HMM

Eg. <s> martin justin can watch will <e>
<s> Spot will watch Martin <e>
<s> will justin spot Martin <e>
<s> Martin will pat spot <e>

n: noun, m: model verb, v: verb.

→ (1) create Emission table. word : tag

<s>	martin N.	justin N	can M	watch V.	will N	<e>
<s>	Spot N	will M	watch V	martin N		<e> 4
<s>	will M	justin N	spot V	martin N		<e>
<s>	martin N.	will M	pat V	Spot N.		<e>.

	N	M	V
martin	4		
justin	2		
will	1	3	
Spot	2		1
can		1	
watch			2
pat			1
Σ	9	4	4

$$P(\text{justin} \mid n) = \frac{2}{9}.$$

(2) Create state Tran^n. Tag : Tag.

	<s>	N	M	V	<e>
<s>	X	3	1		X
N	X	1	3	1	4
M	X	1		3	
V	X	4			
<e>	X	X	X	X	X

∴ # <s> = 4
∴ # N = 9
∴ # M = 4
∴ # V = 4

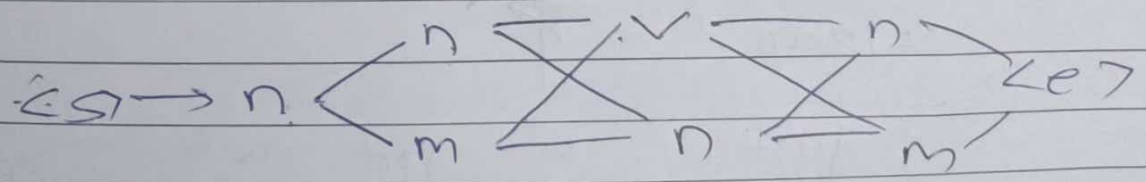
$P(n|s) = S,n = \frac{3}{4}$.

(3) Test Data
<s> justin will spot will <e>

justin	will	spot	will
n	n	v	n
	m	n	m

← Ambiguity

$1 \times 2 \times 2 \times 2 \times 1$
∴ Total ways : 8.



S, n, justin → $P(\text{justin}|n) * P(n|s)$.
1 2 3 3/2 2/1
$= \frac{2}{9} * \frac{3}{4} = \frac{1}{6}$.

=

will as MV(m).

n, will = (will|m) * (n,m)
1 2 3
$= \frac{3}{4} * \frac{3}{9} = \frac{1}{4}$

$= \frac{1}{4} * \frac{1}{6} = \frac{1}{24}$ ✓

will as noun

n, will = (will|n) * (n,n)
$\frac{1}{9} * \frac{1}{9} = \frac{1}{81}$

$\frac{1}{6} * \frac{1}{81} = \frac{1}{486}$

∴ will becomes modal verb

Spot is a verb

∴ m → V → spot

$\frac{3}{16}$

↓

$\frac{1}{128}$ ✓

Spot is a noun

m → N → spot = spot|n * f(n)

$= \frac{2}{9} * \frac{1}{4}$

$\frac{1}{18}$

↓

$\frac{1}{432}$

will is a MV

V → m → will

$\frac{3}{4} * (v,m)$

↓

laplace smoothing.

$\therefore \frac{3}{4} * \frac{1}{1000} = \frac{3}{4000}$

↓ * $\frac{1}{128}$.

$\frac{1}{512000}$.

will is a n

V → n → will.

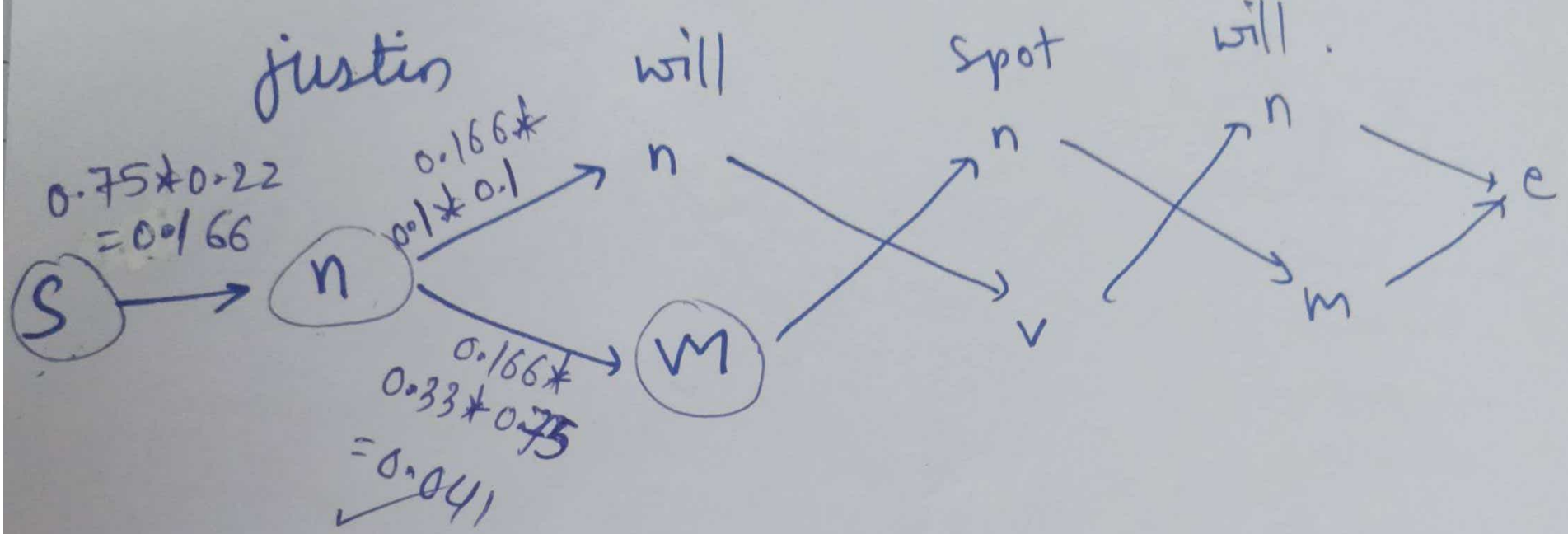
$\frac{1}{9} * \frac{1}{1} = \frac{1}{9}$

↓ * $\frac{1}{128}$.

$\frac{1}{1152}$

∴ justin will spot will

n m v n


justin
will
Spot
will .
0.75*0.22
=0.166
S
n
0.166*
0.1*0.1
n
0.166*
0.33*0.75
=0.041
M
n
v
n
m
e

exp means e raised to

Max. Entropy model

$$p(\text{spam}|x) = \frac{\text{spam}}{\text{total}} = \frac{\exp(0.6+0.7+0.5)}{\exp(0.6+0.7+0.5)+\exp(-0.3+-0.4+-0.2}$$

$$= 0.75 \quad \leftarrow \text{higher}$$

$$p(\text{non spam}|x) = \quad = \frac{\exp(-0.3-0.4-0.2)}{\exp)\ (\quad\quad\quad\quad)}$$

$$= 0.25.$$

∴ Mail classified as spam

+ve weights in spam means spam
+ve weights in non spam means non spam

Some of these features are more inclined towards spam than the others.

We can assign weight to each features depending on the priority. Weight vary from -1 to 1.

Features	Weight	Weight (Spam)	Weight (Not Spam)
F1: Look at who the email is addressed to	W1	-0.2	0.5
F2: Check the "from" email address	W2	-0.2	0.6
F3: Check the greeting	W3	-0.3	0.7
F4: Look at the Subject of the Email	W4	0.6	-0.3
F5: Email content are money, lottery, win, prizes	W5	0.7	-0.4
F6: Bad Grammar and Spelling	W6	0.5	-0.2
F7: No phone number in the signature	W7	-0.2	0.5

2 Moods — Happy, sad.

HMM

Today's mood - Depends on yesterday's.

Based on Her Mood, she wears Diff. Colors ← R, G, B.

Given the E & Ts tables:

E — W:T

	R	G	B
H	0.8	0.1	0.1
S	0.2	0.3	0.5

T:T — prev day Mood vs today

	H	S	
H	0.7	0.3	→ 1
S	0.5	0.5	→ 1

we, as Students won't go and ask tr. if she's happy or sad ∴ These are HIDDEN Layers.

but based on the color of Tshirts they're wearing, we can OBSERVE the Hidden states.

Observations:

Day 1	$C_1 = G$	$M_1 = ?$
Day 2	$C_2 = B$	$M_2 = ?$
Day 3	$C_3 = R$	$M_3 = ?$

Predict Most Likely moods

$\max_{M_1, M_2, M_3} P(C_1 = G, C_2 = B, C_3 = R, M_1 = M_1, M_2 = M_2, M_3 = M_3)$ all happening tg.

Sol?: $P(C_3 | C_2 C_1 M_3 M_2 M_1)$. But C_3 depends only on M_3 ⇒ $P(C_3 | M_3)$

$P(C_2 | C_1 M_3 M_2 M_1)$ $P(C_2 | M_2)$

$P(C_1 | M_3 M_2 M_1)$ $P(C_1 | M_1)$.

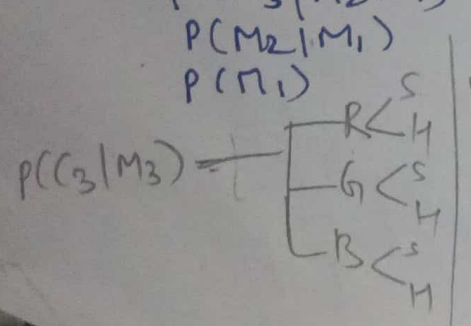
$P(M_3 | M_2 M_1)$. But only 1 history considered ⇒ $P(M_3 | M_2)$.

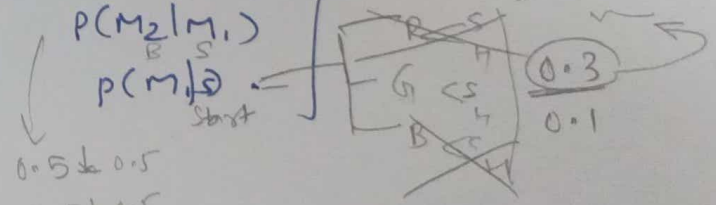
$P(M_2 | M_1)$ $P(M_2 | M_1)$

$P(M_1)$ $P(M_1)$ Start

find prod. of these s.t. its maximized

Assume it gives S, S, H.





Conditionally

Q. I went to the bank to deposit my money
I sat by Riverbank & enjoyed the view
The bank of river was covered in flowers.

word : Bank Sense1 : financial institution Sense2 : RiverBank

test : I deposited my money in bank & left to buy flowers

Assume P(Bank1) = 0.6
P(Bank2) = 0.4

P(Bank1 | Context1) = P("I", "dep...." | Bank1) * P(Bank1)

P(Bank2 | Context2) = P("I", "dep...." | Bank2) * P(Bank2)

	went	bank	deposit	money	River	Enjoyed	view	flowers
Sent1	1	1	1	1	1	1		
Sent2		1						
Sent3	1	1			1	0	1	

$P(C1) = P(Context = 1) * [P(went|C=1) * P(bank|C=1) \dots]$

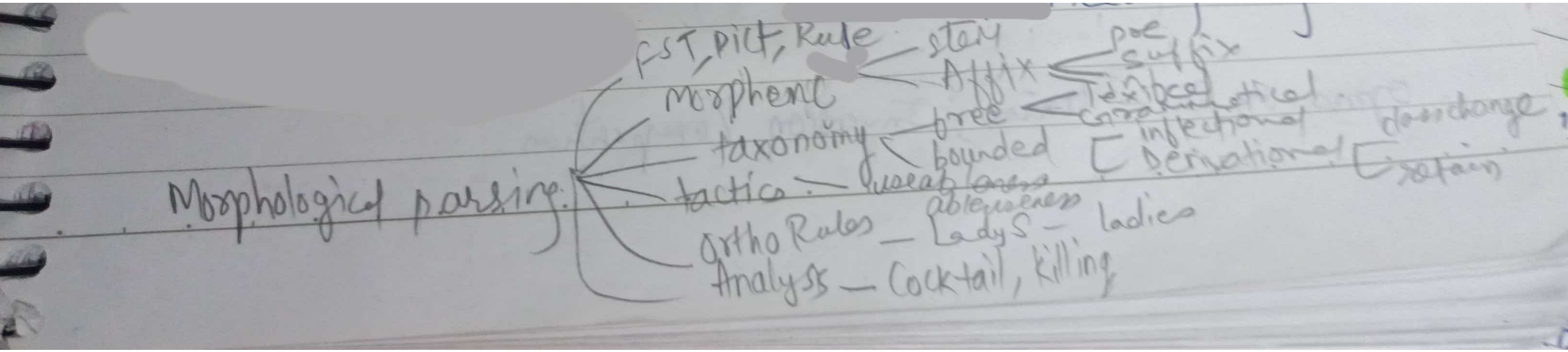
$= \frac{1}{3} * [\frac{2}{3} * \frac{1}{1} * \frac{1}{1} * \frac{1}{4} * \frac{1}{1} \dots]$

$P(C2) = \dots$

14.

Shift Reduce Parsing:

Stack	Input string	Action
$	The dog saw a man in the park $	
$ The	dog saw a man in the park $	Det → "The"
$ Det	dog saw a man in the park $	
$ Det dog	saw a man in the park $	N → "dog"
$ Det N	saw a man in the park $	
$ NP	saw a man in the park $	NP → Det N
$ NP saw	a man in the park $	V → "saw"
$ NP V	a man in the park $	det → "a"
$ NP V a	man in the park $	
$ NP V det	man in the park $	N → "man"
$ NP V det man	in the park $	NP → det N
$ NP V det N	in the park $	
$ NP V NP	in the park $	P → "in"
$ NP V NP in	the park $	Det → "the"
$ NP V NP P the	park $	N → "park"
$ NP V NP P Det park	$	NP → Det N
$ NP V NP P Det N	$	PP → P NP
$ NP V NP P NP	$	S → NP VP P
$ NP VP PP	$	
$ S		


Morphological parsing
FST, PICT, Rule
Morphemic
taxonomy
tactics
Ortho Rules
Analysis
stem
Affix
pre
suffix
free
bounded
Textbook
Grammatical
inflectional
derivational
Lady's
ladies
Cocktail, killing

3. Good Turing Algo.

- #V = 9
- possible word pairs in bigram model (like in the bigram Table count)
  $= 9 * 9 = 81$
- Seen Distinct pairs $= 7 + 2 + 1 = 10$.

Total seen pairs $= 0 \times 71 + 1 \times 7 + 2 \times 2 + 3 \times 1 = 14$

Unseen pairs $= 81 - 10 = 71$.

no. of times pair is repeating

C^*	C	N_C	P	P^*
$(C+1)*\frac{N_{C+1}}{N_C}$			$\frac{C*N_C}{\#\text{seen pairs}}$	C^* * N_C
$1*\frac{7}{71}=\frac{7}{71}$	0	71 (81−7−2−1).	$\frac{0}{14}=0$	$\frac{7}{14}=\frac{7}{71}*\frac{71}{14}*$
$2*\frac{2}{7}=\frac{4}{7}$	1	7 going abroad abroad is going to study study in in the the field.	$\frac{7}{14}$	$\frac{4}{14}$ $\frac{4}{7}*\frac{1}{14}*7$
$3*\frac{1}{2}=\frac{3}{2}$	2	2 is he is going	$\frac{4}{14}$	$\frac{3}{14}$ $\frac{3}{2}*\frac{1}{14}*2$
	3	1 He is	$\frac{3}{14}$	

$C^* = (C+1) * \frac{N_{C+1}}{N_C}$

or $P(\text{going} \mid \text{is})$ $P^* \cdot N_C$

$P(\text{is going}) \cdot$ going abroad $\cdot$ abroad to $\cdot$ to study

$= \left(\frac{3}{14} \div 2\right) \cdot \left(\frac{4}{14} \div 7\right) \cdot \left(\frac{7}{14} \div 71\right) \cdot \left(\frac{4}{14} \div 7\right) =$

possible adjacent pairs ⇒

he is	is he	going abroad	abroad i	is going	going to	to study	study in	in the	the field
3	2	1	1	2	1	1	1	1	1

Write all possible Bigrams Down

he is	III
is he	II
is going	II
going abroad	I
abroad is	I
going to	I
to study	I
study in	I
in the	I
the field	I

→ 3 categories: Count 1 : 7 items; Count 2 : 2 items; Count 3 : 1 item

#words = 9, ∴ 9*9 = 81

seen Distinct pairs = 7+2+1 = 10

Total seen pairs = 7*1 + 2*2 + 3*1 = 14

Unseen pairs = 81 − 10 = 71

N_c, c^*, P, P^*

N_c Count of Bigrams	C Size of Bigrams	C^* $(C+1)*\frac{N_{C+1}}{N_C}$	P $\frac{C^* \cap C}{\#\text{seen pairs}}$	P^* $\frac{C^* * N_C}{\#\text{seen pairs}}$
71	0	$1 * \frac{7}{71} = \frac{7}{71}$	0	$\frac{7}{14}$
7	1	$2 * \frac{2}{7} = \frac{4}{7}$	$\frac{7*1}{14}$	$\frac{4}{14}$
2	2	$3 * \frac{1}{2} = \frac{3}{2}$	$\frac{4}{14}$	$\frac{3}{14}$
1	3	$4 * \frac{1}{1}$	$\frac{3}{14}$	

P(~~(going/is)~~)

= P(is going) . P(going abroad) . P(abroad to) . P(to study)

= ÷2

$P^* \div N_C$

25/10/23

Summary Cheat Sheet

Word Level Analysis

Tokeniz^n. : → stemming (Rule) studies → lemmatiz. (Dict°)

overstemming: univers → [AI, ity, e]

understemming: Alumni → [us, ae, i]

studi (meaning x not imp) .

study

morphemes

- stem
- Affix
  - prefix
  - infix
  - postfix
  - circumfix

- inflectional
- derivational
- compounding / Concatenating — Notebook, Laptop.

regex

/[abcd]/
[a-d]

[^a]	caret
[^A-Z]	
^\.	not a period
sa?j	
o*h	h, oh, ...
ya+	ya, yaa, ...
^[ ]	anchor
[ ]$	

morph. Models

- dict
- FSM — DFA, NDFA, FST

SENTIMENT ANALYSIS

Stemming : get Root stem. only 1 arg. the word u r stemming

① **Porter stemmer**

eating → eat, history → histori, understanding → understand

Congratulations → Congratul, writes → write, fairly → fairli

② **Lancaster stemming** (worse)

writing → writ

history → hist

Finalized → fin

writes → writ

③ **RegEx stemmer**

ing$; suds w ing Remove kare

④ **Snowball stemmer** (Best)

all ✓ history → histori, fairly → Fairli

Appl": CHATBOT

lemmatization : Get Root word

u provide 2 args. 1st word. 2nd. pos tag.

lemmatizer.lemmatize(word, pos='v').

programming, v ⇒ program.

n ⇒ programming.

p. 172

CYK Algorithm

Membership Algorithm —

agar 1-4 mei start symbol aagya
Then ✓ yes

$S \to AB$
$A \to BB \mid a$
$B \to AB \mid b$

check for a b b b.

$\#cells = \frac{n(n+1)}{2}$.

	a (1)	b (2)	b (3)	b (4)
1	A (1,1)	S,B (1,2)	A (1,3)	S,B (1,4)
2	(2,1)	B (2,2)	A (2,3)	S,B (2,4)
3	(3,1)	(3,2)	B (3,3)	A (3,4)
4	(4,1)	(4,2)	(4,3)	B (4,4)

(1,2) = (1,1) + (2,2)
A B

(2,3) = (2,2) + (3,3)
B B

(3,4) = (3,3) + (4,4)
B B

$S \to AB$ ∴ S, B.
$B \to AB$

$A \to BB$ ∴ A.

∴ A.

Bigrams.

(1,3) = (1,1) + (3,3)
A B

(1,4) = (1,1) + (4,4)

(2,4) = (2,2) + (4,4)
B

(1,1) + (2,3) → abb (1 2 3)
A A ✗ ϕ

abbb (1 2 3 4)

bbb (2 3 4)

(2,3) (4) → (4,4)
A or B = AB ✓ S, B.

(2) (3,4)
B A = BA ✗

Trigrams.

(1,2) (3) ϕ
S,B B = SB BB = A

(1,4) ⇒ a b b b (1 2 3 4)

1	2,3,4 (2,4)	A S,B ⇒ (A,S) (A,B) → SB
1,2	3,4	S,B A ⇒ (S,A) (B,A) ✗ ϕ
(1,3) or (1,2,3)	4	A B ⇒ AB. S, B.

Hey Pranav, can you add auto play
next video feature on codebasics.io

[12 1087 7 1870 1 2 ... 9]

Label Encoding

1 2 3 4 5
add, auto, and, assistance, are,
6 7 8 9
business, can, claim, codebsics.io,
10 11 12
data, 55, hey,
...

urgent

2000

VOCABULARY

label encoding if you have done

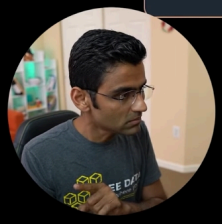
	add,	auto,	and,	assistance,	are,	business,	can,	...	urgent
Hey	[0	0	0	0	0	0	...		0]
Pranav	[0	0	0	0	0	0	...		0]
can	[0	0	0	0	0	0	1		0]
you	[0	0	0	0	0	0	0		0]
add	[1	0	0	0	0	0	0		0]
auto	[0	1	0	0	0	0	0		0]
...									

See here

	add,	auto,	and,	assistance,	need,	I,	immediately,	help,	urgent
I	[0	0	0	0	0	1	0	0	0]
need	[0	0	0	0	1	0	0	0	0]
help	[0	0	0	0	0	0	0	1	0]

I	[0	0	0	0	0	1	0	0	0]
need	[0	0	0	0	1	0	0	0	0]
assistance	[0	0	0	1	0	0	0	0	0]

Similar words do not have similar representation



the disadvantage number one.

add, auto, and, assistance, need, I, immediately, ... urgent (100 k words)

I [0 0 0 0 0 1 0 ... 0] ← vector size=100k

if too many words in Email

Consumes too much memory & compute resources

So it consumes not only too much memory,

	add,	auto,	and,	assistance,	need,	I,	immediately,	help,	urgent
I	[0	0	0	0	0	1	0	0	0]
need	[0	0	0	0	1	0	0	0	0]
help	[0	0	0	0	0	0	0	1	0]
bahubali	??								

Out Of Vocabulary (OOV) problem

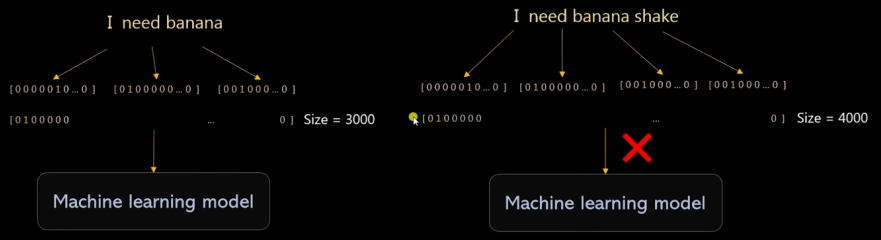
numeric representation

	add,	auto,	and,	assistance,	need,	I,	immediately,	help,	urgent
I	[0	0	0	0	0	1	0	0	0]
need	[0	0	0	0	1	0	0	0	0]
help	[0	0	0	0	0	0	0	1	0]
bahubali	???								

Out Of Vocabulary (OOV) problem

which can even represent the unknown

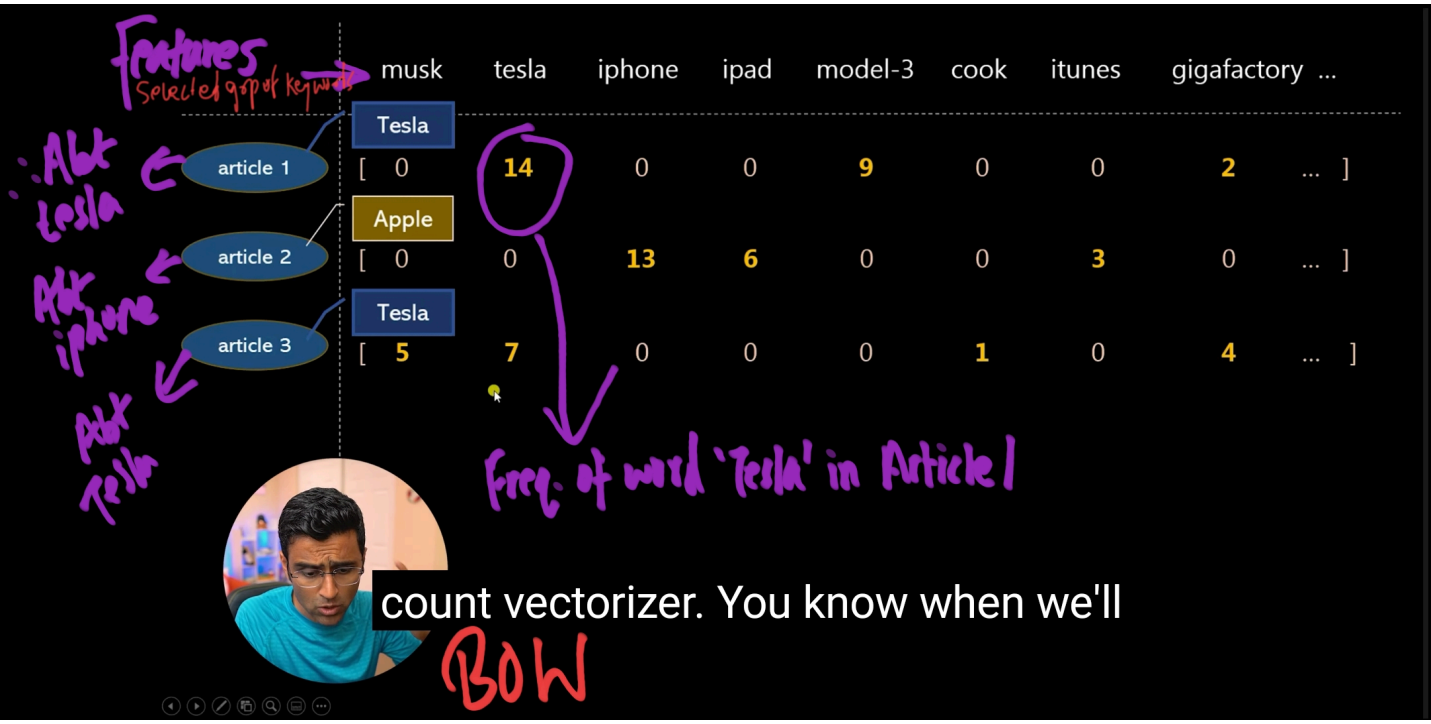
All Unknowns will have Same Numeric Representation.



No fixed length representation

and that's the disadvantage number four.

All Emails have diff. Sizes. A NN has a fixed size i/p.


Features
Selected group of keywords
musk
tesla
iphone
ipad
model-3
cook
itunes
gigafactory ...
Tesla
article 1
[0 14 0 0 9 0 0 2 ...]
Apple
article 2
[0 0 13 6 0 0 3 0 ...]
Tesla
article 3
[5 7 0 0 0 1 0 4 ...]
Abt Tesla
Abt iphone
Abt Tesla
Freq. of word 'Tesla' in Article 1
count vectorizer. You know when we'll
BoW

	claim	free	help	health	hurry	so	play	money	up	volleyball
Doc 1 play volleyball	0	0	0	0	0	0	**1**	0	0	**1**
Doc 2 Claim FREE money. Hurry up!	**1**	**1**	0	0	**1**	0	0	**1**	**1**	0
Doc 3 Playing helps health so play!	0	0	**1**	**1**	0	**1**	**2**	0	0	0

text.

	add,	auto,	and,	assistance,	need,	I,	immediately,	help,	...	urgent (100 k words)	
I need help	[0	0	0	0	1	1	0	1	...	0]	← vector size=100k

Sparse Representation

That's called sparse presentation and
It may consume too much memory & compute resources

	add,	auto,	and,	assistance,	need,	I,	immediately,	help,	urgent
I need help	[0	0	0	0	1	1	0	1	0]
I need assistance	[0	0	0	1	1	1	0	0	0]

emails is different. See here we have 1

Doesn't capture meaning of words properly

	add,	auto,	and,	assistance,	need,	I,	immediately,	help,	urgent
I need help	[0	0	0	0	1	1	0	1	0]
I need assistance	[0	0	0	1	1	1	0	0	0]

numeric representation of both of these

Doesn't capture meaning of words properly

Dhaval sat on a sofa and ate a samosa

Dhaval sat on a samosa and ate a sofa

Meaning of a sentence is determined by order of words

Dhaval sat on a sofa and ate a samosa

BOW

Dhaval | sat | on | a | sofa | and | ate | ...

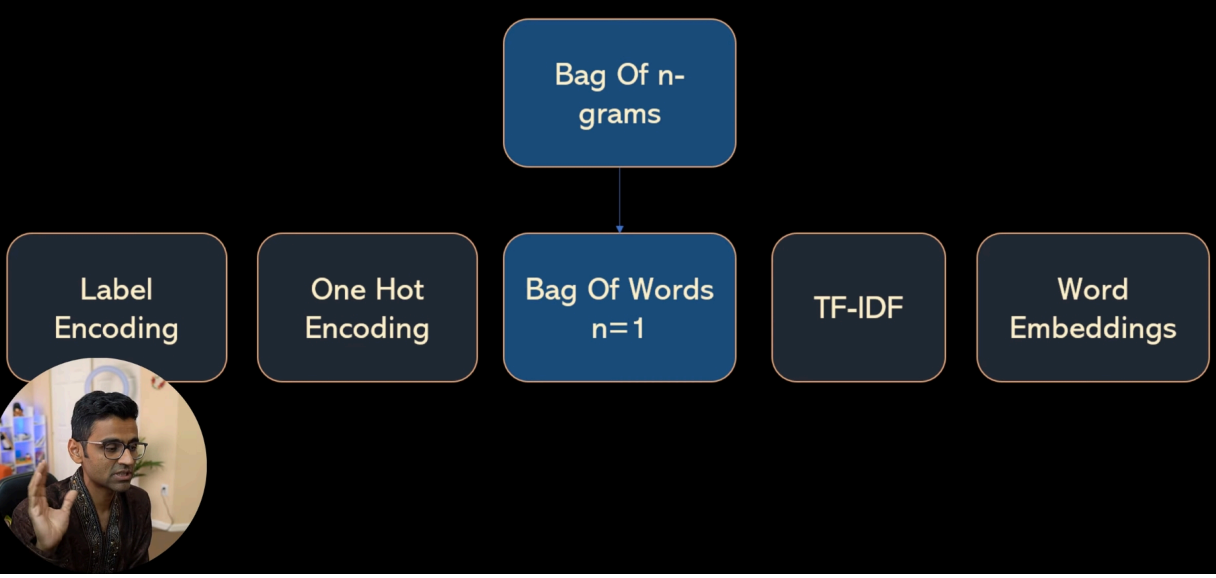
bi-gram

Dhaval sat | sat on | on a | a sofa | sofa and | ...

tri-gram

Dhaval sat on | sat on a | on a sofa | a sofa and | ...

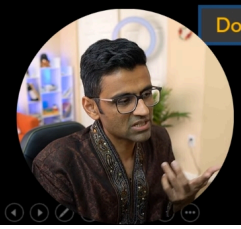
There are various approaches of converting text into vector



Pre-processing

Doc 1	Doc 2	Doc 3
Thor ate pizza	Loki is tall	Loki is eating pizza
Thor eat pizza	Loki tall	Loki eat pizza

	Thor	eat	pizza	Loki	tall
Doc 1	1	1	1	0	0
Doc 2	0	0	0	1	1
Doc 3	0	1	1	1	0



Pre-processing

	Doc 1	Doc 2	Doc 3
	Thor ate pizza	Loki is tall	Loki is eating pizza
	Thor eat pizza	Loki tall	Loki eat pizza

	"Thor eat"	"eat pizza"	"Loki tall"	"Loki eat"
Doc 1	1	1	0	0
Doc 2	0	0	1	0
Doc 3	0	1	0	1

Word2vec

King – man + woman = Queen

We'll look all of this in detail, we'll


King
authority = 1
has tail = 0
rich = 1
gender = -1
[1,0,1,-1]
Horse
authority = 0
has tail = 1
rich = 0
gender = -1
[0,1,0,-1]
CODE BASICS

	battle	horse	king	man	queen	..	woman
authority	0	0.01	1	0.2	1	...	0.2
event	1	0	0	0	0	...	0
has tail?	0	1	0	0	0	...	0
rich	0	0.1	1	0.3	1	...	0.2
gender	0	1	-1	-1	1	...	1

King	- man	+ woman			~	Queen
1	0.2	0.2		1		1
0	0	0		0		0
0	0	0	=	0	~	0
1	0.3	0.2		0.9		1
-1	-1	1		1		1

1. Take a fake problem
2. Solve it using neural network
3. You get word embeddings as a side effect

fake problem: fill in a missing word in a sentence

There lived a king called Ashoka in India. After Kalinga battle, he converted to Buddhism. This mighty king ordered his ministers to put together a peaceful treaty with their neighboring kingdoms. The emperor ordered his ministers to also build stupa, a monument with Buddha's teachings.

Fake problem

king ordered his ministers

emperor ordered his ministers

problem: fill in the blanks

Side effect

king [0.7, 0.4, 1.2, 3.8] emperor [0.7, 0.5, 1.2, 3.8]

king ~ emperor

word embeddings

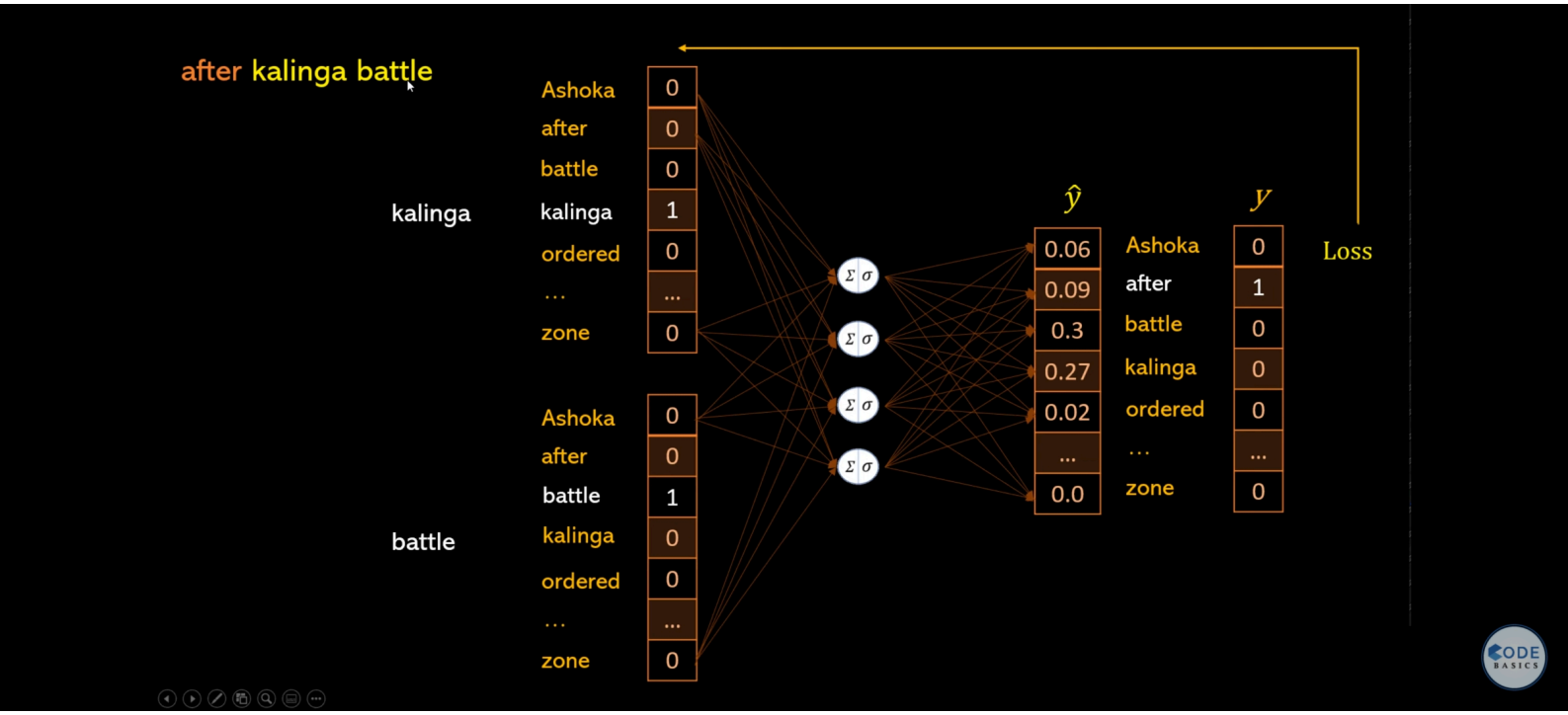
eating ____ is very healthy

table, angry, truck, apple, pizza, walnut

Meaning of word can be inferred by surrounding words

NASA launched ____ last month

table, angry, truck, rocket, apple, pizza


after kalinga battle
kalinga
battle
Ashoka
after
battle
kalinga
ordered
…
zone
0
0
0
1
0
...
0
Ashoka
after
battle
kalinga
ordered
…
zone
0
0
1
0
0
...
0
Σ σ
Σ σ
Σ σ
Σ σ
ŷ
0.06
0.09
0.3
0.27
0.02
...
0.0
Ashoka
after
battle
kalinga
ordered
…
zone
y
0
1
0
0
0
...
0
Loss

The first method, called **Continuous Bag of Words**, increases the context by using the surrounding words to predict what occurs in the middle.

Training Data

Troll 2 is great!
Gymkata is great!