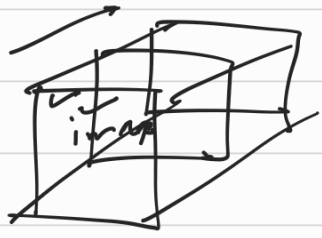
Natural Language Processing

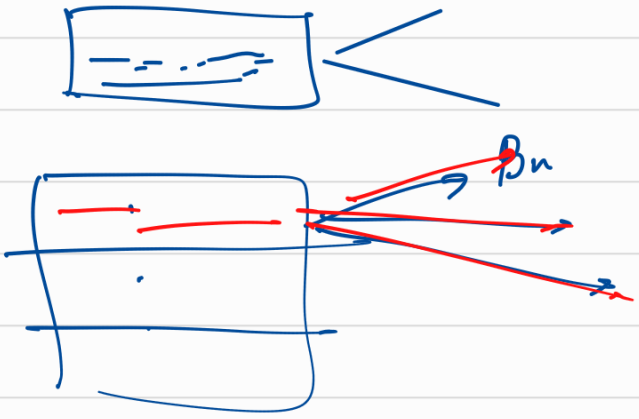
to make machines → human language.



→ doesn't have a universal representation in numbers.

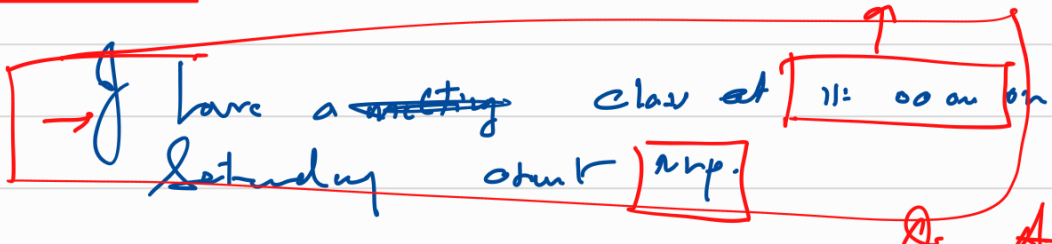
→ modeling.

→ Classification

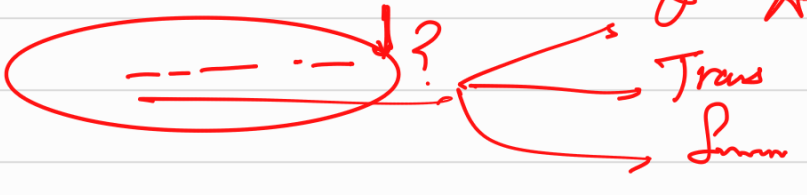


Timing, Topic

→ Named Entity



Large Language modeling



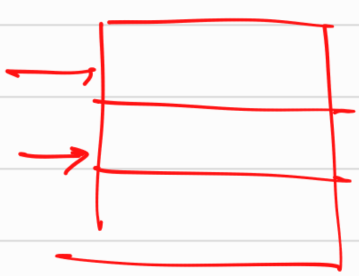
Definitions:

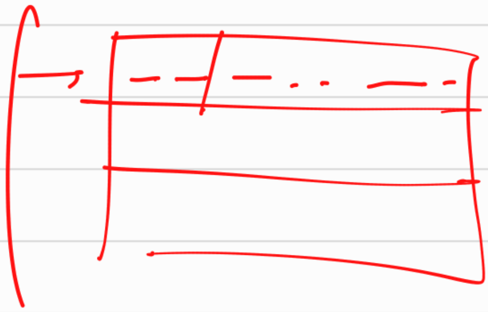
Corpus: ? → set / Union of document.

Document: ??

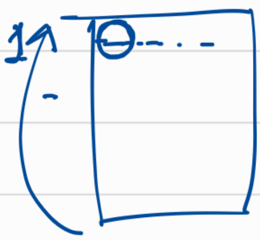
Token

Chatgpt → Corpus → all the internet data.





1 Token: ?? → 1 word



1 Token → Smaller

1 word (Running) → Run, ing

I am hav . break

I am having , breakfast

Vocabulary:

Union / Collection of Tokens

NLP
1. text → numbers
2. Model.

text → numbers (Embedding).

Prediction — Frequency

text → numbers

Frequency

Word

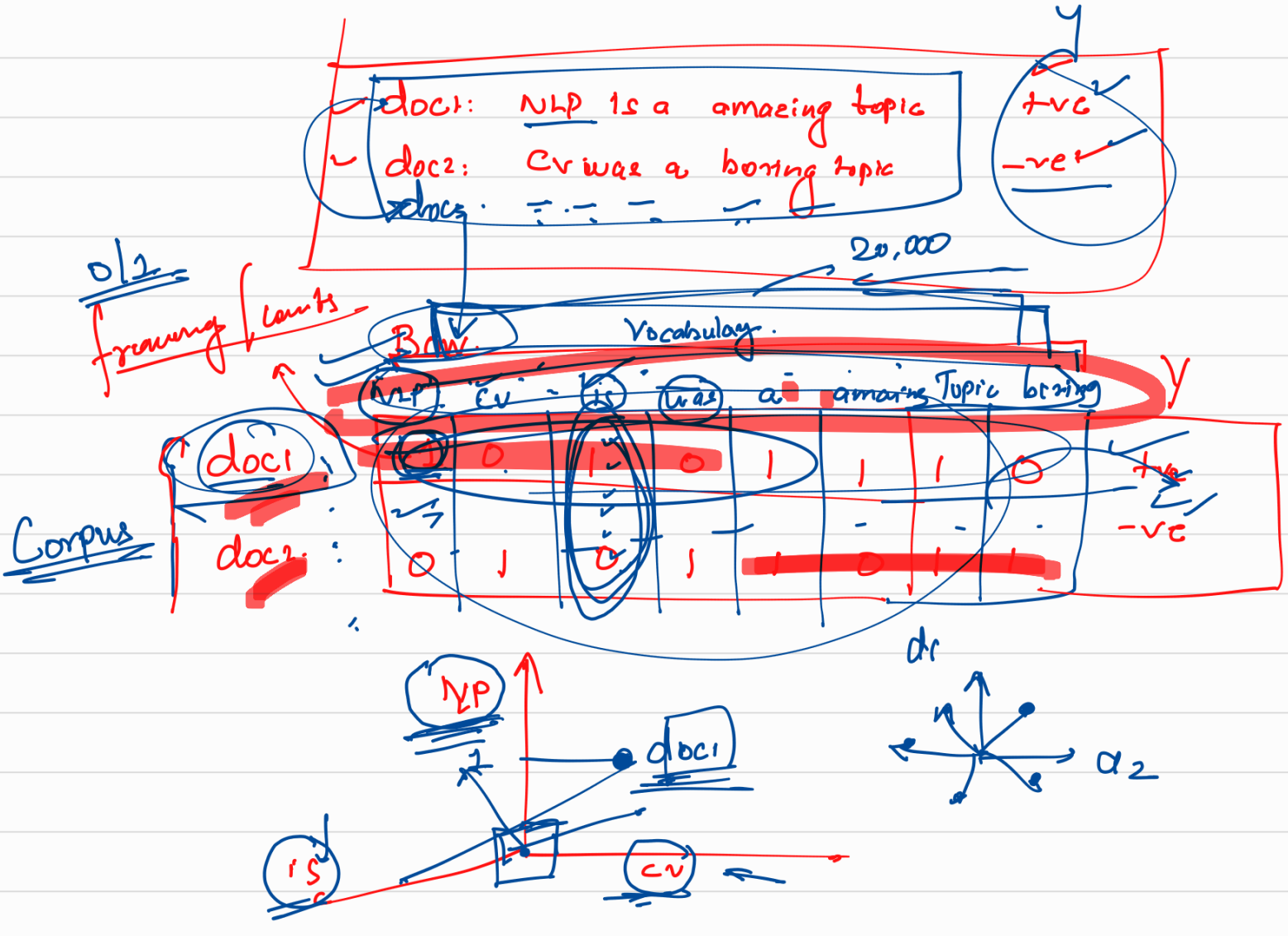
Word →

1 dimensional

dense

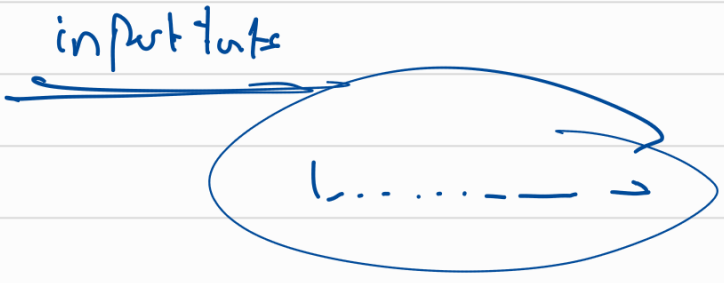
Embedding

Video



High dimensions → Sparsity

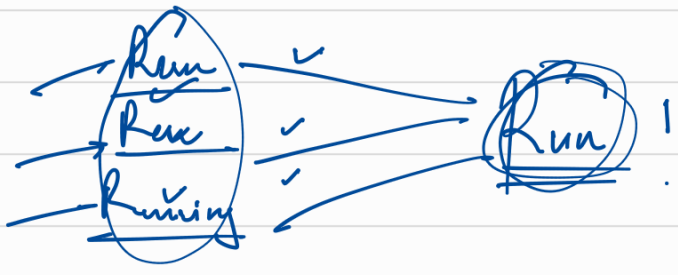
Ignores → order

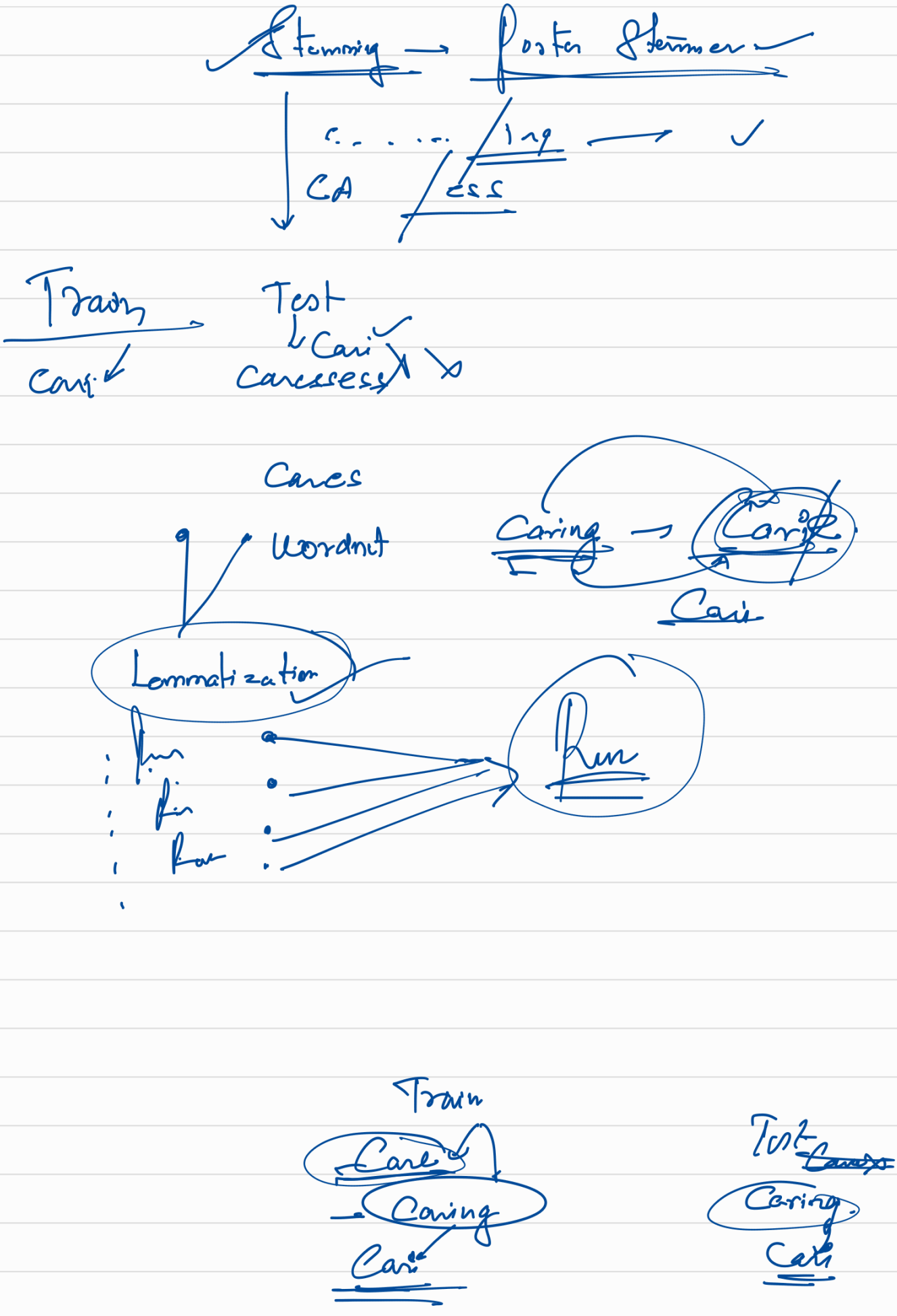


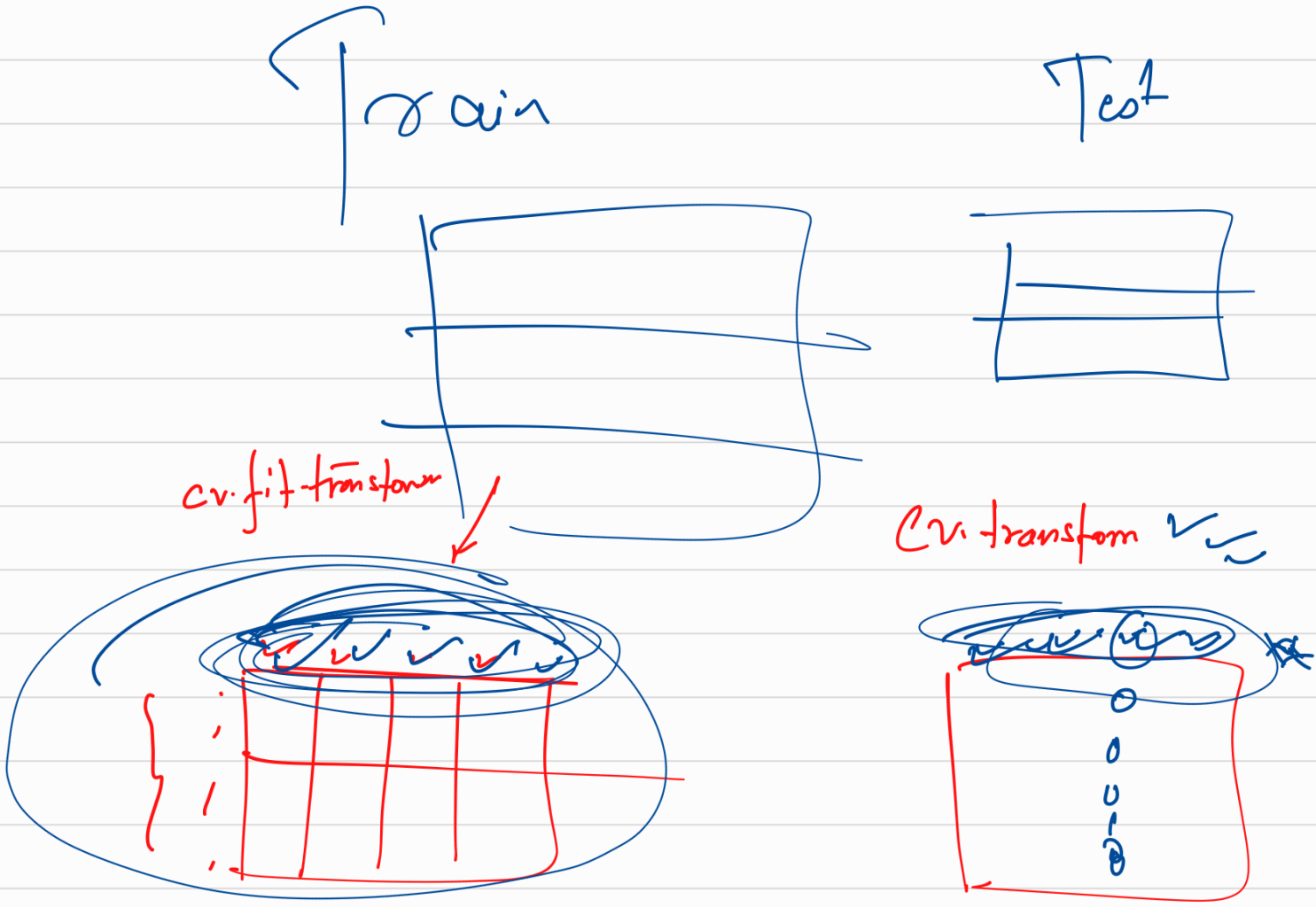
12: IDF

Higher dimensionality →

① I can remove stop words, remove punctuations, remove #, , remove numbers ✓

② 

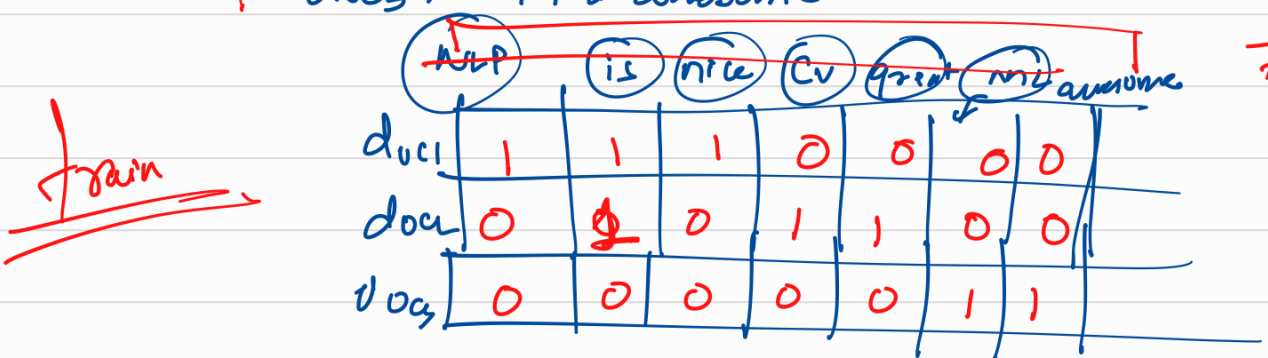

Stemming → Porter Stemmer
CA
ess
Train
Test
Caring → Caring
Caring
Lemmatization
Run
Train
Caring
Caring
Test
Caring
Caring



doc1: NLP is nice

doc2: CV is great.

doc3: ML is awesome

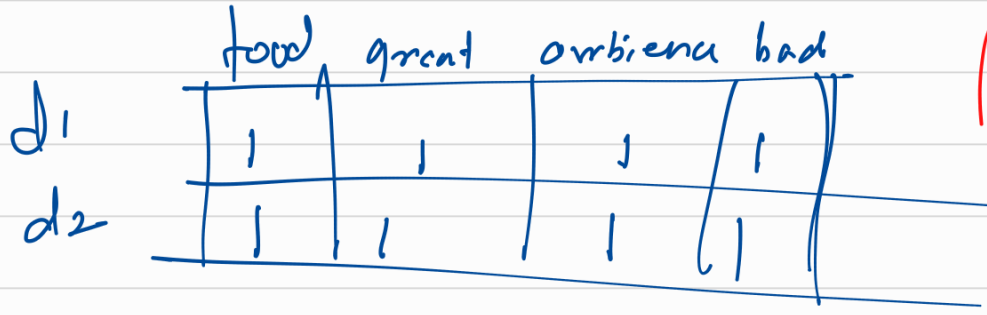


Count

order of the words

doc1 : food great ambience bad

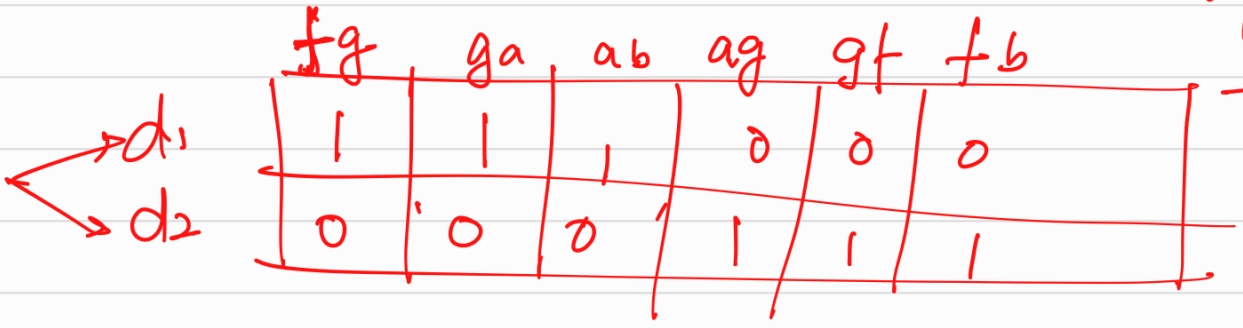
doc2 : ambience great food bad.



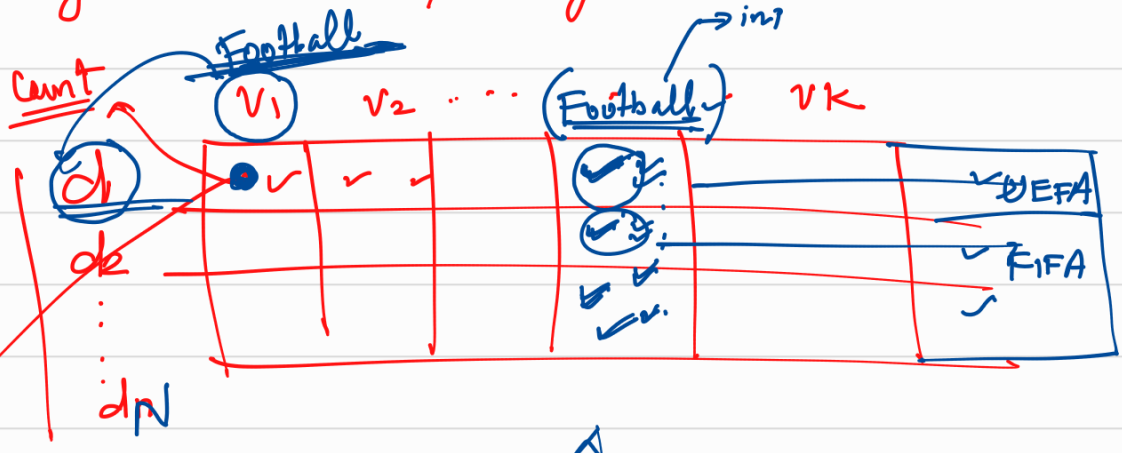
Bigram

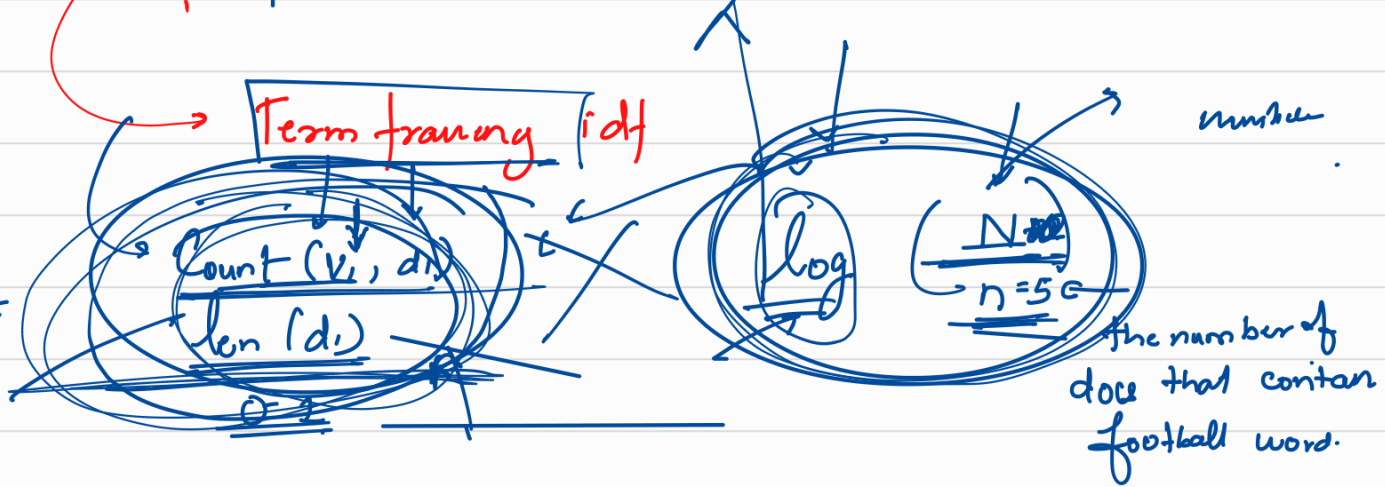
Trigram

gram

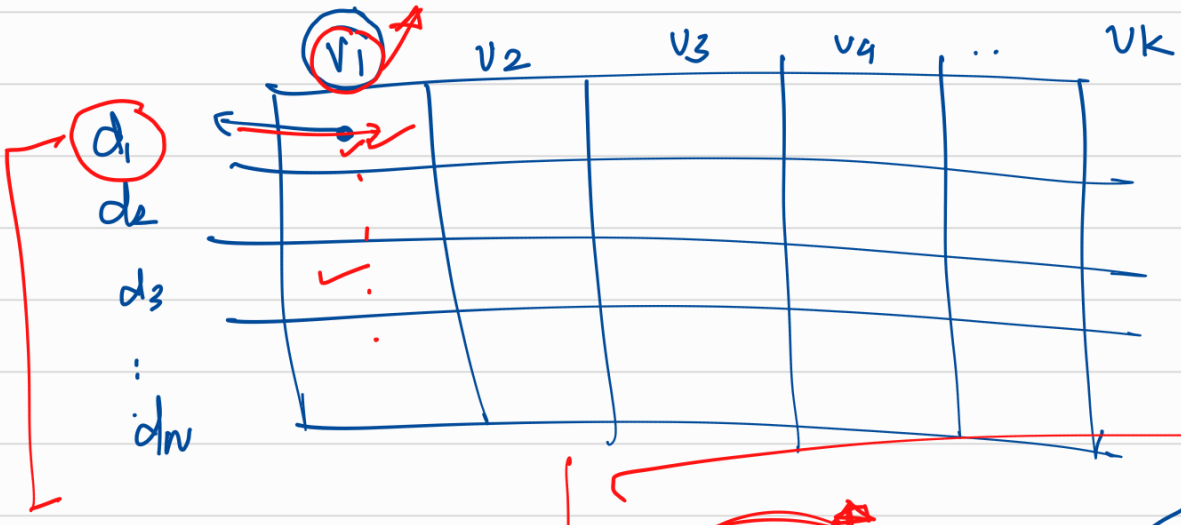


Term frequency inverse document frequency (Tf idf) →



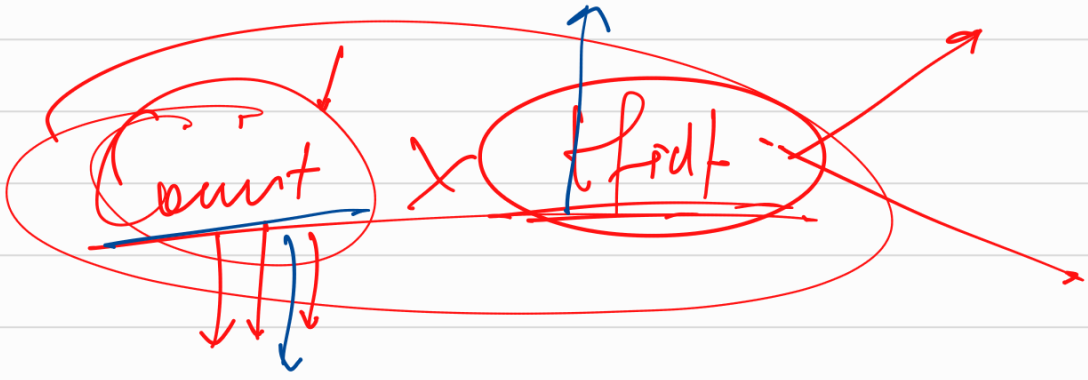


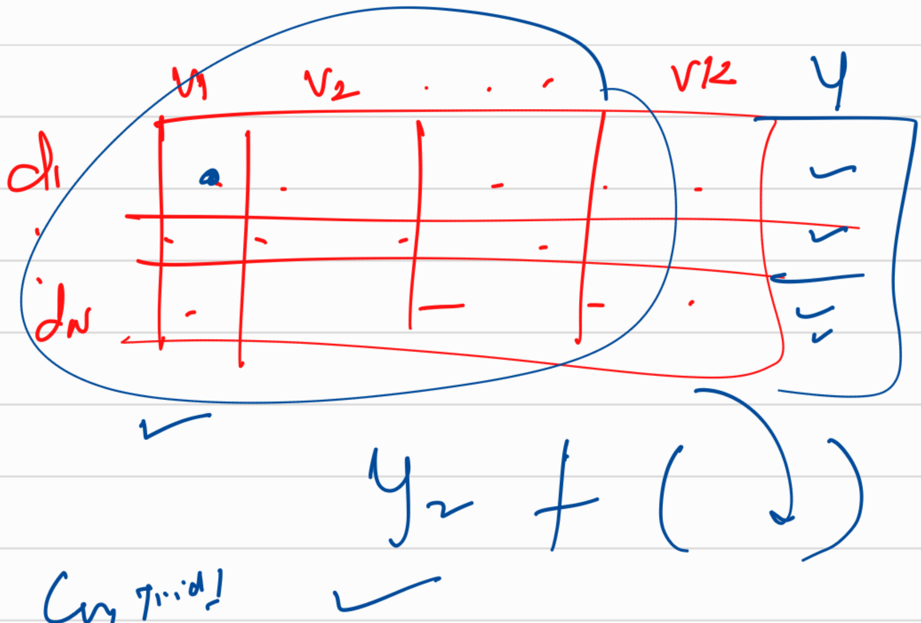
$$\underline{tfidf} = imp(w_i, d_i) \times \log\left(\frac{N}{n}\right).$$



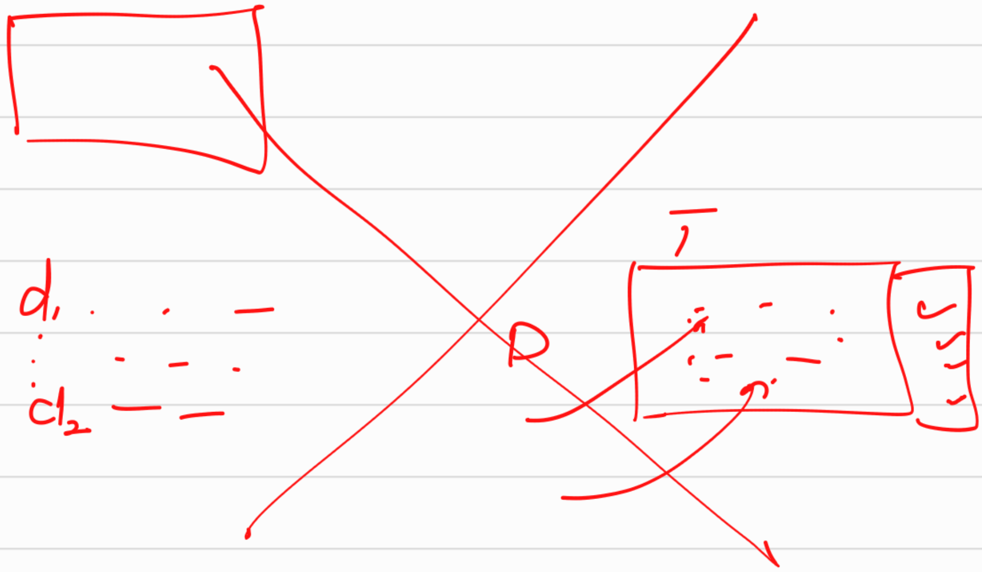
idf

$$tfidf(v_1, d_1) = \frac{Count(v_1, d_1)}{len(d_1)} \times \log \frac{N_{1000+1}}{n}$$

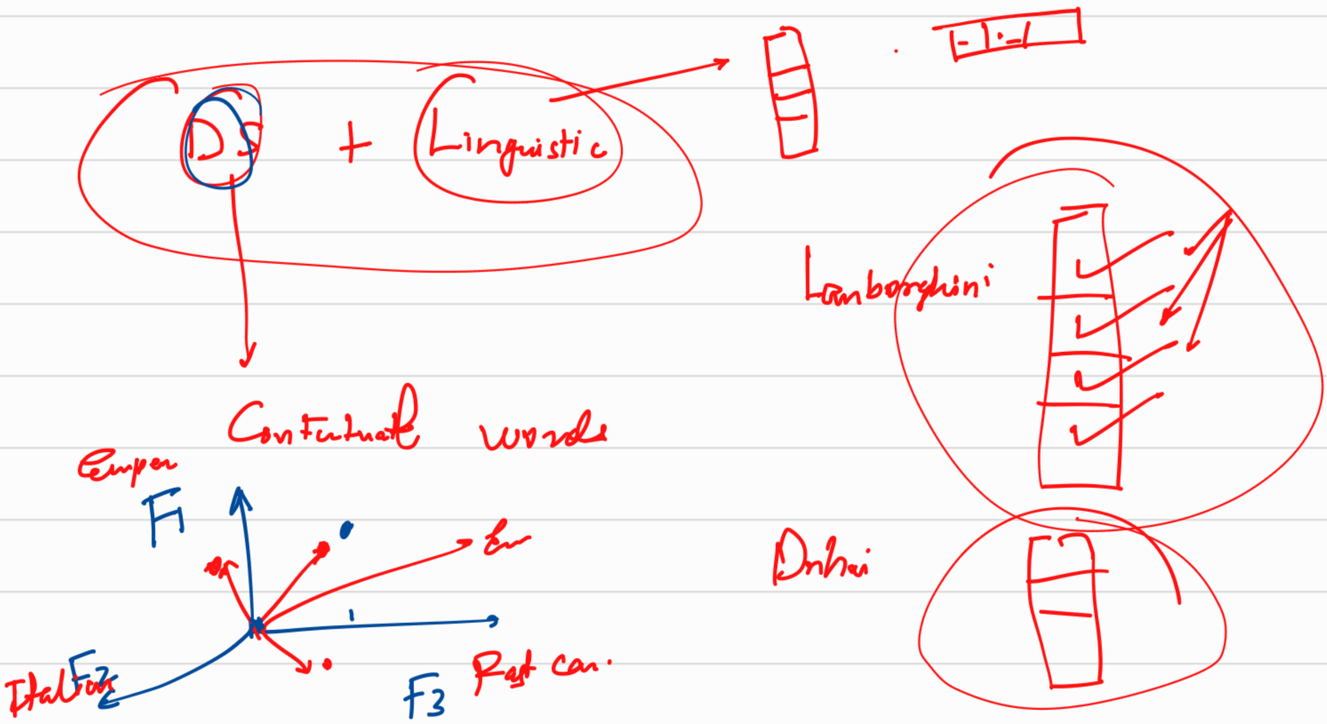




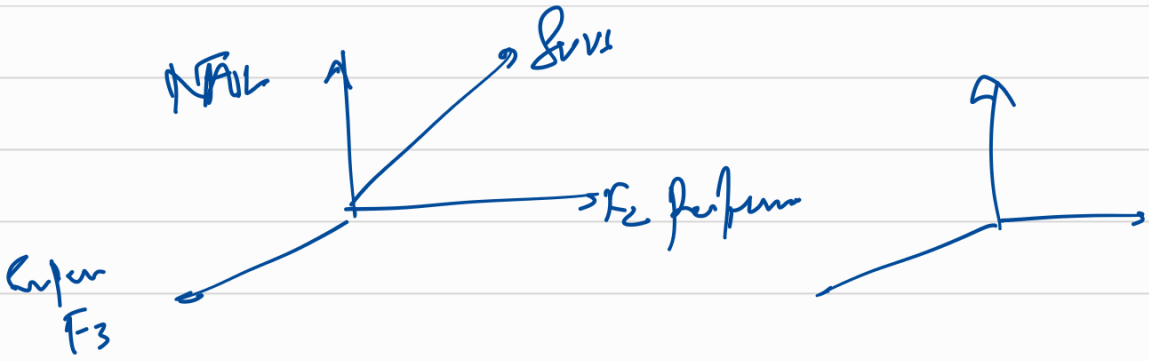
v1
v2
v12
y
d1
dN
y2 + ( )

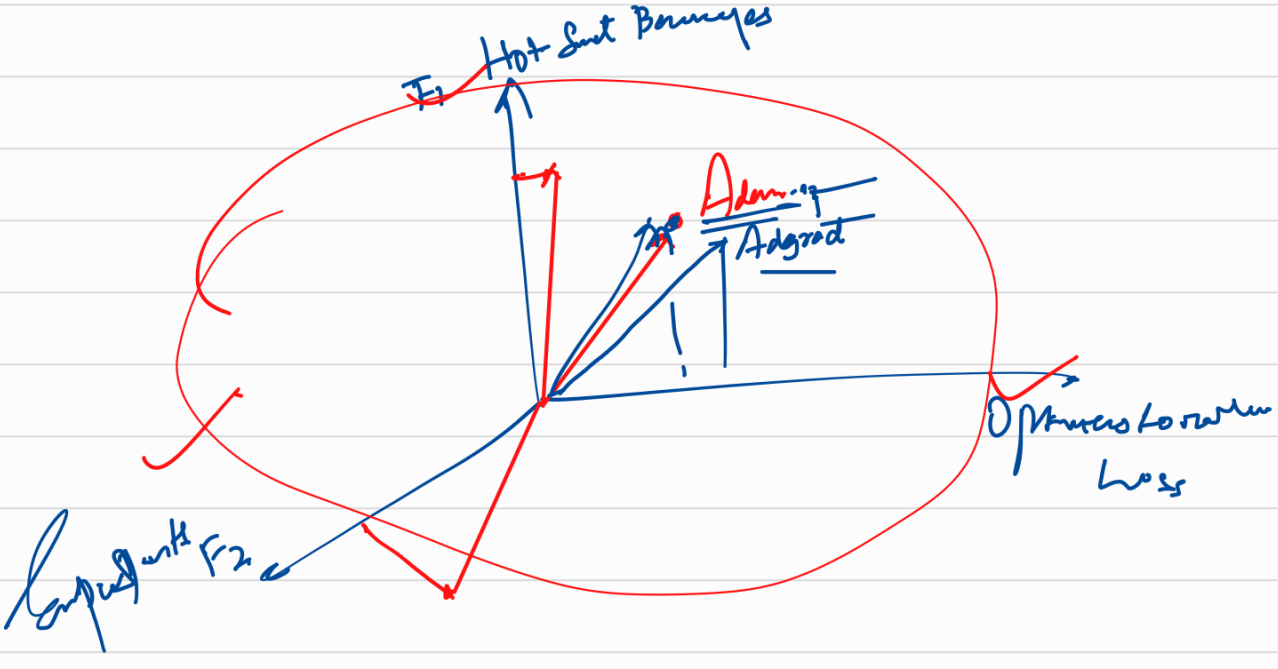


d1
d2
D

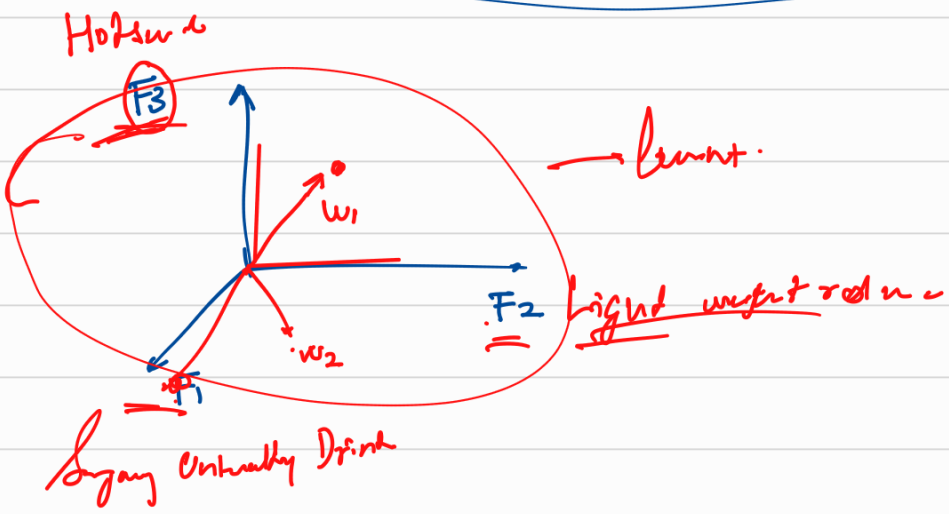


DS + Linguistic
Contextual words
F1
F2
F3
Fast car.
Lamborghini
Dubai

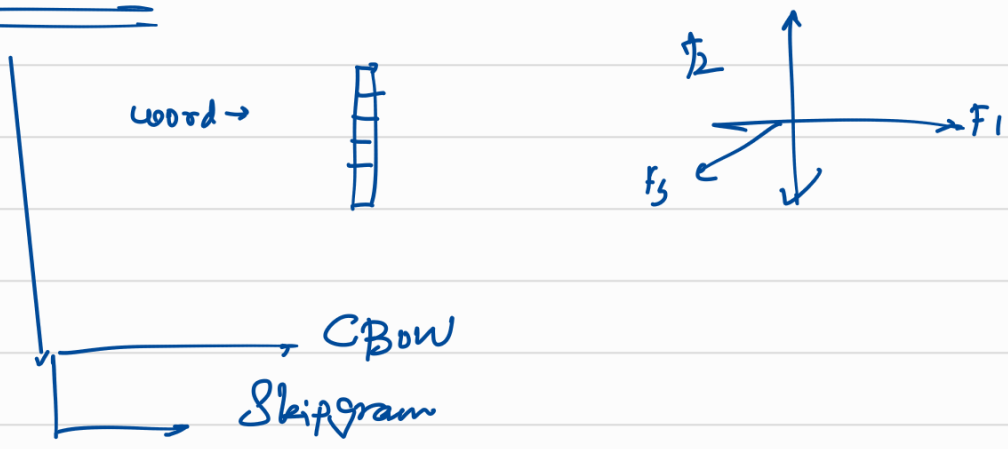



Hot Start Bounces
Adam
Adgrad
Optimizer Loss

Words Similar meaning → high cosine similarity

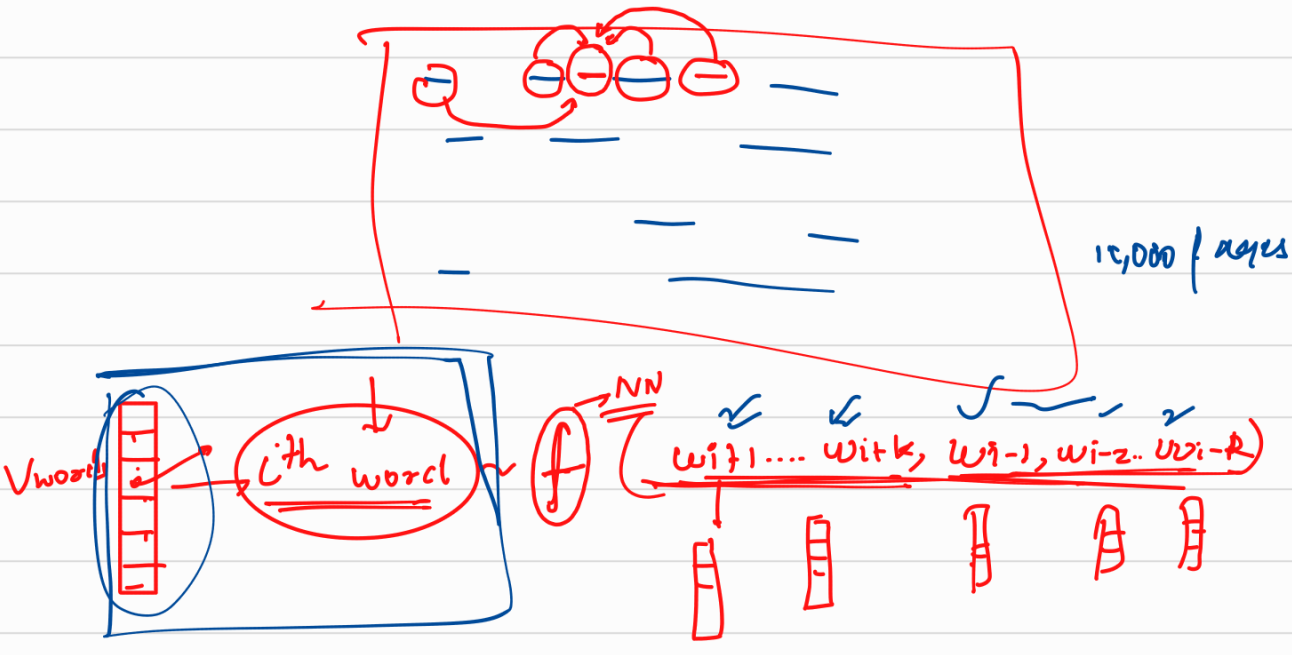


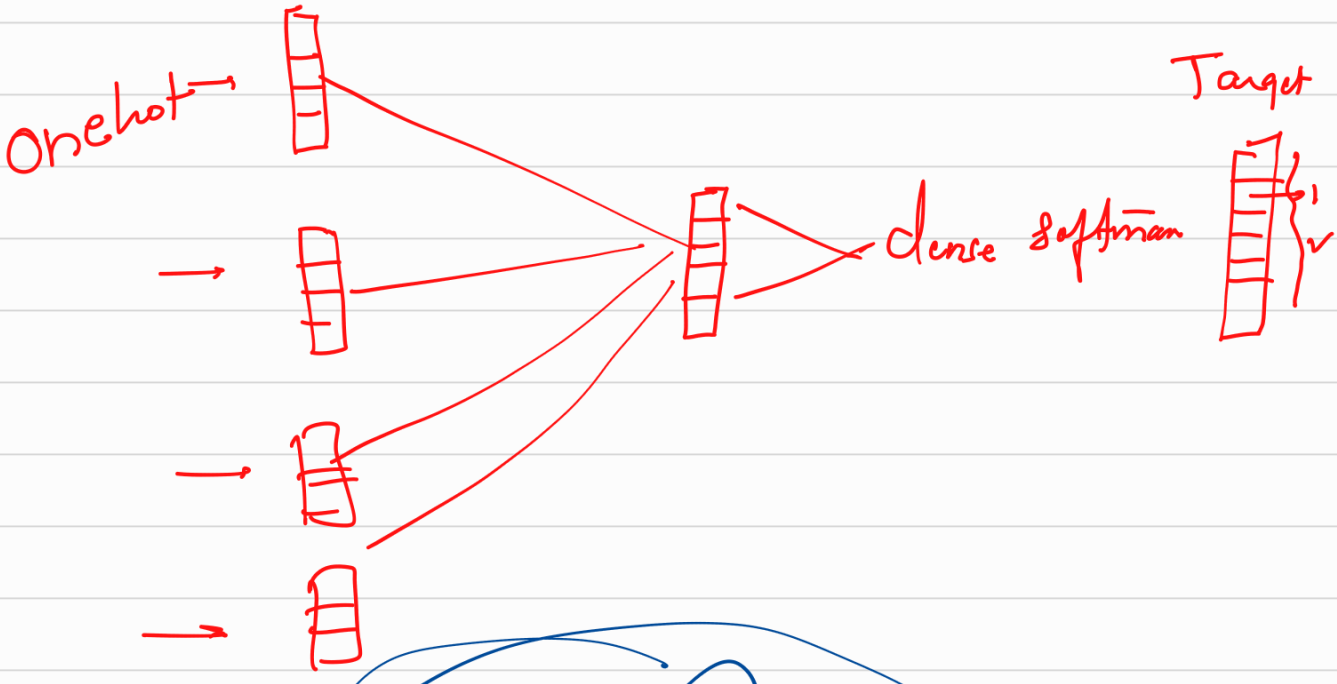
Word2vec

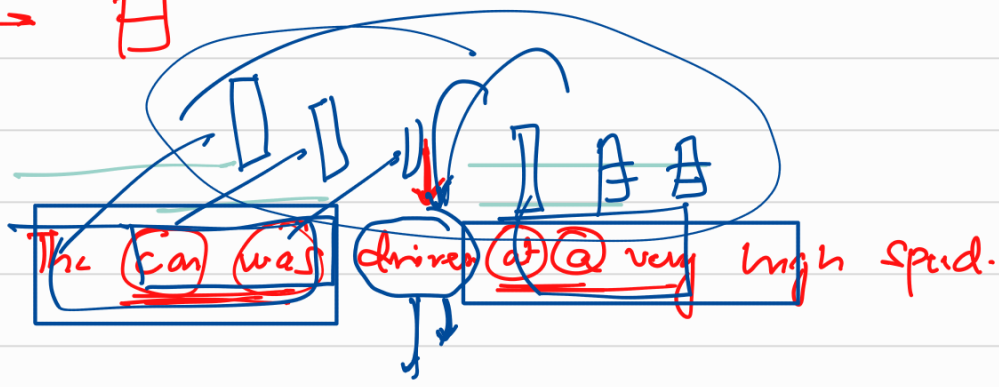


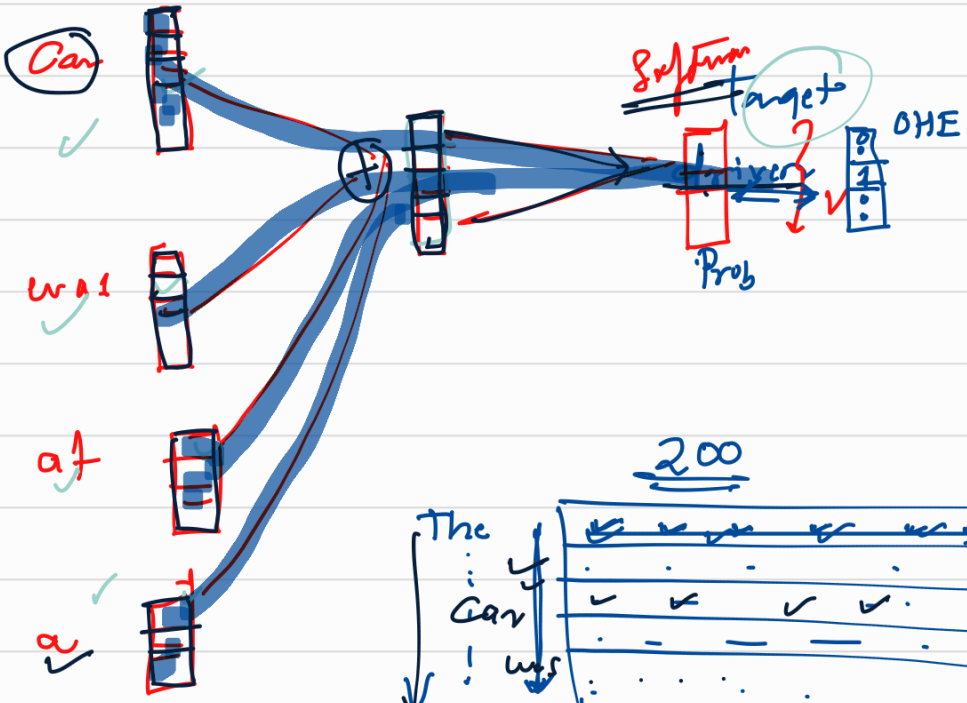
CBOW

Corpus — Wikipedia / Google news

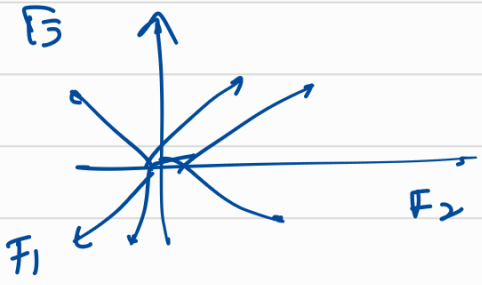


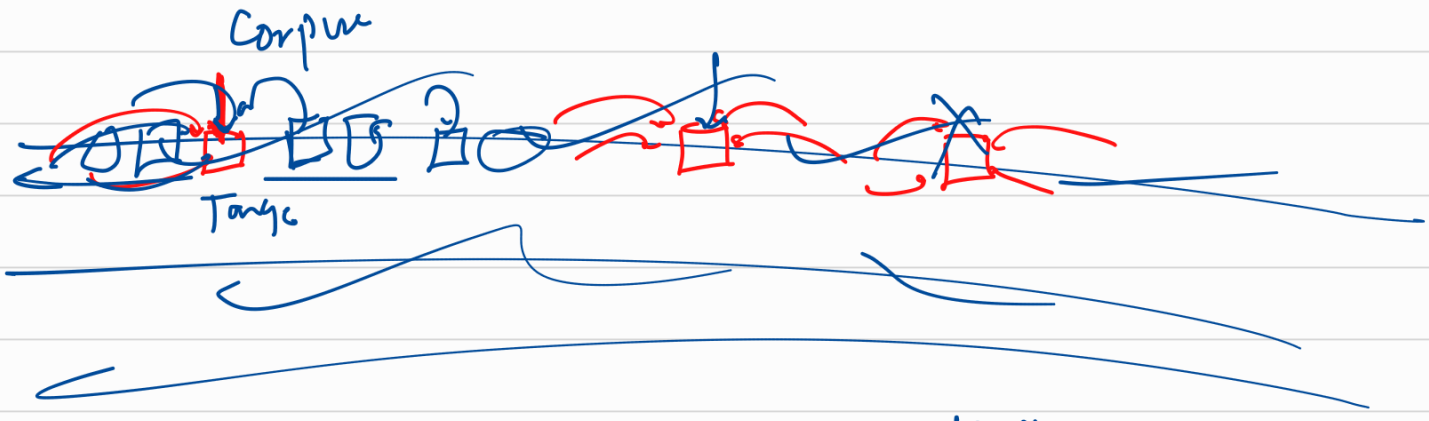




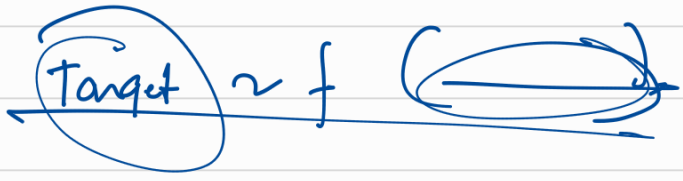


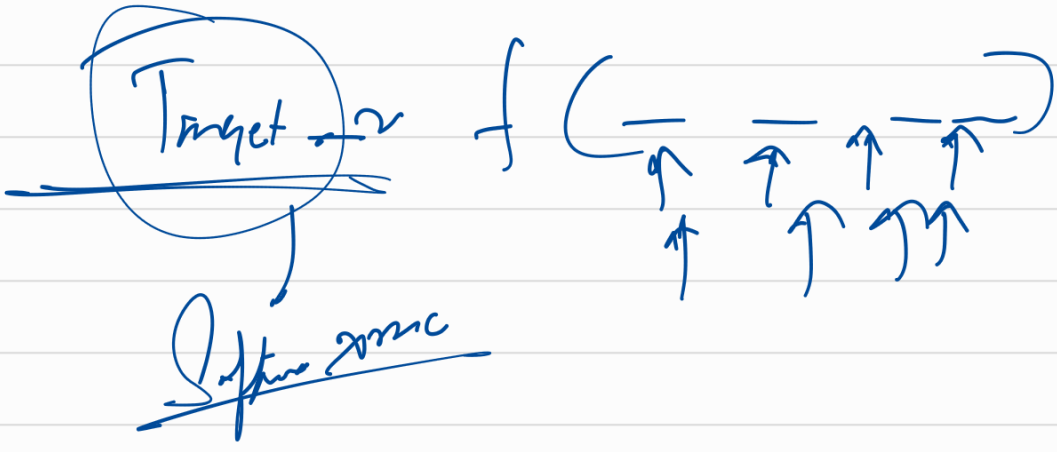
Embedding matrix

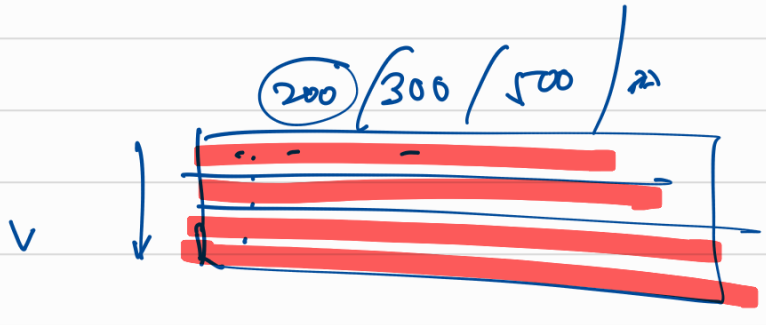


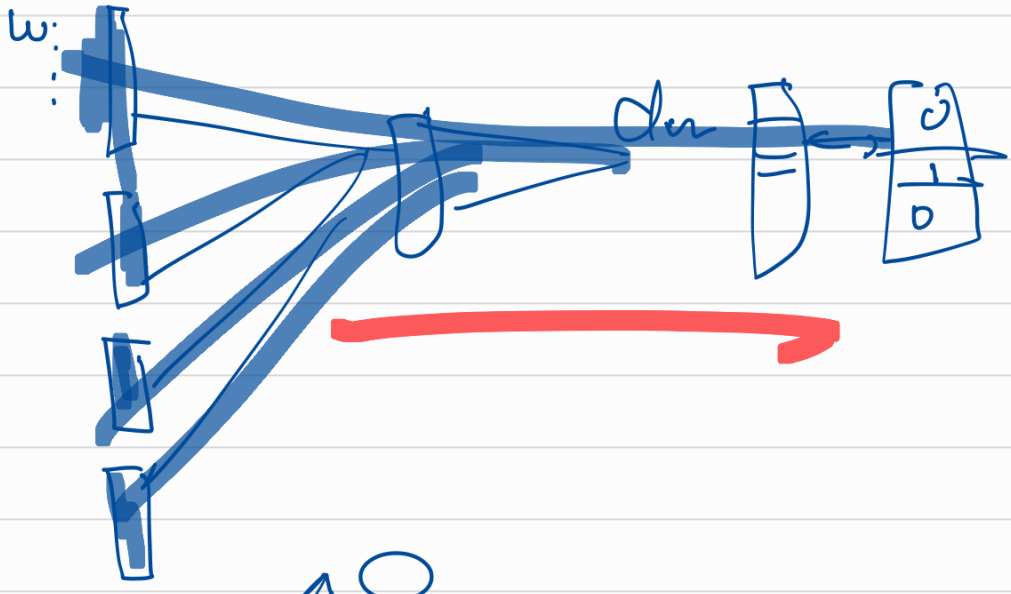


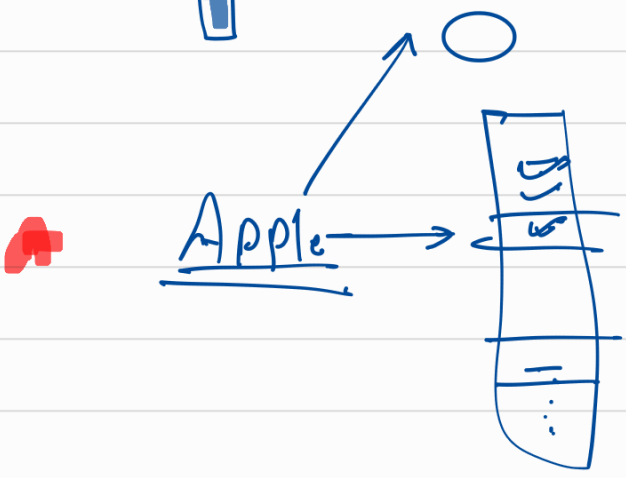
10.000

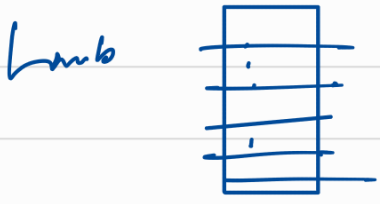


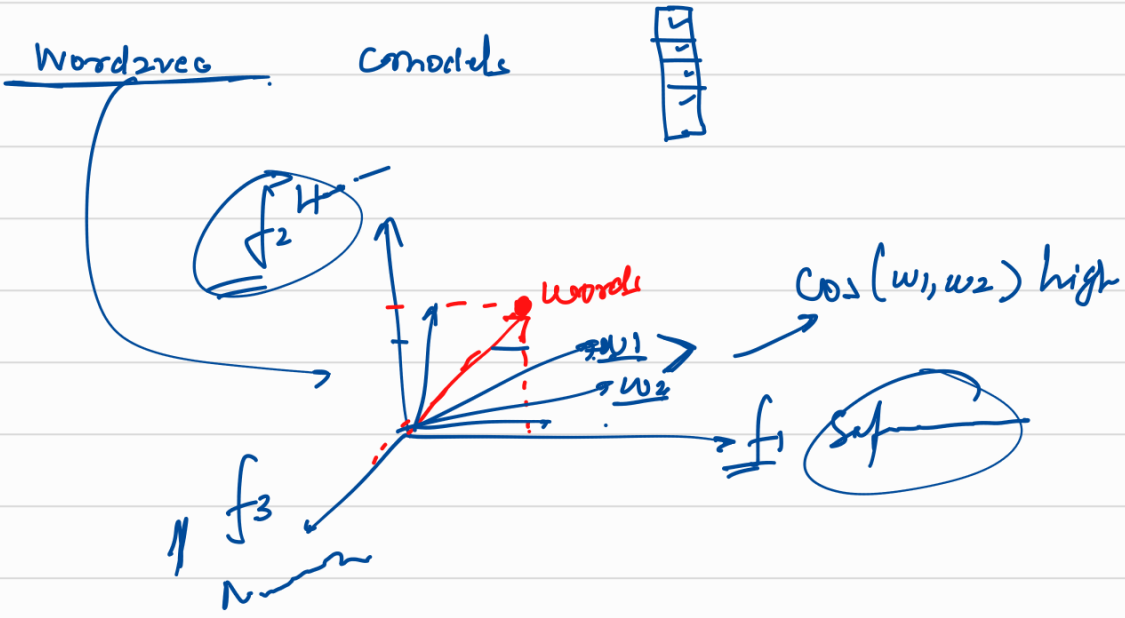





w:
den
c
1
0

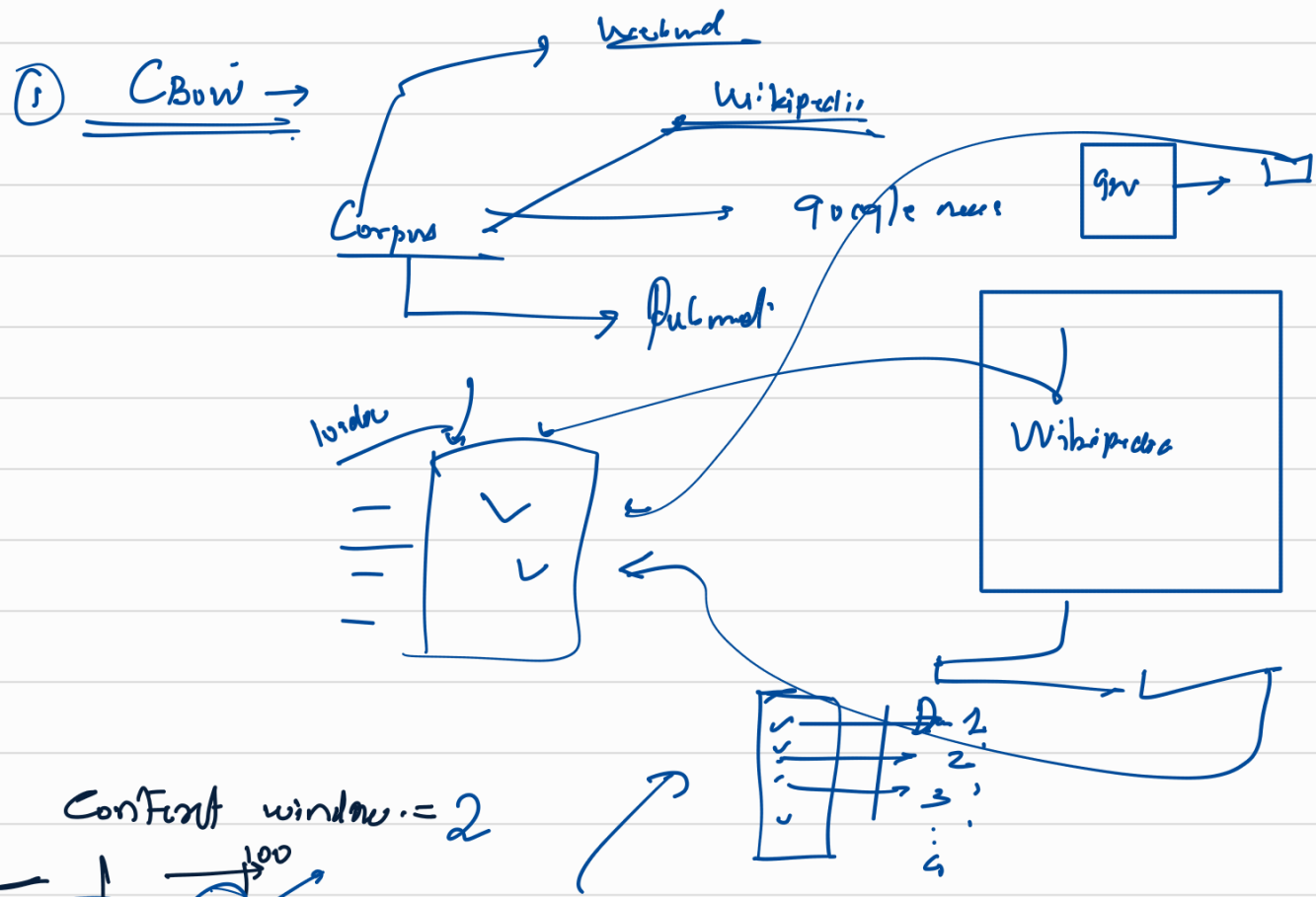

Apple

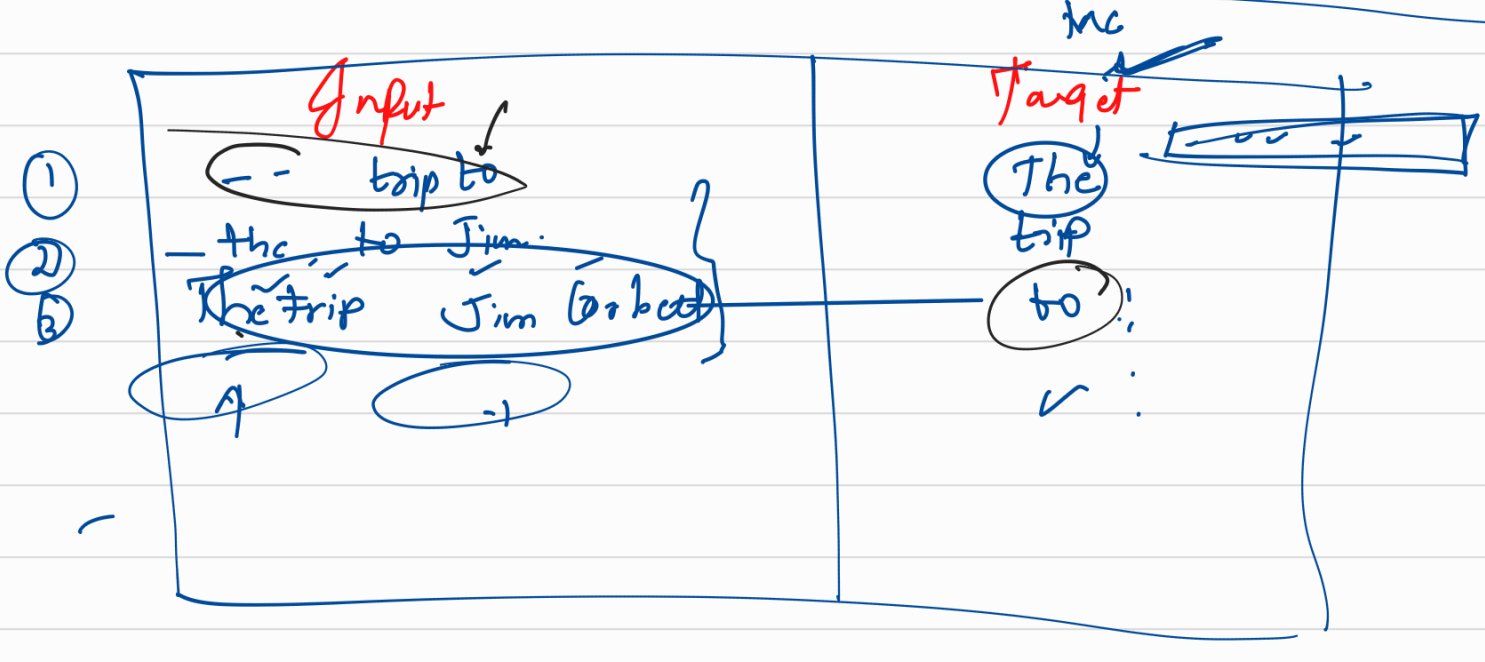

Lamb

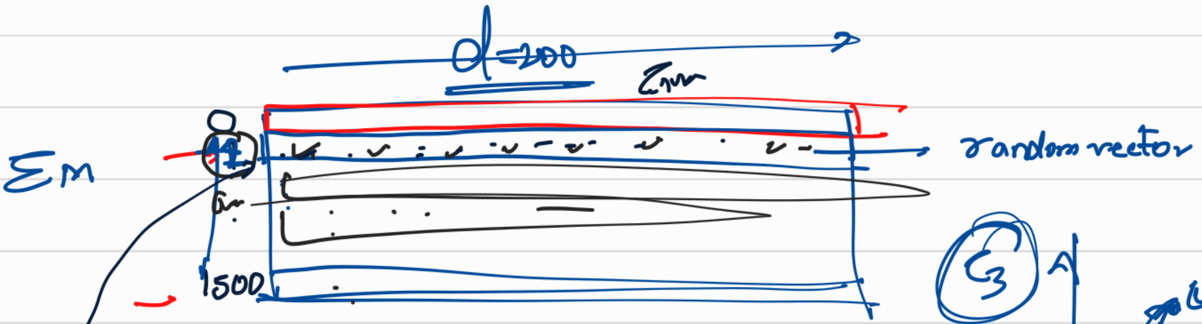

Word2vec
Comodels
words
cos(w1, w2) high
w1
w2
f1
f2
f3

CBOW (Continuous Bag of words model).

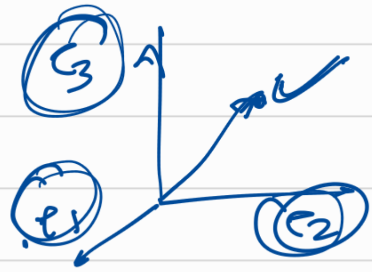
Skipgram model

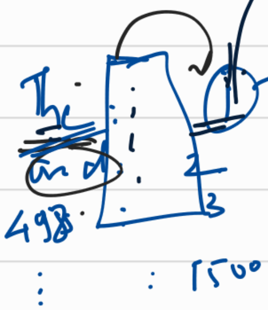




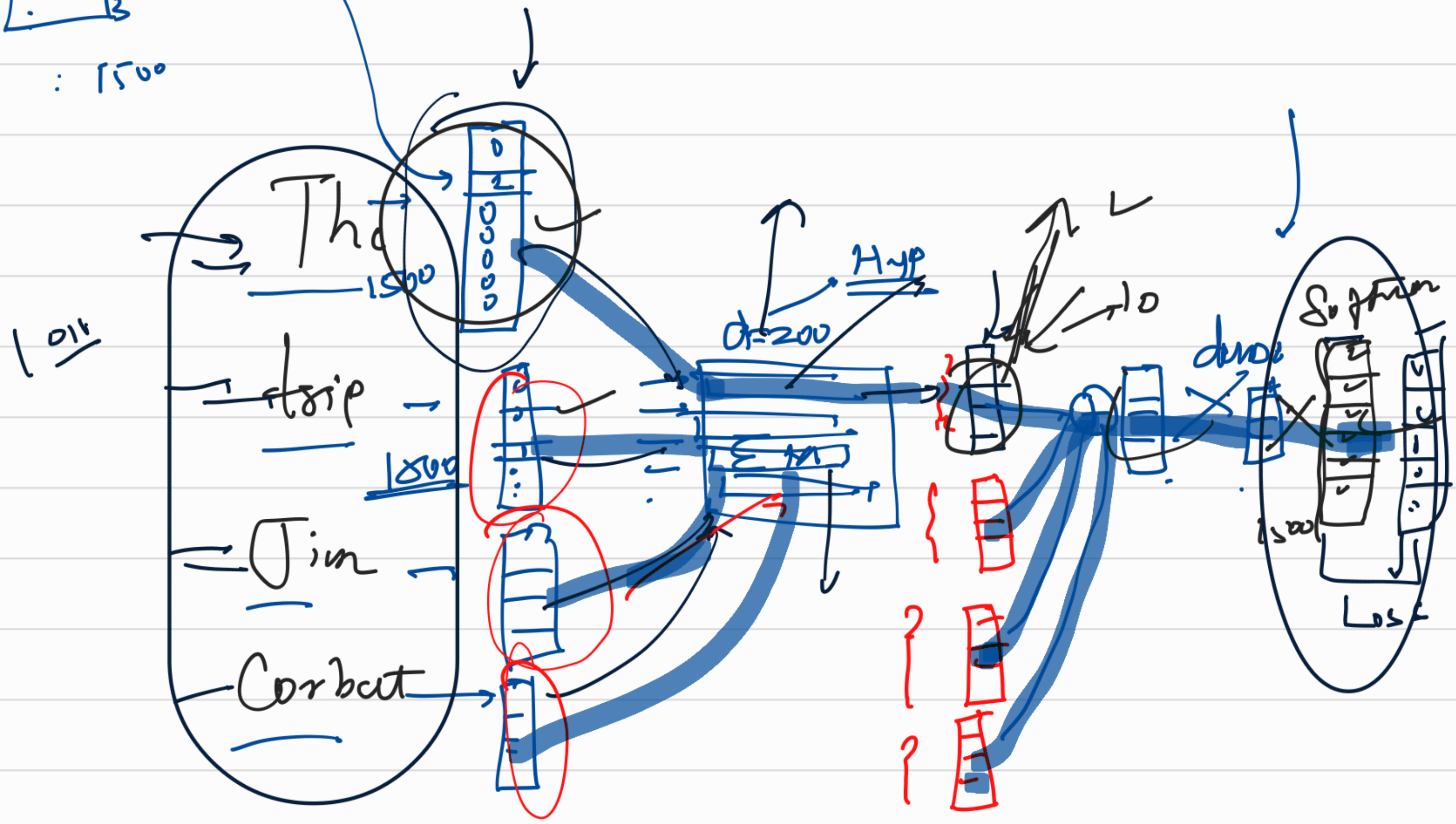


d=200
Em
1500
random vector

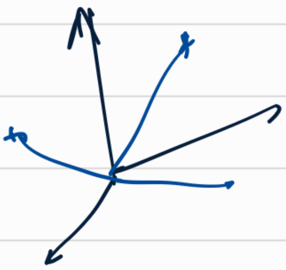


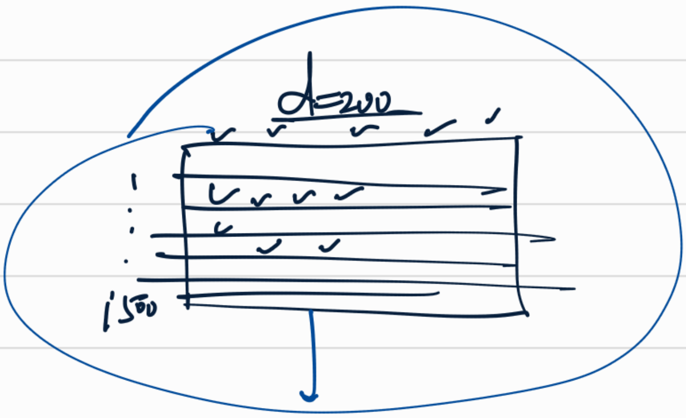


The
and
498
1500



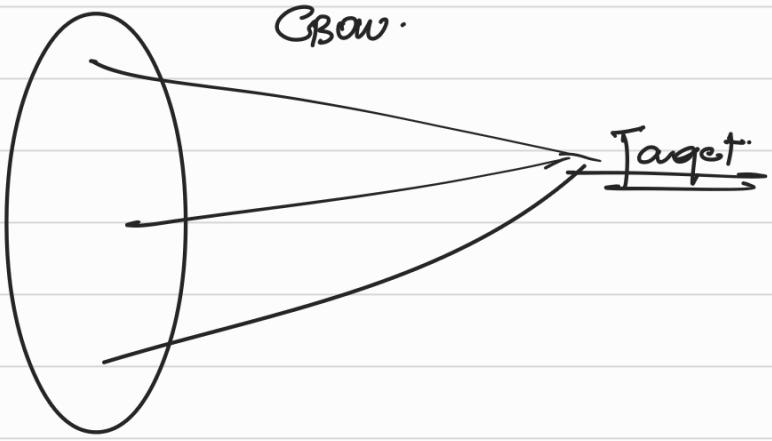
The
trip
Jim
Corbett
1500
d=200
Hyp
Em
dense
Softmax
Loss



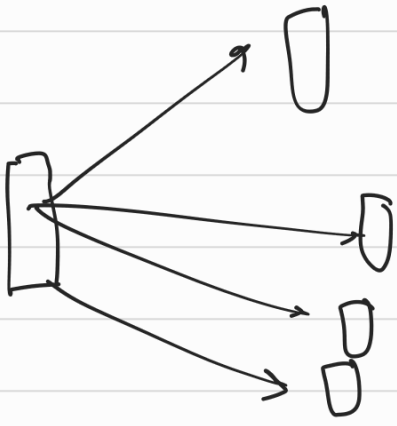


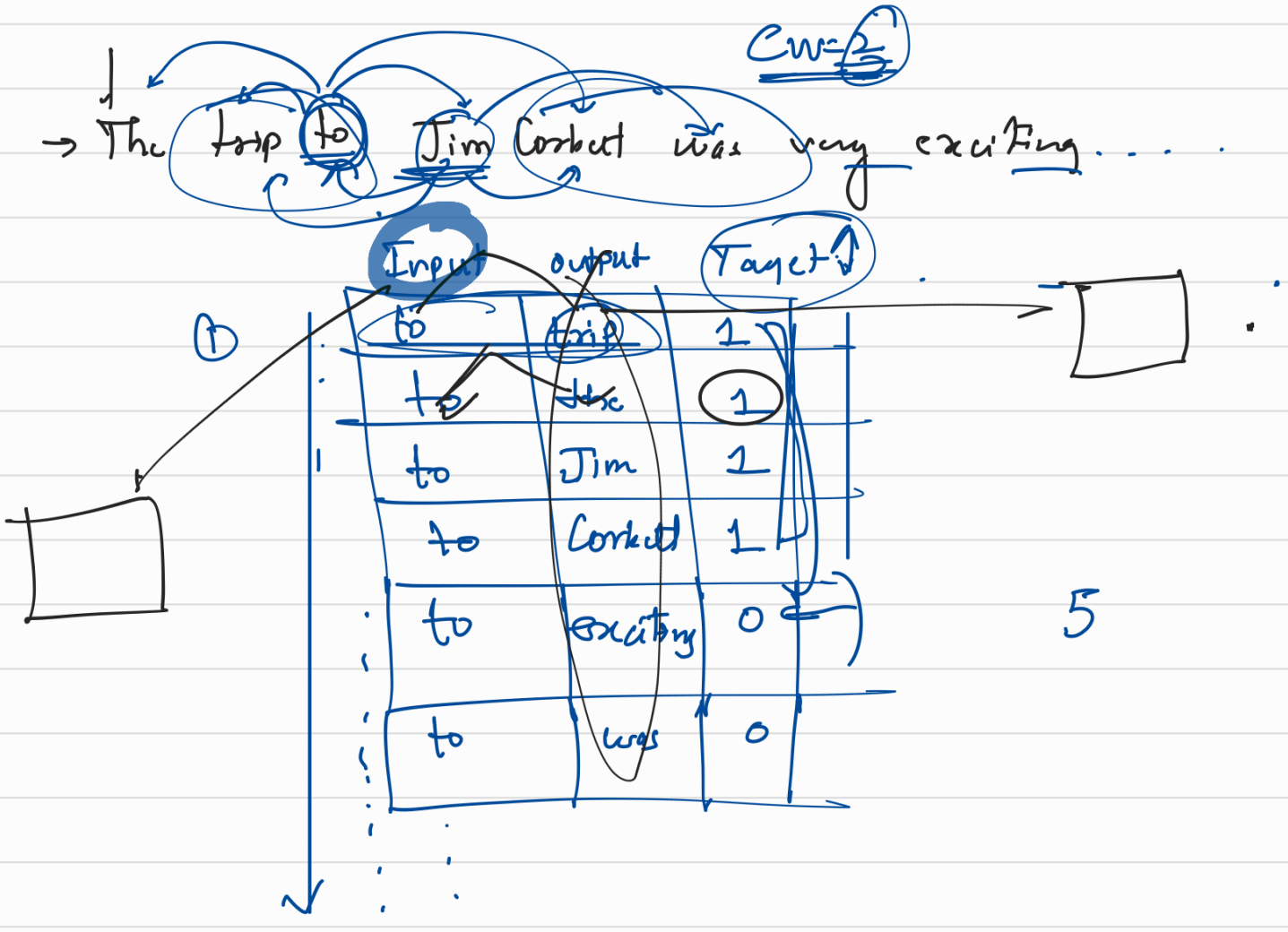
d=200
1500

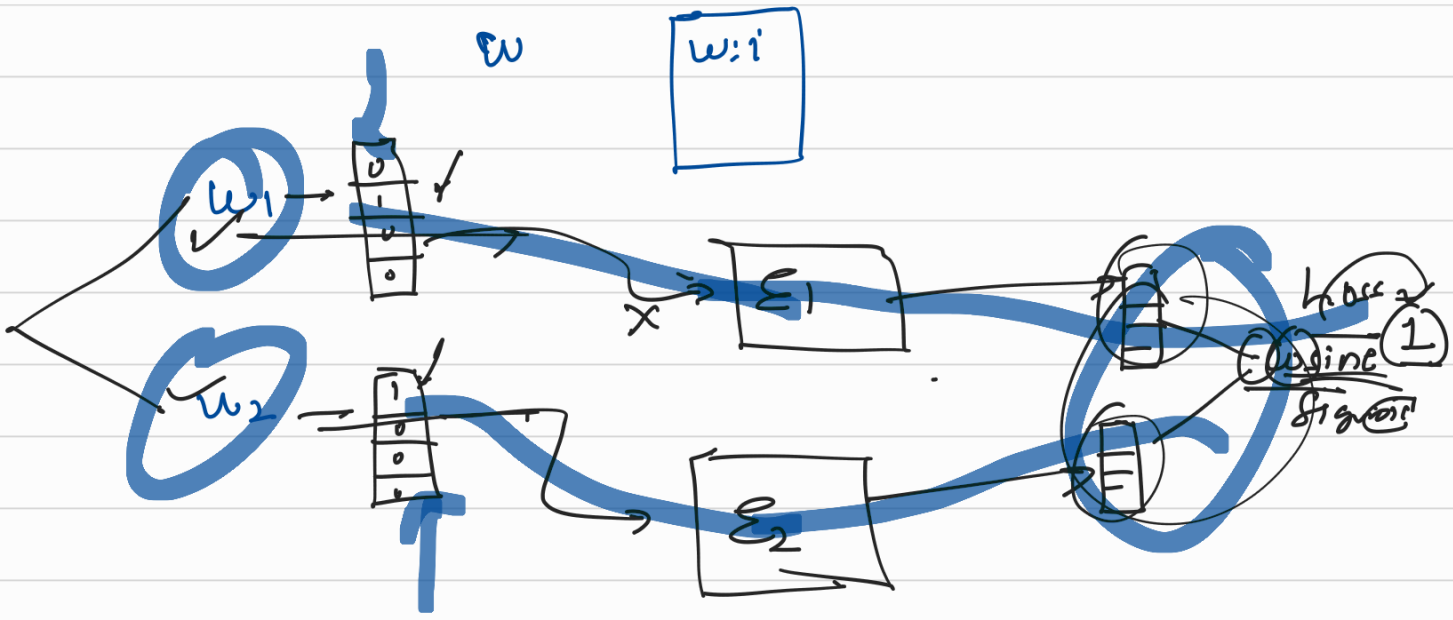
Skipgram →


CBOW.
Target

Skip-gram.




CW=2
The trip to Jim Corbett was very exciting . . .
Input
output
Target
to
trip
1
to
Jim
1
to
Corbett
1
to
exciting
0
to
was
0
5



Recurrent Neural network:

Text → Sequence

$w_t \sim f(w_{t-1}, w_{t-2}, \ldots w_{t-12})$

which is a neural network designed for 'Sequence models'

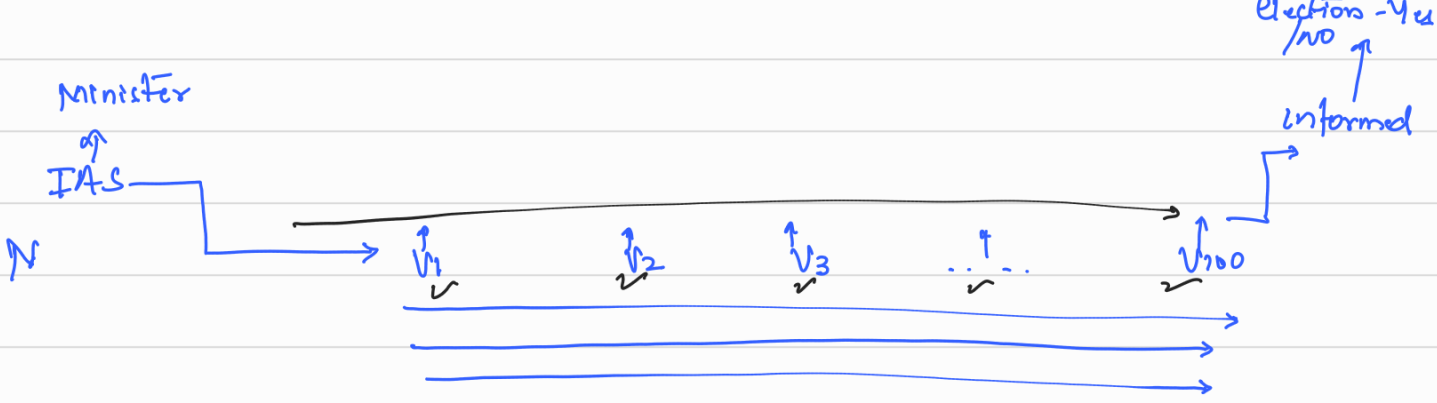
Cross-sectional data.

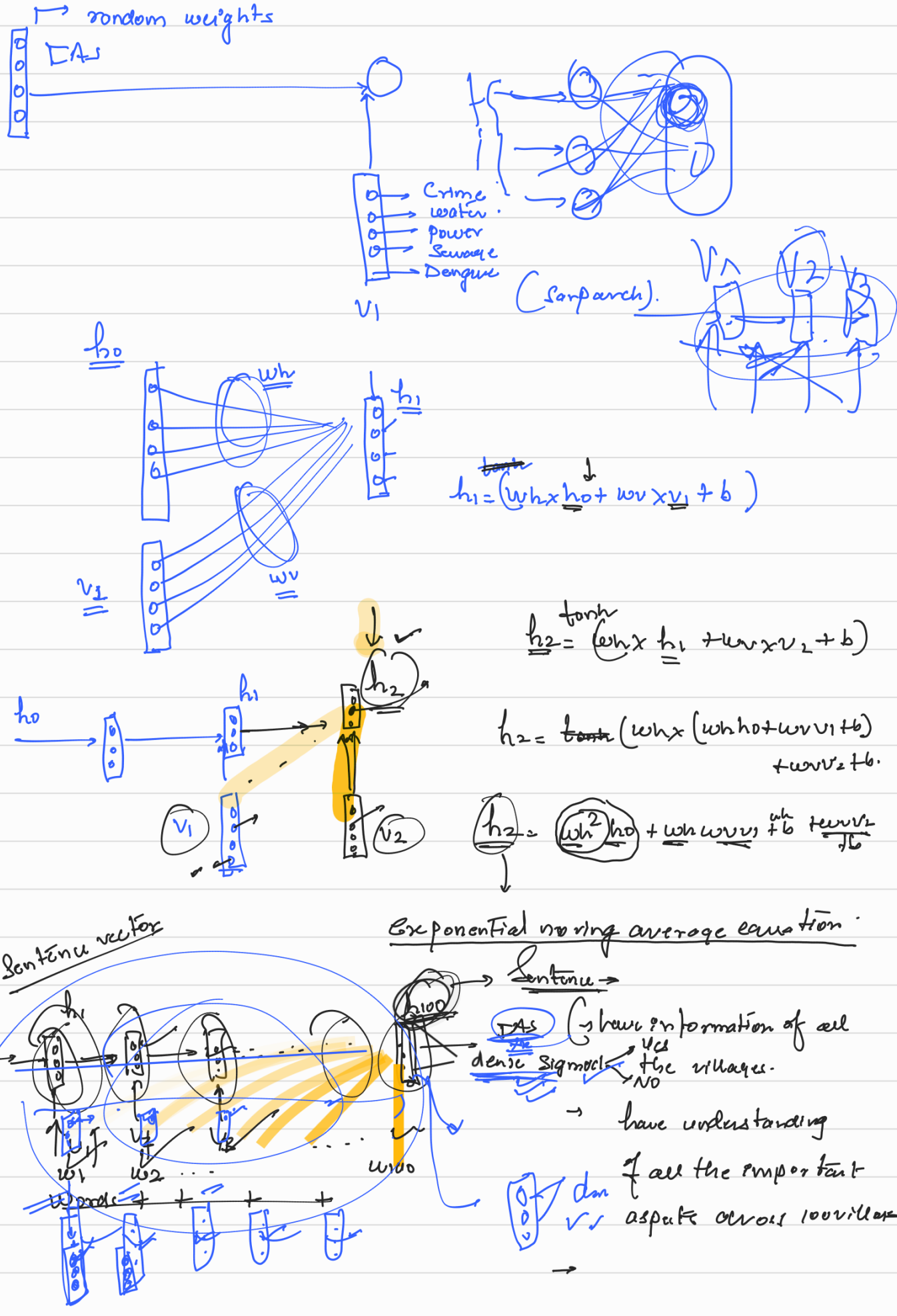
ANN $(w_{t-1}, w_{t-2}, \ldots w_{t-k})$

h_t

Recurrent Neural network

assume there are 100 villages in a town




random weights
DAs
Crime
water
power
Sewage
Dengue
v_1
(Sarpanch).
h_0
w_h
h_1
v_1
w_v
$h_1 = (w_h \times h_0 + w_v \times v_1 + b)$
$h_2 = \tanh(w_h \times h_1 + w_v \times v_2 + b)$
$h_2 = \tanh(w_h \times (w_h h_0 + w_v v_1 + b) + w_v v_2 + b)$
$h_2 = w_h^2 h_0 + w_h w_v v_1 + w_h b + w_v v_2 + b$
Sentence vector
w_1
w_2
w_{100}
Words
Exponential moving average equation.
Sentence
DAs
dense Sigmoid
Yes
No
→ have information of all the villages.
→ have understanding of all the important aspects across 100 villages