MARKOV MODEL

3rd Sem, MCA

Contents

- Markov model
  - Markov Property
  - Markov Chain
  - Transition Matrix
  - Markov Simulation
  - Markov applications
  - Hidden Markov model

Markov model

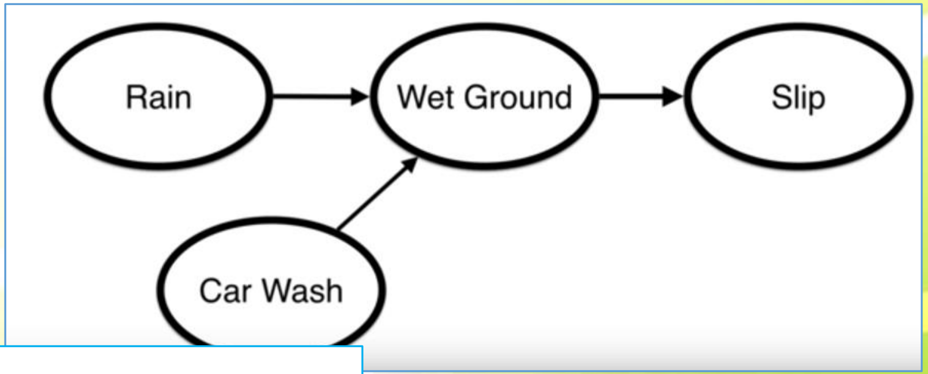
- In probability theory, a Markov model is a stochastic model used to model pseudo-randomly changing systems.
- Stochastic processes are widely used mathematical models of systems that appear to vary in random manner.
- Four common Markov models used in different situations, depending on whether every sequential state is observable or not, and whether the system is to be adjusted on the basis of observations made:

Markov models

	System state is fully observable	System state is partially observable
System is autonomous	Markov chain	Hidden Markov model
System is controlled	Markov decision process	Partially observable Markov decision process

Markov model

- According to markov model, it is assumed that future states depend only on the current state, not on the events that occurred before it
- **Markov property → "memorylessness".** It doesn't matter where we have been, only where we are.
- Probability distribution of future state only depends on present state.



$$\Pr(X_{n+1}=x \mid X_1=x_1, X_2=x_2, \ldots, X_n=x_n) = \Pr(X_{n+1}=x \mid X_n=x_n)$$

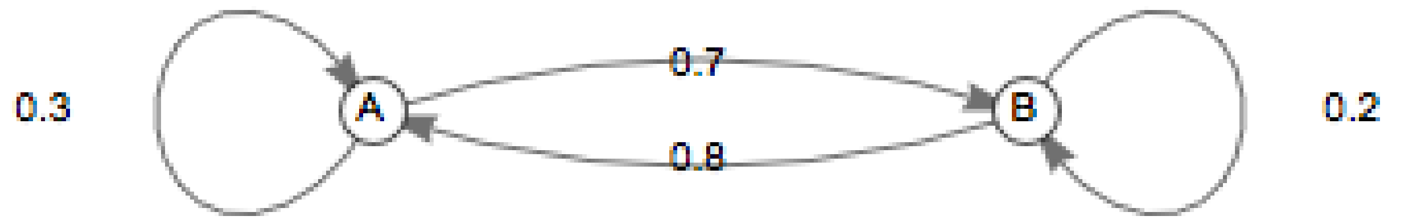
$$\underbrace{\mathbb{P}(X_{t+1}=s}_{\text{distribution of } X_{t+1}} \mid \underbrace{X_t=s_t}_{\text{depends on } X_t},\ \underbrace{\cancel{X_{t-1}=s_{t-1}, X_{t-2}=s_{t-2}, \ldots, X_1=s_1, X_0=s_0}}_{\text{but whatever happened before time } t \text{ doesn't matter.}})$$

Markov property

- Assume that a bag contains two red balls and one green ball. One ball was drawn yesterday, one ball was drawn today, and the final ball will be drawn tomorrow. All of the draws are "without replacement".
- Suppose you know today's ball was red, but *no information about yesterday's ball* → chance that tomorrow's ball will be red is 1/2.
  - Alternatively, if you know that both today and yesterday's balls were red, then you are guaranteed to get a green ball tomorrow.
- This discrepancy shows that the probability distribution for tomorrow's color depends not only on the present value, but is also affected by information about the past.
- This stochastic process of observed colors doesn't have the Markov property.
- Using the same experiment above, if sampling "without replacement" is changed to sampling "**with replacement**," the process of observed colors will have the **Markov property**.

Markov Chain

- Simplest Markov model is the Markov chain or markov process.
- It models the state of a system with a random variable that changes through time.
- Here, Markov property suggests that distribution for this variable depends only on distribution of previous state.
- Markov chains are used to calculate the probability of an event occurring by considering it as a state transitioning to another state or a state transitioning to the same state as before.



State Transition Diagram

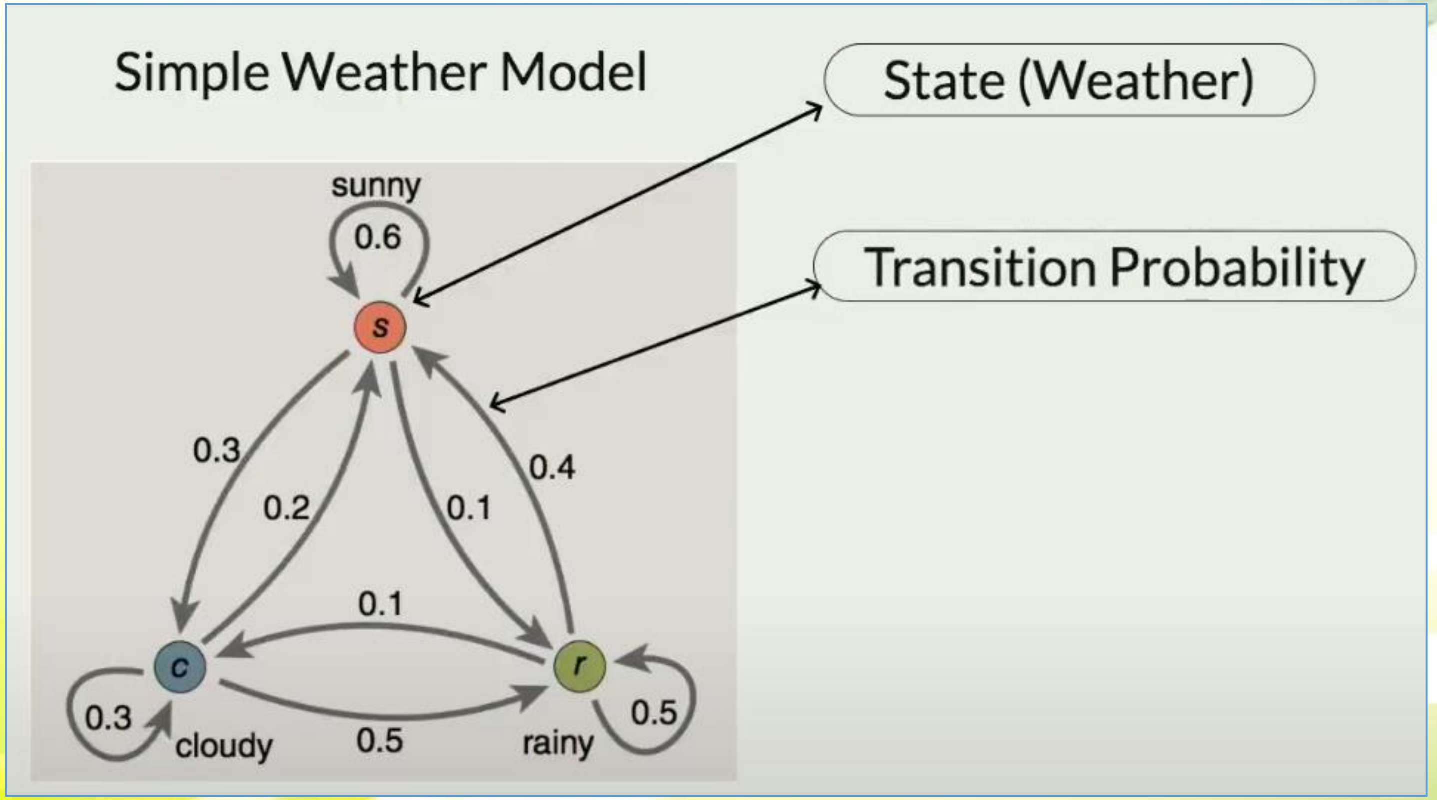
Markov Transition Matrix

- A Transition matrix is a square matrix used to describe the transitions of a Markov chain.
- Also called a probability matrix, stochastic matrix, substitution matrix, or Markov matrix.
- Each of its entries is a nonnegative real number representing a probability.
- Its a square matrix describing the probabilities of moving from one state to another in a dynamic system.
- In each row are the probabilities of moving from the state represented by that row, to the other states.
- Ideally, the rows / columns of a Markov transition matrix each add to one.

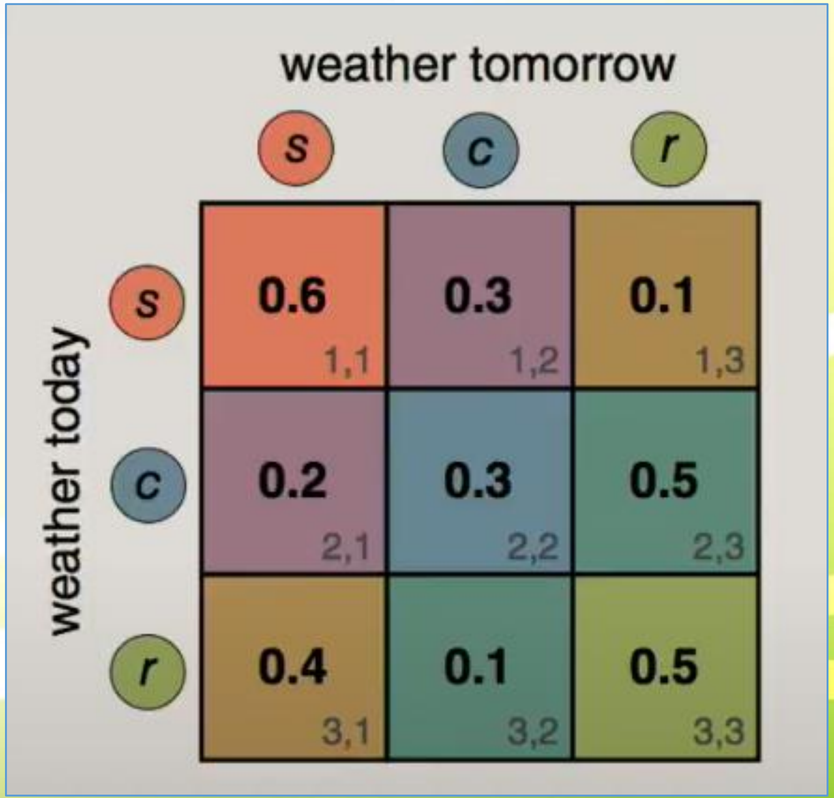
$$\sum_{j=1}^{\alpha} P_{i,j} = 1;$$

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,j} & \dots & P_{1,\alpha} \\ P_{2,1} & P_{2,2} & \dots & P_{2,j} & \dots & P_{2,\alpha} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{i,1} & P_{i,2} & \dots & P_{i,j} & \dots & P_{i,\alpha} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{\alpha,1} & P_{\alpha,2} & \dots & P_{\alpha,j} & \dots & P_{\alpha,\alpha} \end{bmatrix}.$$

Markov model Example

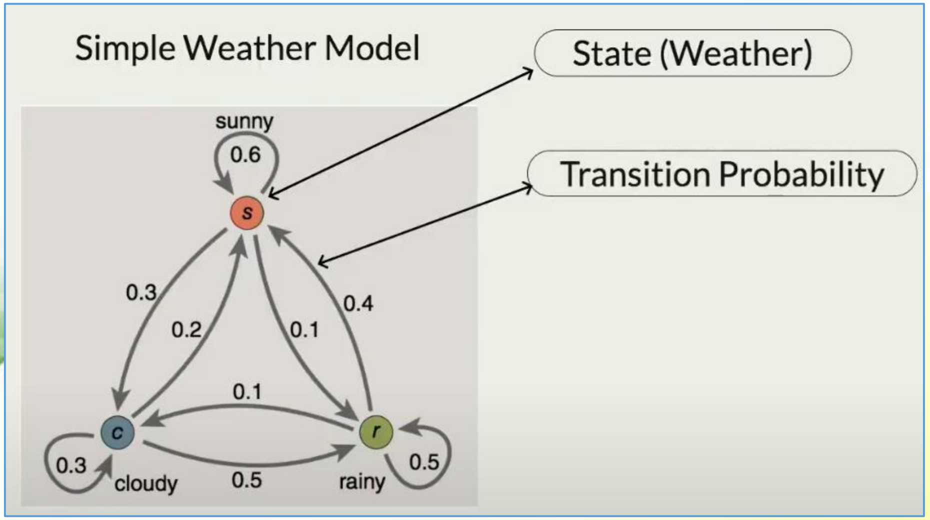


Markov Chain



Transition Matrix

Markov model Example



probability of rain in two days if it's cloudy today

$$(P_{2,3})^2 = c \begin{bmatrix} 0.2_{2,1} & 0.3_{2,2} & 0.5_{2,3} \end{bmatrix} \times \begin{array}{c} r \\ \begin{bmatrix} 0.1_{1,3} \\ 0.5_{2,3} \\ 0.5_{3,3} \end{bmatrix} \end{array}$$

$$(P_{2,3})^2 = (P_{2,1} \times P_{1,3}) + (P_{2,2} \times P_{2,3}) + (P_{2,3} \times P_{3,3})$$

$$(P_{2,3})^2 = (0.2 \times 0.1) + (0.3 \times 0.5) + (0.5 \times 0.5) = 0.42$$

weather tomorrow

weather today	s	c	r
s	0.6 (1,1)	0.3 (1,2)	0.1 (1,3)
c	0.2 (2,1)	0.3 (2,2)	0.5 (2,3)
r	0.4 (3,1)	0.1 (3,2)	0.5 (3,3)

Markov model Example

Earlier.. After choosing	Your Choice	# number of time
Rock	Rock	8
	Paper	16
	Scissors	15
Paper	Rock	19
	Paper	6
	Scissors	5
Scissors	Rock	13
	Paper	8
	Scissors	10

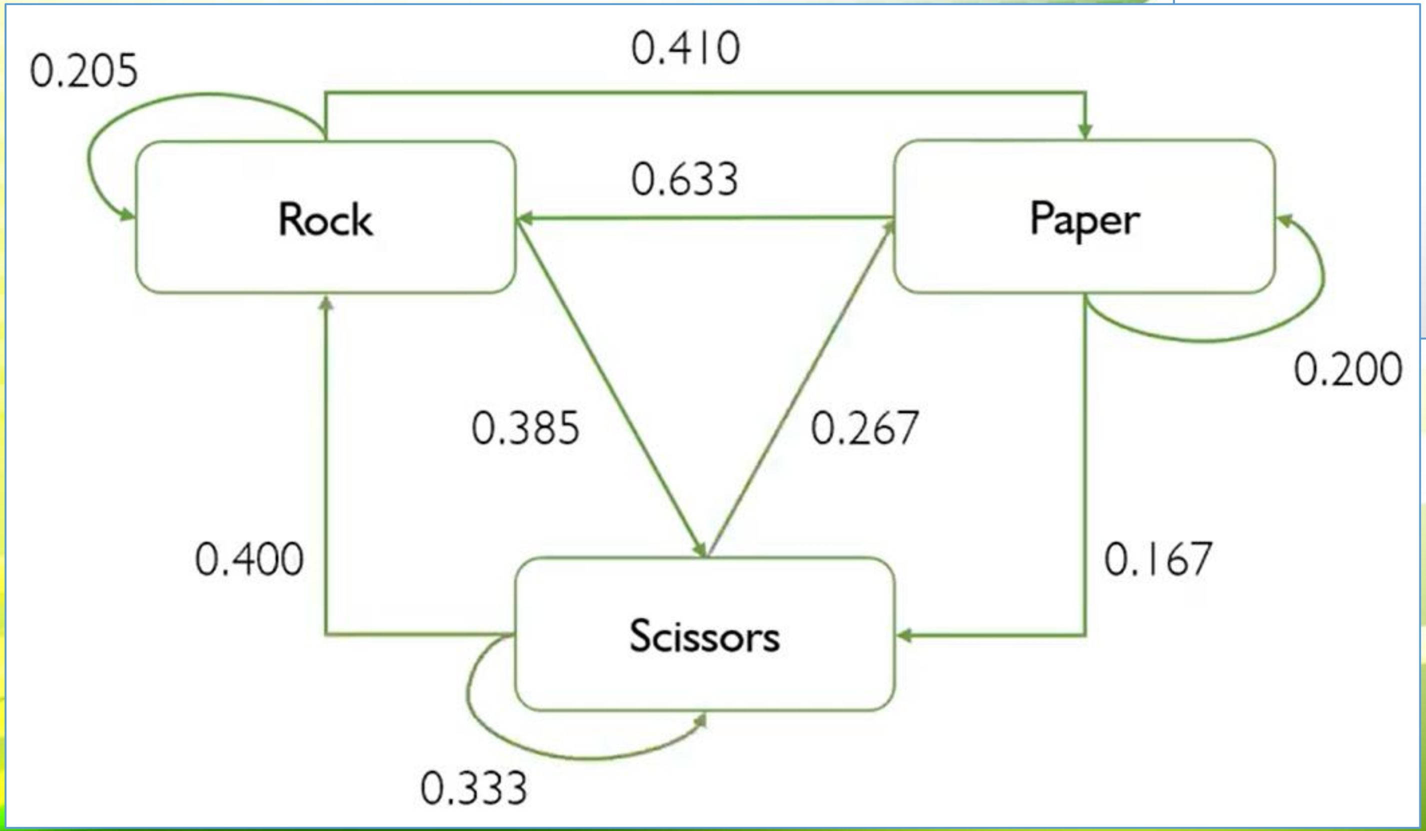
Your Choice	# number of time
Rock	40
Paper	30
Scissors	30

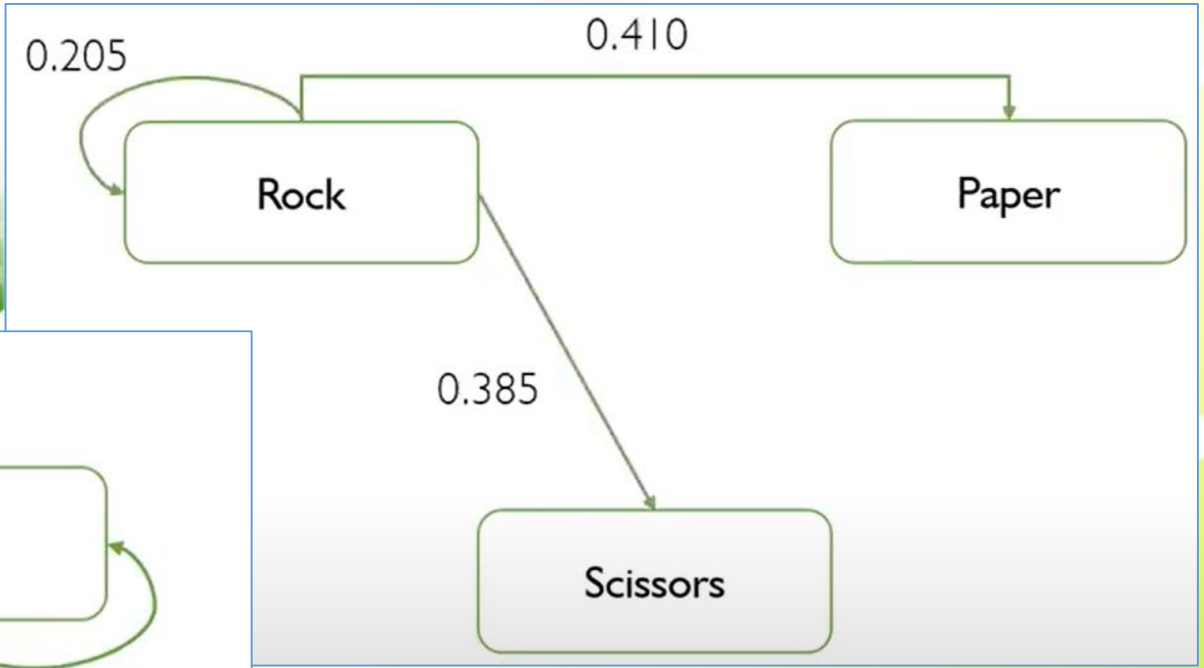
To…. From…	Rock	Paper	Scissors
Rock	8/39 = 20.5%	41%	38.5%
Paper	63.3%	20%	16.7%
Scissors	41.9%	25.8%	32.2%

Transition Matrix

Markov model Example

- Player's behavior described with 3 states here.
- Diagram to show Transition between states.





To.... From...	Rock	Paper	Scissors
Rock	20.5%	41%	38.5%
Paper	63.3%	20%	16.7%
Scissors	41.9%	25.8%	32.2%

Markov model Example

- One can simulate from a Markov chain by noting that the collection of moves from any given state (the corresponding row in the probability matrix) form a multinomial distribution.
- One can thus simulate from a Markov Chain by simulating from a multinomial distribution.
- Types of Markov Simulation:
  - **Microsimulation**
  - Cohort simulation

RSRSRSPRSRSRRRPPPRRSSSSPRRPRPR
PRPPSRSRPSSPRPPSRSRPRRPRRPRSSP
RPRPRPRRSSSRSSPPSSRSRSPRPRPSPP
RSPRSSRPRR...

Markov model Example

Markov Simulation

- Microsimulation
- **Cohort simulation**

Game	Rock	Paper	Scissors
1	1000	0	0
2	205	410	385
3	42 + 260 + 162 = 464	265	271
4	381	311	308
5	410	295	295
…	…	…	…
infinite	400	300	300

Stationary condition / equilibrium state

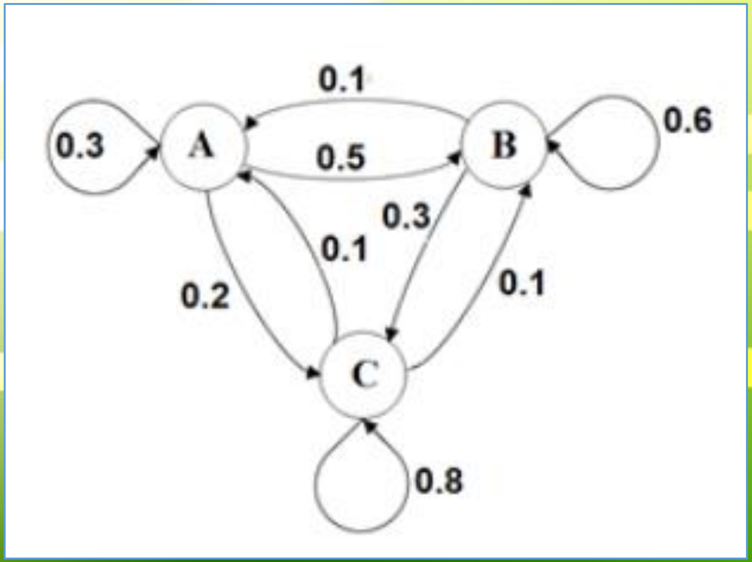
Your Choice	# number of time
Rock	40
Paper	30
Scissors	30

To.... From...	Rock	Paper	Scissors
Rock	20.5%	41%	38.5%
Paper	63.3%	20%	16.7%
Scissors	41.9%	25.8%	32.2%

Markov model Example

- Consider bike sharing system across. A customer pays a fee to avail the service and can borrow a bicycle from any bike share station and then can return it to the same or another system. There are 3 stations (A, B, C). Its found for any specific day,
    - of the bikes borrowed from station A, 30% are returned to station A, 50% end up at station B, & 20% end up at station C.
    - of the bikes borrowed from station B, 10% end up at station A, 60% returned to station B, & 30% end up at station C
    - of the bikes borrowed from station C, 10% end up at station A, 10% end up at station B, & 80% are returned to station C.
- Prepare the State transition matrix and diagram for the above scenario.

$$T = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$



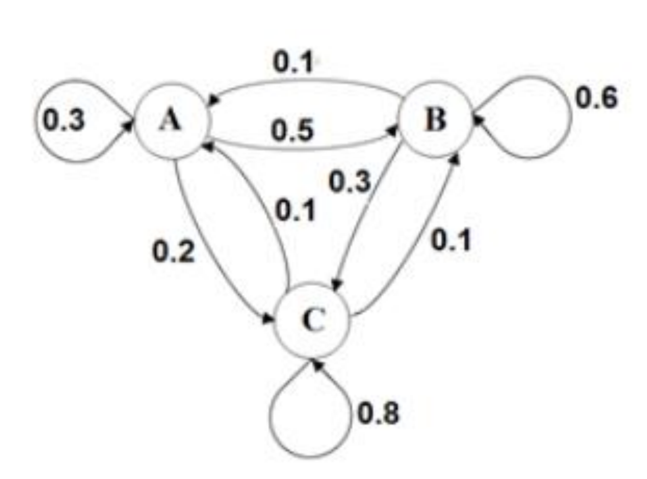
Markov model Example

- In the beginning, 30% of the bikes are at station A, 45% at station B, & 25% are at station C.
- Initial state vector is;

$$\begin{array}{c} \\ V_0 = \end{array}\begin{array}{ccc} A & B & C \\ \left[0.30\right. & 0.45 & \left.0.25\right] \end{array}$$

$$T = \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$

$$V_1 = V_0 T$$



- after one transition, V_1 is computed by multiplying V_0 by the transition matrix T,

$$\begin{aligned} V_1 &= V_0 T \\ &= [0.30 \quad 0.45 \quad 0.25] \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \\ &= [.30(.3) + .45(.1) + .25(.1) \quad .30(.5) + .45(.6) + .25(.1) \quad .30(.2) + .45(.3) + .25(.8)] \\ &= [.16 \quad .445 \quad .395] \end{aligned}$$

After 1 day (1 transition), 16 % of the bikes are at station A, 44.5 % are at station B and 39.5% are at station C

Markov model Example

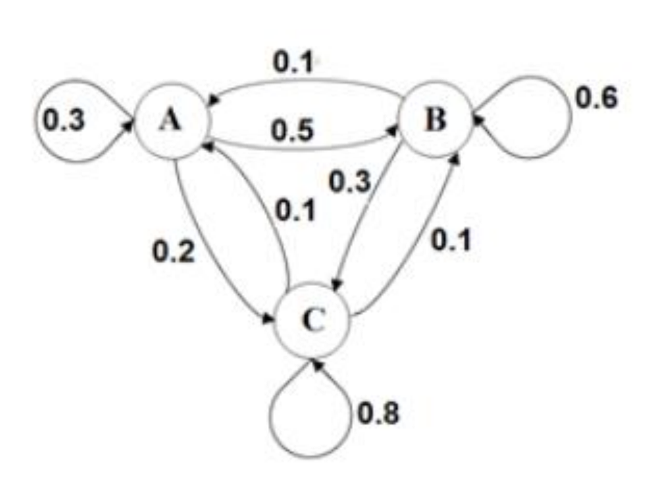
- Initial state vector is;

$$\begin{array}{cccc} & A & B & C \\ V_0 = & [0.30 & 0.45 & 0.25] \end{array}$$

$$V_1 = [.16 \quad .445 \quad .395]$$

$$T = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$

- after one transition, V_1 is computed by multiplying V_0 by the transition matrix T,
- V2, the state vector after two transitions is computed by multiplying the state vector after one transition V1 by the transition matrix T.



$$V_2 = V_1 T = [.16 \quad .445 \quad .395] \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} = [.132 \quad .3865 \quad .4815]$$

$$V_1 = V_0 T, \text{ so } V_2 = V_1 T = (V_0 T)\, T = V_0 T^2$$

After 2 days (2 transitions), 13.2 % of the bikes are at station A, 38.65 % are at station B & 48.15% are at station C.

Markov model Example

$$V_0 = \begin{array}{c} \begin{matrix} A & B & C \end{matrix} \\ \begin{bmatrix} 0.30 & 0.45 & 0.25 \end{bmatrix} \end{array}$$

$$T = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0.3 & 0.5 & 0.2 \\ B & 0.1 & 0.6 & 0.3 \\ C & 0.1 & 0.1 & 0.8 \end{array}$$

$$V_1 = [.16 \quad .445 \quad .395]$$

$$V_1 = V_0 T, \text{ so } V_2 = V_1 T = (V_0 T) T = V_0 T^2$$

$$V_2 = V_1 T = [.16 \quad .445 \quad .395] \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} = [.132 \quad .3865 \quad .4815]$$

$$T^2 = TT = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.16 & 0.47 & 0.37 \\ 0.12 & 0.44 & 0.44 \\ 0.12 & 0.19 & 0.69 \end{bmatrix}$$

$$V_2 = V_0 T^2 = [0.30 \quad 0.45 \quad 0.25] \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}^2$$

$$= [0.30 \quad 0.45 \quad 0.25] \begin{bmatrix} 0.16 & 0.47 & 0.37 \\ 0.12 & 0.44 & 0.44 \\ 0.12 & 0.19 & 0.69 \end{bmatrix}$$

$$V_2 = V_0 T^2 = [.132 \quad .3865 \quad .4815]$$

Multiplying initial state vector V_0 by T^n ($V_n = V_0 T^n$) is the distribution of bicycles after n transitions.

Markov model Example

- A city is served by two cable TV companies, BestTV and CableCast.
    - Due to their aggressive sales tactics, each year 40% of BestTV customers switch to CableCast; the other 60% of BestTV customers stay with BestTV.
    - On the other hand, 30% of the CableCast customers switch to Best TV.
- Express the information above as a transition matrix which displays the probabilities of going from one state into another state.

$$
\begin{array}{ccc}
 & & \textbf{Next year} \\
 & & \begin{array}{cc} \text{BestTV} & \text{CableCast} \end{array} \\
\textbf{This year} & \begin{array}{c} \text{BestTV} \\ \text{CableCast} \end{array} & \begin{bmatrix} .60 & .40 \\ .30 & .70 \end{bmatrix}
\end{array}
$$

Markov model Example

$$\begin{array}{ccc} & & \textbf{Next year} \\ & & \text{BestTV} \quad \text{CableCast} \\ \textbf{This year} & \begin{array}{r}\text{BestTV} \\ \text{CableCast}\end{array} & \begin{bmatrix} .60 & .40 \\ .30 & .70 \end{bmatrix} \end{array}$$

- Suppose that today 1/4 of customers subscribe to BestTV and 3/4 of customers subscribe to CableCast. After 1 year, what percent subscribe to each company?

$$V_1 = V_0 T$$

- Suppose instead that today of 80% of customers subscribe to BestTV and 20% subscribe to CableCast. After 1 year, what percent subscribe to each company?

$$T = \begin{bmatrix} .60 & .40 \\ .30 & .70 \end{bmatrix}$$

initial distribution given by the initial state vector $V_0 = [1/4 \quad 3/4] = [.25 \quad .75]$

After 1 year, the distribution of customers is

$$V_1 = V_0 T = [.25 \quad .75] \begin{bmatrix} .60 & .40 \\ .30 & .70 \end{bmatrix} = [.375 \quad .625]$$

After 1 year, 37.5% of customers subscribe to BestTV and 62.5% to CableCast.

$$V_1 = V_0 T = [.8 \quad .2] \begin{bmatrix} .60 & .40 \\ .30 & .70 \end{bmatrix} = [.54 \quad .46]$$

After 1 year, 54% of customers subscribe to BestTV and 46% to CableCast.

Markov model Example

Professor Symons either walks to school, or he rides his bicycle. If he walks to school one day, then next day, he will walk or cycle with equal probability. But if he bicycles one day, then probability that he will walk the next day is 1/4.

- Express this information in a transition matrix.
- if it is assumed that initial day is Monday, prepare the matrix that gives probabilities of a transition from Monday to Wednesday.

$$\begin{array}{ccc} & & \textbf{Next Day} \\ & & \begin{array}{cc}\text{Walk} & \text{Bicycle}\end{array} \\ \textbf{Initial Day} & \begin{array}{c}\text{Walk} \\ \text{Bicycle}\end{array} & \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} \end{array}$$

$$\begin{aligned} \mathrm{T}^2 = \mathrm{T} \times \mathrm{T} &= \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} \begin{bmatrix} 1/2 & \\ 1/4 & 1/2 \end{bmatrix} \\ &= \begin{bmatrix} 1/4 + 1/8 & 1/4 + 3/8 \\ 1/8 + 3/16 & 1/8 + 9/16 \end{bmatrix} \\ &= \begin{bmatrix} 3/8 & 5/8 \\ 5/16 & 11/16 \end{bmatrix} \end{aligned}$$

If today is Monday, then Wednesday is two days from now, representing two transitions. We need to find the square, T^2, of the original transition matrix T, using matrix multiplication.

$$\begin{array}{ccc} & & \textbf{Wednesday} \\ & & \begin{array}{cc}\text{Walk} & \text{Bicycle}\end{array} \\ \textbf{Monday} & \begin{array}{c}\text{Walk} \\ \text{Bicycle}\end{array} & \begin{bmatrix} 3/8 & 5/8 \\ 5/16 & 11/16 \end{bmatrix} \end{array}$$

Markov model Example

How to calculate the probability of the states.

Ex. $P(X_2 = 3)$; $P(X_1 = 4)$ etc

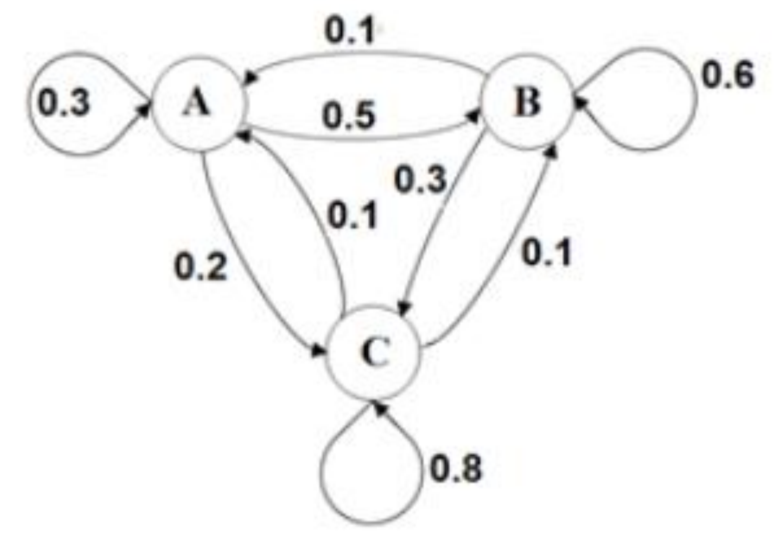
How to calculate the probability after n-steps.

Ex. $p_{ij}^{(2)}$, $p_{23}^{(3)}$, $P(X_3 = 4 | X_1 = 2)$ etc

$P(X_3 = 3 | X_1 = 1)$ or

like $p_{13}^{(2)}$, $p_{23}^{(3)}$ etc

$$T = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$



Markov model Example

The probability of movement of state 1 to 3 after two time periods/steps,

$$\boxed{\begin{aligned} &P(X_3 = 3 | X_1 = 1) \\ &= p_{13}^{(2)} \end{aligned}}$$

For $p_{13}^{(2)}$, Calculate P^2 by using

$$P^2 = P.P$$

$$= \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix} \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}$$

$$= \begin{bmatrix} \square & \square & p_{13} \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}$$

Similarly, for $p_{23}^{(3)}$, calculate P^3 as

$P^3 = P^2 P$ and get

$$P^3 = \begin{bmatrix} \square & \square & \square \\ \square & \square & p_{23} \\ \square & \square & \square \end{bmatrix}$$

Markov model Example

i.e., $P(X_4 = C | X_0 = T)$

Since $P(X_4 = C | X_0 = T) = p_{TC}^{(4)}$.

Compute P^4 and look at its value of p_{TC}

$$P^2 = P.P$$

$$= \begin{bmatrix} 0 & 1 \\ 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1/2 & 1/2 \end{bmatrix}$$

$$= \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix}$$

Hence, $P^4 = P^2 P^2$

$$= \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix}$$

$$= \begin{array}{cc} & \begin{array}{cc} T & C \end{array} \\ \begin{array}{c} T \\ C \end{array} & \begin{bmatrix} 3/8 & 5/8 \\ 5/16 & 11/16 \end{bmatrix} \end{array}$$

Hence, the required answer is $p_{TC}^{(4)} = 5/8$.

Markov model Example

Example6: The TPM of the Markov chain with three states 1, 2, 3 is

$$P = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

And the initial probability is $(0.7, 0.2, 0.1)$. Calculate

(i) $P(X_2 = 1)$

(ii) $P(X_3 = 2, X_2 = 3, X_1 = 3, X_0 = 2)$.

Markov model Example

Consider the Markov chain with three states, S={1,2,3}, that has given transition matrix

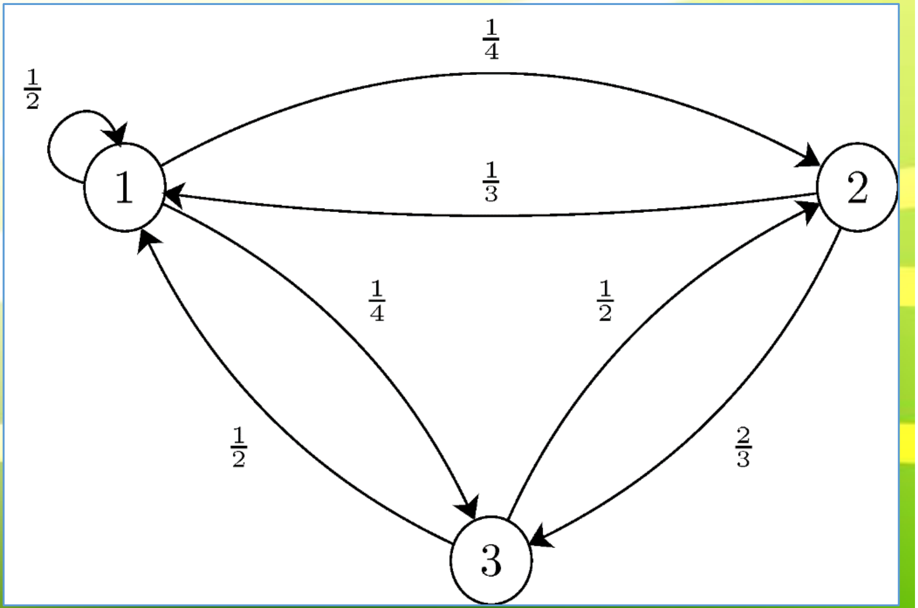
$$P = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

a) Draw the state transition diagram for this chain.

b) If we know $P(X_1=1) = P(X_1=2) = 1/4$, then find $P(X_1=3, X_2=2, X_3=1)$

$$\begin{aligned} P(X_1 = 3) &= 1 - P(X_1 = 1) - P(X_1 = 2) \\ &= 1 - \frac{1}{4} - \frac{1}{4} \\ &= \frac{1}{2}. \end{aligned}$$

$$\begin{aligned} P(X_1 = 3, X_2 = 2, X_3 = 1) &= P(X_1 = 3) \cdot p_{32} \cdot p_{21} \\ &= \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{3} \\ &= \frac{1}{12}. \end{aligned}$$



Markov model Example

Given the transition probabilities matrix below:

$$P = \begin{pmatrix} 1/3 & 2/3 & 0 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}$$

and $\pi_0 = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$

Find:

(a) P^2

(b) P^3

(c) $P(X_2 = 2)$ — $\boxed{q_2 = q_0 P^2}$

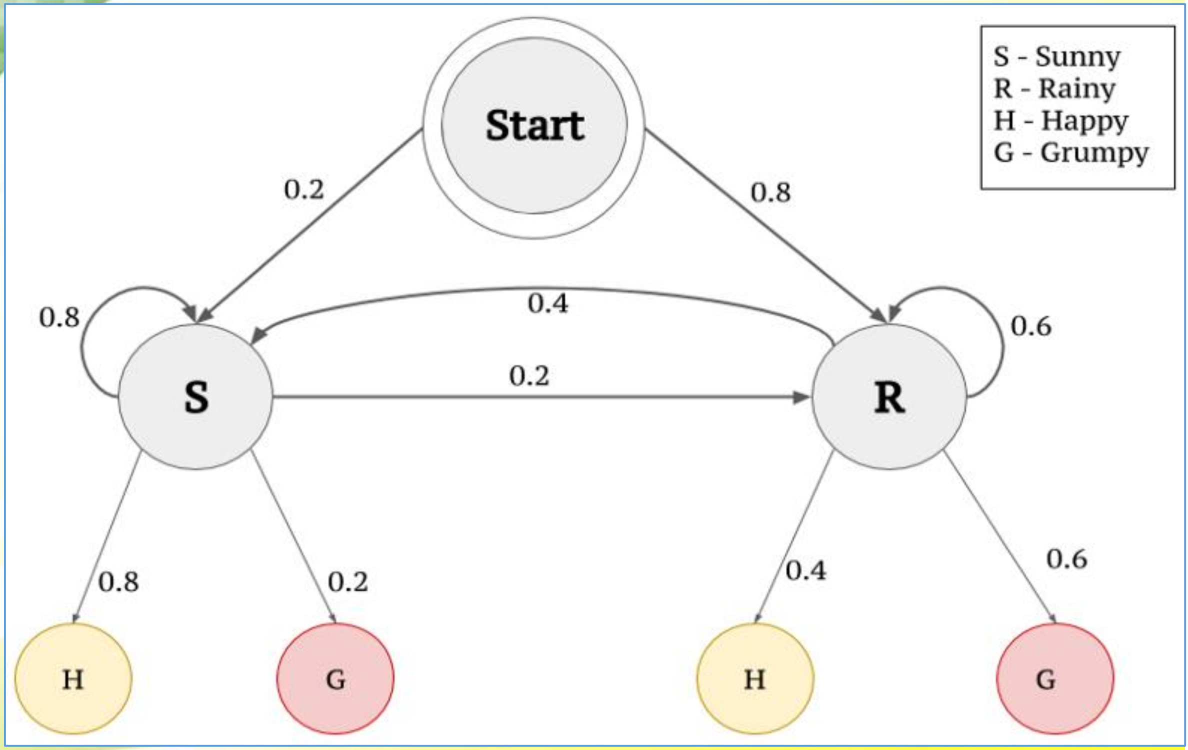
(d) $P(X_0 = 1, X_3 = 3)$ — $\boxed{P^3_{(13)}}$

(e) $P(X_1 = 2, X_2 = 3, X_3 = 1 | X_0 = 1)$ — $\boxed{Q_0(1) * P_{(12)} * P_{(23)} * P_{(31)}}$

(f) $P(X_2 = 3 | X_1 = 3)$ — $\boxed{P^1_{(33)}}$

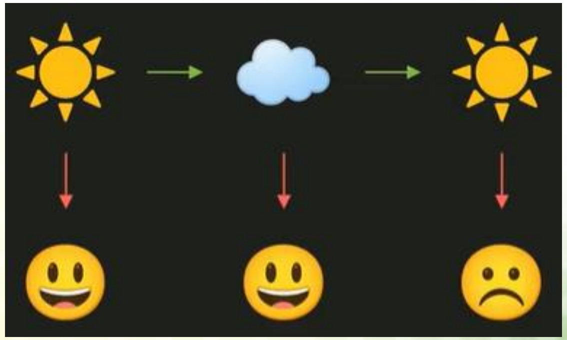
Hidden Markov model (HMM)

- HMM is a statistical Markov model in which system being modeled is assumed to be a Markov process with unobserved (hidden) states & observed (sequence of) outputs.
  - **Markov Model**: Series of (hidden) states $z=\{z_1, z_2 \ldots\ldots\ldots\}$ drawn from state alphabet $S = \{s_1, s_2, \ldots\ldots s_i \,|S\}$ where z_i belongs to S.
  - **Hidden Markov Model**: Series of observed output $x = \{x_1, x_2, \ldots\ldots\}$ drawn from an output alphabet $V= \{v_1, v_2, \ldots, v_n |v\}$ where x_i belongs to V.



Set of observed states (S) = {Happy, Grumpy}
Set of hidden states (Q) = {Sunny , Rainy}

Hidden Markov model

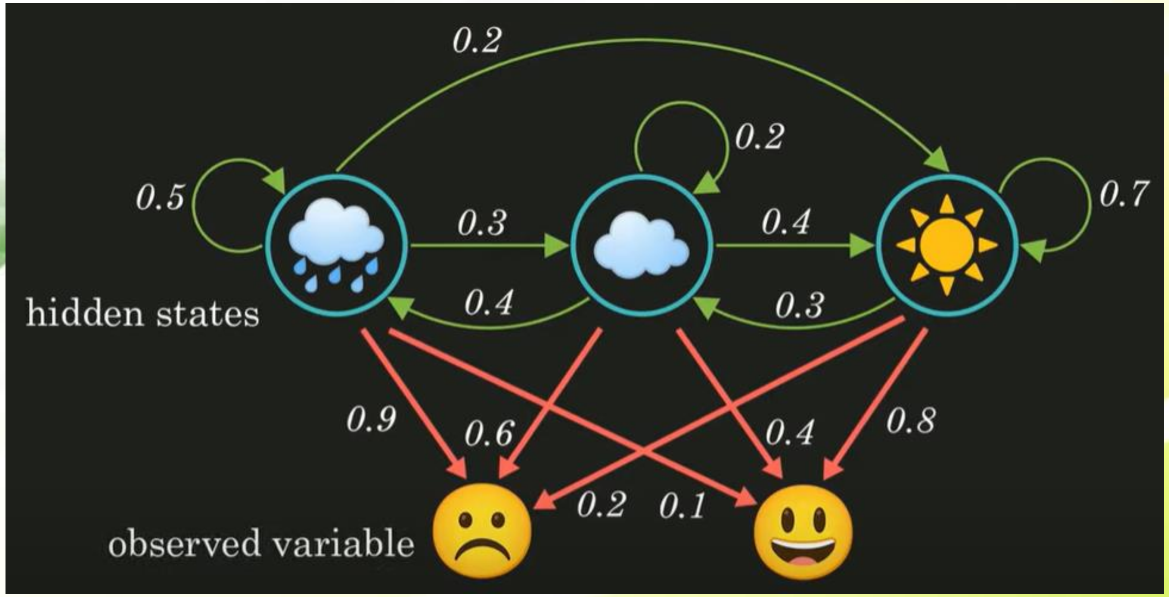


$P($Y = 😃 😃 🙁 , X = ☀️ ☁️ ☀️ $)$

$P(X_1 =$ ☀️$)$ $P(Y_1 =$ 😃 $\mid X_1 =$ ☀️$)$

$P(X_2 =$ ☁️ $\mid X_1 =$ ☀️$)$ $P(Y_2 =$ 😃 $\mid X_2 =$ ☁️$)$

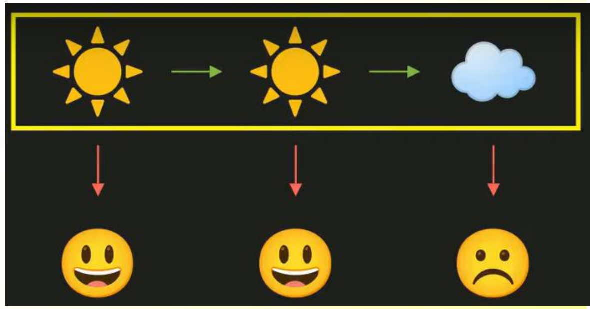
$P(X_3 =$ ☀️ $\mid X_2 =$ ☁️$)$ $P(Y_3 =$ 🙁 $\mid X_3 =$ ☀️$)$



	🌧️	☁️	☀️
🌧️	0.5	0.3	0.2
☁️	0.4	0.2	0.4
☀️	0.0	0.3	0.7

	🙁	😃
🌧️	0.9	0.1
☁️	0.6	0.4
☀️	0.2	0.8

Hidden Markov model



$$\underset{X=X_1,X_2,\ldots X_n}{\arg\max} \; P(X = X_1, X_2, \ldots X_n \mid Y = Y_1, Y_2, \ldots Y_n)$$

$$\underset{X=X_1,X_2,\ldots X_n}{\arg\max} \; \frac{P(Y|X)P(X)}{P(Y)}$$

$$P(Y|X) = \prod P(Y_i \mid X_i)$$

$$P(X) = \prod P(X_i \mid X_{i-1})$$

$$\underset{X=X_1,X_2,\ldots X_n}{\arg\max} \; \prod P(Y_i \mid X_i) \; P(X_i \mid X_{i-1})$$

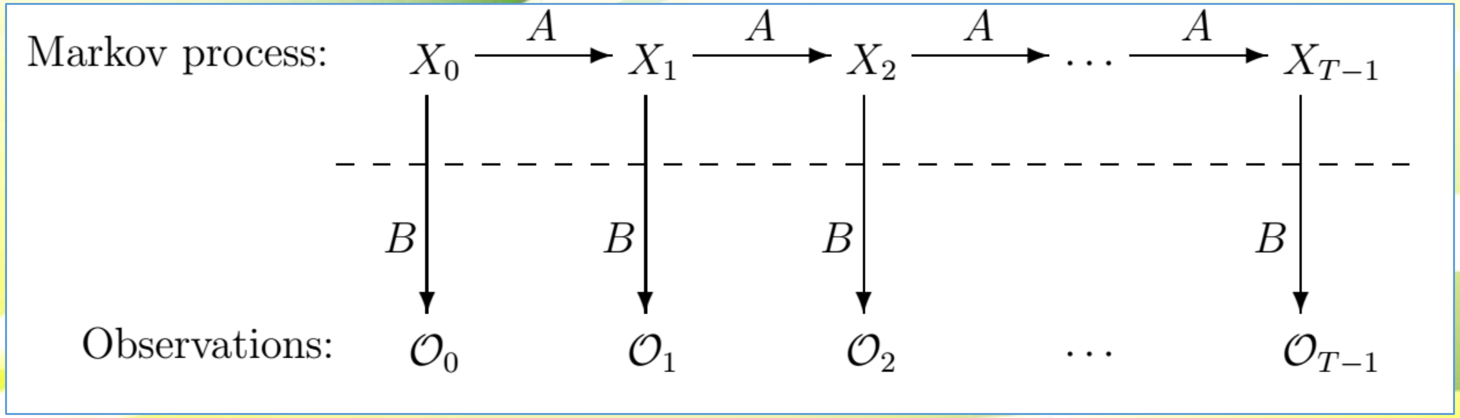
Hidden Markov model

- **HMM Inference:** For given a sequence, infer the hidden state path.
    - There are potentially many state paths that could generate the same sequence

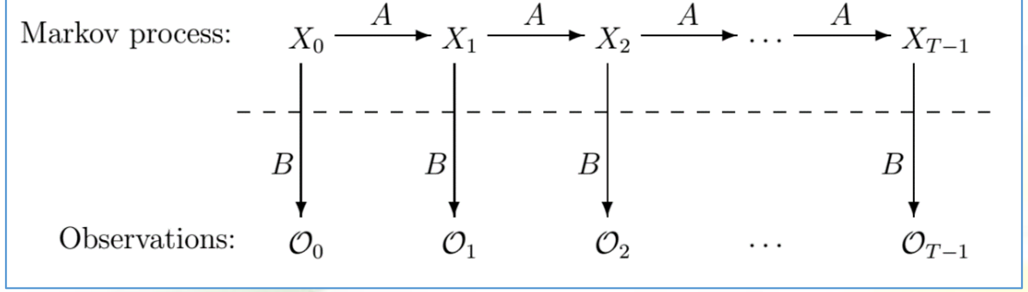
      → find the one with the highest probability.
- The **Viterbi Algorithm** uses dynamic programming to solve the HMM problem.
- **Dynamic programming** involves breaking down a complex problem into simpler sub-problems using a recursive approach. It includes the steps of Initialization, Recursion, and Termination to find the sequence of the hidden states.

Hidden Markov model

- **Transition Probabilities:** Probability of moving from one state to another. For N states, this is an NxN matrix.
- **Emission Probabilities**: Probability of a particular output given a particular state. Given a choice of M possible observation symbols, this is an NxM matrix. This is also called output or observation probabilities.
- **Initial Probabilities:** Probability of being in a state at the start, say, yesterday or ten days ago.



Hidden Markov model



Let A, B and π denote the transition matrix, observation matrix and initial state distribution respectively.

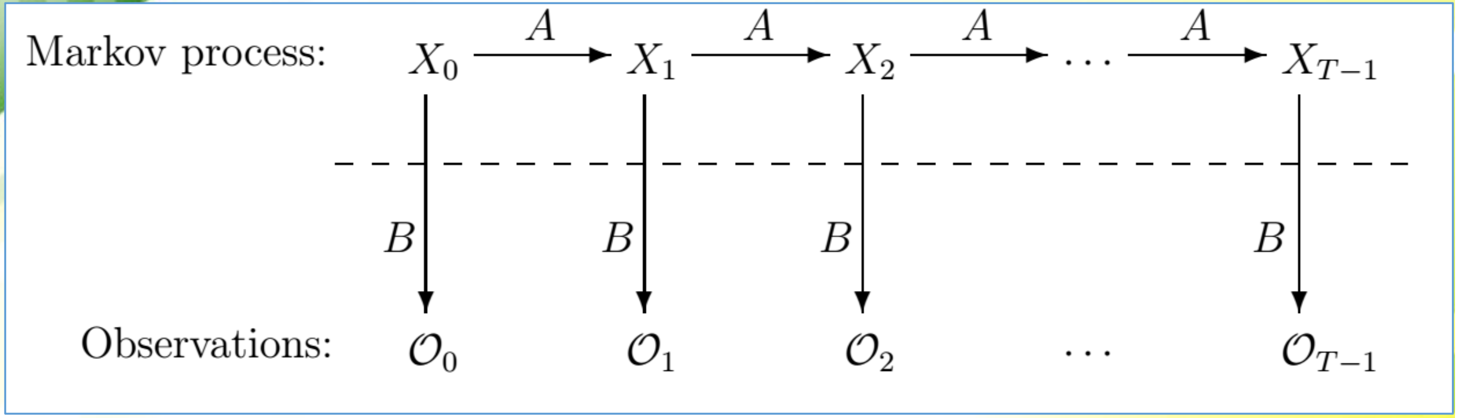
HMM can be represented as λ = (A, B, π). Let observation sequence be O and state sequence be Q.

HMM can be used to solve three types of problems:

- **Likelihood Problem**: Given O and λ, find the likelihood P(O|λ). How likely is a particular sequence of observations? Forward algorithm solves this problem.
- **Decoding Problem**: Given O and λ, find the best possible Q that explains O. Given the observation sequence, what's the best possible state sequence? Viterbi algorithm solves this problem.
- **Learning Problem**: Given O and Q, learn λ, perhaps by maximizing P(O|λ). What model best maps states to observations?
  - Baum-Welch algorithm, also called forward-backward algorithm, solves this problem. In the language of machine learning, we can say that O is training data and the number of states N is the model's hyperparameter.

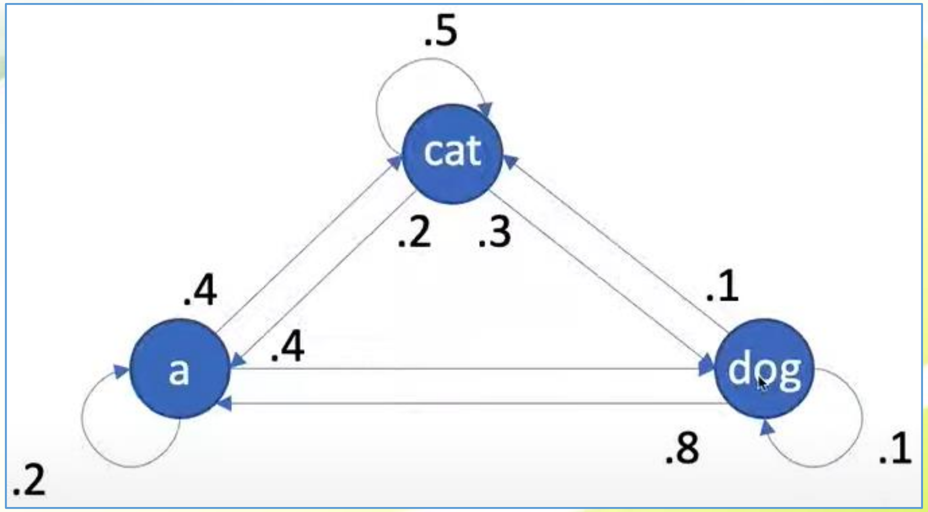
HMM Applications

- An application, where HMM is used, aims to recover the data sequence where the next sequence of the data can not be observed immediately but the next data depends on the old sequences. Taking the above intuition into account the HMM can be used in the following applications:
  - Computational finance
  - speed analysis, Transportation forecasting
  - Speech recognition, Speech synthesis
  - Part-of-speech tagging
  - Document separation in scanning solutions
  - Machine translation, Handwriting recognition
  - Genetic analysis, Time series analysis



HMM (Language modelling)

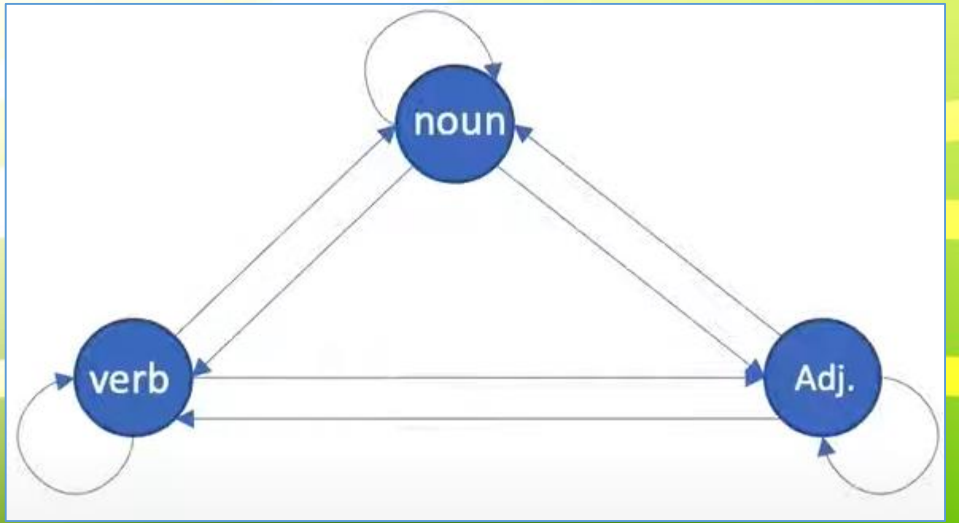
A **language model** is a probability distribution over a sequence of words (phrase / sentence).



P(a, cat, dog) = p(a) x p(cat | a) x p(dog | cat)

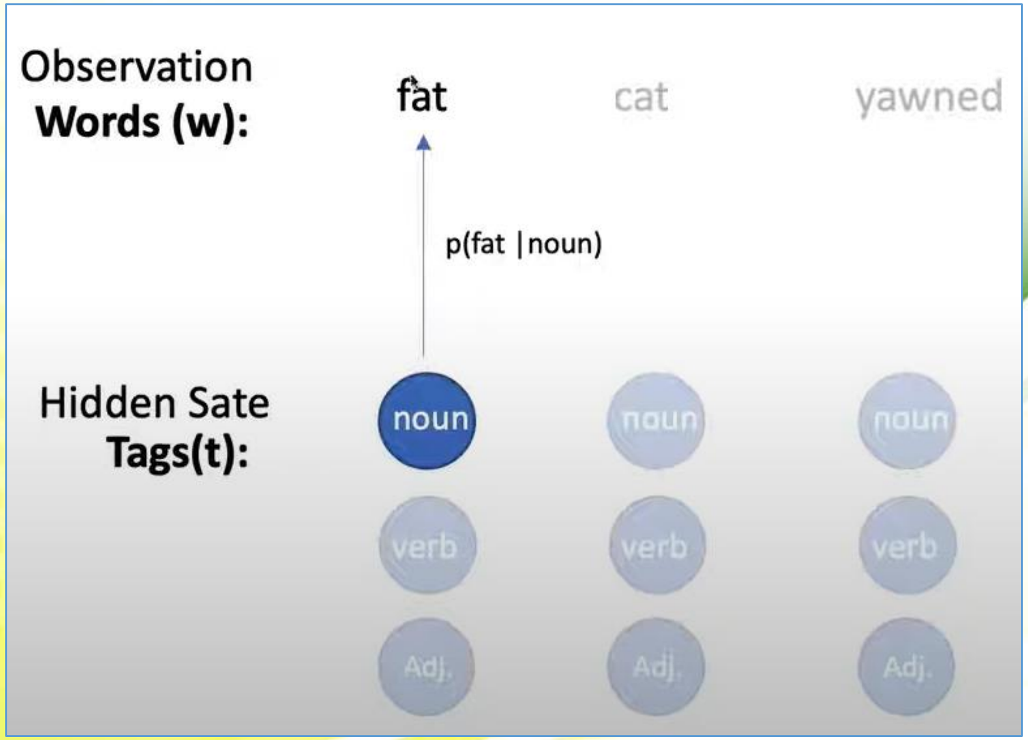
P(a, cat, dog) = p(a) x p(cat | a) x p(dog | ~~a,~~ cat)

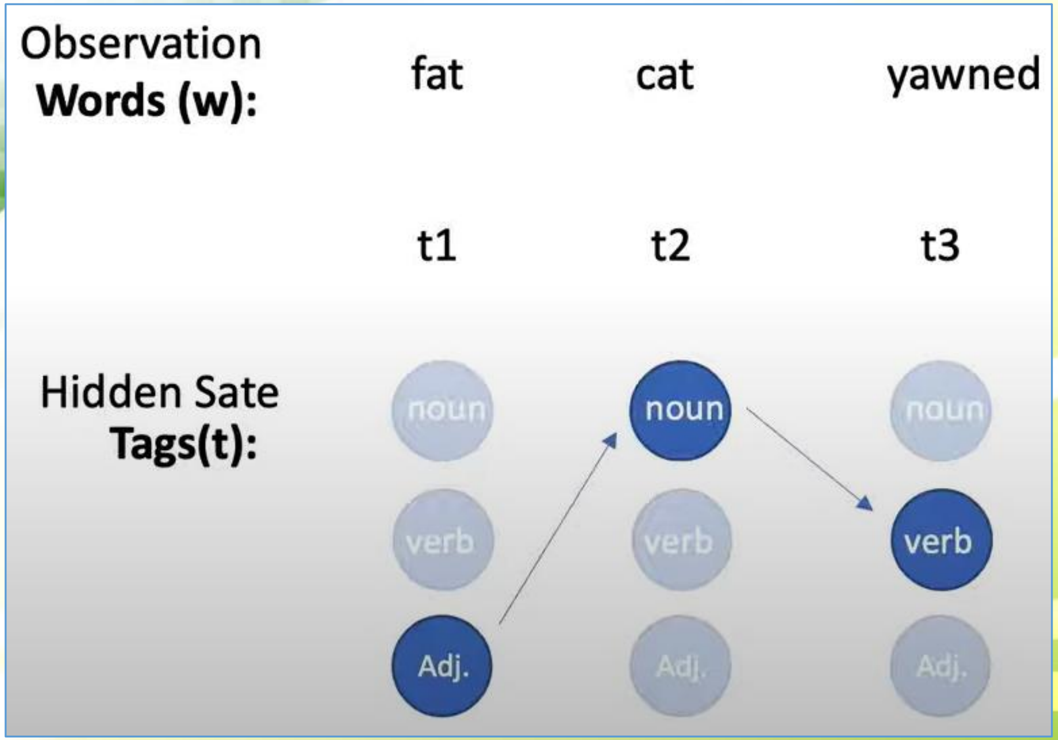
Hidden property of the word: noun, verb, adj.v



A **hidden Markov model (HMM)** accounts for the temporal relationship between hidden state elements, and how those hidden states emit "observations" - or data that we tend to collect in the real world.

HMM (Language modelling)





$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

Language modelling (NLP)

Natural Language

日本語で

冬は世界各地でさまざまなお祝いが行われる時期です。ほんのいくつか例を挙げるだけでも、ハナカ、クリスマス、クワンザ、新年などさまざまなお祝いがあります。各文化によってその祝い方はさまざまですが、ほとんどのお祝いにはごちそうが欠かせません。

Artificial Language

```
try {
    cMessage = messageQueue.take();
    for (AsyncContext ac : queue) {
        try {
            PrintWriter acWriter = ac.g
            acWriter.println(cMessage);
            acWriter.flush();
        } catch(IO
            System
            queue.
        }
    }
} catch(Interrupte
```

```
append(CharSequenc
append(char c)
append(CharSequenc
format(String form
format(Locale l, S
printf(String form
```

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"];' % ast[1]
        else:
            print '"]'
    else:
        print '"];'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print '    %s -> {' % nodename,
        for name in children:
            print '%s' % name,
```

Languages

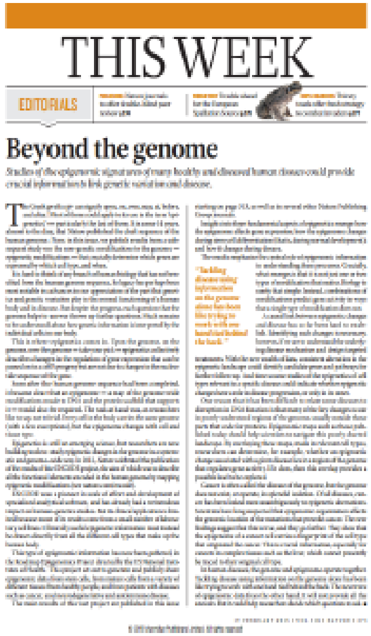
Representation
- Syntax (Grammar/Well formed Sentence)
- Semantic (Meaning/Truth value of the Sentence)

Natural Language

A vocabulary consists of a set of words (w_i)

(http://learnenglish.britishcouncil.org/en/vocabulary-games)

A text is composed of a sequence of words from a vocabulary



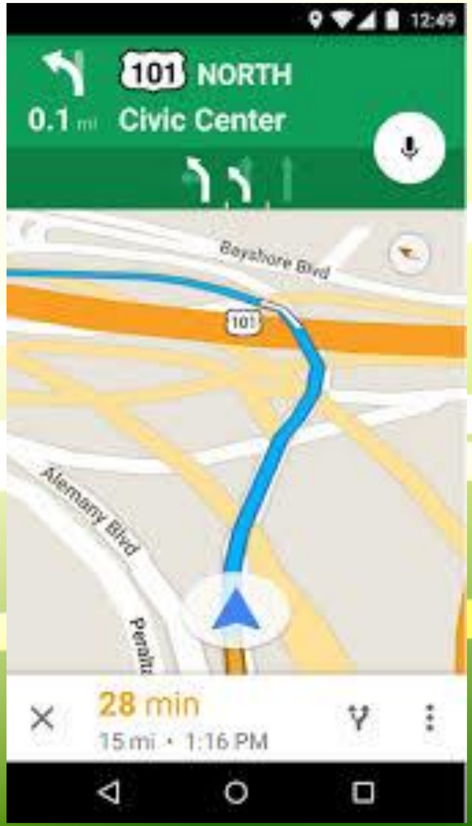
(http://www.nature.com/polopoly_fs/1.16929!/menu/main/topColumns/topLeftColumn/pdf/518273a.pdf)

A language is constructed of a set of all possible texts

(http://www.old-engli.sh/language.php)

Language model (NLP)

- Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages.



NLP

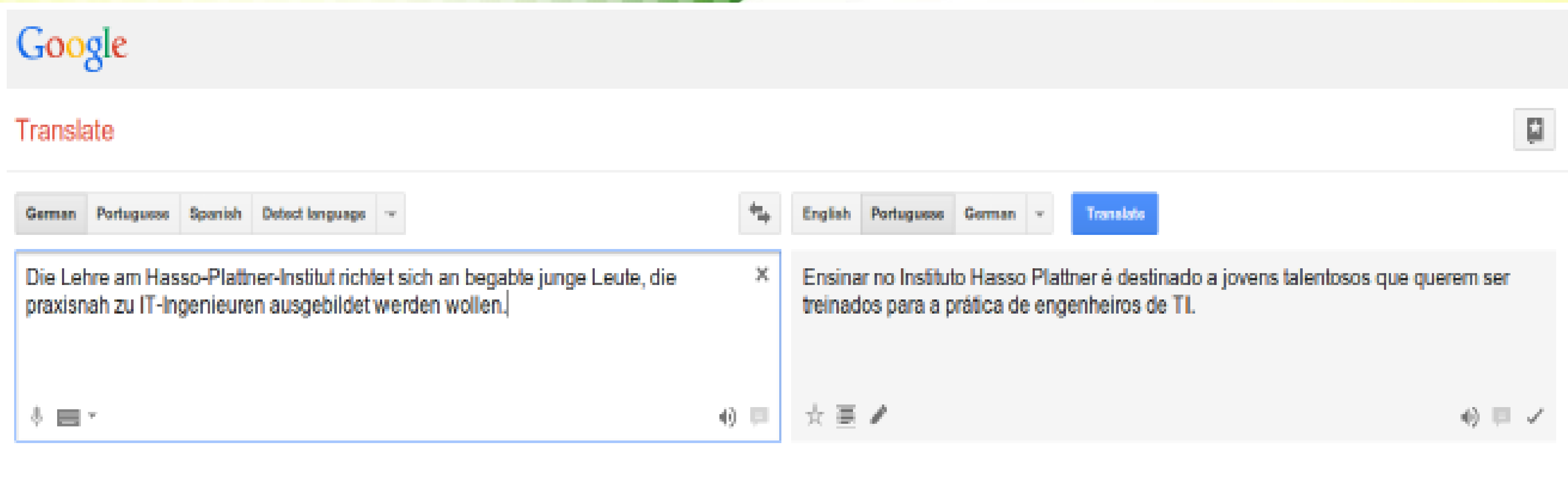
- Study that focuses on interactions between human language and computers.
- NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way.
- With NLP, developers can organize & structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.
- NLP considers the hierarchical structure of language: several words make phrase, several phrases make sentence &, ultimately, sentences convey ideas.
- Reduce words to their root, or break up text into **tokens.**

N-gram	Words/Sentences
Unigram	I enjoyed the coffee
Bigram	I enjoyed enjoyed the the coffee
Trigram	I enjoyed the enjoyed the coffee

Introduction to NLP Applications

Machine Translation

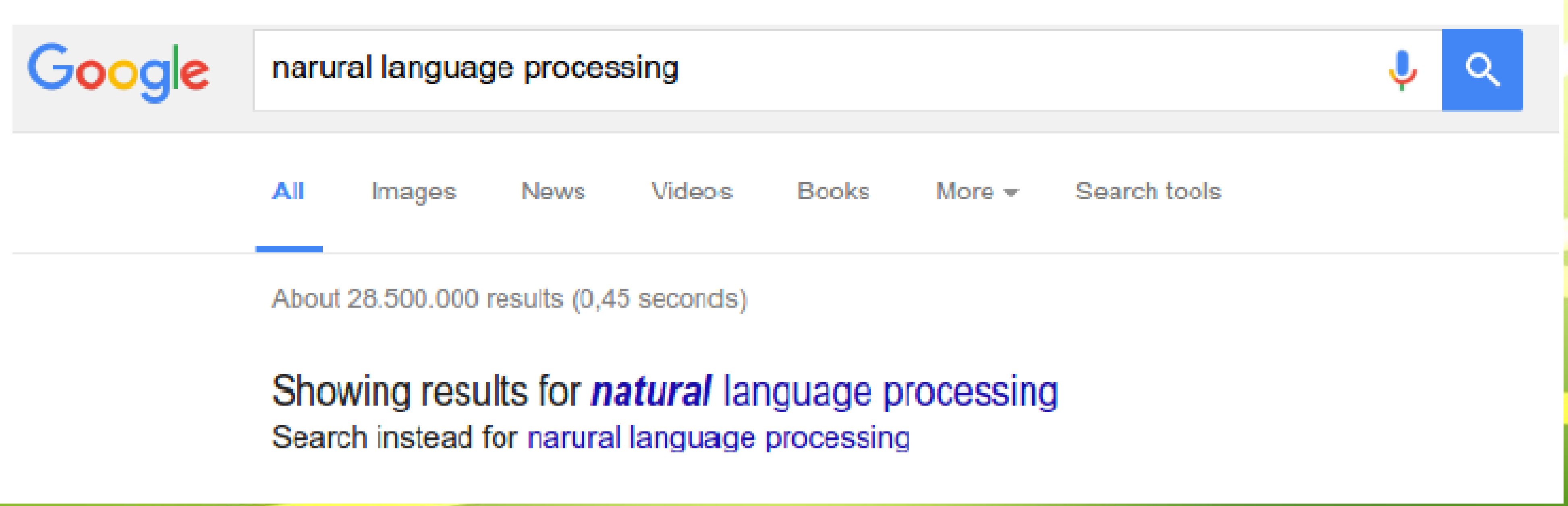
- Translating a text from one language to another



Introduction to NLP Applications

Spelling & Grammar Checking

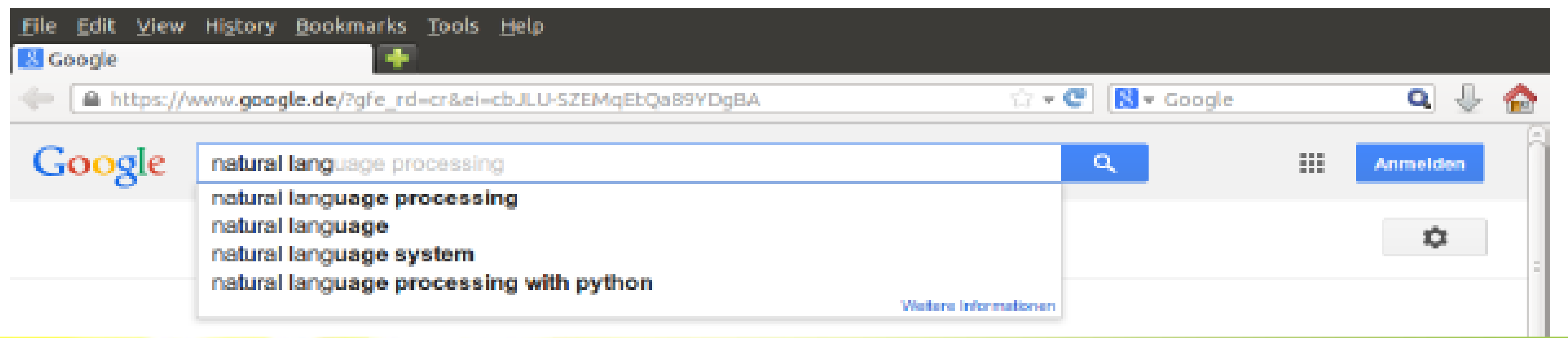
- Checking spelling and grammar
- Suggesting alternatives for errors



Introduction to NLP Applications

Word Prediction

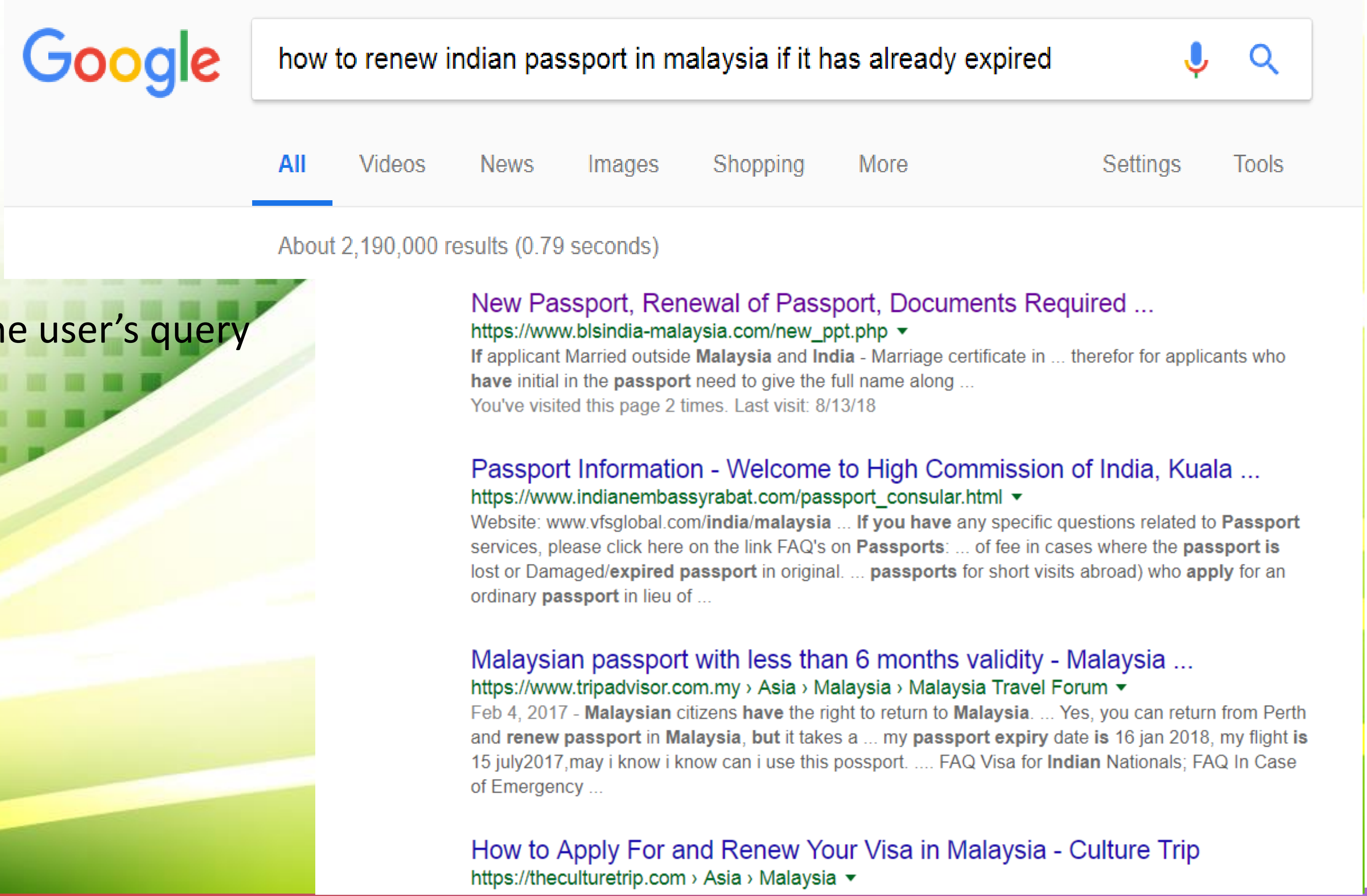
- Predicting next word that is highly probable to be typed by user



Introduction to NLP Applications

Information Retrieval

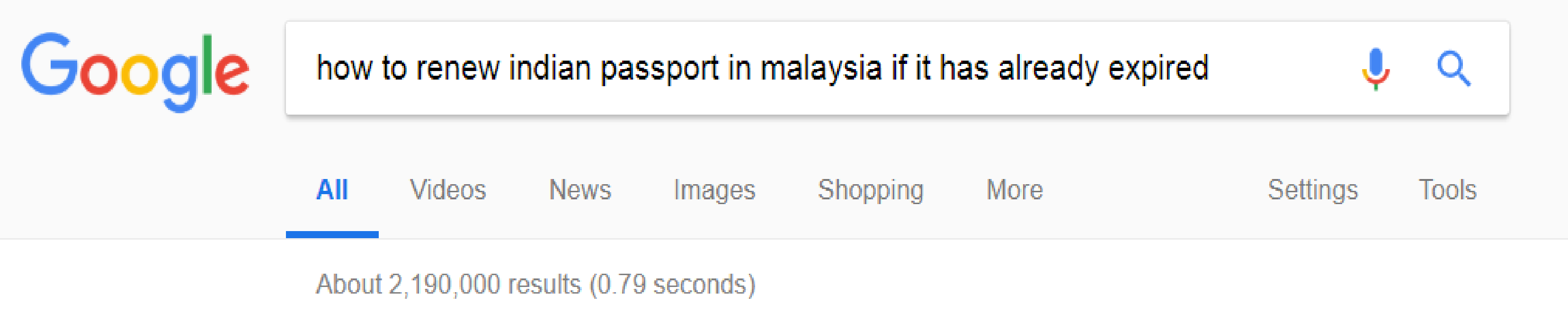
- Finding relevant information to the user’s query



Introduction to NLP Applications

Information Retrieval: Stop words

- **Pre-processing:** Process of converting data to something computer can understand.
- One major forms of pre-processing is to filter out **useless data**.
- **STOP WORDS**: Useless words (data) in NLP.
- A commonly used word ("the", "a", "an", "in") that search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as result to a search query.



Introduction to NLP Applications

Information Extraction

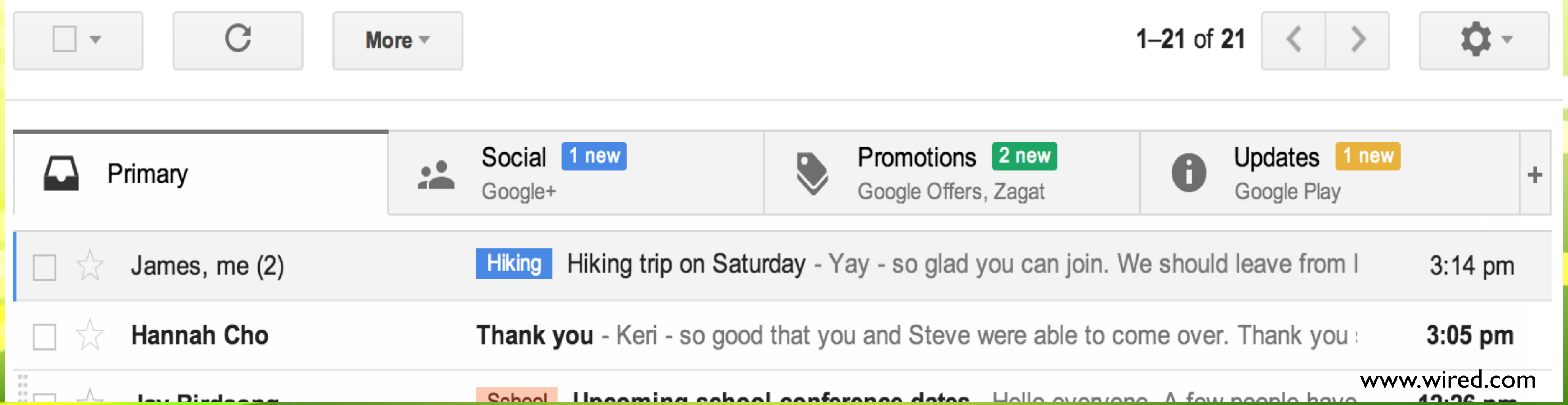
- Unstructured text to database entries.

New York Times Co. named Russell T. Lewis, 45, president and general manager of its flagship New York Times newspaper, responsible for all business-side activities. He was executive vice president and deputy general manager. He succeeds Lance R. Primis, who in September was named president and chief operating officer of the parent.

Person	Company	Post	State
Russell T. Lewis	New York Times newspaper	president and general manager	start
Russell T. Lewis	New York Times newspaper	executive vice president	end
Lance R. Primis	New York Times Co.	president and CEO	start

Introduction to NLP Applications

Text Classification



www.wired.com

Introduction to NLP Applications

Text Categorization

- Assigning one (or more) pre-defined category to a text

Summarization

- Generating a short summary from one or more documents, sometimes based on a given query

Language Comprehension

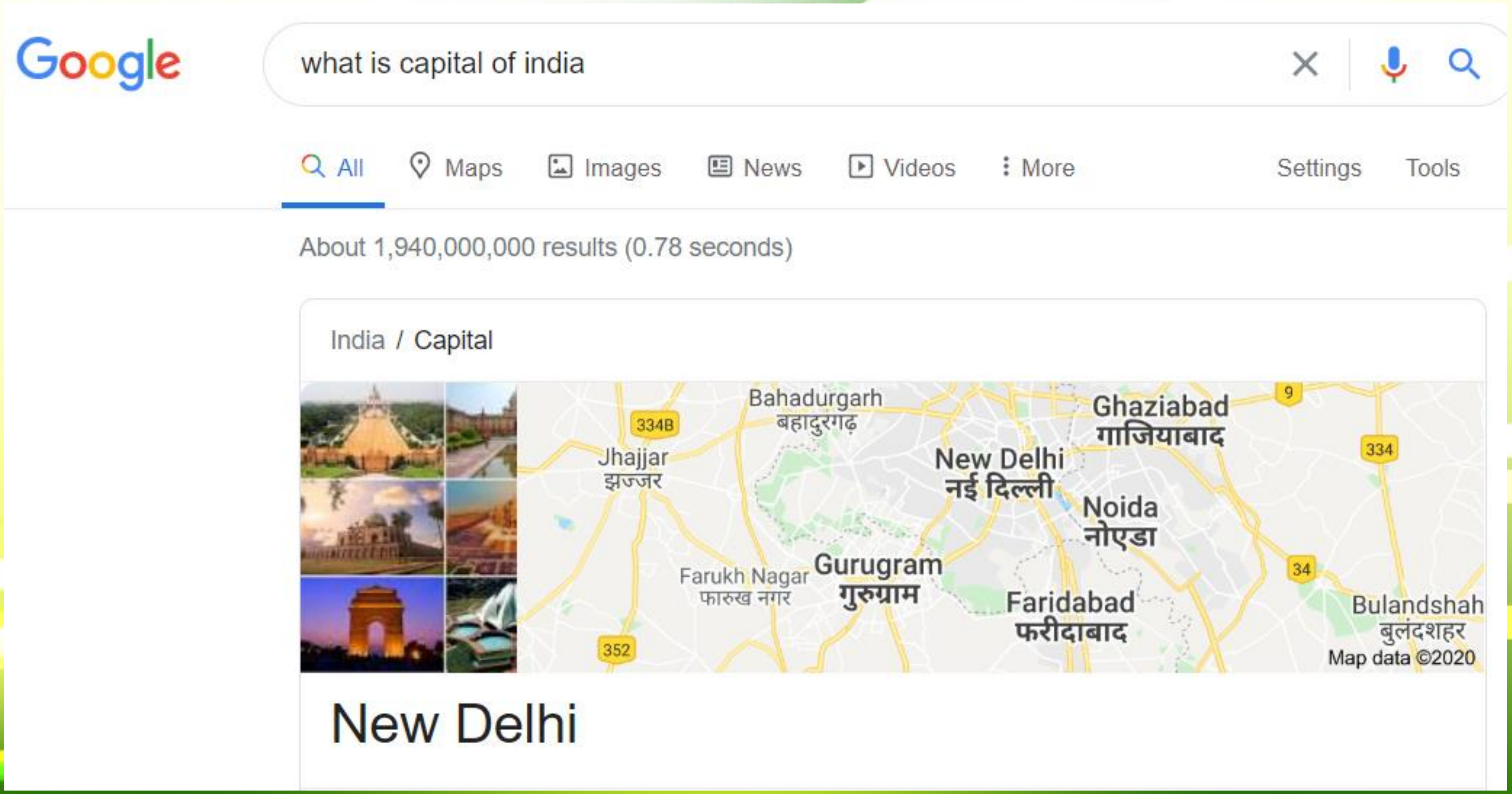
Q: who wrote Winnie the Pooh?
Q: where is Chris lived?

Christopher Robin is alive and well. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book

Introduction to NLP Applications

Question answering.

- Go beyond search



Introduction to NLP Applications

Information Extraction

- Extracting important concepts from texts and assigning them to slot in a certain template

WIKIPEDIA
The Free Encyclopedia

Angela Merkel

Merkel at the EPP Summit, March 2016

Chancellor of Germany

Incumbent

Assumed office
22 November 2005

President	Horst Köhler Christian Wulff Joachim Gauck
Deputy	Franz Müntefering Frank-Walter Steinmeier Guido Westerwelle Philipp Rösler Sigmar Gabriel
Preceded by	Gerhard Schröder

Leader of the Christian Democratic Union

Incumbent

Assumed office
10 April 2000

Preceded by	Wolfgang Schäuble

Minister for the Environment

In office
17 November 1994 – 26 October 1998

Chancellor	Helmut Kohl
Preceded by	Klaus Töpfer
Succeeded by	Jürgen Trittin

Minister for Women and Youth

In office
18 January 1991 – 17 November 1994

Chancellor	Helmut Kohl
Preceded by	Ursula Lehr
Succeeded by	Claudia Nolte

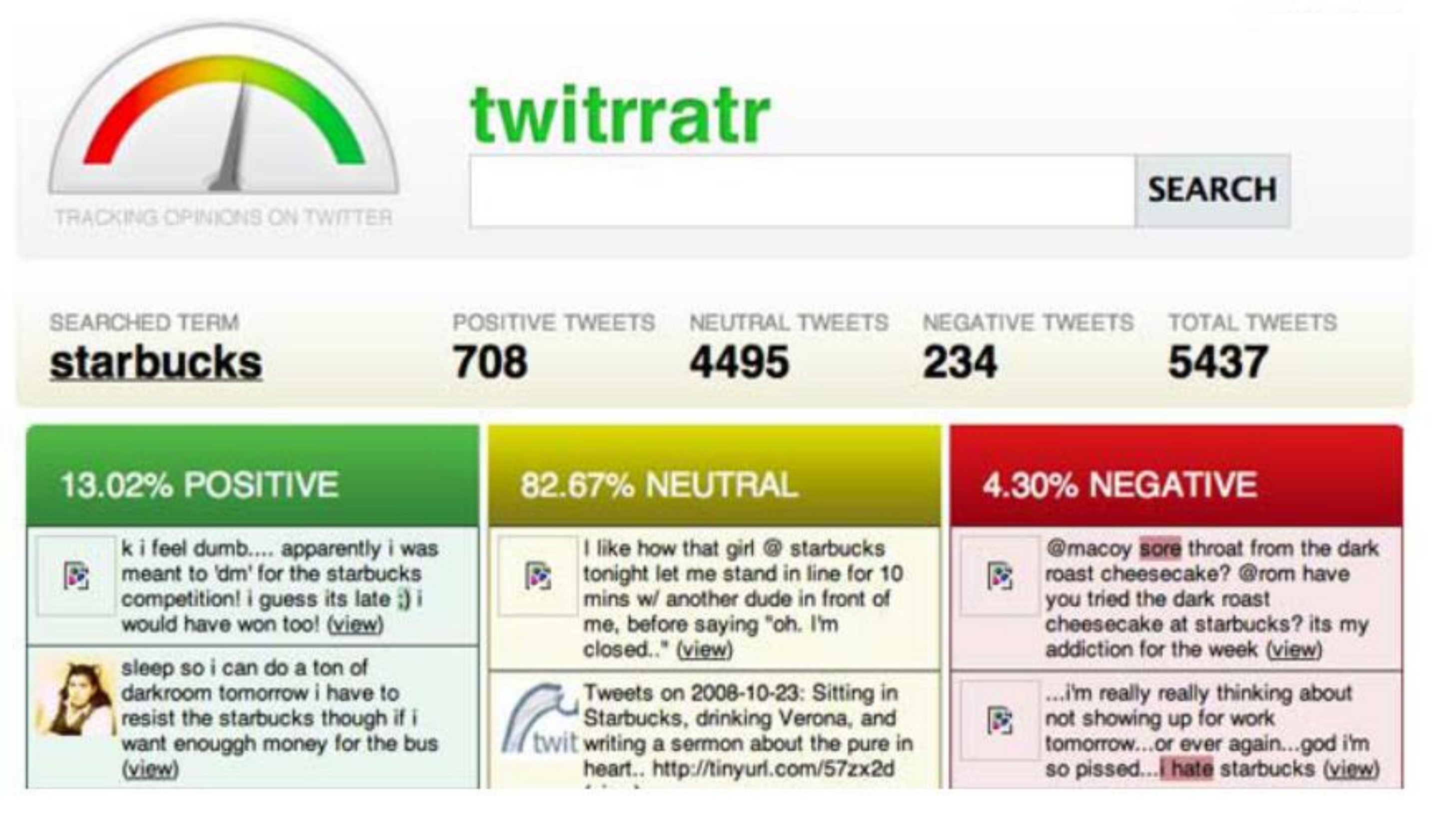
Personal details

Born	Angela Dorothea Kasner 17 July 1954 (age 61) Hamburg, West Germany
Political party	Democratic Awakening (1989–1990) Christian Democratic Union (1990–present)
Spouse(s)	Ulrich Merkel (1977–1982) Joachim Sauer (1998–present)
Alma mater	Leipzig University
Religion	Lutheranism (within Evangelical Church)
Signature	

Introduction to NLP Applications

Sentiment Analysis

- Identifying sentiments and opinions stated in a text



Hybrid NLP Applications

Optical Character Recognition

- Recognizing printed or handwritten texts and converting them to computer-readable texts

Hybrid NLP Applications

Speech recognition

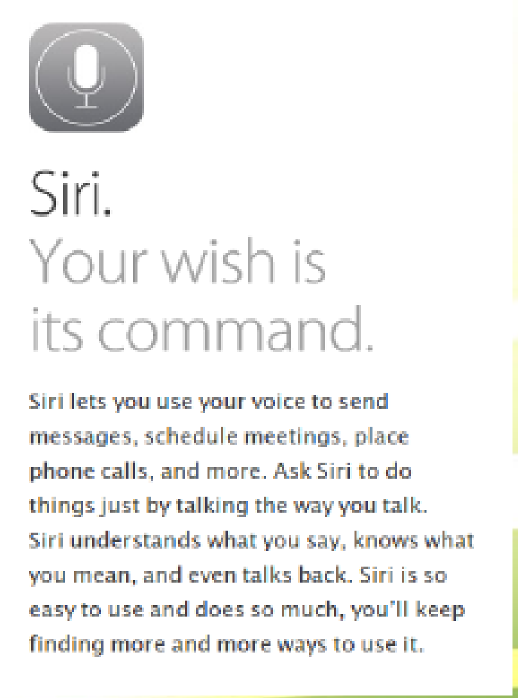
- Recognizing a spoken language and transforming it into a text

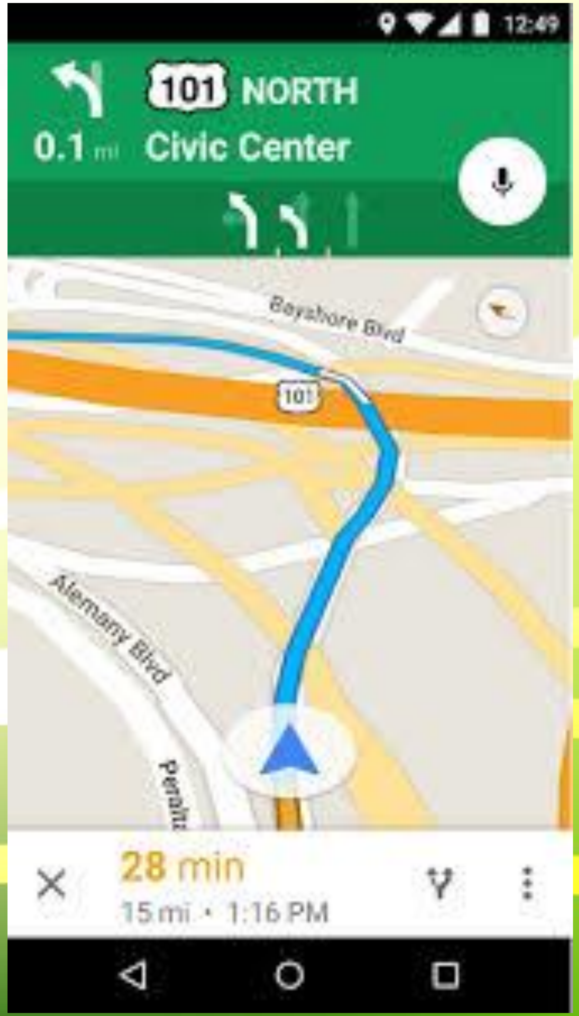
Speech synthesis

- Producing a spoken language from a text

Spoken dialog systems

- Running a dialog between the user and the system





Levels to NLP Applications

Easy (mostly solved)

– Spell and grammar checking

– Some text categorization tasks

– Some named-entity recognition tasks

Intermediate (good progress)

– Information retrieval

– Sentiment analysis

– Machine translation

– Information extraction

Difficult (still hard)

– Question answering

– Summarization

– Dialog systems

Introduction to NLP_Challenges

Paraphrasing

- Different words/sentences express the same meaning
  - – Season of the year
    - Fall
    - Autumn
  - – Book delivery time
    - When will my book **arrive**?
    - When will I **receive** my book?

Introduction to NLP_Challenges

Ambiguity

- One word/sentence can have different meanings
  - – Fall
    - The third season of the year
    - Moving down towards the ground or towards a lower position
  - – The door is open.
    - Expressing a fact
    - A request to close the door

Introduction to NLP_Challenges

Semantics.. Word sense / meaning ambiguity

- The astronomer loves the star.
  - Star in the sky
  - Celebrity

Introduction to NLP_Challenges

Pronoun reference ambiguity

Ambiguous Statements:

- ISIS head seeks arms.
- Local high school dropouts cut in half.
- Army has fired 7 Foot soldiers.
- Safety experts say school bus passengers should be belted.

Language is not static

- Language grows and changes: e.g., cyber lingo

LOL	Laugh out loud
G2G	Got to go
BFN	Bye for now
B4N	Bye for now
Idk	I don't know
FWIW	For what it's worth
LUWAMH	Love you with all my heart

Introduction to NLP_Challenges

Language is compositional

小心:
Carefully
Careful
Take
Care
Caution

地滑:
Slide
Landslip
Wet
Floor
Smooth

Introduction to NLP_Challenges

Scale

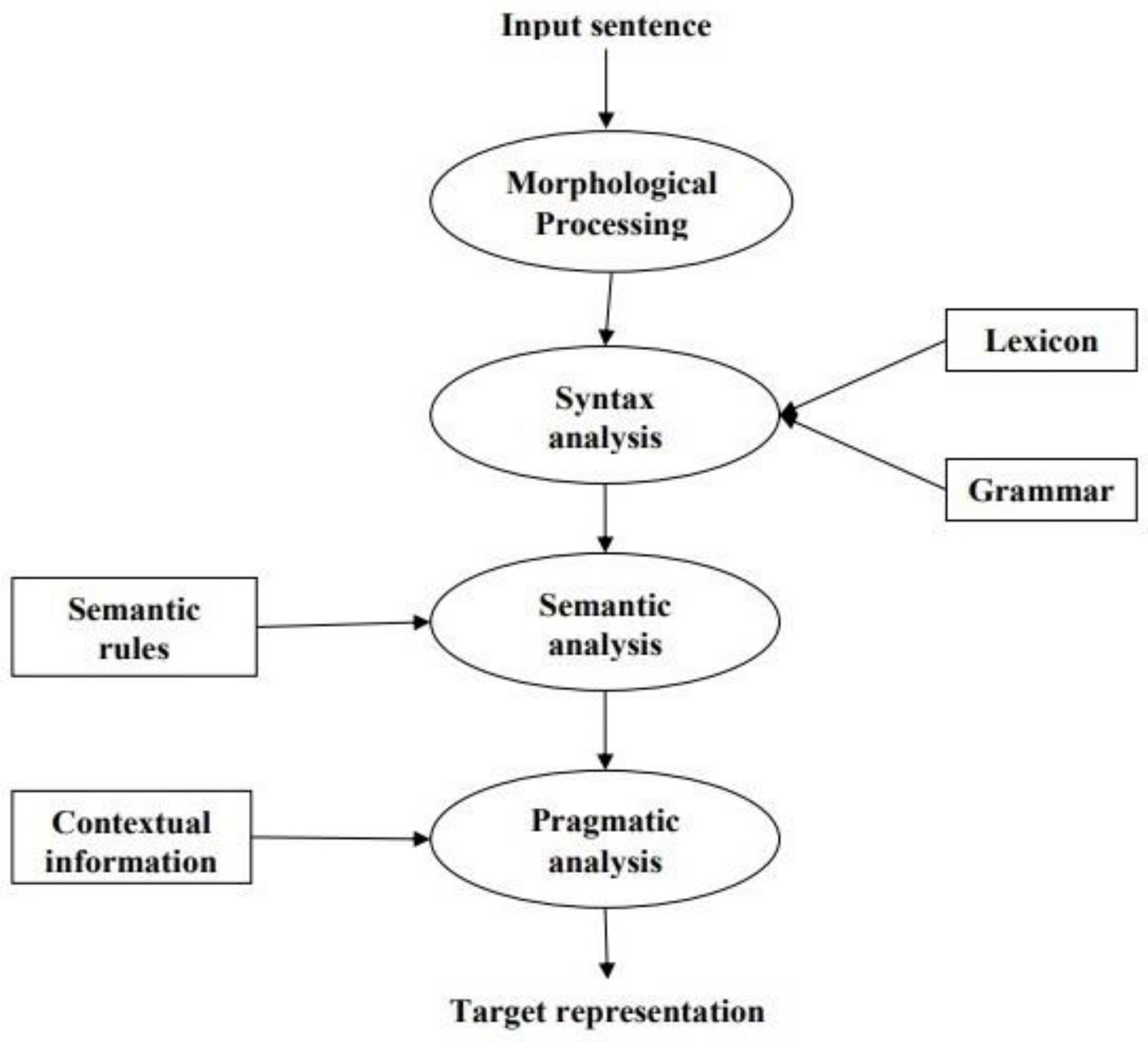
- Bible (King James version): ~700K
- Penn Tree bank ~1M from Wall street journal
- Newswire collection: 500M+
- Wikipedia: 2.9 billion word (English)
- Web: several billions of words

Introduction to NLU

- Natural Language Understanding or interpretation (NLU/NLI)
- Subtopic of natural-language processing in artificial intelligence
- Deals with machine reading comprehension.
- NLU is considered an AI-hard problem.
- **AI-complete** or **AI-hard** are the most difficult problems in AI.
- Required for NLP; *automated reasoning, machine translation, question answering, news-gathering, text categorization, voice-activation, archiving, and large-scale content analysis* etc.

Introduction to NLP _ Explained

Phases or logical steps in NLP



Introduction to NLP _ Explained

Morphological Processing: First phase of NLP.

- Break chunks of language input into sets of tokens corresponding to paragraphs, sentences and words.
- N-gram, Unigram, Bigram, Trigram, etc.
- Ex, word "uneasy" can be broken into two **sub-word tokens** as "un, easy".

Syntax Analysis: Second phase of NLP.

- Check sentence is well formed or not, and break it up into a structure that shows the syntactic relationships between different words.
- Ex, sentence "The school goes to the boy" would be rejected by syntax analyzer or parser.

Introduction to NLP _ Explained

Semantic Analysis: Third phase of NLP.

- Draw exact (dictionary) meaning from text.
- Text is checked for meaningfulness.
- Ex, semantic analyzer would reject sentence “Hot ice-cream”.

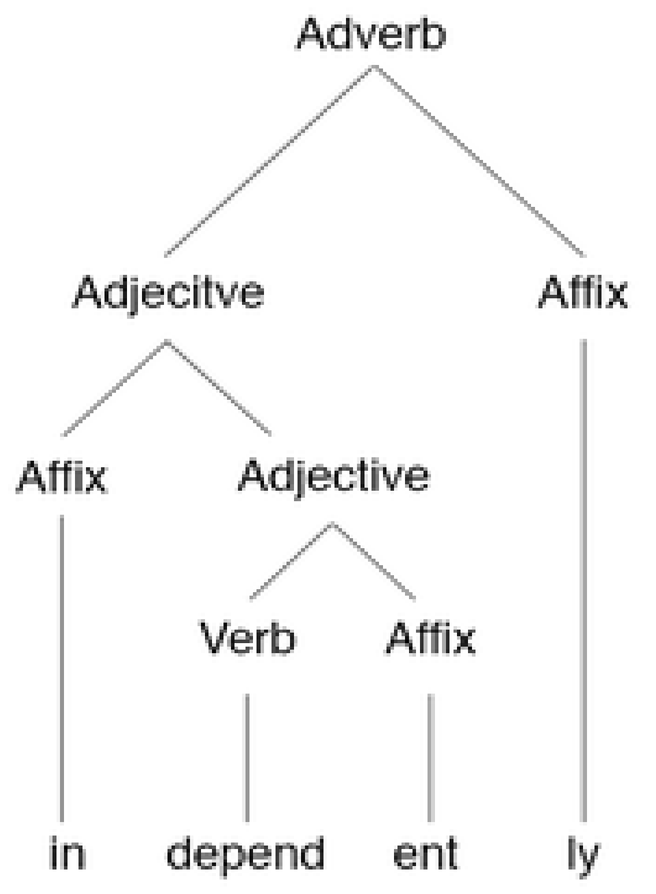
Pragmatic Analysis: Fourth phase of NLP.

- Simply fits the actual objects/events, which exist in given context with object references obtained during the previous phase (semantic analysis).
- Ex, sentence “Put the banana in the basket on the shelf” can have two semantic interpretations and pragmatic analyzer will choose one between these two possibilities.

Introduction to NLP _ Explained

Morphological Parsing

- **Morphology:** Study of internal structures of words and how they can be modified
- **Parsing**: Breaking complex words into their smaller meaningful components
- In linguistics, morphology is the study of words, how they are formed, and their relationship to other words in the same language.
- Analyzes structure of words and parts of words (**stems, root words, prefixes**, and **suffixes**)



Introduction to NLP _ Explained

MORPHEME = smallest meaningful unit of language *(any part of a word that cannot be broken down further into smaller meaningful parts, including the whole word itself).*

- Word "**items**" can be broken down into two meaningful parts: '**item**' and plural suffix '**-s**';
- Neither of these can be broken down into smaller parts that have a meaning.
- Therefore 'item' and '-s' are both morphemes.
- Word "items" includes "it" – which is spelled same as word "it", which is a meaningful unit of language.
  - But "it" in "items" is used differently, and does not in any way share function or distribution of word "it."

Introduction to NLP _ Explained

- **FREE MORPHEME** = a morpheme that can stand alone as an independent word (e.g. 'item').
- **BOUND MORPHEME** = a morpheme that cannot stand alone as an independent word, but must be attached to another morpheme/word *(plural '-s', are always bound)*
- **AFFIX** = a bound morpheme which attaches to a base (root or stem).
- **PREFIXES** attach to the front of a base;
- **SUFFIXES** to the end of a base;
- **INFIXES** are inserted inside of a root.

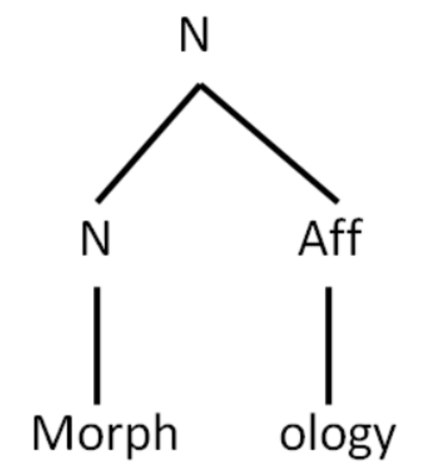
Example: Prefix: 're-' of 'rewrite'; || Suffix: '-al' of 'critical';

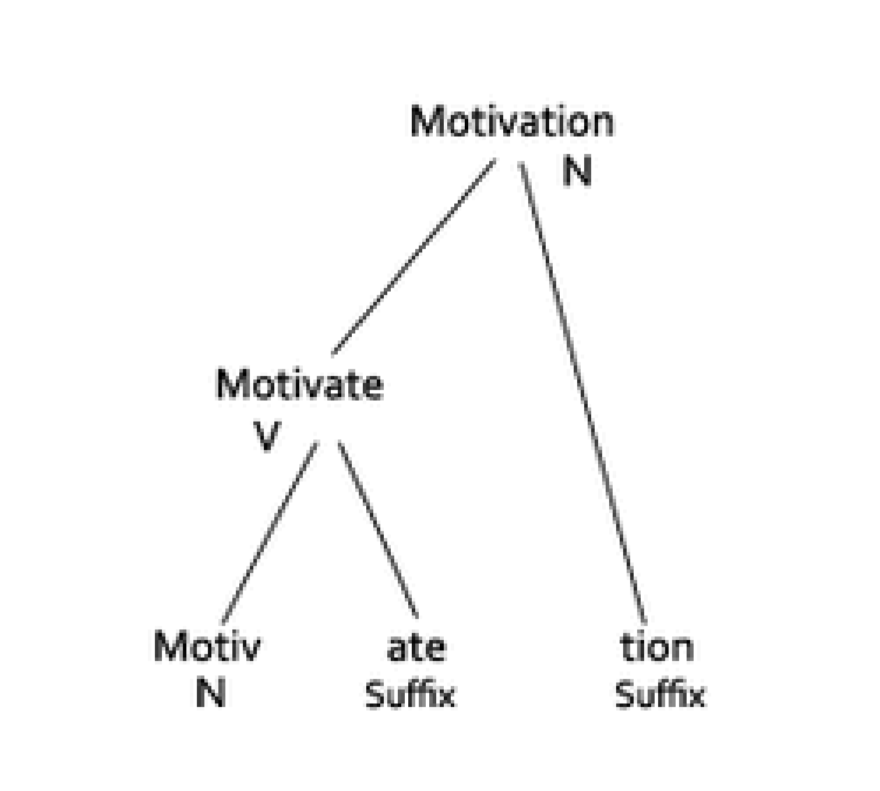
- Infix: (rare in English) is the morpheme “freaking” in the slang term “abso-freaking-lutely” or “in-freakingcredible”

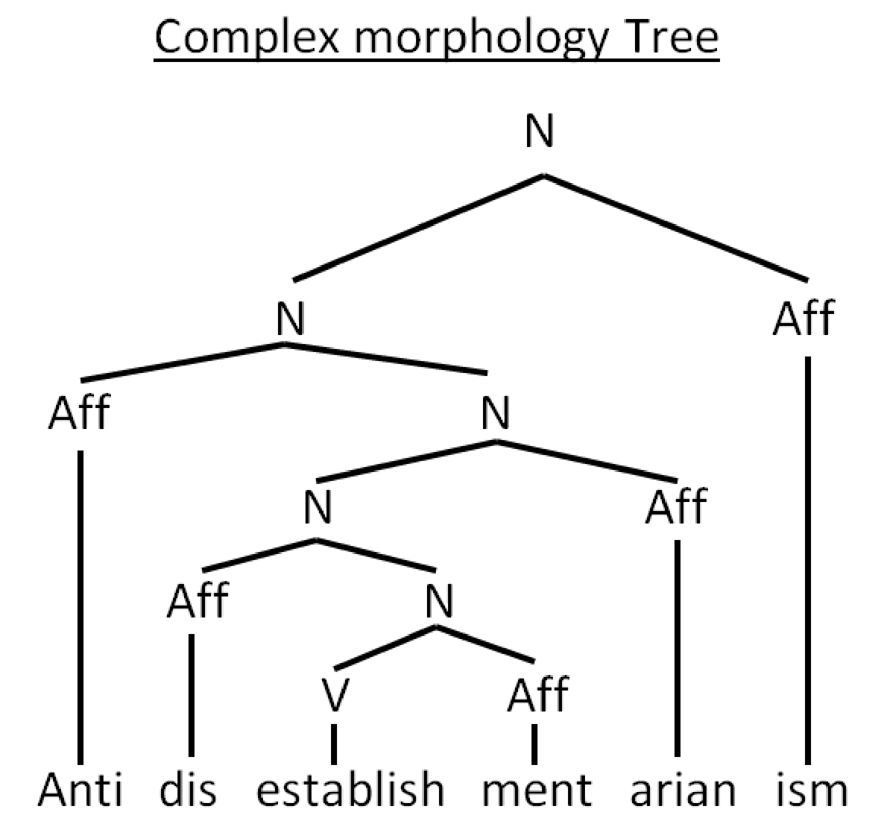
Introduction to NLP _ Explained

Morphology

- study of internal structures of words





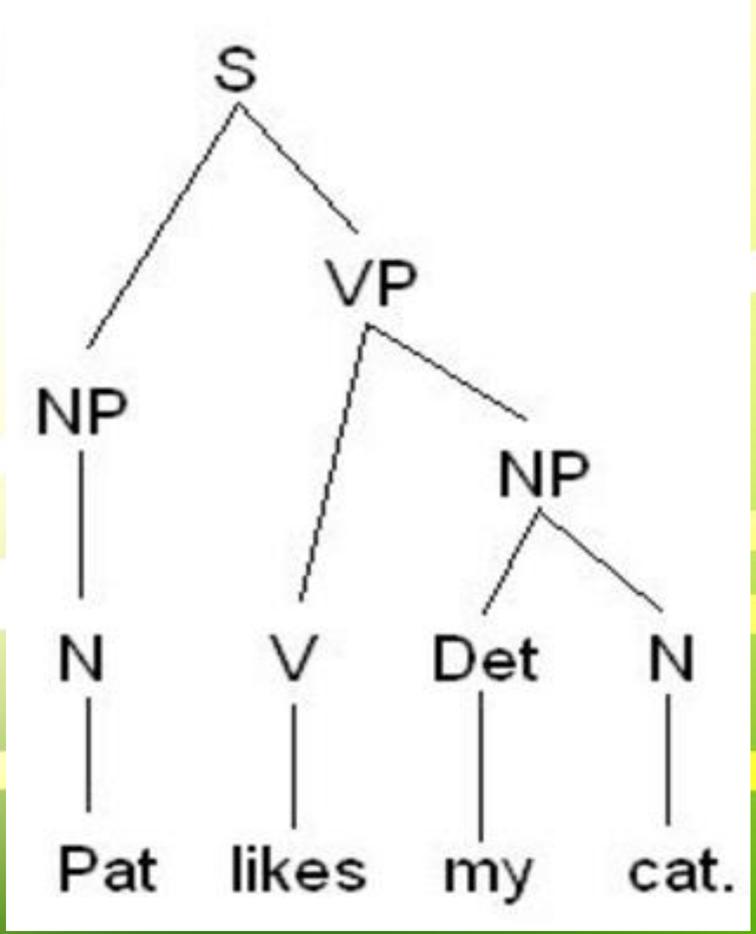


Introduction to NLP _ Explained

Syntax: Study of structural relationships between words in a sentence.

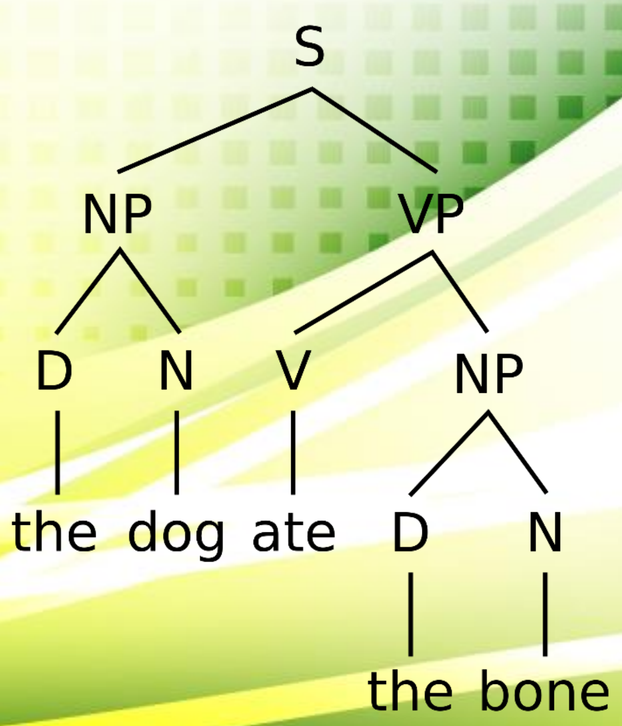
- S- Stem
- N- Noun
- V- Verb
- NP- Noun phrase / pronoun
- VP- Verb phrase
- D- Determiner / determinative

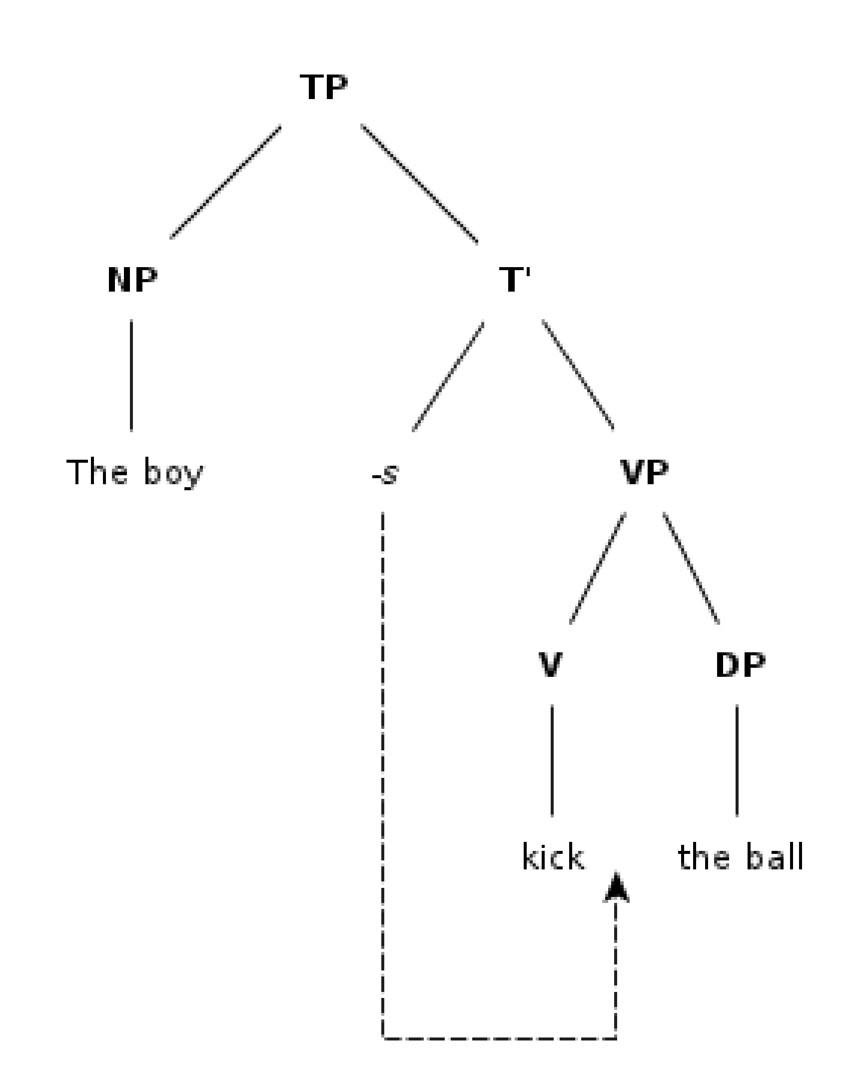
  (a, the, my, some)
- DP- Determiner phrase
- Adv- Adverb
- AdvP- Adverb phrase



Introduction to NLP _ Explained

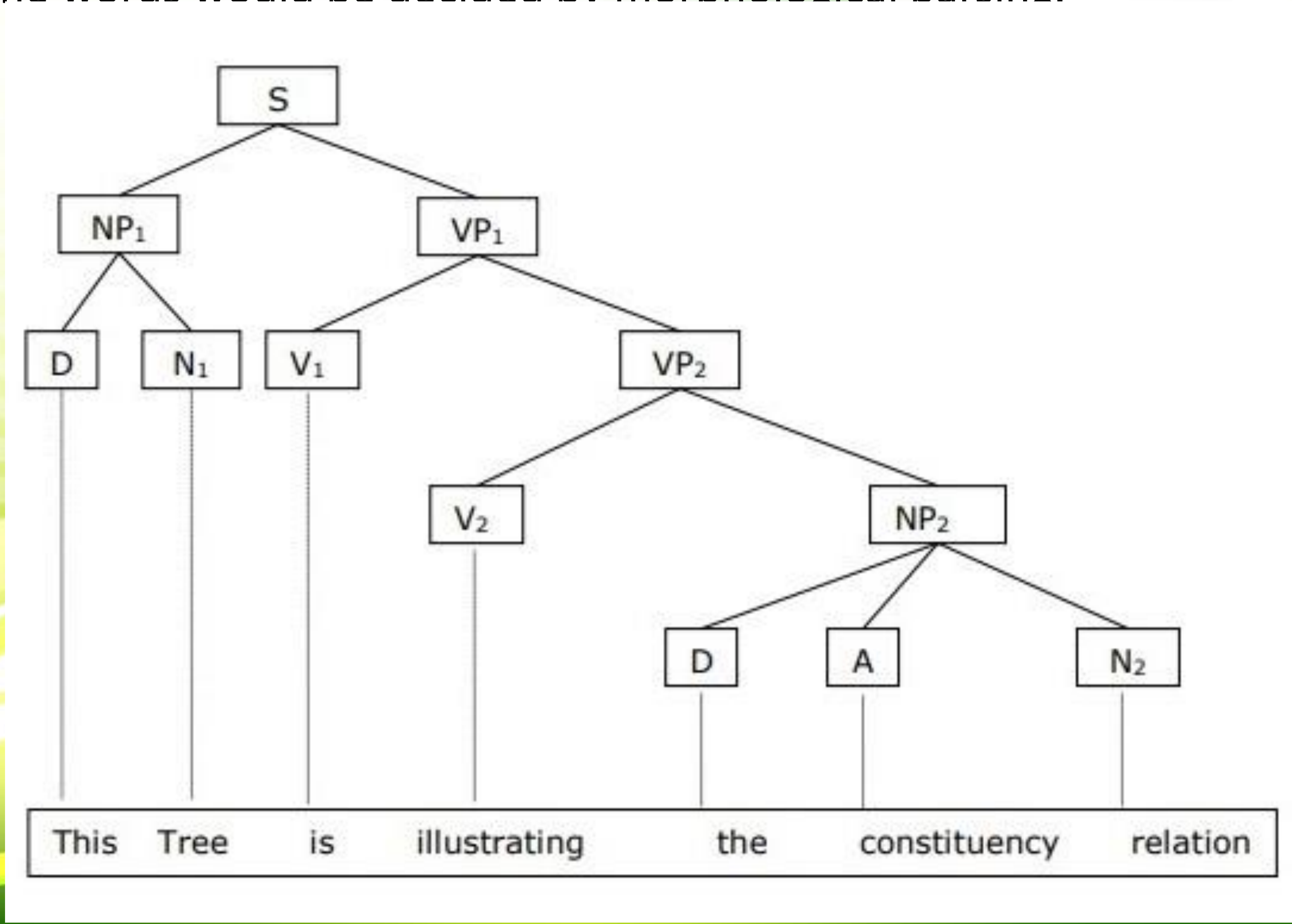
Syntax: Study of the structural relationships between words in a sentence



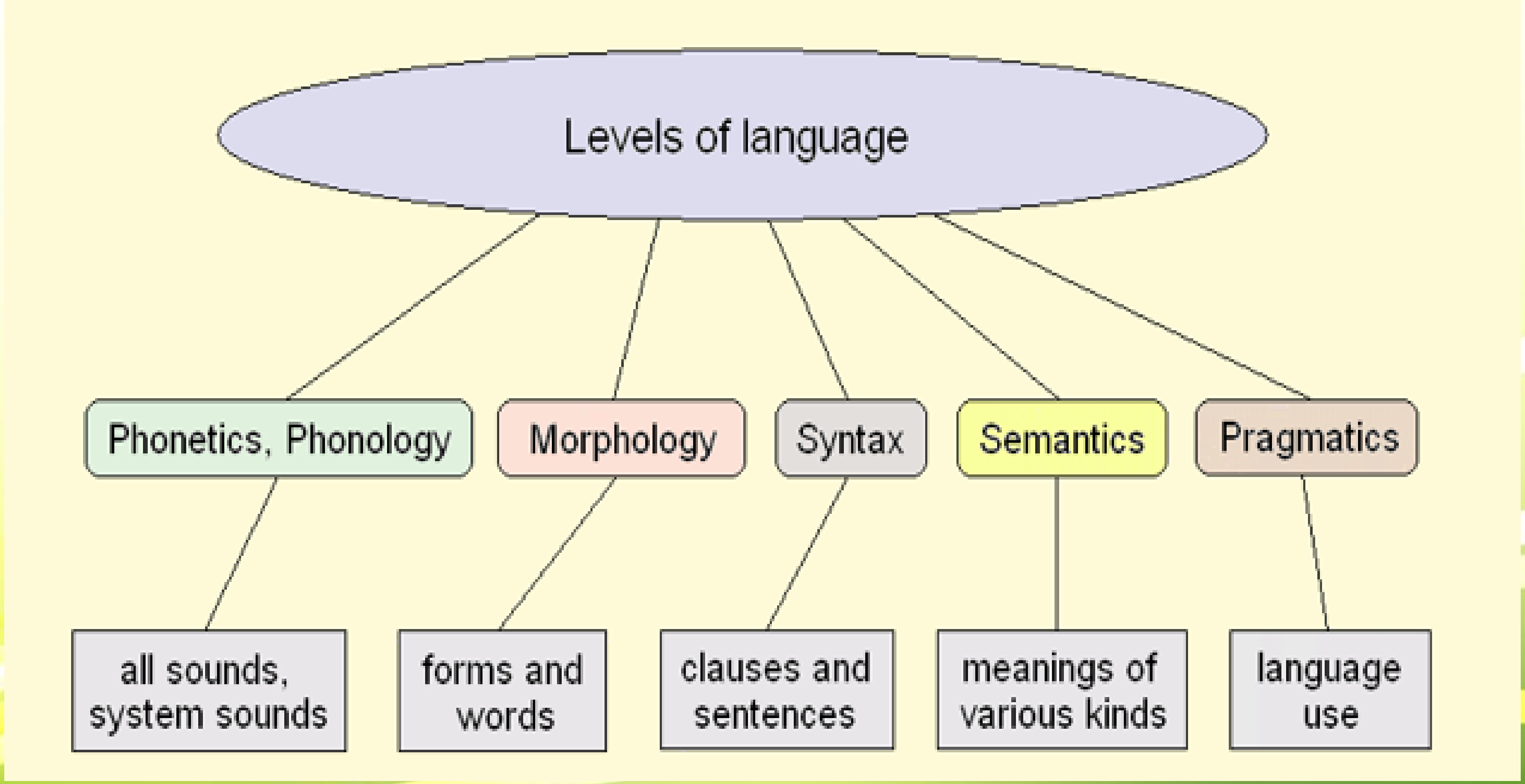


Introduction to NLP _ Explained

Word Order: Order of the words would be decided by morphological parsing.



Introduction to NLP _ Explained



Introduction to NLP _ Explained

Language Processing

- *Level 1* – Speech sound (*Phonetics & Phonology*)
- *Level 2* – Words & their forms (*Morphology, Lexicon*)
- *Level 3* – Structure of sentences (*Syntax, Parsing*)
- *Level 4* – Meaning of sentences (*Semantics*)
- *Level 5* – Meaning in context & for a purpose (*Pragmatics*)
- *Level 6* – Connected sentence processing in a larger body of text (*Discourse*)

Introduction to NLP _ Explained

Phonetics and phonology: Study of linguistic sounds and their relations to words

- *Phonetics:* Study of human sounds.
- *Phonology:* Study of the sound system of a language or languages.

Semantics: Study of meaning of words, and how these combine to form meanings of sentences.

– Hypernymy & hyponymy (is a): animal & dog

– Meronymy (part of): finger & hand

– Homonymy: fall (verb & season)

– Synonymy: fall & autumn

– Antonymy: big & small

Introduction to NLP _ Explained

Pragmatics

- Social use of language
- Study of how language is used to accomplish goals, and influence of context on meaning
- Understanding the aspects of a language which depends on situation and world knowledge

Give me the salt!

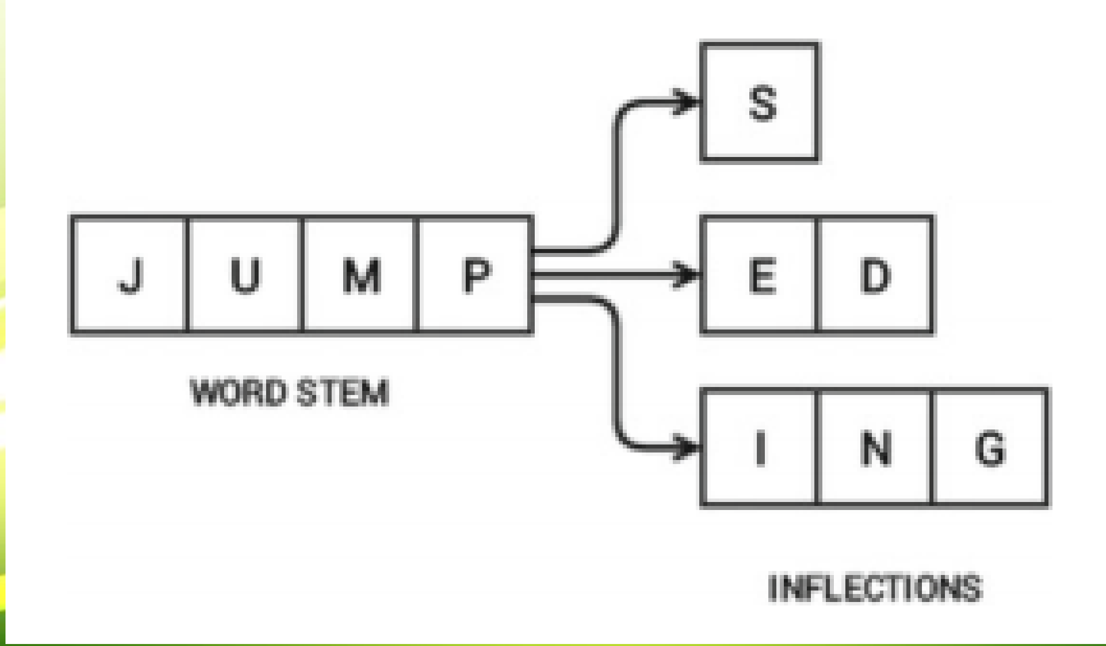
Could you please give me the salt?

Discourse: Study of linguistic units larger than a single statement

Introduction to NLP _ Explained

Stemming

- Reverse process of obtaining base form of a word from its inflected form.
- Stemming helps in standardizing words to their base or root stem, irrespective of their inflections.
- Helps in many applications like classifying or clustering text, and even in information retrieval.



Introduction to NLP _ Explained

Lemmatization

- Very similar to stemming.
- Remove word affixes to get to base form of word.
- Base form in this case is known as root word, but not root stem.
- Root word is always a lexicographically correct word (present in dictionary).
- Root stem may not be so.
- Root word (lemma) will always be present in dictionary.

Word	***Lemmatization***	***Stemming***
was	be	wa
studies	study	studi
studying	study	study

NLP _ Techniques

Stemming and Lemmatizing: Reducing different forms of word to a common base form. E.g. In "*I am a student*"; the process would result in "*I be a student*"

$$\text{I } \textbf{am} \text{ a student} \rightarrow \{\,[\,\text{am} \mid \text{are} \mid \text{is}\,] \Rightarrow \text{be}\} \rightarrow \text{I } \textbf{be} \text{ a student}$$

In "*My dog's fur is dark*"; the process would result in "*My dog fur be dark*".

$$\text{My } \textbf{dog's} \text{ fur is dark} \rightarrow \{\,[\,\text{dog} \mid \text{dogs} \mid \text{dog's} \mid \text{dogs'}\,] \Rightarrow \text{dog}\} \rightarrow \text{My } \textbf{dog} \text{ fur is dark}$$

- ***Stemming***- crude process that chops off the ends of words for achieving this goal.
- The obtained element is known as the **stem**.

NLP _ Techniques

- ***Lemmatization-*** doing things properly with use of vocabulary and morphological analysis of words, to return the base or dictionary form of a word (**lemma)**.
- Process of grouping together the inflected forms of a word so they can be analysed as a single item

$$\text{I } \textbf{saw} \text{ an amazing thing} \xrightarrow{\text{lemma}} \text{I } \textbf{see} \text{ an amazing thing}$$

- Both techniques may remove important information, but also help to normalize corpus.

Introduction to NLP_Techniques

Part-of-speech tagging

- Determine part of speech for each word by assigning syntactic category in sentence.
- Many (common) words can serve as multiple parts of speech.

  E.g. "book" can be a noun ("the book on the table") or verb ("to book a flight").

- Syntactic category;
  - *Closed class types*: **preposition**, **determiner**, **pronoun**, **conjunction**, **auxiliary verb**, **particle**, **numeral**.
  - *Open class categories/types* (relatively fixed)

    noun, **verb**, **adjective**, **adverb**.

"I want to play the piano"

```
I  (Preposition)

want  (Verb)

play  (Verb)

piano  (Noun)
```

NLP _ Techniques

Morphological segmentation

- Separate words into individual morphemes and identify class of the morphemes.
- Difficulty depends greatly on complexity of the morphology (i.e. structure of words) of the language being considered.

Word segmentation

- Separate a chunk of continuous text into separate words.
- English language is very easy, since words are separated by spaces.
- Some written languages (Chinese, Japanese, Thai) don't mark word boundaries in such a fashion, and hence text segmentation is a significant task.
- Requires knowledge of vocabulary and morphology of words in the language.

NLP _ Techniques

Sentence breaking (***sentence boundary disambiguation***)

- Given a chunk of text, find the sentence boundaries.
- Often marked by periods or other punctuation marks, but these same characters can serve other purposes.

xyz@gmail.com

NLP _ Techniques

Parsing

- Determine parse tree (grammatical analysis) of a given sentence. Grammar for natural languages is ambiguous. typical sentences have multiple possible analyses.
- Two types of parsing.
  - **Dependency Parsing** focuses on relationships between words in a sentence (marking things like Primary Objects and predicates).
  - **Constituency Parsing** focuses on building out Parse Tree using a Probabilistic Context-Free Grammar (PCFG).

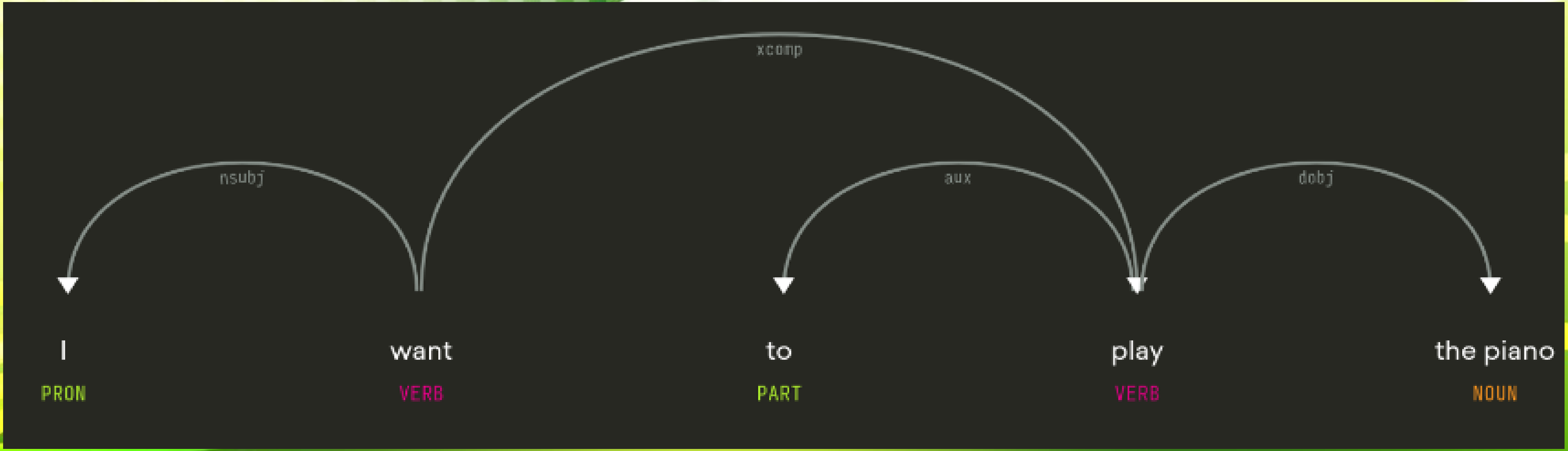
NLP _ Techniques (Parsing)

- Dependencies general

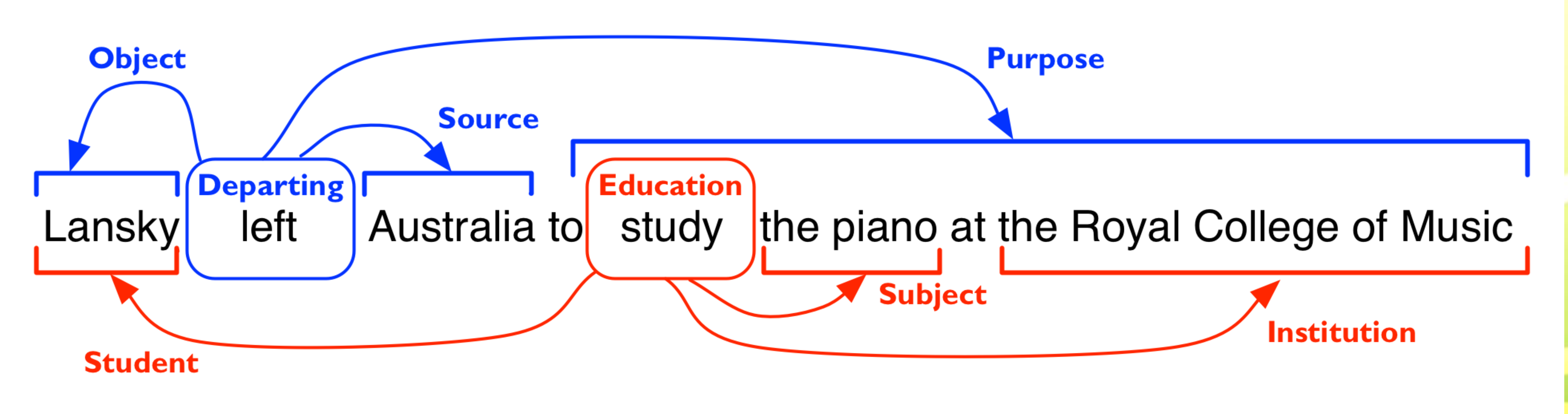
  terms: **subject**, **object**, **complement** and **modifier** relations.

  “I want to play the piano”

```
want — I: nominal subject.

want — play: open clausal complement.

play — to: auxiliary verb.

play — the piano: direct object
```



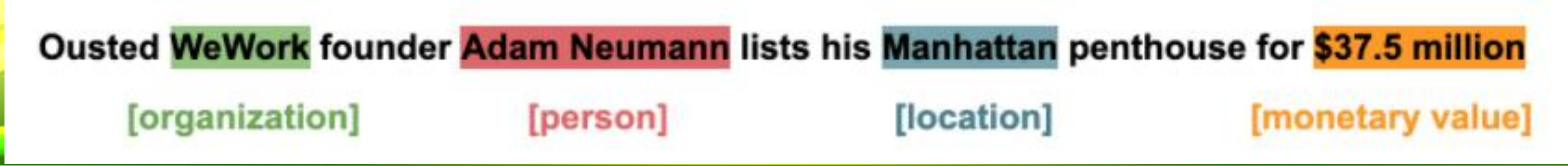
NLP _ Techniques (Parsing)



NLP _ Techniques

Named Entity Recognition (NER)

- Entity chunking, extraction, or identification
- Task of identifying and categorizing key information (entities) in text.
- NER is a form of natural language processing.
- An entity can be any word or series of words that consistently refers to the same thing.
- Every detected entity is classified into a predetermined category.



NLP _ Techniques

Named Entity Recognition (NER)

- NER model is a two step process:
  1. **<u>Detect a named entity</u>**
     - Detecting a word or string of words that form an entity.
     - Each word represents a token:
     - “The Great Lakes” is a string of three tokens that represents one entity.
  2. **<u>Categorize the entity</u>**
     - Creation of entity categories.
- Some common entity categories:
  - Person, Organization, Time, Location, Work of art, and lot more.

NLP _ Techniques

Language Modelling (LM)

- Use of various statistical and probabilistic techniques to determine probability of a given sequence of words occurring in a sentence.
- LM analyzes bodies of text data to provide a basis for their word predictions.
- Popularly used in machine translation and question answering, etc.
- LM determines word probability by analyzing text data.
  - Interpret this data by feeding it through algorithm that establishes rules for context in natural language.
  - Then, the model applies these rules in language tasks to accurately predict or produce new sentences.
  - Model learns the features and characteristics of basic language and uses those features to understand new phrases.

NLP _ Techniques

Unigram

- Unigram is the simplest type of language model.
- Doesn't look at any conditioning context in its calculations.
- Evaluates each word or term independently.
- Handles language processing tasks such as information retrieval.

NLP _ Techniques

N-gram

- N-grams are relatively simple approach to language models.
- Create probability distribution for a sequence of n entities.
- n can be any number, and defines size of the "gram", or sequence of words being assigned the probability.
  - If n = 5, a gram might look like this: "can you please call me."
- The model then assigns probabilities using sequences of n size.
- Some types of n-grams are **unigrams, bigrams, trigrams** and so on.

NLP _ Techniques

Bidirectional

- n-gram models analyze text in one direction (backwards).
- Bidirectional models analyze text in both directions, backwards and forwards.
- Predict any word in a sentence or body of text by using other word in the text (in bidirectional).
- Examining text bidirectionally increases result accuracy.
- Popularly used in machine learning and speech generation applications.

NLP _ Techniques

Machine Translation (MT)

- Use of software to translate text or speech from one language to another.
- MT performs mechanical substitution of words in one language for words in another.
  - Not sufficient to produce good translation,
  - Recognition of whole phrases and their closest counterparts in the target language is also needed.
  - All words in one language do not have equivalent words in another language,
  - Many words have more than one meaning.
  - Sentence structure in different languages need not be same.

NLP _ Techniques

Machine Translation (MT)

- Language translation (normally by Human) follows two steps;
  - Decoding the meaning of the source text;
  - Re-encoding this meaning in the target language.
- While decoding translator must interpret and analyse all the features of the text
- In-depth knowledge of the grammar, semantics, syntax, idioms, etc. are required for the source language.
- Same level of in-depth knowledge in target language needed to re-encode the meaning.
- MT approached;
  - Rule-based, Statistical (SMT), Example-based, Hybrid (HMT), Neural (NMT).

NLP _ Techniques

Machine Transliteration

- Process of automatically transforming ***(NOT translating)*** the script of a word from source language to target language, while **preserving pronunciation**.

- Preserves the phonetic and orthographic aspects of the transliterated words.

- Dilwale Dulhania Le Jayenge → Hindi to English Transliteration
- दिलवाले दुल्हनिया ले जायेंगे → Original in Hindi
- Hearted will take the bride → Hindi to English Translation

NLP _ Techniques

Code Mixing (Code switching)

- Use of multiple (generally two) different grammatical languages using one single (Roman/English) alphabet writing system.
- Common practice among the modern social media users.
- Popularly practices by groups that do not share a common language; may occur within a multilingual setting where speakers share more than one language.
- Such a complicated practice makes the context parsing, sentiment analysis and machine translation even more challenging.
    - Let's eat the homecooked chawal. → Hinglish; in common use
    - Let's eat the homecooked rice. → Actual English representation

NLP _ Techniques

Code Mixing (Code switching)

- Challenges
  - Identifying languages
  - Differentiating which entity in which language
  - Use of transliteration (sometimes)
  - Spelling
  - Grammar

NLP _ Techniques

TF-IDF

- Statistical measure that evaluates how relevant a word is to a document in a collection of documents.
- Two metrics considered:
  - how many times a word appears in a document (TF), and
  - inverse document frequency of the word across a set of documents (IDF).
- Popularly used for document search and information retrieval.

NLP _ Techniques

TF-IDF

- Increases proportionally to the number of times a word appears in a document,
- but offset by the number of documents that contain the word.

➔ words that are common in every document (this, what, if, etc) rank very low even though they may appear many times, since they don’t mean much to that document in particular.

- If the word appears many times in a document, while not appearing many times in others, it probably means that it’s very relevant.
- TF-IDF score for the word “t” in document “d” from the document

$$tf\,idf(t, d, D) = tf(t, d) \,.\, idf(t, D)$$

$$tf(t, d) = \log(1 + freq(t, d))$$

$$idf(t, D) = \log\left(\frac{N}{count(d \in D : t \in d)}\right)$$

NLP _ Techniques

Word Embedding

- A type of word representation that allows words with similar meaning to have a similar representation.
- Allows words that are used in similar ways to result in having similar representations.
- Contrasting to fragile representation in a bag of words model where, unless explicitly managed, different words have different representations, regardless of how they are used.

Word2Vec

- Word2Vec is one of the most popular technique to learn word embeddings using shallow neural network.
- Developed at Google (2013).

NLP _ Techniques

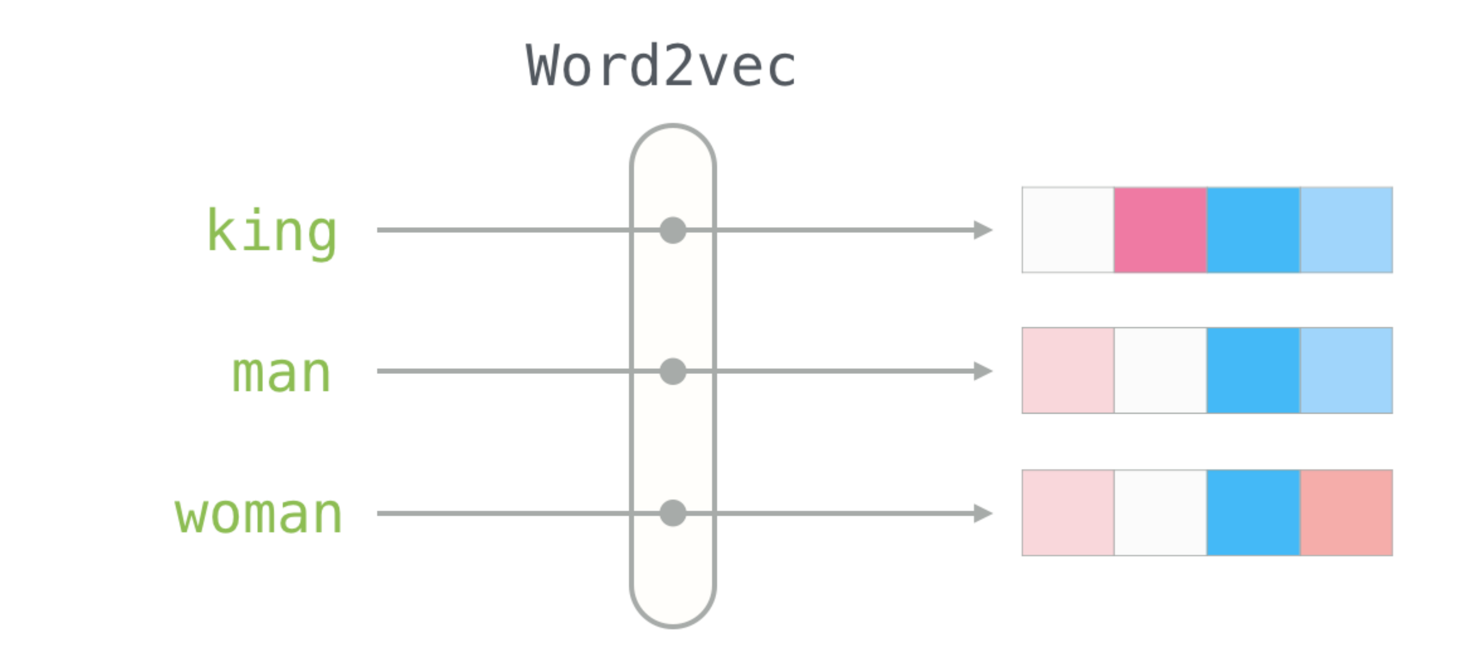
Word2Vec

- Two similar sentences:
  - Have a good day and
  - Have a great day.
- Vocabulary (V) = {Have, a, good, great, day}.
- One-hot encoded vector representation for each words in V.
- Have = [1,0,0,0,0]; a=[0,1,0,0,0] ; good=[0,0,1,0,0] ; great=[0,0,0,1,0] ; day=[0,0,0,0,1]
- Visualize these encodings in 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest.

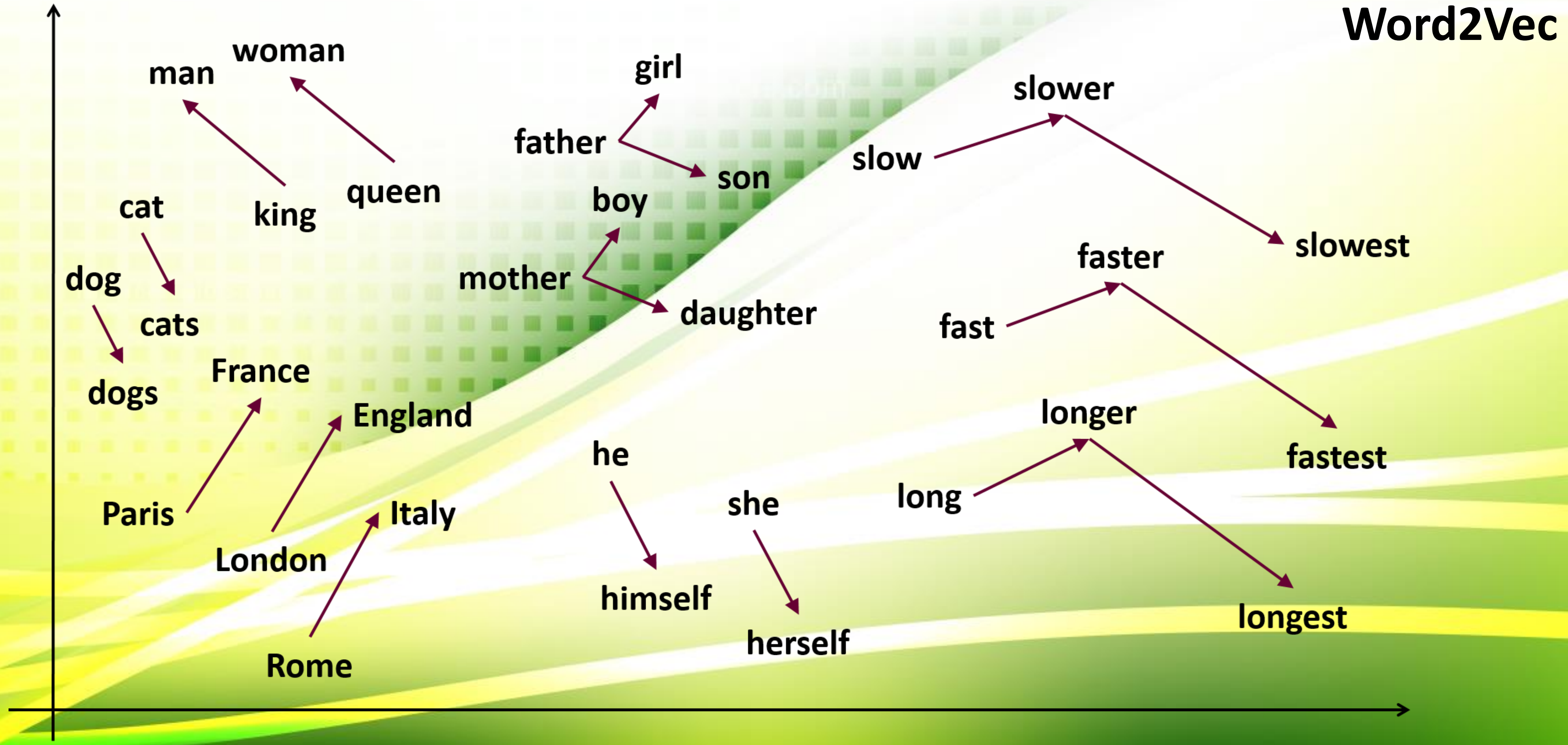
  → This means 'good' and 'great' are as different as 'day' and 'have', which is not true.
- Words with similar context should occupy close spatial positions.

NLP _ Techniques

Word2Vec



NLP _ Techniques



NLP _ Techniques

- GloVe (Global Vector)
  - A model for distributed word representation
- LSTM (Long Short-Term Memory)
  - A type of recurrent NN (RNN) capable of learning order dependence in sequence prediction problems.
- BERT (Bidirectional Encoder Representations from Transformers)
  - A Transformer-based ML technique for NLP pre-training developed at Google (2018).
  - ALBERT
  - ROBERT
  - DISTILLBERT
- GPT-n (Generative Pre-Training 2, 3)
  - An autoregressive language model that uses deep learning to produce human-like text.

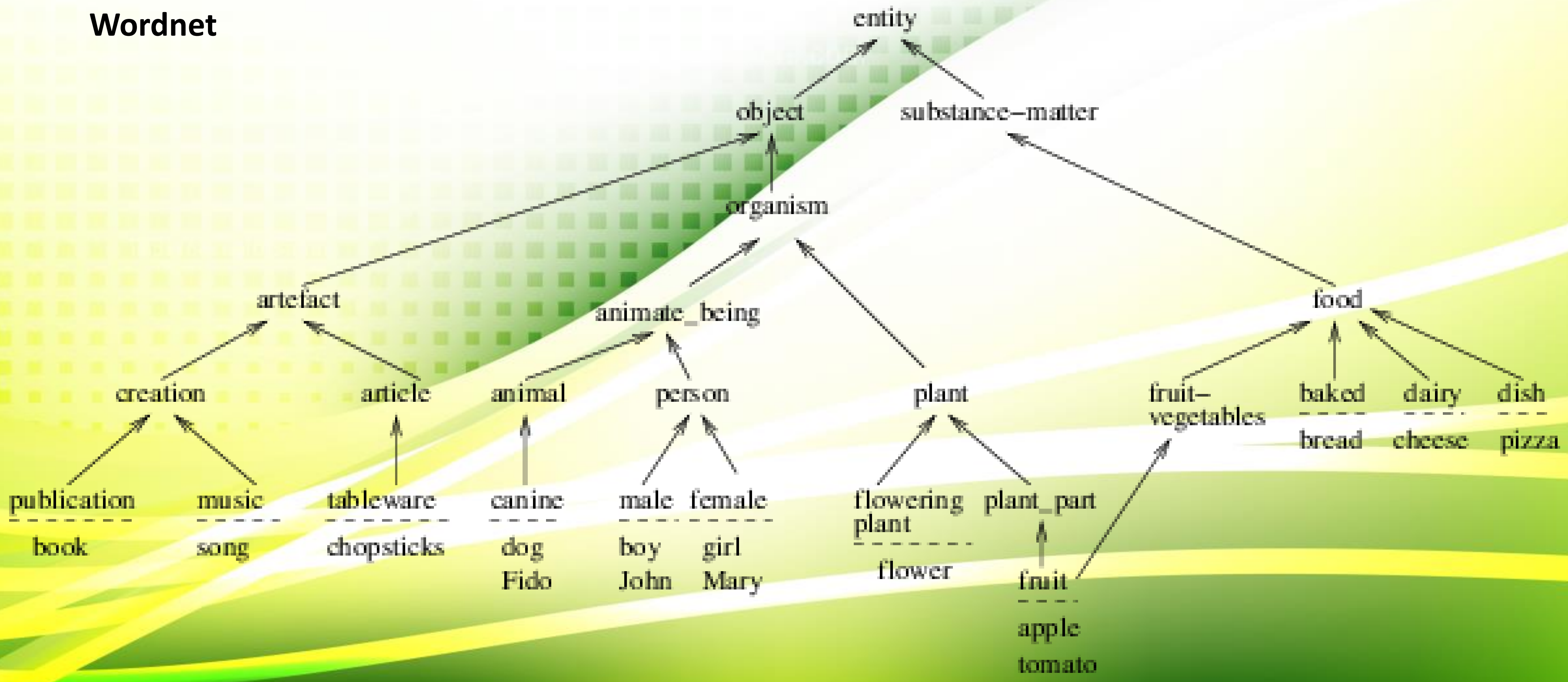
NLP _ Techniques

Wordnet

- A lexical database of semantic relations between words in more than 200 languages.
- Links words into semantic relations including synonyms, hyponyms (sub/super type), and meronyms (part of relationship).
- WordNet superficially resembles a thesaurus → groups words together based on their meanings.
- Important distinctions:
  - WordNet interlinks not just word forms—strings of letters—but specific senses of words
  - WordNet labels the semantic relations among words.

NLP _ Techniques

Wordnet



NLP _ Techniques

Word Cloud (Tag cloud)

- A visual representation of text data.
- Used to depict keyword metadata (tags) on websites, or to visualize free form text.
- Useful for quickly perceiving the most prominent terms to determine its relative prominence.
- Tags are usually single words, and importance of each tag is shown with font size or color.
- Bigger term means greater weight.

NLP _ Techniques

Word Cloud (Tag cloud)