Jay Alammar

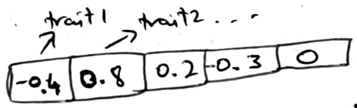
The Illustrated Word2Vec

-> Word2Vec
↳ efficiently create embeddings

Example:

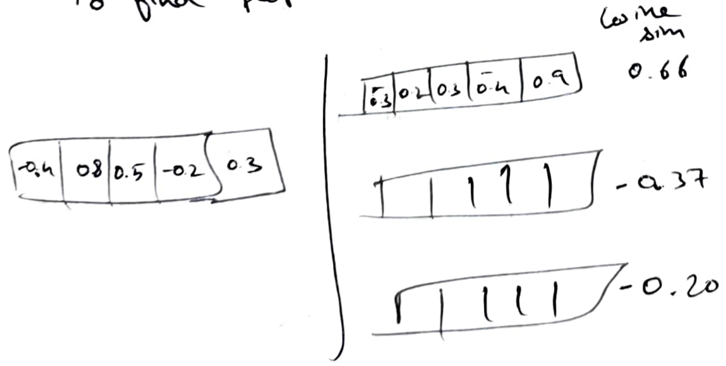
-> personality trait
any person's personality can be expressed in 5 numbers
features are i) extraversion
ii) negative emotionally
⋮

You can express a person in a 5 dimensional vector space



to detect 2 persons with similar personality we plot them in a 5 dimensional vector space and use algorithms to calculate how close each person is for example using (cosine similarity)

To find people similar to you,



0.66 is the guy most similar to you

Word embeddings

GloVe vector trained on wikipedia has 50 numbers to define a word

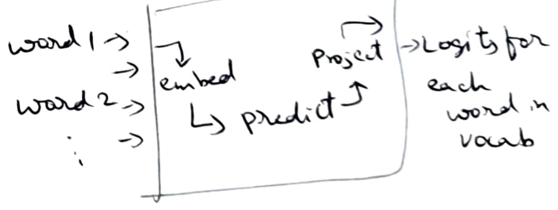
we don't need what those 50 numbers represents

vector embedding of

king + woman - man
gives queen
throne
prince
⋮

now looking at training the word embedding model

Language model training



data:
we get a paragraph
run a window over the text and ~~take~~ that tex will be ~~output~~ input to the model when
X = text window
y = window

Skipgram dataset

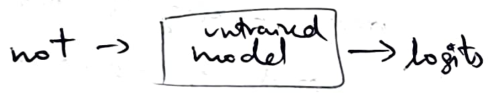
-> in a text blob, we get a sliding window say with 5 words as window size, ~~[illegible]~~ we take the middle text as ~~[illegible]~~ x and the remaining 4 as y for the model, in this text the skipgram will look something like

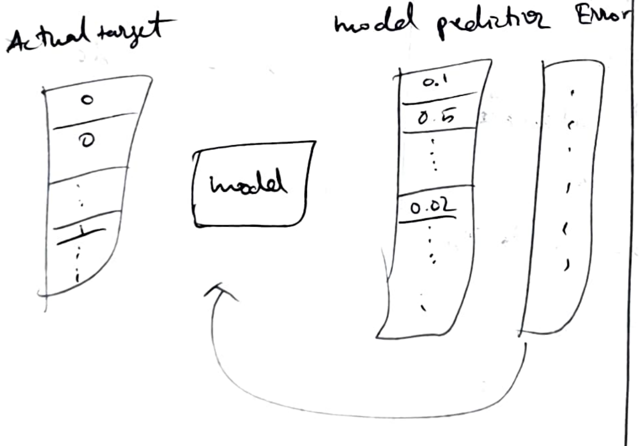
~~text~~	~~in~~
~~text~~	~~a~~
~~text~~	~~blob~~
~~text~~	~~we~~
~~text~~	
~~blob~~	~~a~~
~~blob~~	~~text~~

text	in
text	a
text	blob
text	we
blob	a
blob	text
blob	we
blob	get

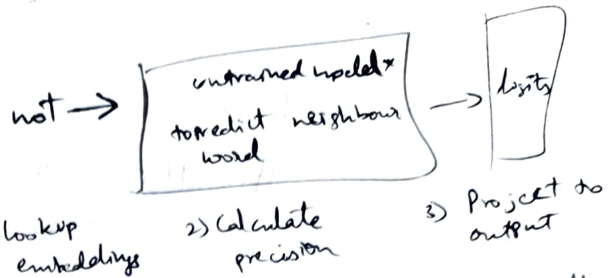
⋮

Training process





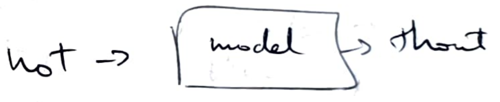
This model is the embeddings matrix



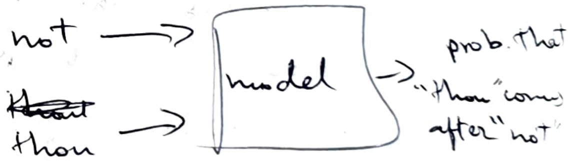
The third step is computationally expensive, to do this we are change our task into 2 steps

1) Generate high quality word embeddings
2) Use this high quality embed.. to train our model

) change the task objective



to



So we change the dataset to

~~[illegible]~~ ~~text~~

input		output
text	in	1
text	a	1
text	blob	1
text	we	1

but the model will collapse at always predicting '1' so we introduce negative samples by randomly sampling words from the vocabulary

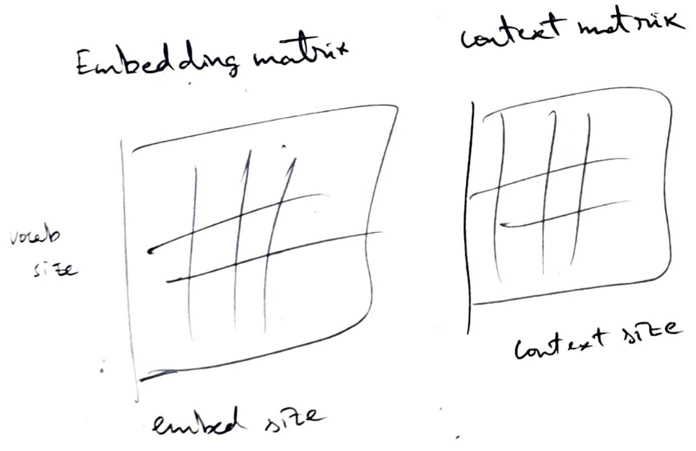
now finally,

actual word2vec

traing process

define ~~ve~~ vocab_size = 10,000

create embedding matrix, context matrix

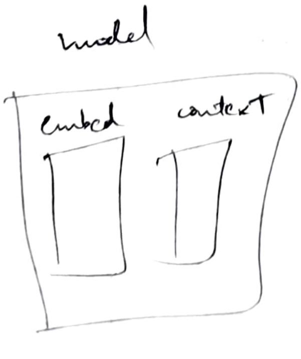


Initialize both with random

dataset

text	in	1
text	a	1
text	blab	1
text	we	1
text	sun	0
text	moon	0

↓ input word ↓ context word



~~for~~

training step

for input word look in embedding matrix, context word look in context matrix

dot product of

~~input · output~~

Signoid (input · output) = model output

calculate error use error to update embed & context table

discard the context table

hyperparameters
→ window size
→ num negative samples

less than 15
=> words are interchangable

>15
=> related words

the dataset always contain

1 +ve
n -ve
1 +ve
n -ve
⋮

and you select n+1 samples per training step