	17.4

Total count = 204,400 documents

$$P_{\text{prof}\rightarrow\text{kid}} = \frac{1.5}{204.4} = 0.007$$

prof	kid	maths	sky	is	flies	down	weird	cool
P_{ij}	0.01							

What follows after "prof"?

We saw: $|\text{prof} \rightarrow \text{kid}|$ = 1,530 = 1.5K

Now, we do the same thing for all 7 other words:

prof	kid	maths	sky	is	flies	down	weird	cool
count	1.5	170	5.4	5.5	0.3	3.4	0.9	17.4

Total count = 204,400 documents

$$P_{\text{prof}\rightarrow\text{kid}} = \frac{1.5}{204.4} = 0.007$$

prof	kid	maths	sky	is	flies	down	weird	cool
P_{ij}	0.01	0.83	0.03	0.03	0.00	0.02	0.00	0.09

Now we can do the same thing for other ?→? transitions.

Count Matrix

$$C = \begin{bmatrix} 0 & 2 & 170 & 5 & 6 & 0 & 3 & 1 & 17 \\ 2 & 0 & 6 & 98 & 22 & 306 & 192 & 14 & 145 \\ 117 & 37 & 0 & 4 & 12 & 0 & 7 & 1 & 6 \\ 10 & 162 & 4 & 0 & 116 & 27 & 157 & 44 & 400 \\ 2 & 34 & 1 & 10 & 0 & 1 & 258 & 3880 & 1090 \\ 1 & 2 & 2 & 36 & 15 & 0 & 666 & 3 & 8 \\ 23 & 123 & 12 & 260 & 492 & 155 & 0 & 181 & 170 \\ 1280 & 713 & 6 & 22 & 85 & 4 & 44 & 0 & 115 \\ 24 & 2740 & 96 & 221 & 115 & 14 & 60600 & 655 & 0 \end{bmatrix}$$

By dividing each row by its row sum, we get transition matrix.

Transition Matrix

$$P = \begin{bmatrix}
0.00 & 0.01 & 0.83 & 0.03 & 0.03 & 0.00 & 0.02 & 0.00 & 0.09 \\
0.00 & 0.00 & 0.01 & 0.13 & 0.03 & 0.39 & 0.24 & 0.02 & 0.18 \\
0.64 & 0.20 & 0.00 & 0.02 & 0.06 & 0.00 & 0.04 & 0.01 & 0.03 \\
0.01 & 0.18 & 0.00 & 0.00 & 0.13 & 0.03 & 0.17 & 0.05 & 0.43 \\
0.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 & 0.74 & 0.21 \\
0.00 & 0.00 & 0.00 & 0.05 & 0.02 & 0.00 & 0.91 & 0.00 & 0.01 \\
0.02 & 0.09 & 0.01 & 0.18 & 0.35 & 0.11 & 0.00 & 0.13 & 0.12 \\
0.56 & 0.31 & 0.00 & 0.01 & 0.04 & 0.00 & 0.02 & 0.00 & 0.05 \\
0.00 & 0.04 & 0.00 & 0.00 & 0.00 & 0.00 & 0.94 & 0.01 & 0.00
\end{bmatrix}$$

Generating Text using Markov Chains:

To generate text, we repeat the following steps:

1. Set $t = 1$
2. Start with a seed word w_1 (our choice!).
3. Use Markov chain to *sample* next word based on seed.

$$w_{t+1} = w \quad \text{with probability } p_{w_t \to w}$$

4. Set $t \leftarrow t + 1$ and return to step 3 to generate a sentence (until t=17).

Our first haiku (first three words)

Let's start with

$$W_1 = \text{sky}$$

Our first haiku (first three words)

Let's start with

$$W_1 = \text{sky}$$

Then transition probabilities are:

sky	prof	kid	maths	is	flies	down	weird	cool
P_{ij}	0.01	0.18	0.00	0.13	0.03	0.17	0.05	0.43

Our first haiku (first three words)

Let's start with

$$W_1 = \text{sky}$$

Then transition probabilities are:

sky	prof	kid	maths	is	flies	down	weird	cool
P_{ij}	0.01	0.18	0.00	0.13	0.03	0.17	0.05	0.43

We randomly select with probability 0.43:

$$W_2 = \text{cool}$$

Our first haiku (first three words)

Let's start with

$$W_1 = \text{sky}$$

Then transition probabilities are:

sky	prof	kid	maths	is	flies	down	weird	cool
P_{ij}	0.01	0.18	0.00	0.13	0.03	0.17	0.05	0.43

We randomly select with probability 0.43:

$$W_2 = \text{cool}$$

Then transition probabilities are:

cool	prof	kid	maths	sky	is	flies	down	weird
P_{ij}	0.00	0.04	0.00	0.00	0.00	0.00	0.94	0.01

We randomly select:

Our first haiku (first three words)

Let's start with

$$W_1 = \text{sky}$$

Then transition probabilities are:

sky	prof	kid	maths	is	flies	down	weird	cool
P_{ij}	0.01	0.18	0.00	0.13	0.03	0.17	0.05	0.43

We randomly select with probability 0.43:

$$W_2 = \text{cool}$$

Then transition probabilities are:

cool	prof	kid	maths	sky	is	flies	down	weird
P_{ij}	0.00	0.04	0.00	0.00	0.00	0.00	0.94	0.01

We randomly select:

$$W_3 = \text{down}$$

with probability 0.94

Full haiku

sky cool down weird cool

down kid seems weird kid flies down

seems weird prof maths prof

If we want to write *in style of Oscar Wilde*?

If we want to write like Oscar Wilde,...

... then what transition matrix should we use?

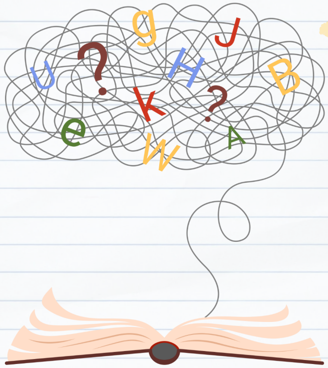
- Each author has their own transition matrix P.
- Matrix P is large $(15,000 \times 15,000)$

We *estimate* or *train* P_{Wilde} from data:

$$\hat{p}_{ij} = \frac{\text{\# instances of word } i \text{ followed by word } j}{\text{\# instances of word } i}$$

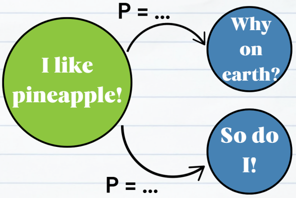
by scanning through books of Oscar Wilde.

Let's go deeper

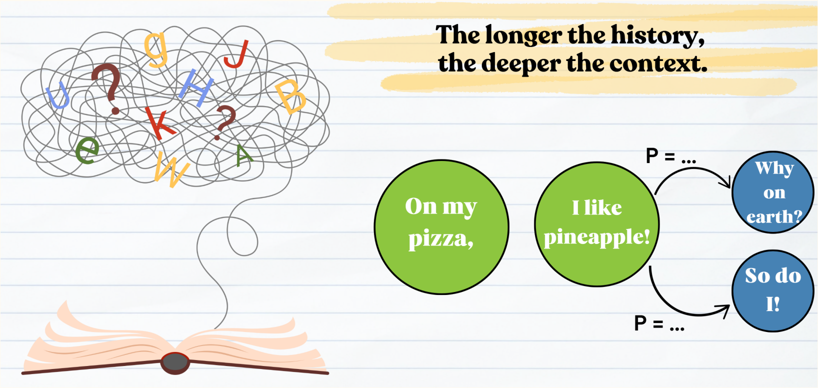


Besides limited vocabulary, sense and context were lacking...

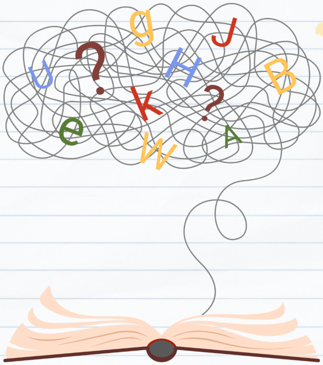
Again, choosing the next word **only looking at the previous one,** what did you expect...



Let's go deeper



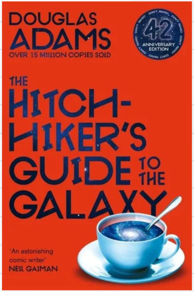
Let's go deeper



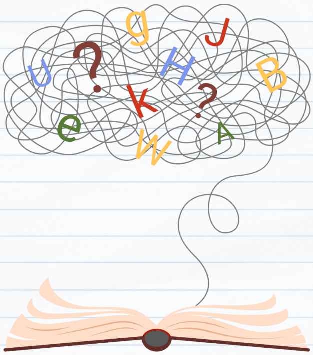
The longer the history, the deeper the context. BUT...

15759 words

How many transitions probabilities if we look back....



Let's go deeper



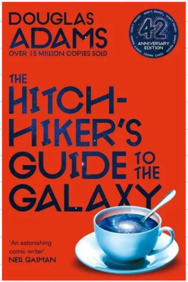
The longer the history, the deeper the context. BUT...

15759 words

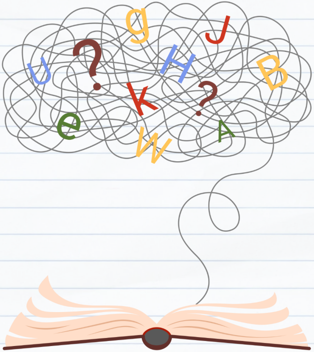
How many transitions probabilities if we look back....

one word?

$$15759^2 \approx 248 \text{ million}$$



Let's go deeper



The longer the history, the deeper the context. BUT...

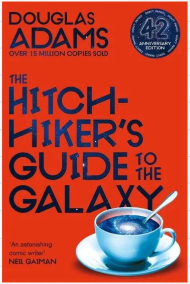
15759 words

How many transitions probabilities if we look back....

two words?

$$15759^3 \approx 391 \text{ billion}$$

bigger matrices !!



Next steps...

We can extend method to get better results:

1. Take more history (condition on more than 1 word)
2. Tokenize words
3. Contextualize previously observed words

Leap in complexity and capability of ChatGPT is significant...

Next steps...

We can extend method to get better results:

1. Take more history (condition on more than 1 word)
2. Tokenize words
3. Contextualize previously observed words

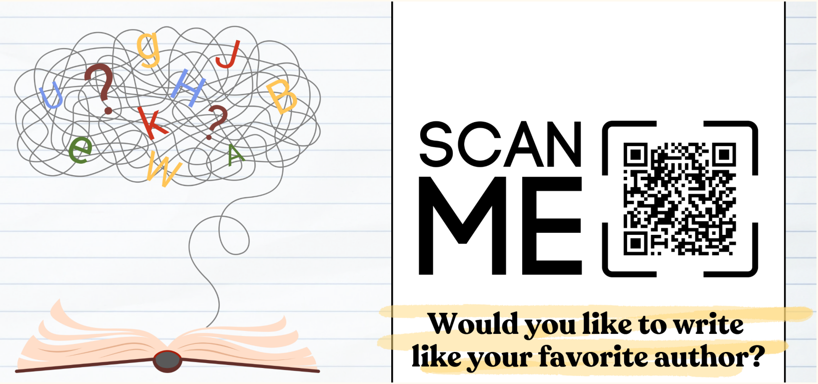
Leap in complexity and capability of ChatGPT is significant...

...but built on idea of sampling random next word in a sequence.

Summary

1. Words in sentence depend on context
2. Markov Chains
   - take into account context (more realistic)
   - but in a limited way (computationally efficient)
3. Transition matrix P describes Markov Chain:
   - Rows add up to 1
   - p_{ij} = probability of going from word i to word j
4. P can be learned from data.

An App



Cracking the code

How data science powers large language models

Writing
as traveling on a network of words with probability transitions

Human actions
-like writing- can be simulated via well-trained probabilities

ChatGpt
does something very similar

If you have any questions...



melania.lembo@usi.ch

martina.boschi@usi.ch

wite@usi.ch

If you are interested...

SCAN ME

The future needs Data Scientists

Bachelor in Data Science
3 years, 180 ECTS

Discover our innovative programme in data science