

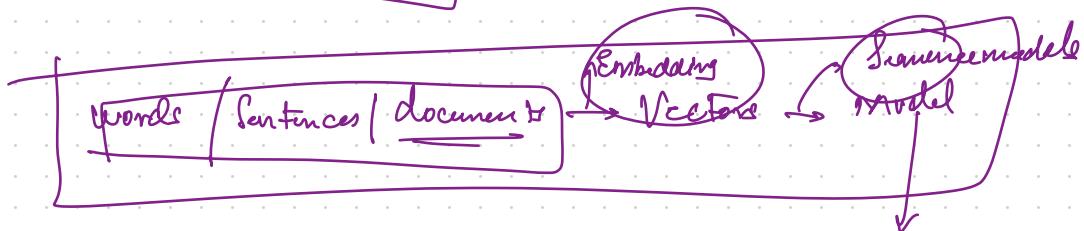
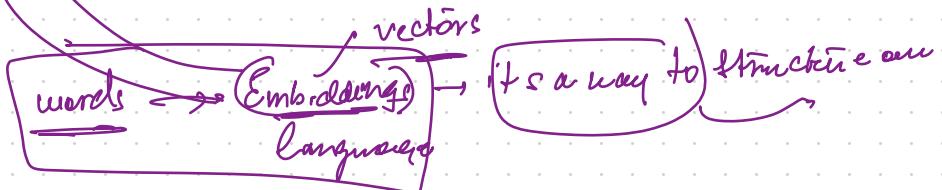
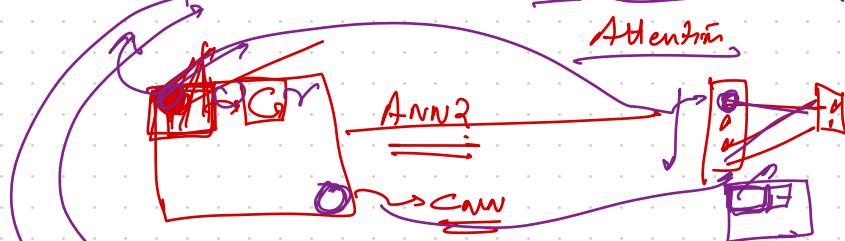

Natural language Processing

① NLP problems are easel.

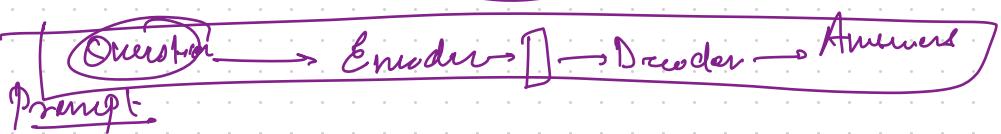
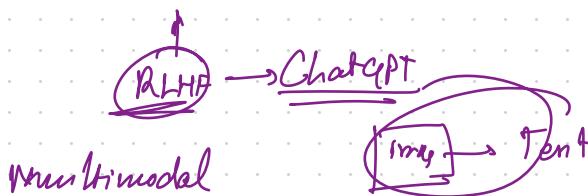
② How do we structure words / sentence

Feature engineering

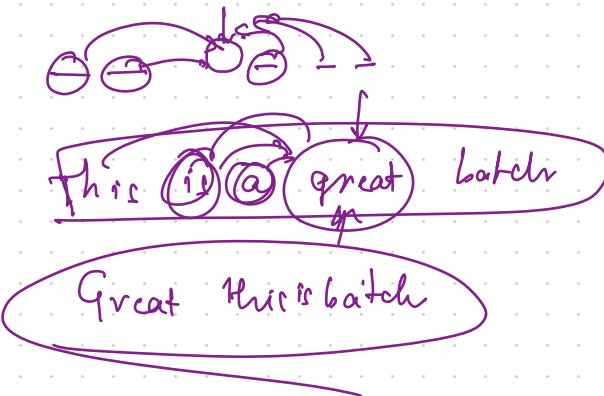
③ Same models → RNN, LSTM, GRU, Transformer



large language models



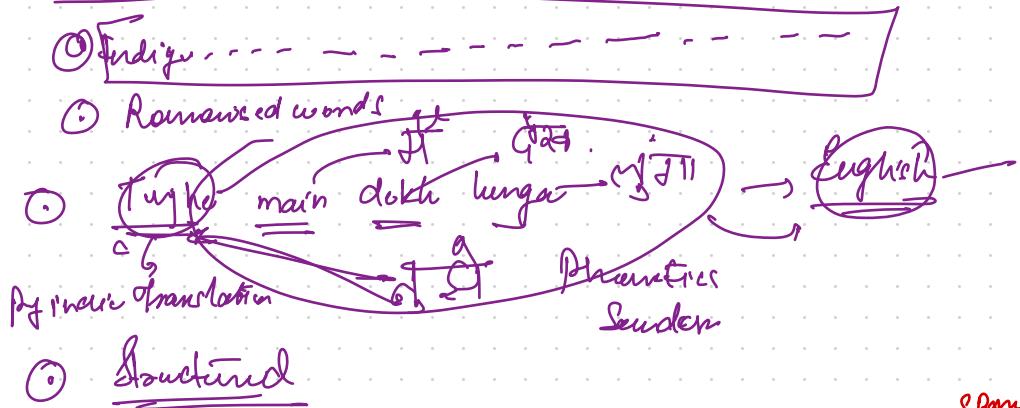
Natural Language - is a Sequence



Natural language Processing

Understanding information from text

- ① Unstructured
- ② Context / Sentiment / Change / Sarcasm



- ③ Structured

Use Cases:

Translation

gmail auto type → auto fill

Search engines

Sentiment analysis

Gmail → spam
→ ham

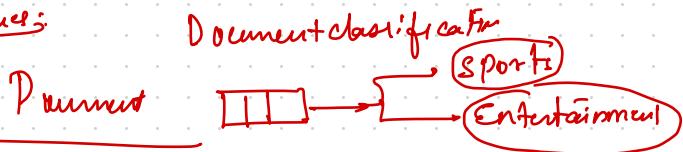
Chatbot

Text classification, Sum
NLP, information

$$\textcircled{1} = f(x) \quad \begin{matrix} \text{Elemental} \\ \uparrow \uparrow \uparrow \end{matrix} \quad \begin{matrix} \text{language model} \\ \boxed{\text{language model}} \end{matrix}$$

Sentences are very unclear

Preprocessing techniques:



Special characters

Removal of unwanted Characters or patterns of characters from our source sentence.

Ticket classifier

Sentiment classification

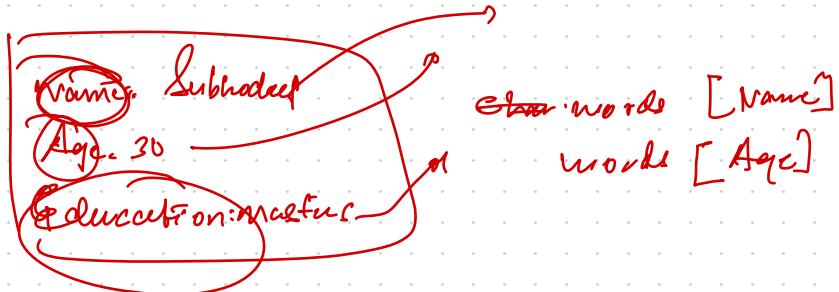
Ticket

Regex → a process of defining certain patterns using certain rules.

"I am very hungry. I will order and give to my friend!"

Region → Extract some defined patterns
from sentences

[w]
[s]



Stopwords

(an), a, the
because and Then

Translation

, what bot → Tomo tan 25°C
→ I am com - - - - -
What's the temp. phenomenon

It's time to go home → Tom (S) air going home

Stemming / lemmatization

Words → Embedding
→ Vectors → Model

Embedding

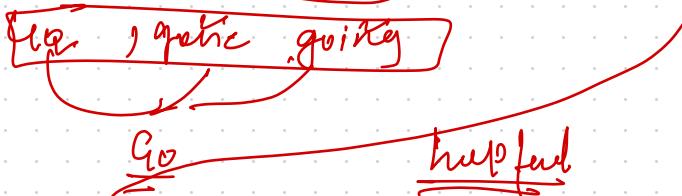
✓ Frequency based
✓ Count of the words
in the sentence

Context?

Randomness based:
vectors generated
through LNN.

If am going home, home is where heart is

g:1, am:1, going:1, home:2, ts:2
where:1, heart:1



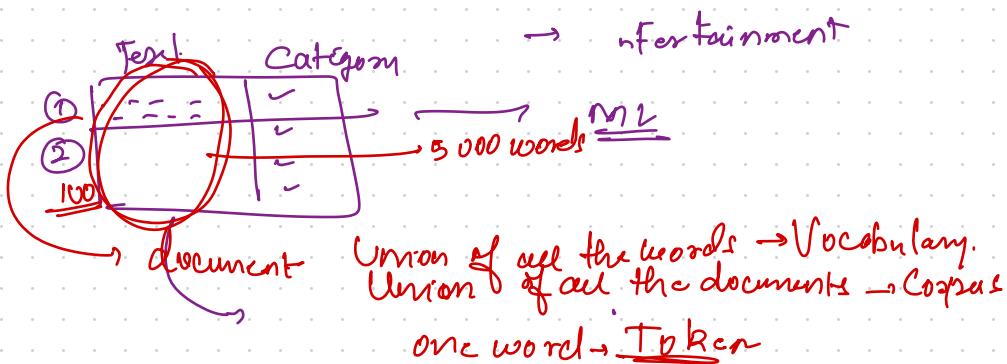
Sentences
vectors
form

10:33 → 10:40pm

↓

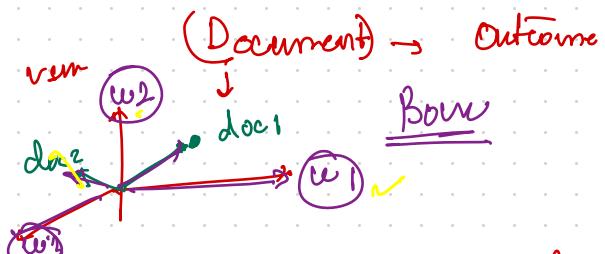


How to represent sentences/documents in a vectorised form using frequency embeddings.



Page of words model

Document classification

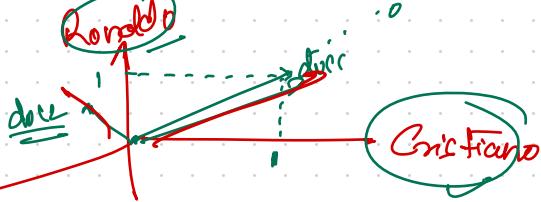


A hand-drawn diagram on lined paper. A large speech bubble shape is drawn with a red marker. Inside the bubble, the word 'clothes' is written in red. Two arrows point from the word 'fabrics' to the top left of the bubble and from the word 'words' to the top right of the bubble.

- Christiano Ronaldo got transferred from Sports to Cristoforo's outfit

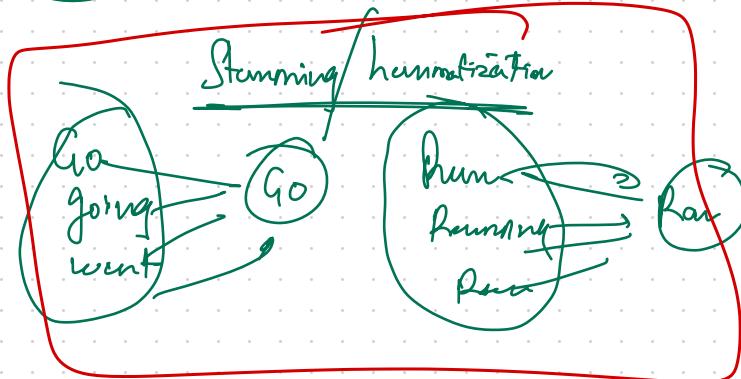
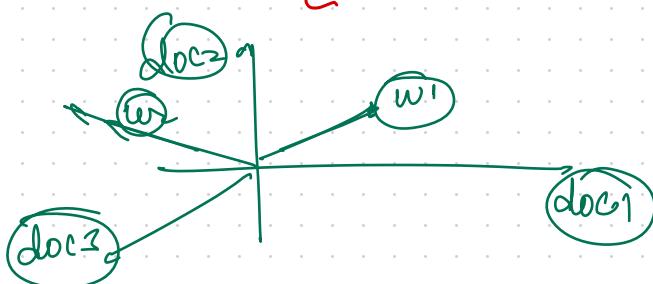
⑥ Pucca began performances in Cannes.

	Cristiano	Ronaldo	got	transferred	Dna	lipa	Performs in	Comme
①	1	1	1	1	0	0	0	0
②	0	0	0	0	1	1	1	1



Problem: ① semantic dimensions

- ① does not maintain context
 - order not maintained
 - By definition of semantics all words are independent





Lernmaterialien

A handwritten diagram on lined paper. On the left, there are two arrows pointing from the word 'Running' to the word 'Run'. The top arrow originates from the first 'R' in 'Running' and points to the underlined 'Run'. The bottom arrow originates from the 'ing' suffix in 'Running' and points to the underlined 'Run'.

Going to go

Spreading ✓
Specific ✓
Conscious ✓
Grey → Grey

Diagram illustrating Verb Phrases and Dependencies:

Verb Phrase Analysis:

- Root node: "will make" (verb phrase)
- Children: "will" and "make"
- "make" has children: "visit my home" and "and"
- "visit my home" has children: "visit" and "my home"
- "visit" has children: "I" and "go"
- "go" has children: "to" and "my home"
- "my home" has children: "my" and "home"
- "my" has children: "I" and "ac"
- "home" has children: "and" and "will make" (referred to as a recursive node)

Dependency Parse:

```

graph TD
    Root --- NP1[my own]
    Root --- NP2[is awesome]
    NP1 --- P1[is]
    NP1 --- NP3[my]
    NP1 --- NP4[home]
    NP3 --- P2[my]
    NP3 --- NP5[home]
    NP5 --- P3[and]
    NP5 --- VP[will make]
    NP4 --- P4[and]
    NP4 --- VP
    VP --- V[will]
    VP --- V2[make]
    V --- T1[to]
    V --- NP6[visit my home]
    NP6 --- V3[visit]
    NP6 --- NP7[my home]
    V3 --- T2[I]
    V3 --- NP8[go]
    NP8 --- T3[to]
    NP8 --- NP9[my home]
    NP9 --- T4[visit]
    NP9 --- NP10[my home]
    T4 --- T5[visit]
    T5 --- T6[my]
    T6 --- T7[home]
    T7 --- T8[and]
    T8 --- VP2[will make]
    VP2 --- V4[will]
    VP2 --- V5[make]
    V4 --- T9[visit]
    V5 --- T10[my]
    V5 --- T11[home]
  
```

The diagram shows the hierarchical structure of the sentence "I am going to visit my home and will make" and its dependencies. The root node is "will make". It branches into "will" and "make". "make" branches into "visit my home" and "and". "visit my home" branches into "visit" and "my home". "visit" branches into "I" and "go". "go" branches into "to" and "my home". "my home" branches into "my" and "home". "my" branches into "I" and "ac". "home" branches into "and" and "will make" (a recursive node). The dependency parse provides a detailed breakdown of these relationships.

"our great at public Speaking"

our (word) Stemming

great lemmatization humantech

at public Speaking word tokeniser

Speaking → words

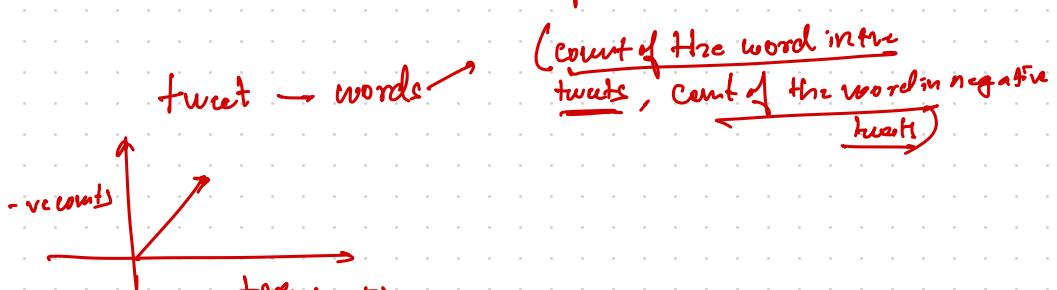
Sentence Tokenizer
Documents \rightarrow Sentence

• split('')

"I | am | going | to | my | home". Split("")

[I, am, going, to, my, home]

write \rightarrow word-tokenizer



tweet \rightarrow (Count of the word, Count of the -ve words, +ve words)

tweet: The Game

+ve 50 10
+ve 80 60
+ve 100 -ve 100

tweet (110, 110)

f_{game}

$$\left\{ \begin{array}{l} (\text{game}, 0) : 20 \\ (\text{game}, 1) : 40 \\ (\text{awesome}, 0) : 50 \\ (\text{awesome}, 1) : 90 \end{array} \right.$$

fact: game \neq awesome

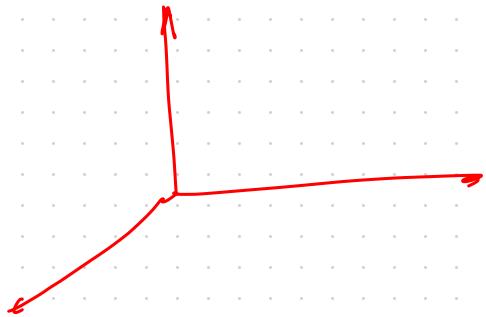
game awesome

$$\begin{aligned} \text{pos} &= \text{f}_{\text{game}}[(\text{game}, 1)] = 40 + \text{f}_{\text{game}}[(\text{awesome}, 1)]^{\rightarrow, 90} \\ \text{neg} &= \text{f}_{\text{game}}[(\text{game}, 0)] = 20 + \text{f}_{\text{game}}[(\text{awesome}, 0)]^{\rightarrow, 50} \end{aligned}$$

$$\text{pos} = 130$$

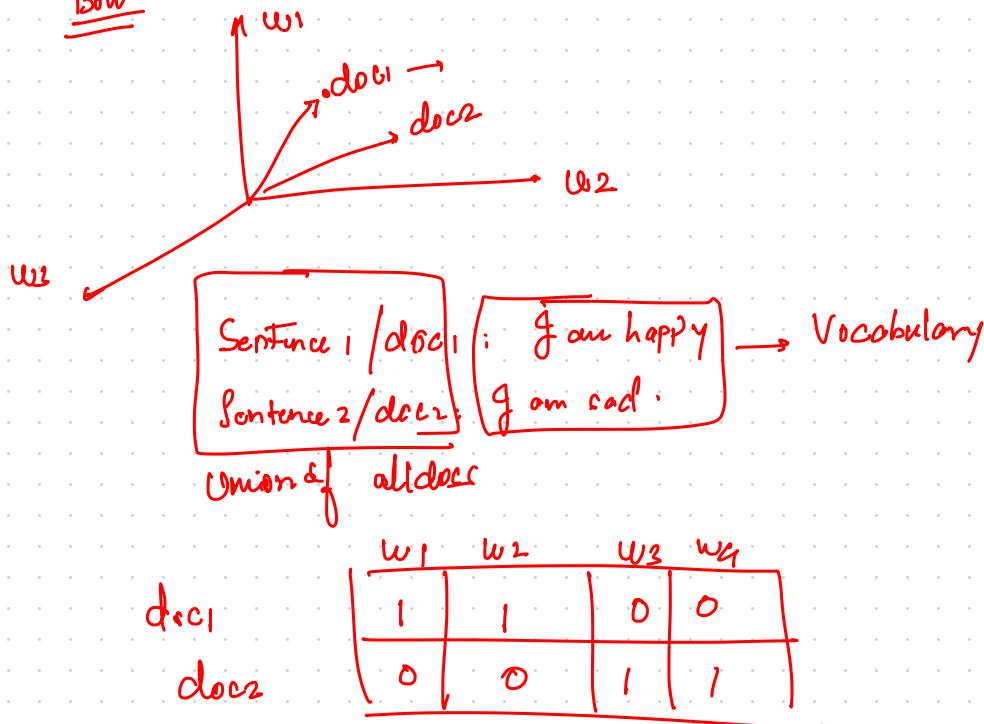
$$\text{neg} = 70$$

$$\text{f}_{\text{total}} = (1, 130, 70)$$

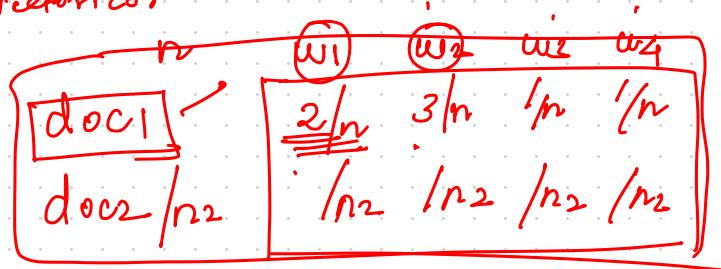


NLP \rightarrow words / docs into vector space models

BOW



Count vectorizer



① do not maintain any order

doc1 : I am happy today. today help I am
 I am happy today

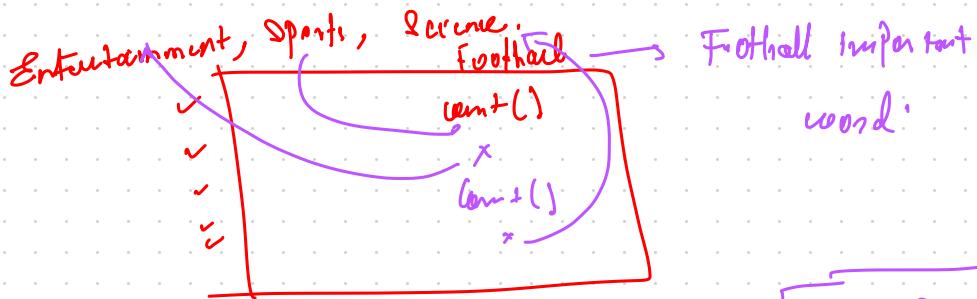
doc1 :

1	1	1	1	1
---	---	---	---	---

frequency based embeddings don't capture the context

Term Frequency vs Document Frequency

Football



$$\text{tf}(\text{Term frequency}) \times \text{idf}$$

$$\text{tf}(w_2, \text{doc}_1) \times \frac{\log\left(\frac{N}{n}\right)}{\text{Total number of tokens in doc}_1}$$

where $w_1, w_2, w_3, \dots, w_m$

$N = \{ \text{doc}_1, \text{doc}_2, \text{doc}_3, \dots \}$

number of documents in which w_i occurs.

doc1 : Ronaldo is lovely, Ronaldo is rich.

doc2 : Ronaldo is famous.

doc3 : Messi is famous.

doc1: Mondo is lovely, Ronaldo is rich.

doc2: Ronaldo is famous.

doc3: Mussi is homely.

Normalized term frequency $\times \log \left(\frac{N+1}{n} \right)$

$\frac{1}{3}$

doc1

doc2

doc3

Mondo

Mussi

lovely

rich

famous

$\log \left(\frac{4}{2} \right)$

$\log \left(\frac{4}{2} \right)$

$\log \left(\frac{4}{2} \right)$

$\frac{2}{4} \times \log \left(\frac{4}{2} \right)$

$\frac{1}{4} \times \log \left(\frac{N+1}{n} \right) = \frac{1}{4} \times \log \left(\frac{3}{2} \right) = n \times \log \left(\frac{1}{n} \right) \rightarrow 0$

$\uparrow \log \left(\frac{N+1}{n} \right) \rightarrow n \approx N$

$\frac{3+1}{2}$

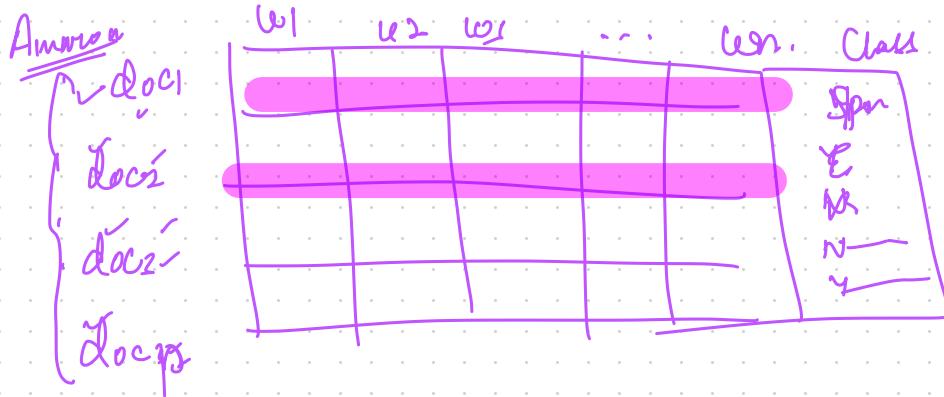
$\log \left(\frac{N+1}{n} \right) \approx 1$

TF-IDF (t_i)_{doc1}

= Normalised term frequency $\times \log \left(\frac{N+1}{n} \right)$

$N \rightarrow$ total number of docs in corpus.

$n \rightarrow$ total number of docs in which term occurs.



Find out Similar Documents

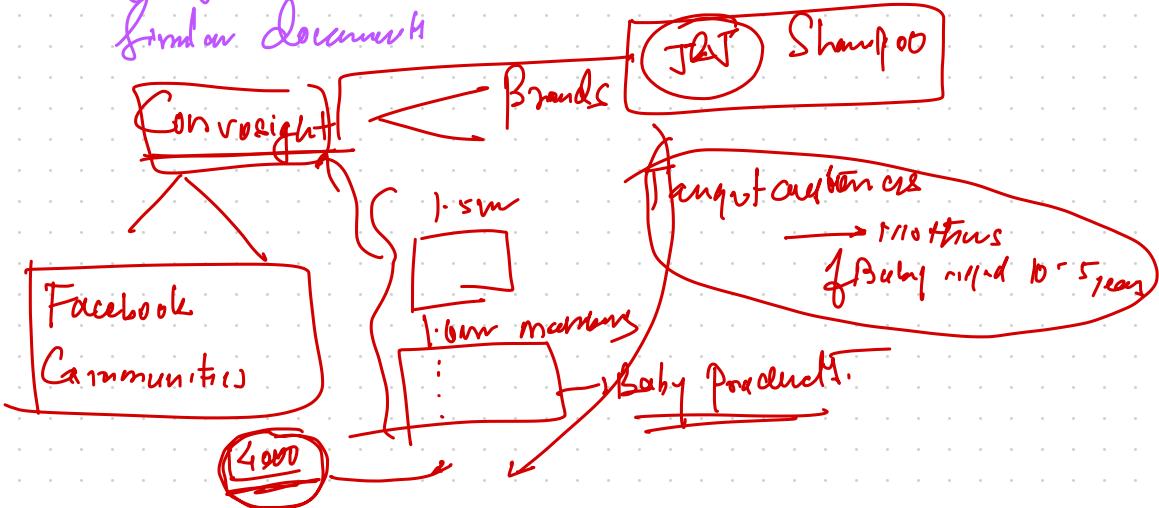
doc1 (Product)
doc3 (Product) ← → :

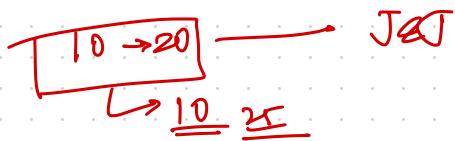
addresses



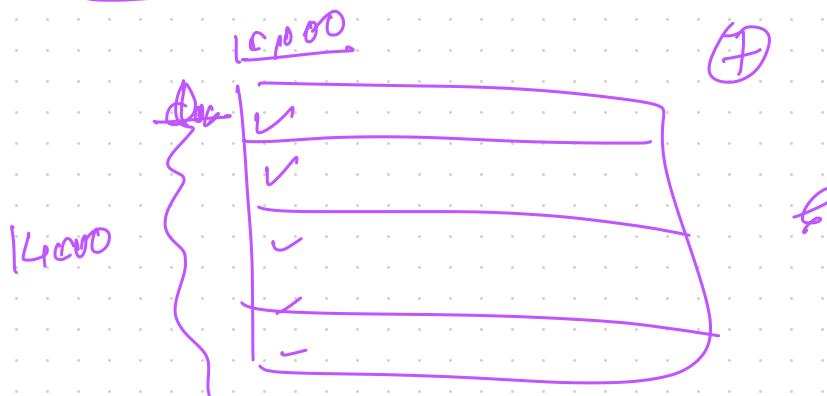
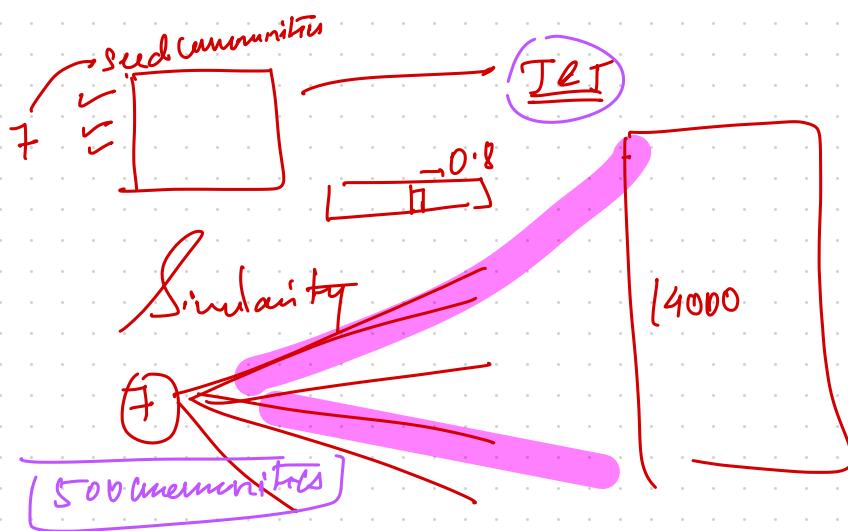
Using document based embedding of word2vec

Find our documents





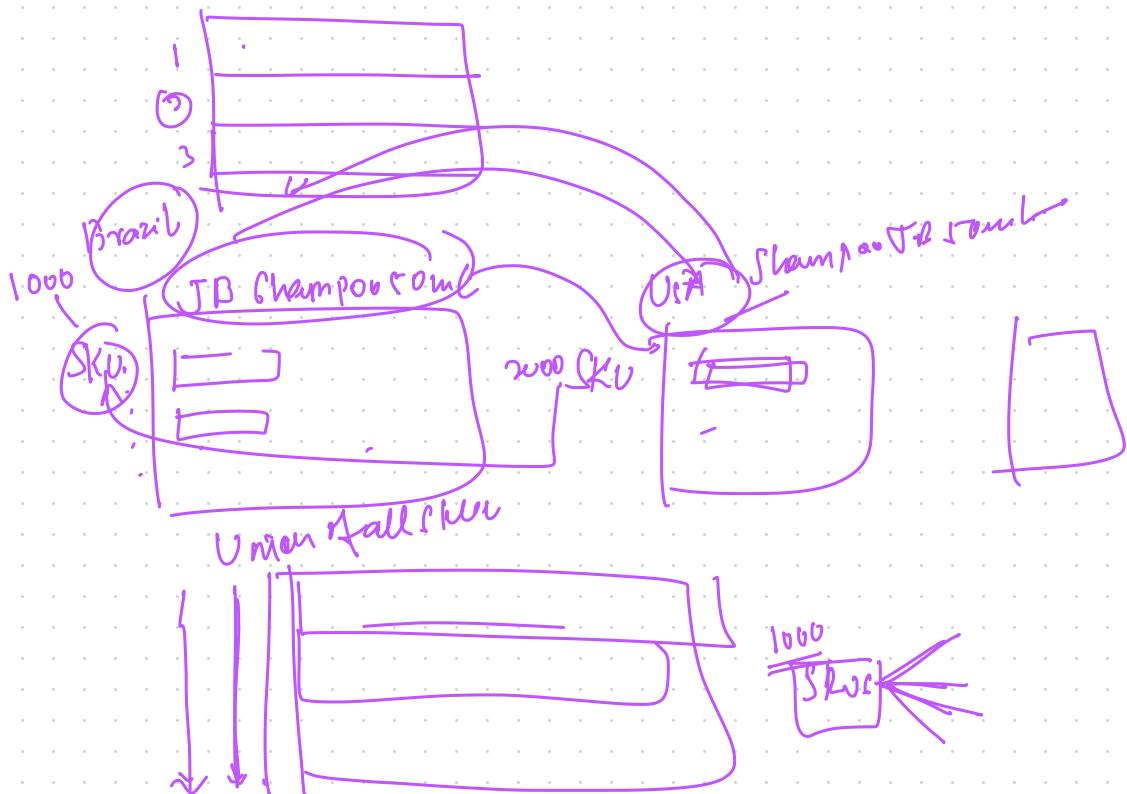
1000 users
↳ 1.6 billion conversations

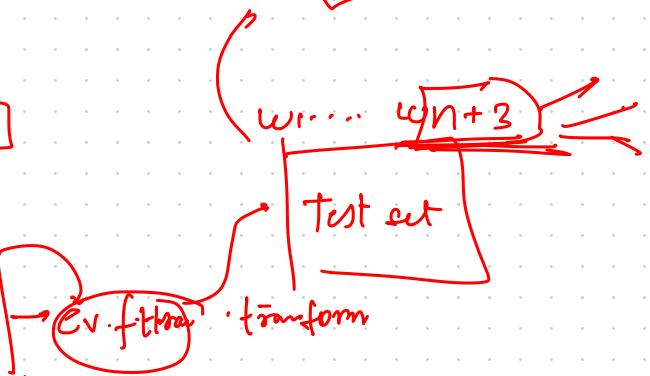
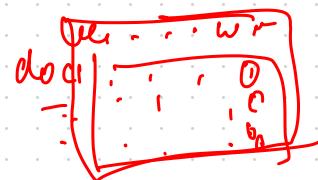
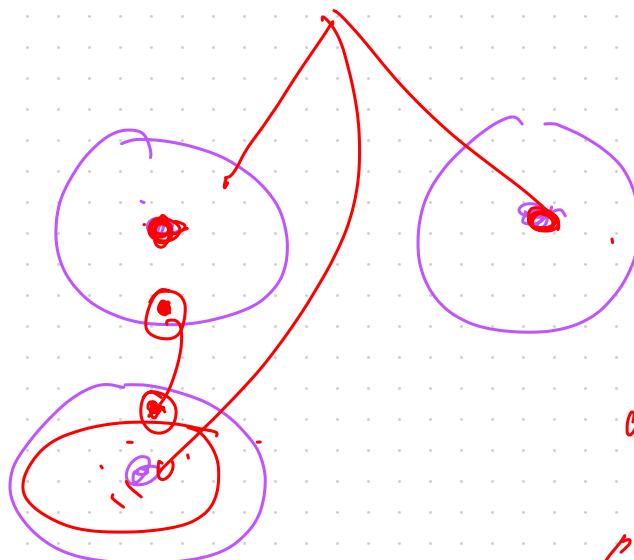


fan  happy to teach you

Food and drops for
weather, nutrients for babies

  Hey mother, see this new recipe
of created





corpus: [" Jan - . , " - ...]

2 for sentence in corpus

log(n/m)

sentence

two happy friends

doc1

$\times \log(n/m)$

doc2

$\times \log(M/m)$

|ʃ| |ən| |'gret|, thank you.

Undergroup:

|ʃ| |ən| |'gret| |θank| |yου|

Bigram: |ʃ| |ən| |'gret| |θank| |yου|

|ʃ| |ən| |ən| |'gret|

Embedding \rightarrow wordvec

Frequency based embedding

words

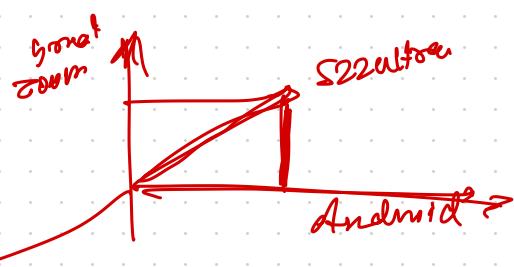
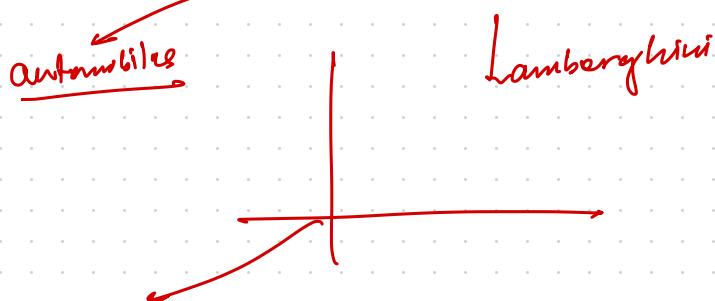
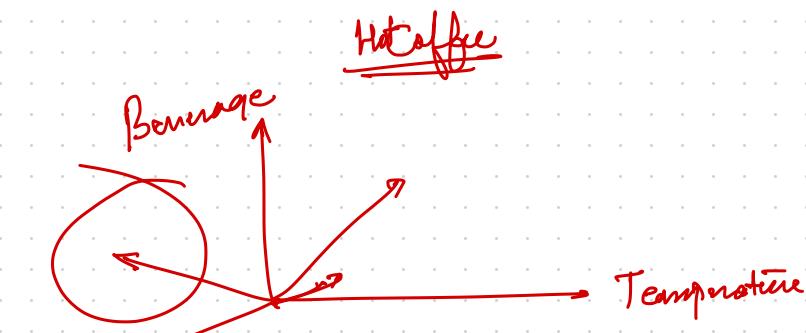
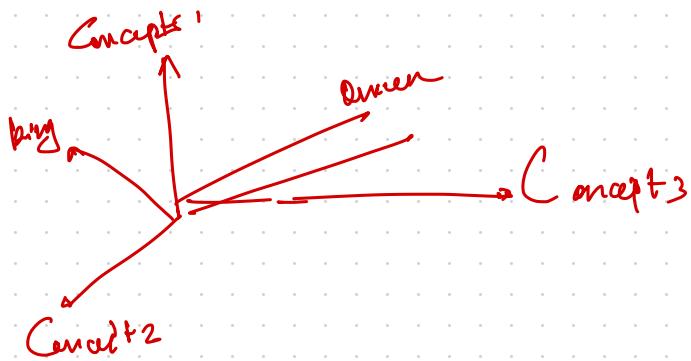
docs

v	v	v	f
v	-	-	-
.	-	-	-
.	-	-	-

Prediction based Embedding



Prediction based embedding



Embeddings are dense representation of concepts

Abstract nonlinear combinations
of multiple words.

Embedding's Dimensionality

Word are an model

Embedding SVD $\propto \propto x$

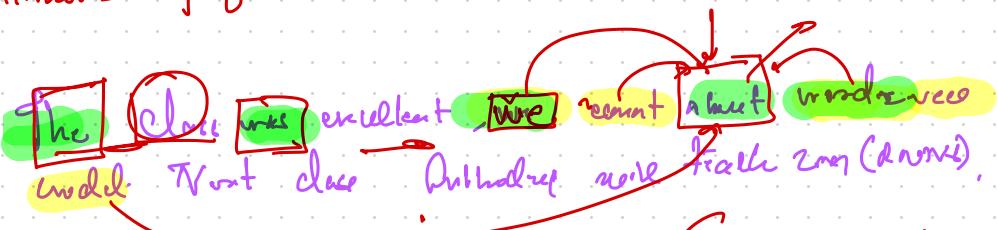
Embedding from sequences models \rightarrow (Supervised way)

Lm. Seqs (p_{lm}, h_{lm})

Word2vec

cbow
continuous Bag of words

Skip gram model



Context words
The, was
close, excellent
was, we

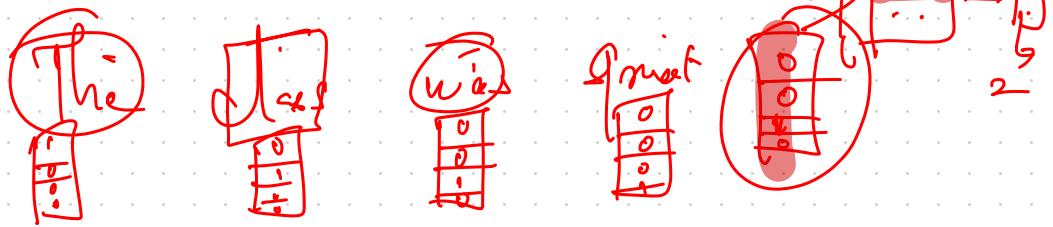
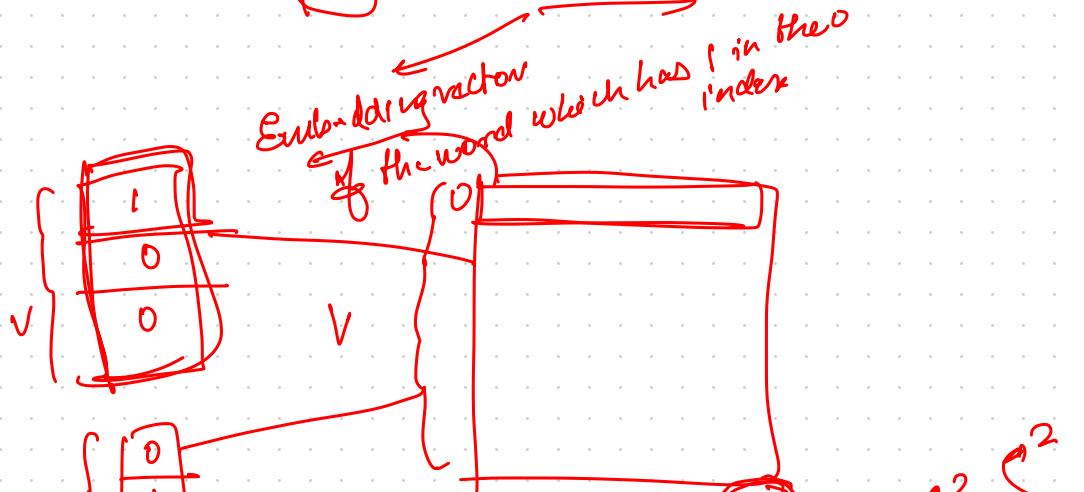
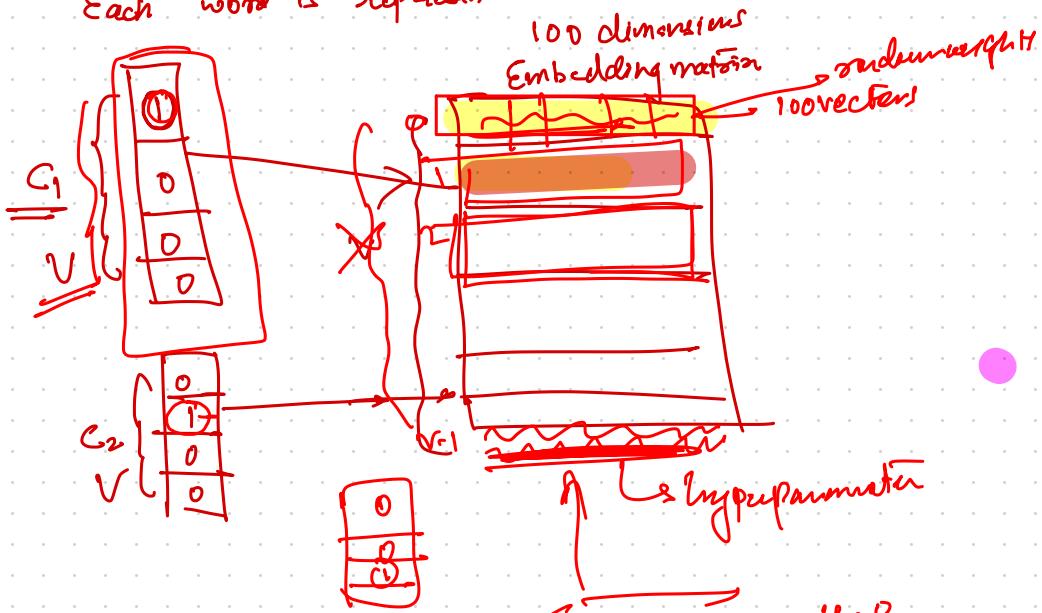
Target word
close
was
excellent

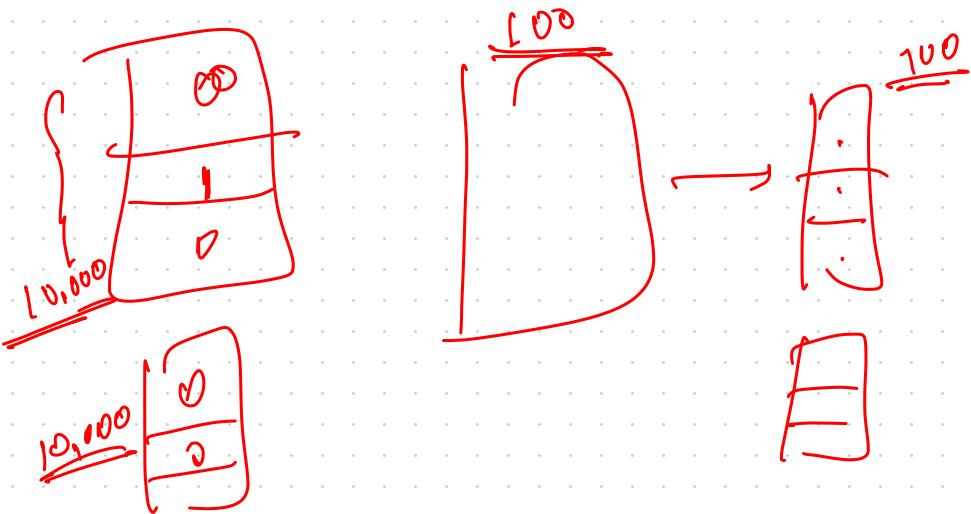
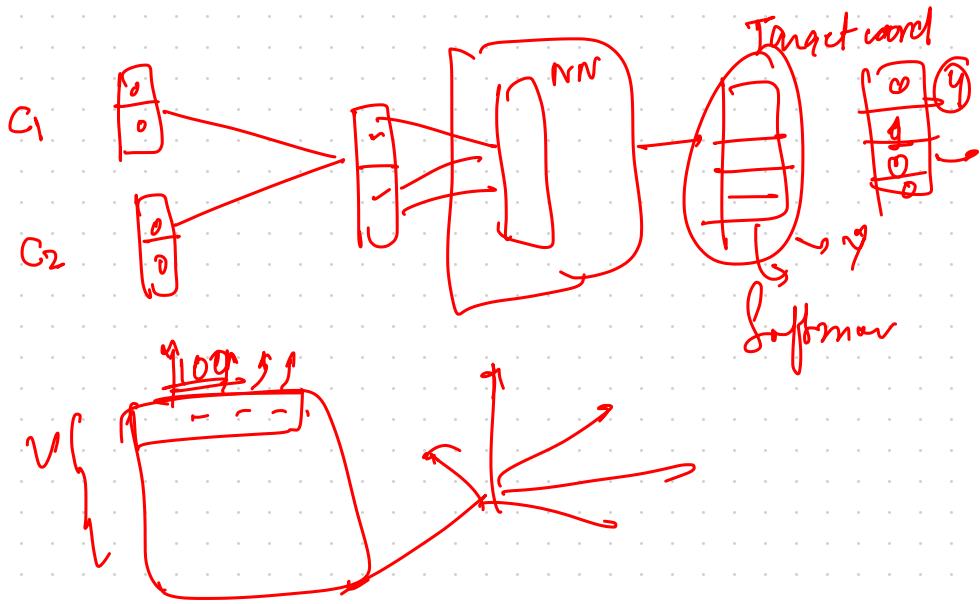
Context window
 $= 1$
Context window
 ≤ 2

close

The

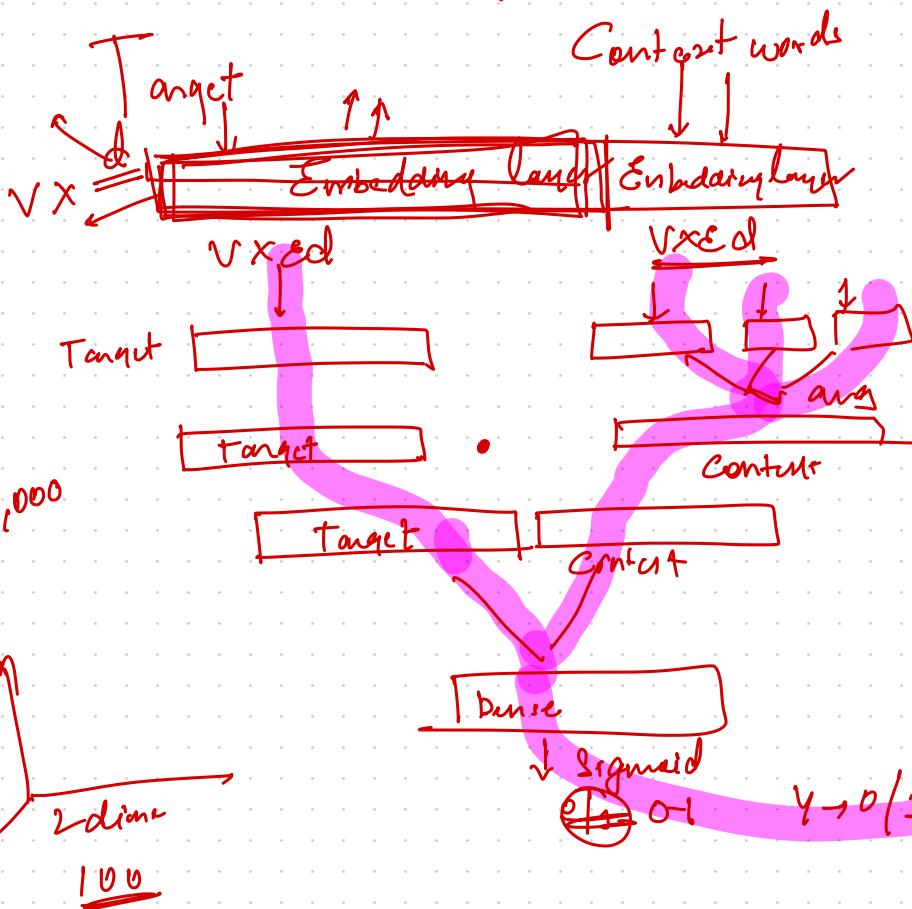
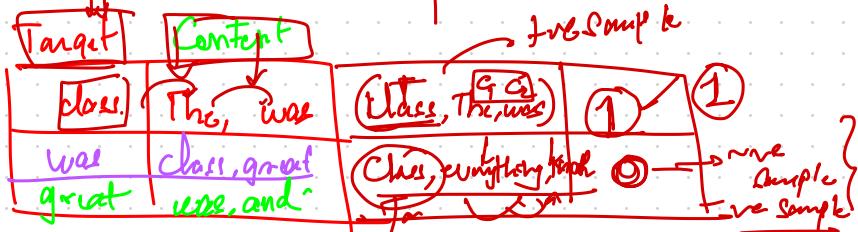
Each word is represented as a one-hot encoded vector

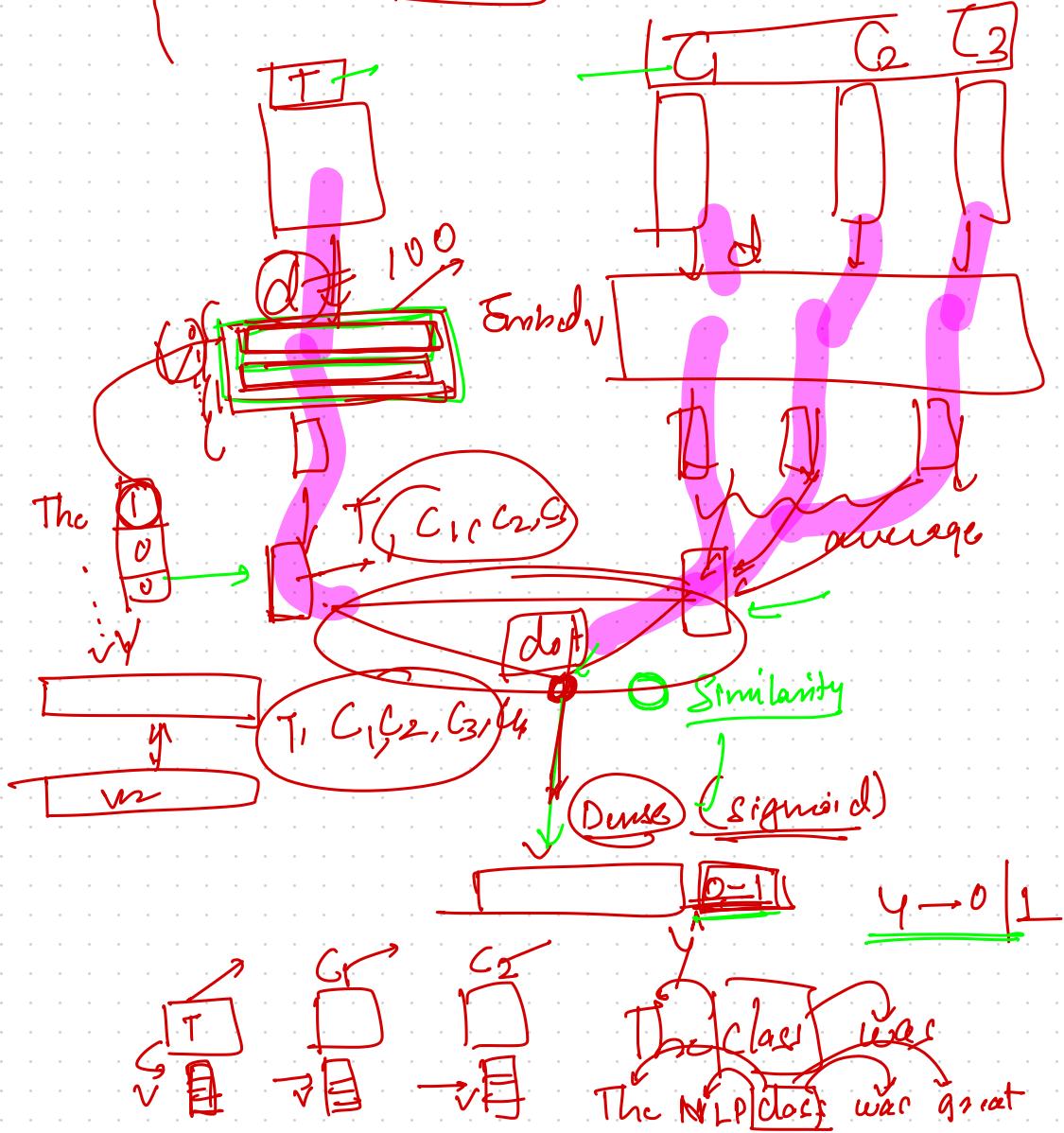
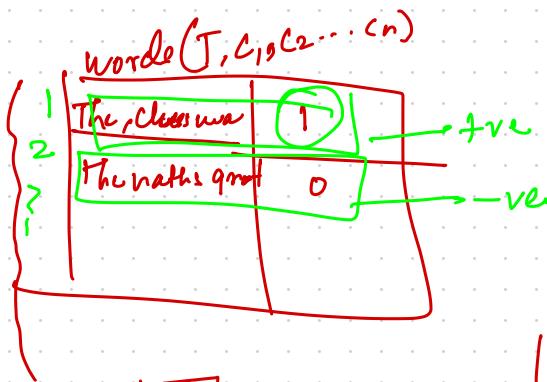




Skip-gram model

The class was great and we got to know new concepts





Target

Class

Content words

The NLP was great

v

v

v

v

v

(SxV)

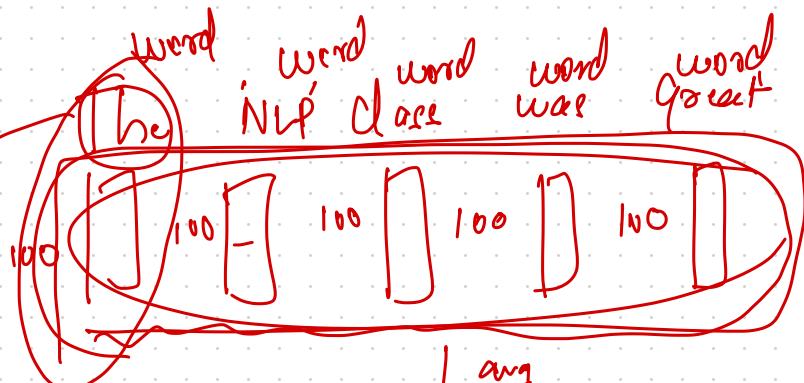


The class was great

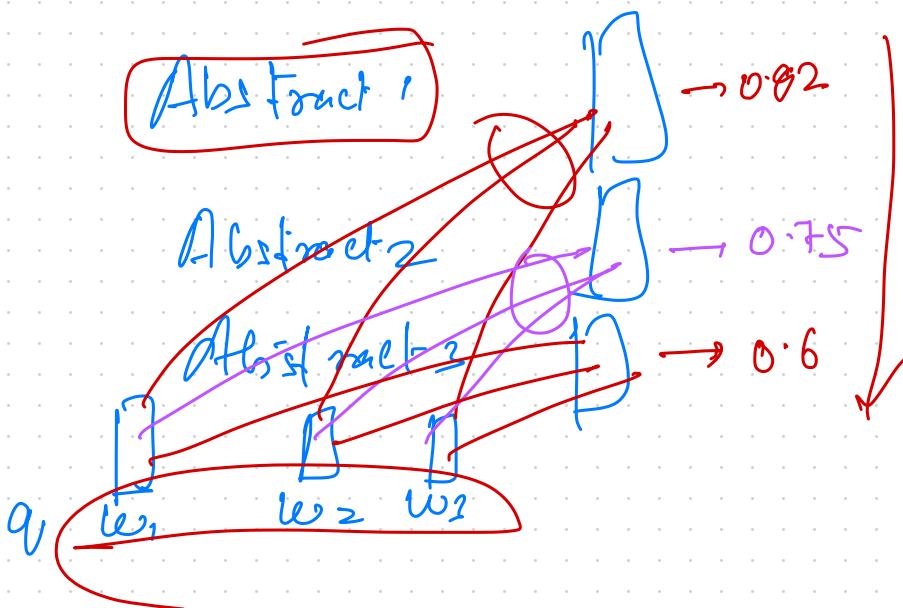
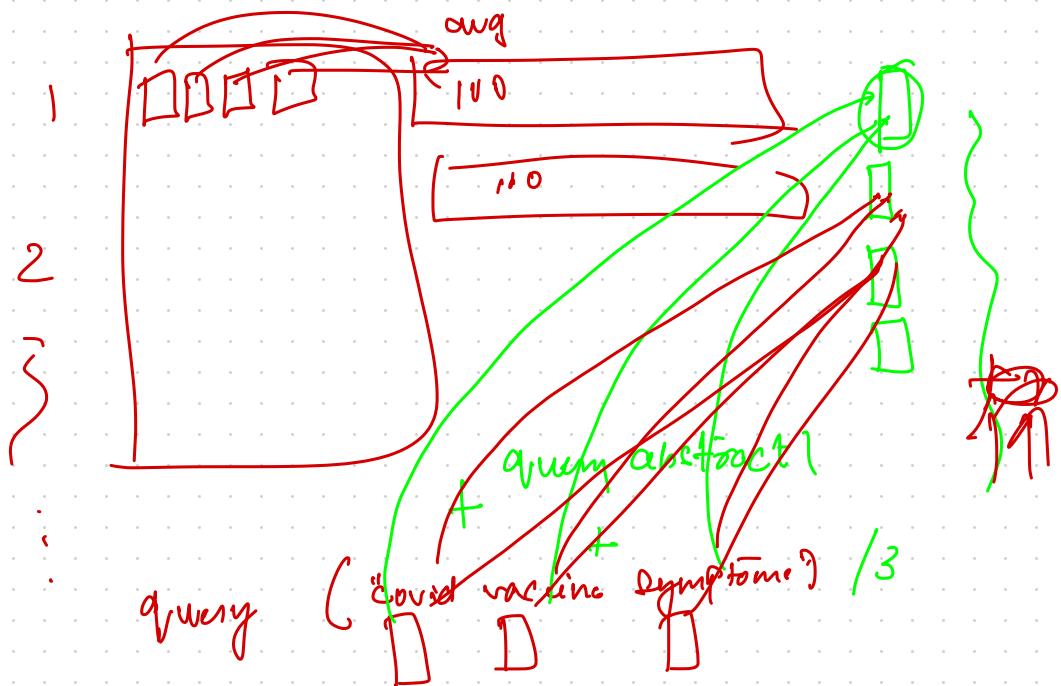
The session was great

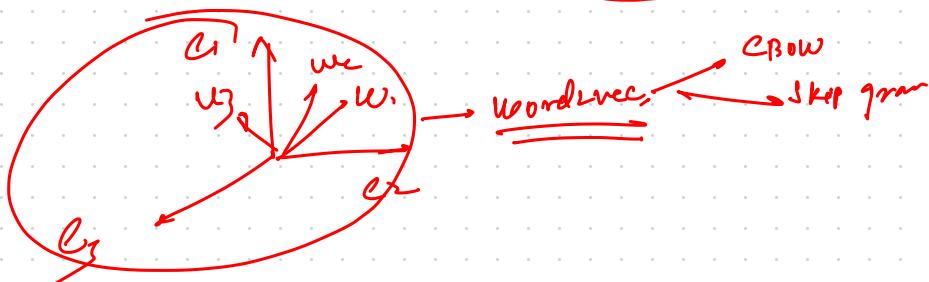
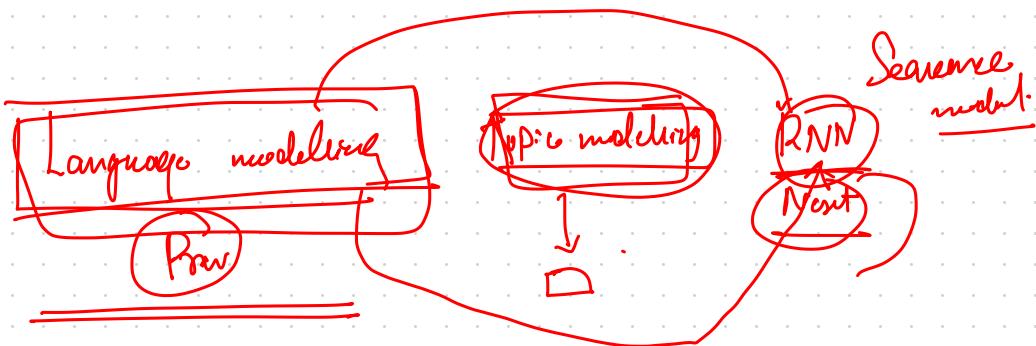
"The class was great."
The NLP is interesting

[the, class, was, great], [the, NLP, is, interesting]

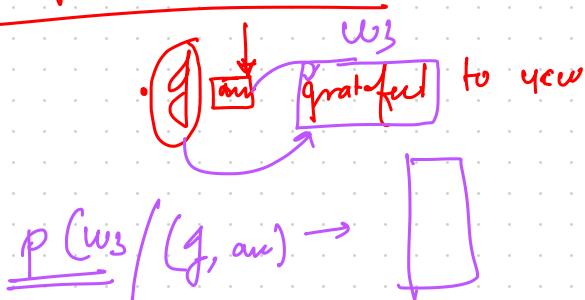


model. wv ['the'] 100 → $\sum_{i=1}^n \text{word}_i$ \rightarrow avg \rightarrow Representative vector for the sentence

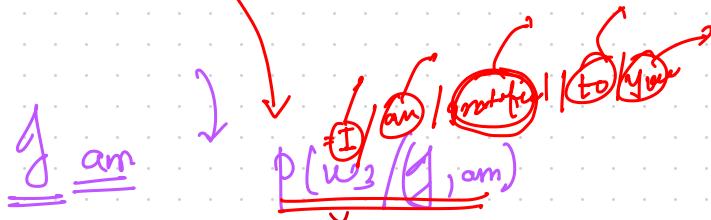




language model



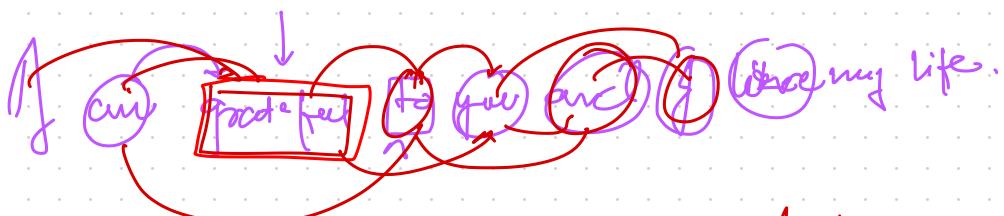
I am grateful to you



I am thankful for this life

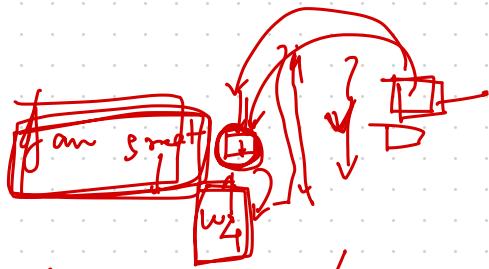


Marcovian A seqm芬



$K=1$ $K=2$

If a particular word at
is dependant for previous two words



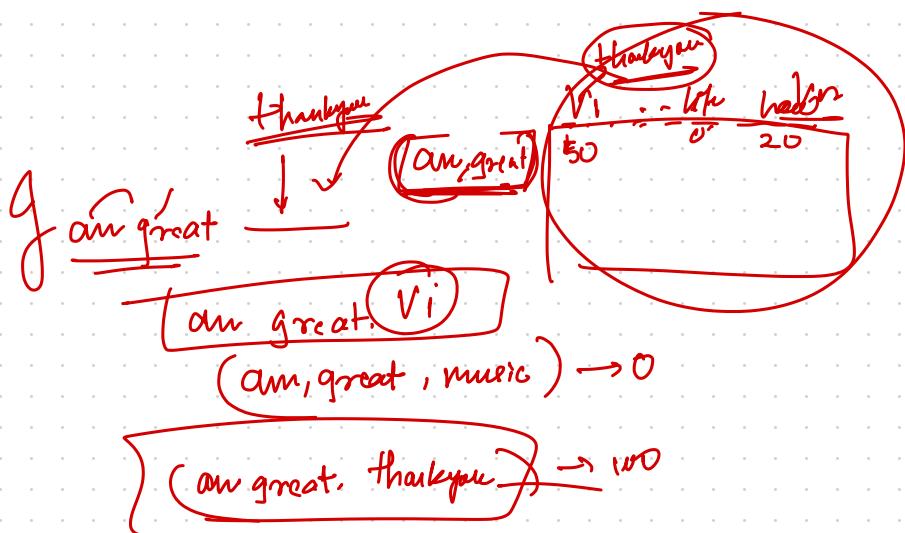
$$P \left(\text{---} \right) / \underline{(G, \text{am}, \text{great})}$$

$$K=0 \quad \sim P \left(\text{---} \right) \text{ Unigram}$$

$$K=1 \quad \sim P \left(\underline{w_4} \text{ } Vi \text{ } / \underline{\text{great}} \right) \text{ Bigram} \rightarrow$$

$$K=2 \quad \sim P \left(\underline{w_4} = Vi / \underline{(\text{great}, \text{am})} \right) \text{ Trigram}$$

$$K=3 \quad \sim P \left(\underline{w_4} = Vi / \underline{(\text{great}, \text{am}, f)} \right) \text{ Quadgram}$$



Handwritten notes from a lesson on 'How are you?'. The notes include:

- A drawing of a television set with the number '2' below it.
- A circle containing the text 'our good' with the number '2' below it.
- The text 'good how' with a bracket underneath and the number '1' below it.
- The question 'how are' underlined with the number '1' below it.
- The question 'are you' underlined with the number '1' below it.
- The number '1' at the bottom center.

good

g x
are x
good
have x
are x
good x
too x

$f_{\text{an}} = p(-/\underline{\text{an}})$

P ($w_3 = v_i$ / Jan)

I want good people to love my country. Jhore India

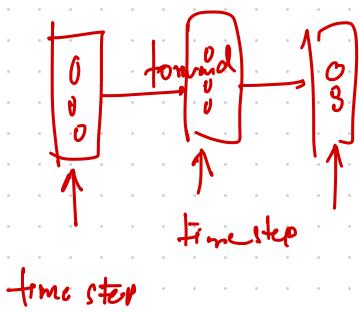
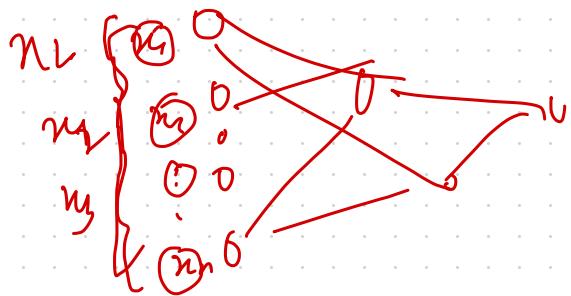
K gran

P C

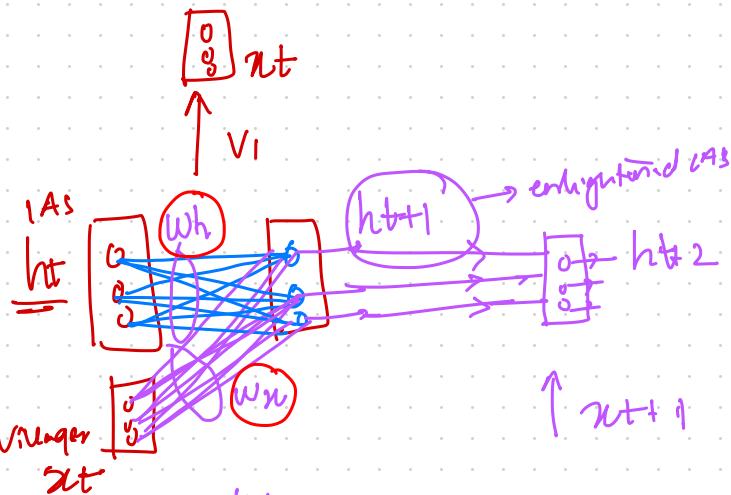
100

Recurrent Neural Networks

9:51 → 9:55 10:00 a.m.



1AS



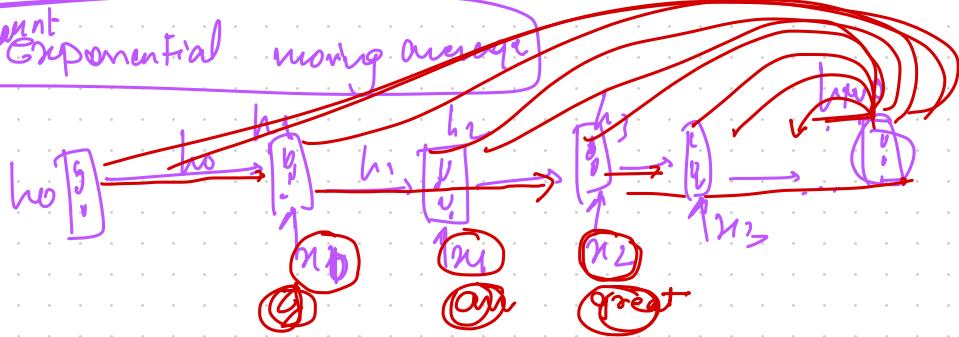
$$\underline{h_{t+1}} = \text{tanh}(\underline{W_h} \times \underline{h_t} + \underline{W_x} \underline{x_{t+b}} + b)$$

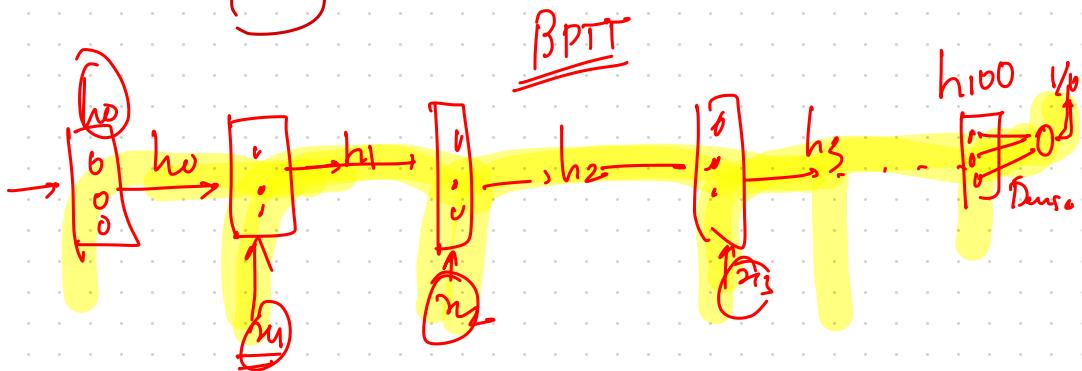
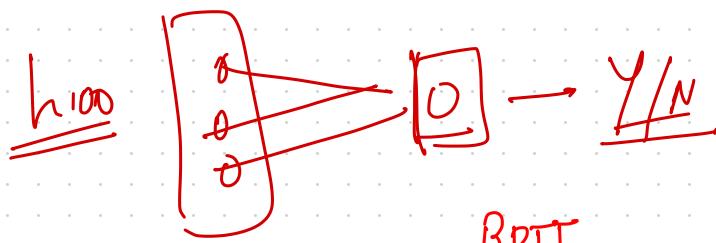
$$\underline{h_{t+2}} = \text{tanh}(\underline{W_h} \times \underline{h_{t+1}} + \underline{W_x} \underline{x_{t+1+b}} + b)$$

$$h_{t+2} = \text{tanh}(\underline{W_h} \times \underline{\text{tanh}(W_h \times h_t + W_x x_t + b)} + W_x x_{t+1+b}) + b$$

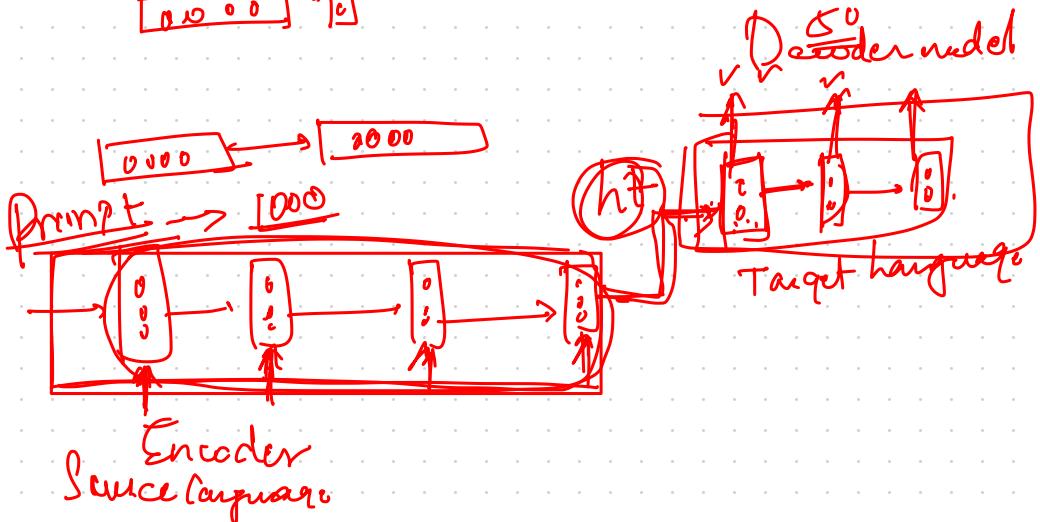
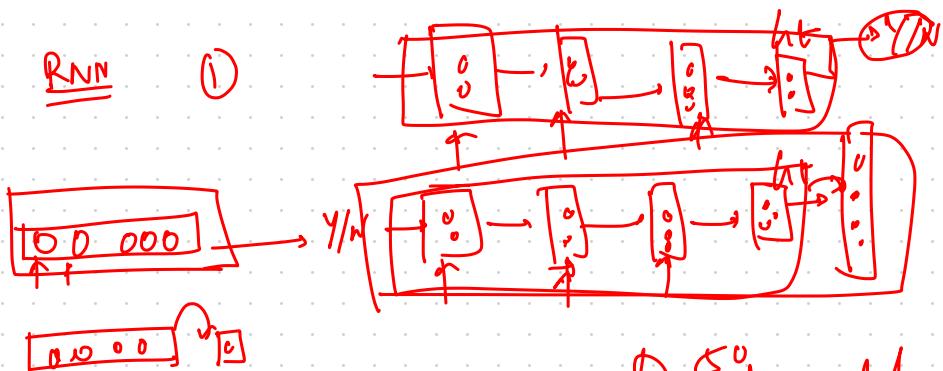
$$h_{t+2} = \text{tanh}(\underline{W_h} \times \underline{h_{t+1}} + \underline{W_x} \underline{x_{t+1+b}} + b)$$

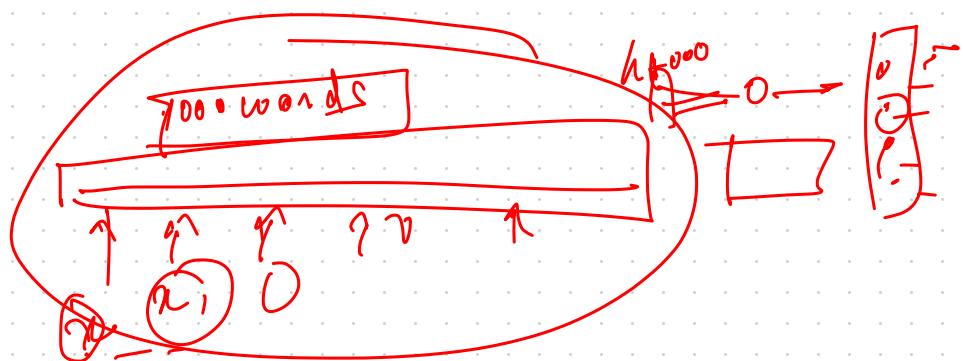
Non-linear moving average



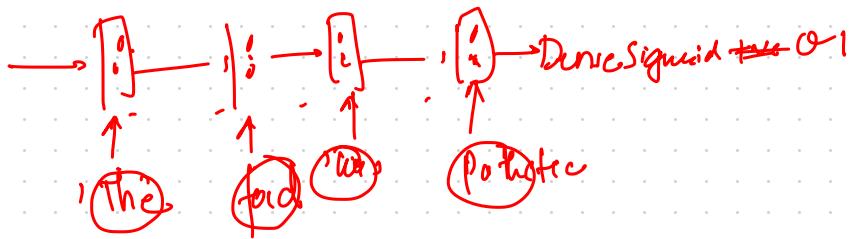


RNN ①





NLP



1000x10

π	γ
1 2 3 4 0 0 0 0 0 0	
1 5 3 6 7 8 9 0 0 0	
..... 0 0 0	

1000

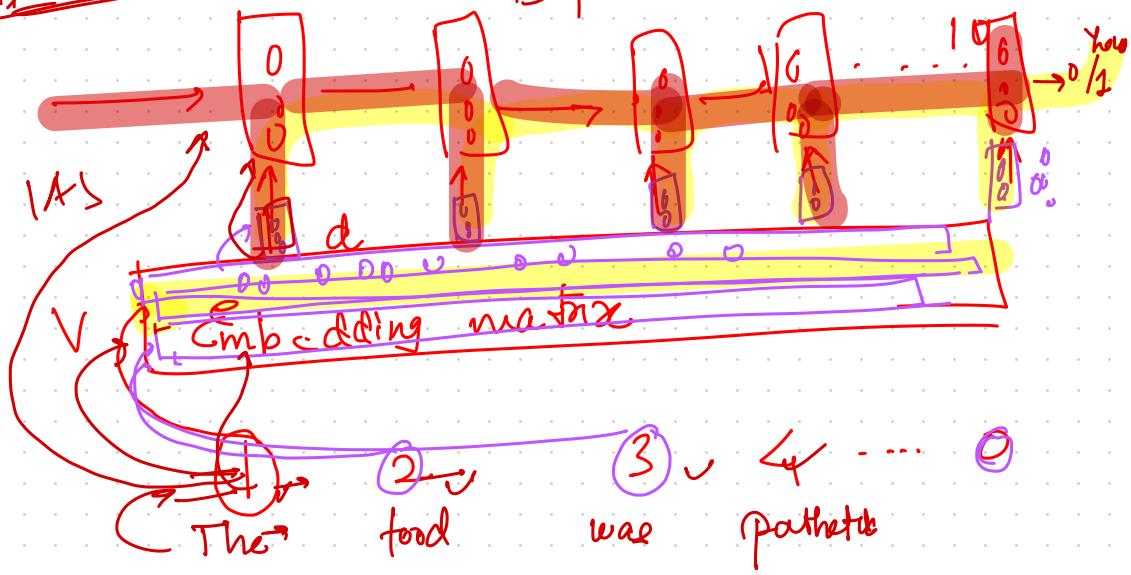
\rightarrow max-length = 10

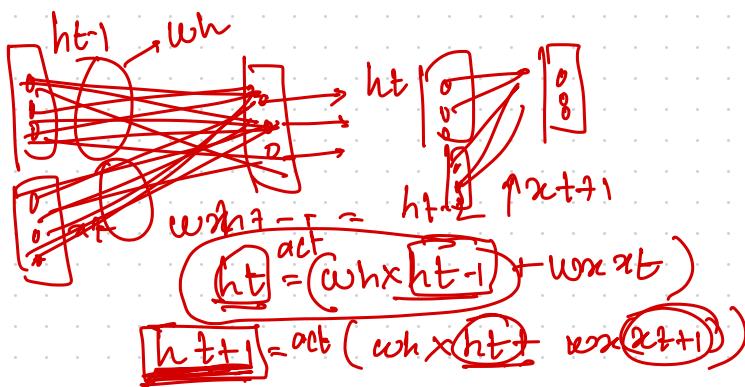
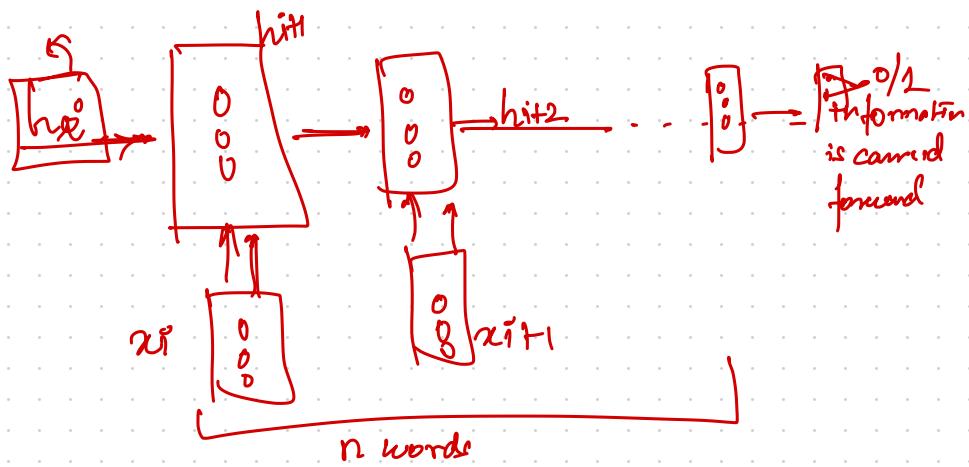
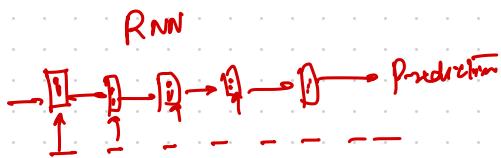
①
②

(The food was pathetic)	0 0 0
The audience was chicken, definitely a result	hot
3	

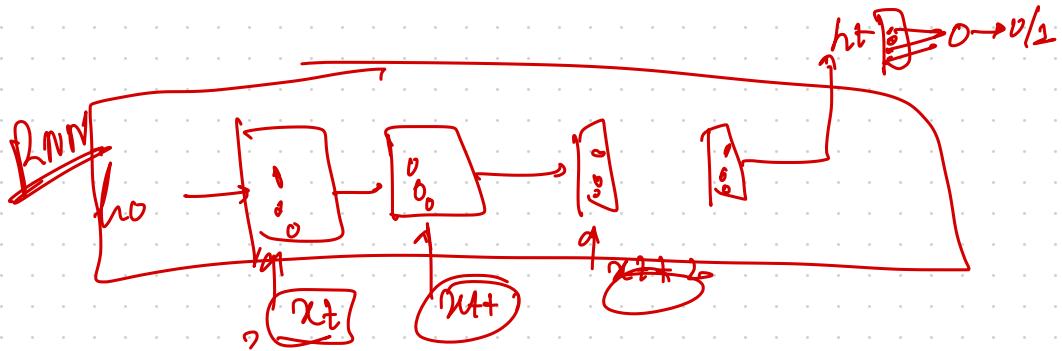
BPTT case

- BPTT -





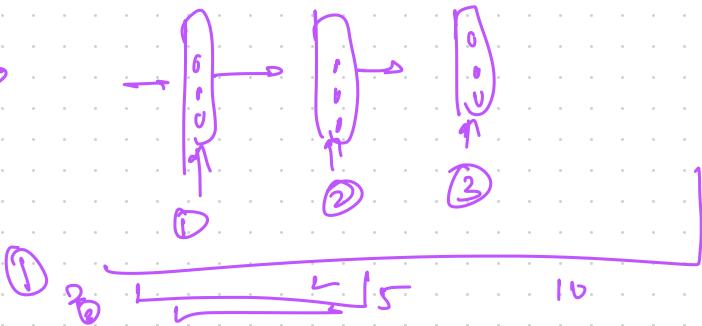
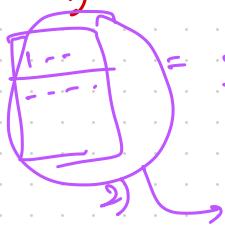
-	+rc
-	-rc
-	+rc



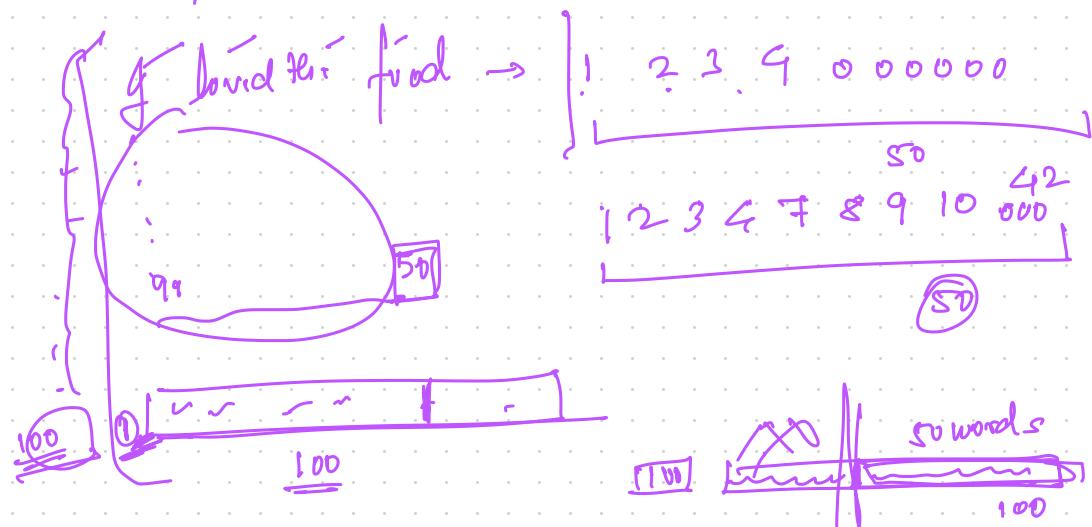
100 words

1 I found the food	+rc
2 the mother	-rc
3 the chicken was late	-rc

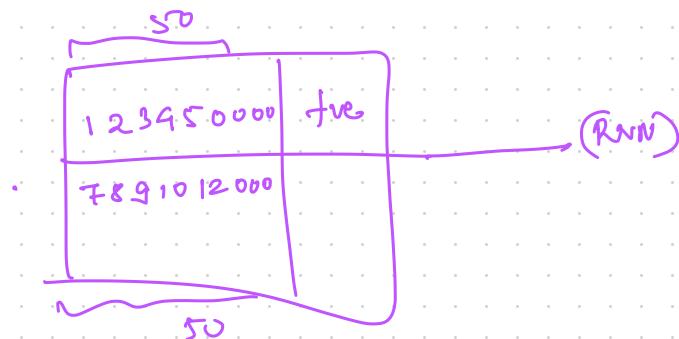
i) We will ~~convert~~ ^{map} each word into an integer
tokenizer. text -> sequences ($df['txt']$)

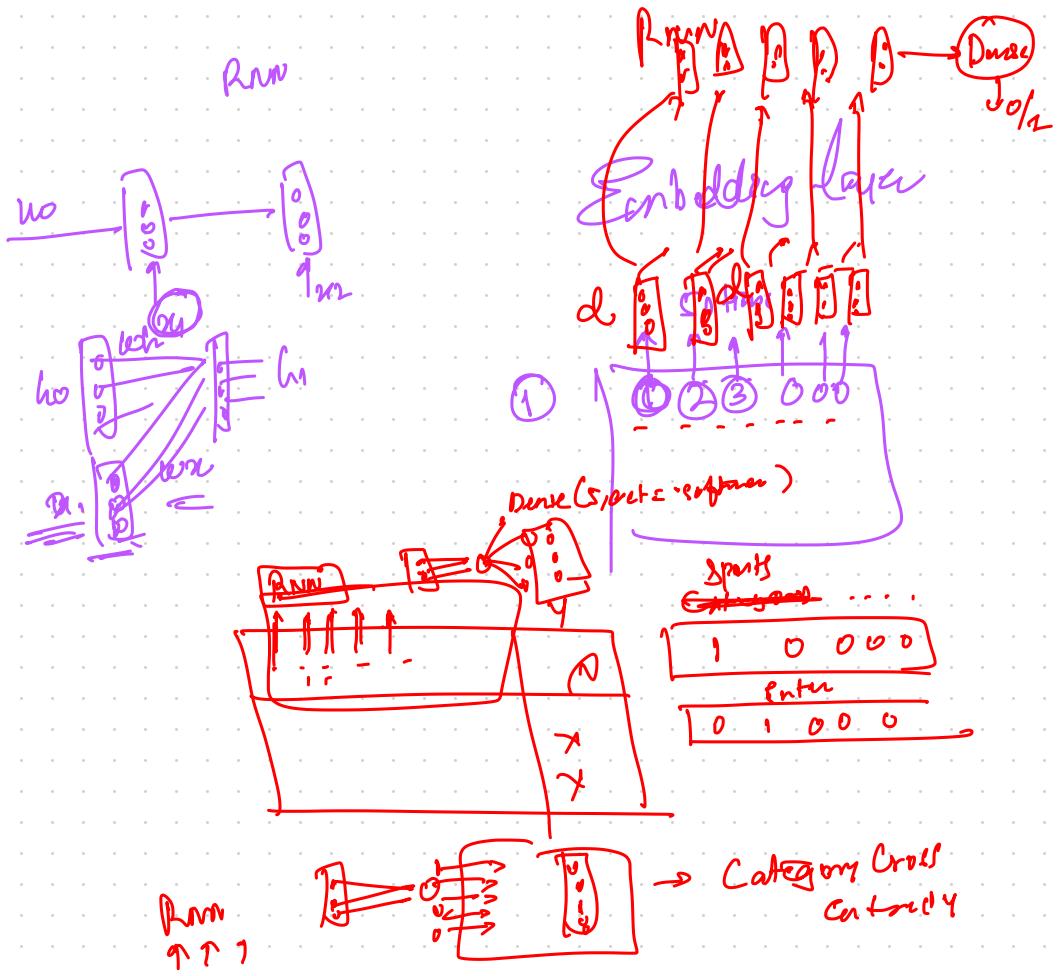


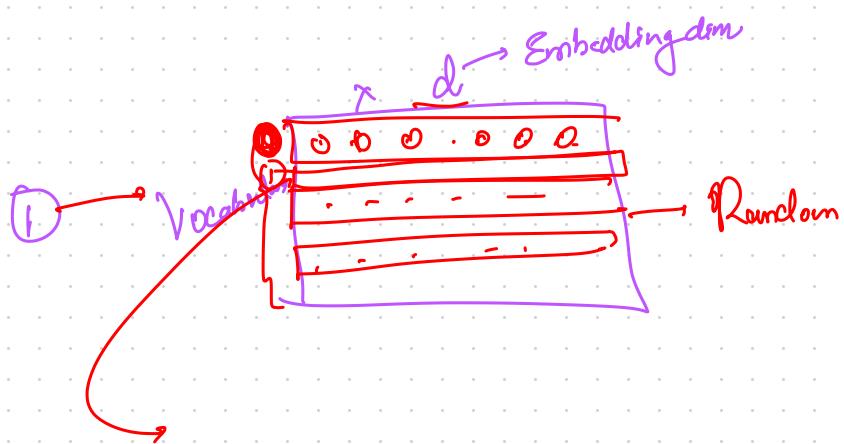
Padding

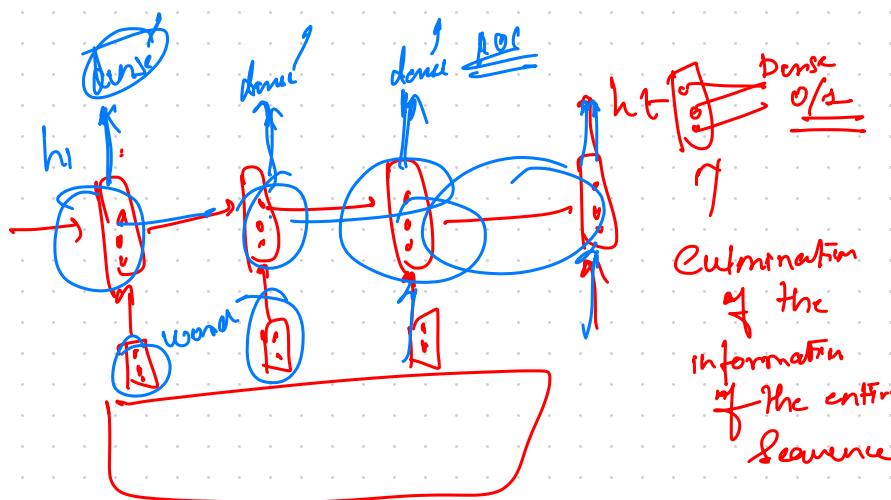
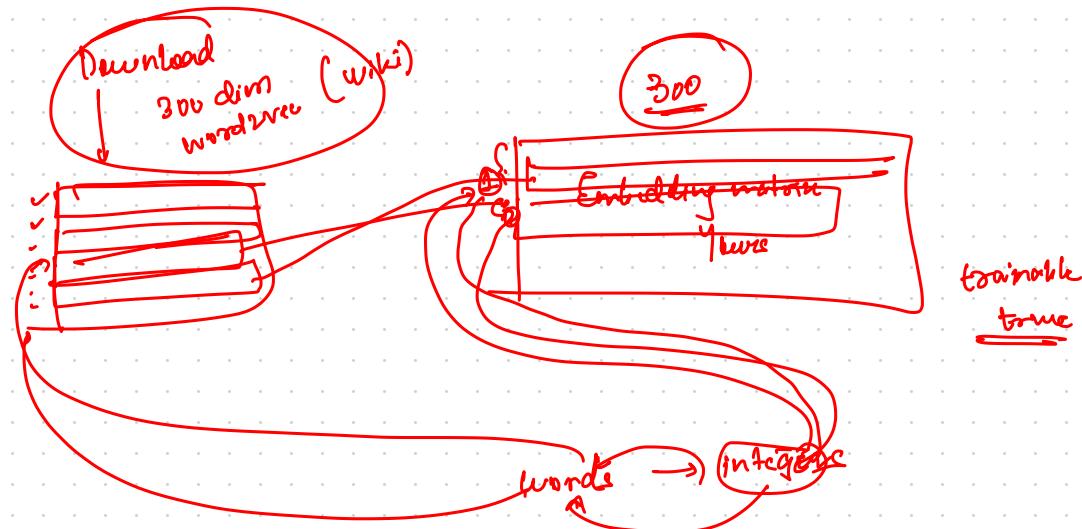


= Pool-sequences (dtC: 'integer test')

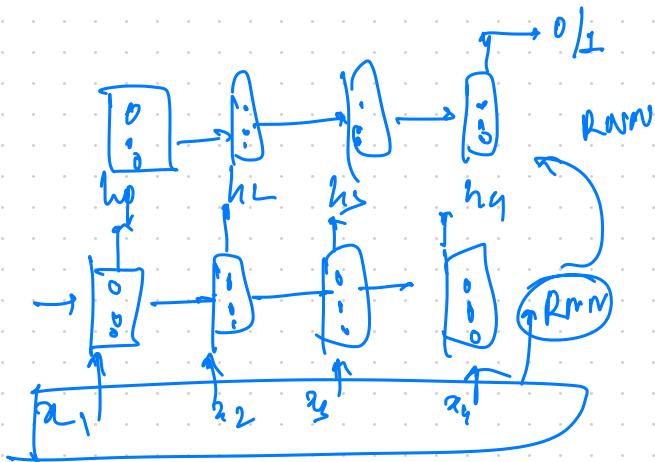






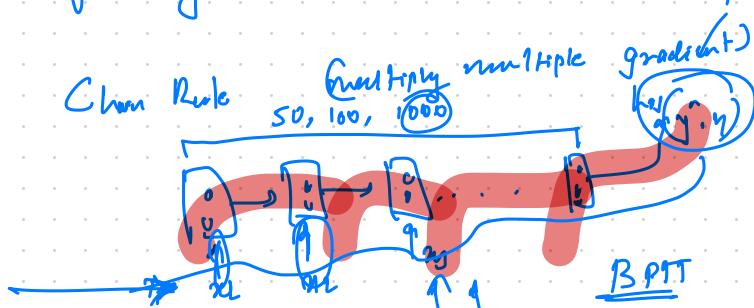


fwr

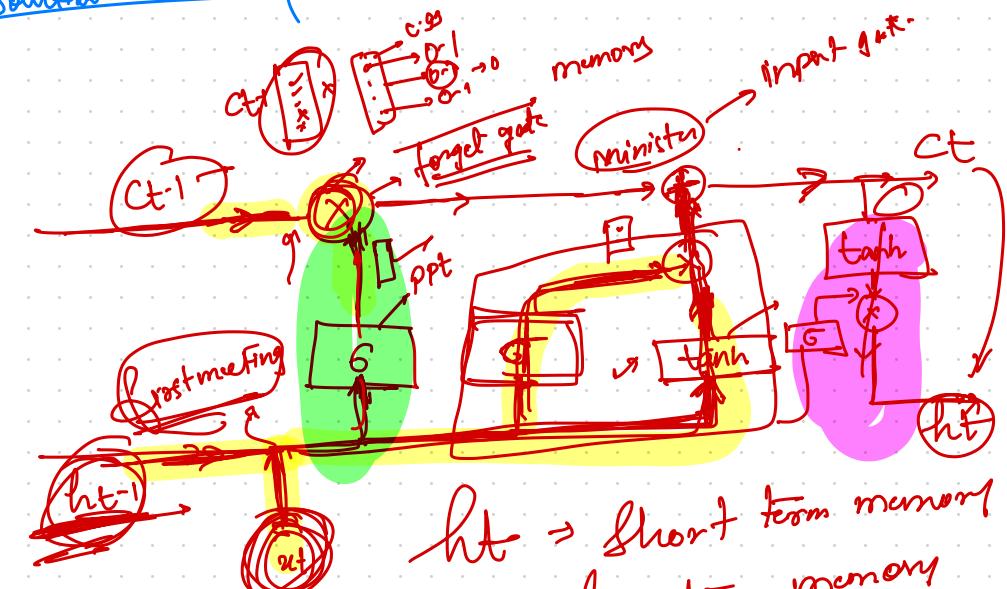


RNN → some problems

Vanishing Grad



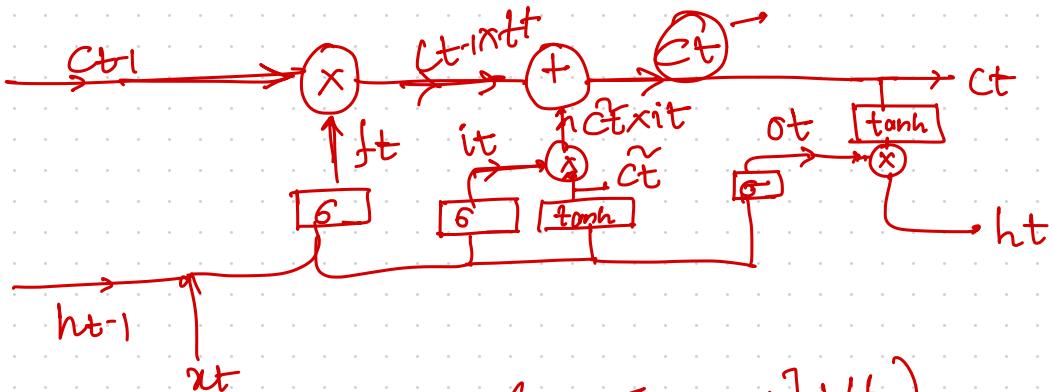
Solution to RNN's Problem:



$ht \rightarrow$ short term memory
 $ct \rightarrow$ long term memory

$ht/ct \rightarrow$ memory manager
for ct

[task] → word



$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(W_c [h_{t-1}, x_t] + b_c)$$

$$c_t = c_{t-1} \times f_t + i_t \times \tilde{c}_t$$

$$h_t = \tanh(c_t) \times o_t$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$