Word Embedding

講者：Isaac

Outline

- Bag of Words
- TF-IDF vector
- N-Gram
- Skip-gram and CBOW
- GloVe
- Doc2vect

What’s word Embedding

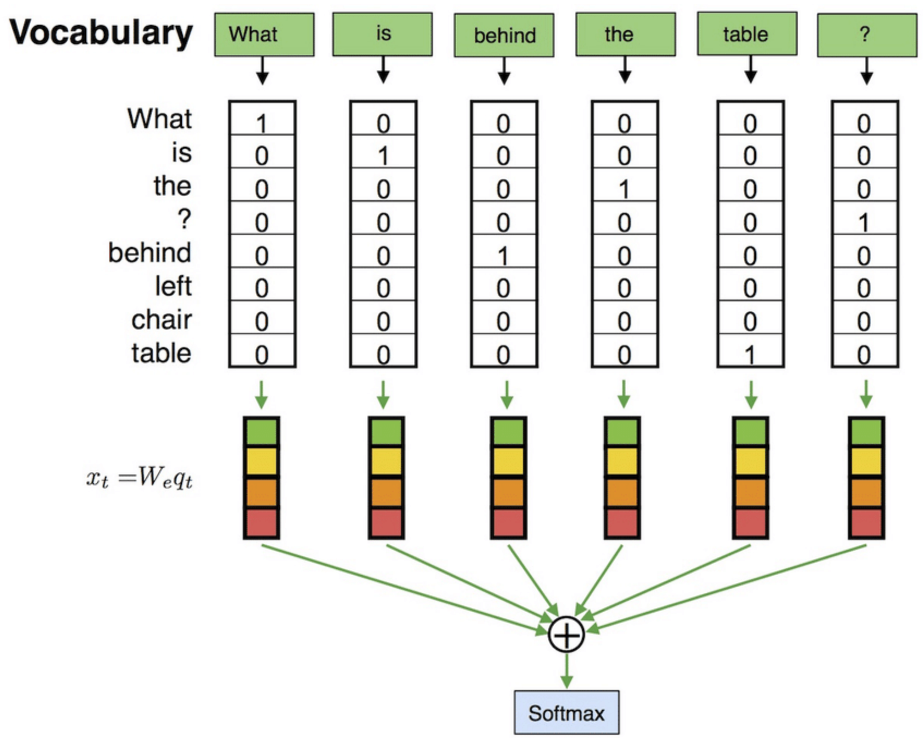
- Word embedding is the collective name that words or phrases from the vocabulary are mapped to vectors of real numbers

“hello” $\Rightarrow$ $\begin{bmatrix} 0.3 \\ -0.2 \\ \vdots \\ 0.8 \\ -0.5 \end{bmatrix}$

Bag of Words

What's Bag of Words

- one of the simplest methods of embedding words into numerical vectors



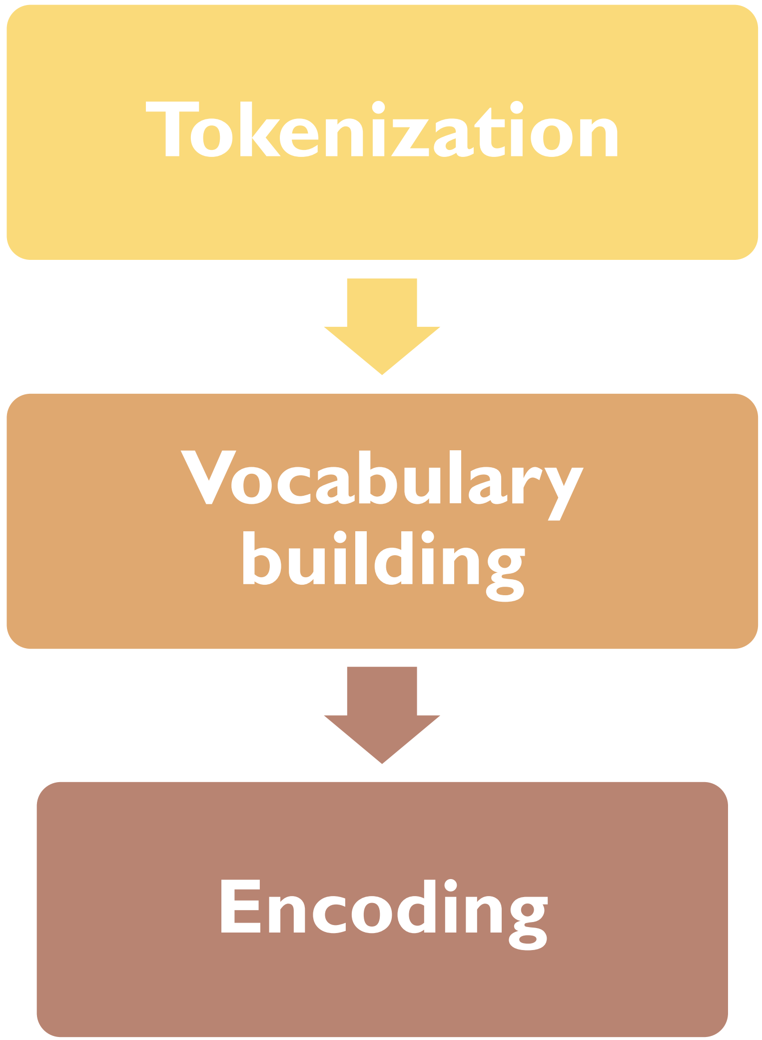
What's Bag of Words

Document 1	High five!
Document 2	I am old.
Document 3	She is five.

	Document 1	Document 2	Document 3
High	1	0	0
Five	1	0	1
I	0	1	0
am	0	1	0
old	0	1	0
She	0	0	1
is	0	0	1

"Five":[1,0,1]
"Document 2":[0,0,1,1,1,0,0]

What's Bag of Words



Bag of Words - Tokenization

Document	Content
1	[“it”, “was”, “the”, “best”, “of”, “times”]
2	[“it”, “was”, “the”, “worst”, “of”, “times”]
3	[“it”, “was”, “the”, “age”, “of”, “wisdom”]
4	[“it”, “was”, “the”, “age”, “of”, “foolishness”]

Bag of Words - Vocabulary building

"it"
"was"
"the"
"best"
"of"
"times"
"worst"
"age"
"wisdom"
"foolishness"

word	word ID
it	1
was	2
the	3
best	4
of	5
times	6
worst	7
Age	8
wisdom	9
foolishness	10

assuming ignore case and punctuation

Bag of Words - Encoding

word	word ID
it	1
was	2
the	3
best	4
of	5
times	6
worst	7
Age	8
wisdom	9
foolishness	10

"it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
"it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
"it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

Drawback of Bag of Words

- ignores word order
- sparse vector problems
  - solved by filtering out stop word, stemming, lemmazation, ignoring case etc … …

What’s tf–idf?

- tf–idf(term frequency–inverse document frequency) is a method that reflect how important a in a collection of corpus
  - tf means term “frequency”
  - idf means term “inverse document frequency”

Term Frequency (TF)

- In a single document
  - measures how frequently a term occurs in a document

TF = (Number of time the word occurs in the text) / (Total number of words in text)

Inverse Data Frequency (IDF)

- Among multiple documents

IDF = log(Total number of documents / Number of documents with word t in it)

tf–idf example

Document	Content
Document 1	It is going to rain today.
Document 2	Today I am not going outside.
Document 3	I am going to watch the season premiere.

tf–idf example

Words / Documents	Document 1	Document 2	Document 3
going	0.16	0.16	0.12
to	0.16	0	0.12
today	0.16	0.16	0
i	0	0.16	0.12
am	0	0.16	0.12
it	0.16	0	0
is	0.16	0	0
rain	0.16	0	0

calculate TF on each document

tf–idf example

Words / Documents	IDF
going	Log(3/3)
to	Log(3/2)
today	Log(3/2)
i	Log(3/2)
am	Log(3/2)
it	Log(3/1)
is	Log(3/1)
rain	Log(3/1)

calculate TF on each document

tf–idf example

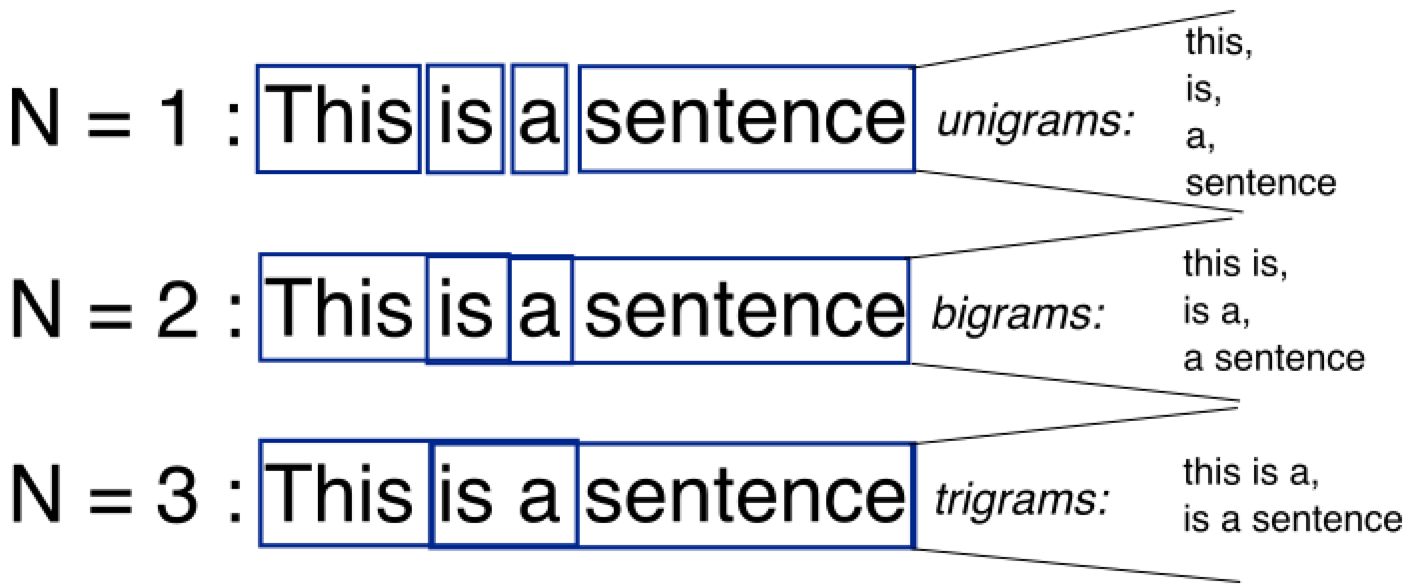
Words/ Documents	going	to	today	I	am	it	is	rain
Document1	0	0.07	0.07	0	0	0.17	0.17	0.17
Document2	0	0	0.07	0.07	0.07	0	0	0
Document3	0	0.05	0	0.05	0.05	0	0	0

Document1 = [0 , 0.07 , 0.07 , 0 , 0 , 0.17 , 0.17 , 0.17]

N-Gram

What's N-Gram

- n-gram is a contiguous sequence of n items from a given sample of text or speech



What's N-Gram

- Use N-gram to form word vector
  - same as bag of word

word	word ID
this is	1
is a	2
a sentence	3
a cat	4
a dog	5

"this is a dog"

[1,1,0,0,1]

Language model

- Probability a sentence occur in a text

Language model

the large green __ . Possible answer may be “mountain” or “tree” ?

Kate swallowed the large green __ . Possible answer may be “pill” or “broccoli” ?

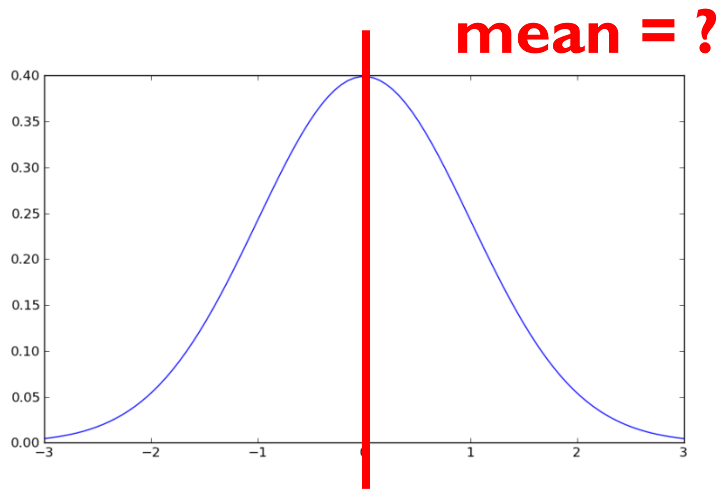
S1= “我剛吃過晚飯”
S2= “剛我過晚飯吃”

Which sentence is more reasonable?

Likelihood Estimation

- Assume students' height is normal distribution

Student ID	1	2	3	4	5
Height (cm)	162	164	170	168	166



What's mean of the normal distribution?

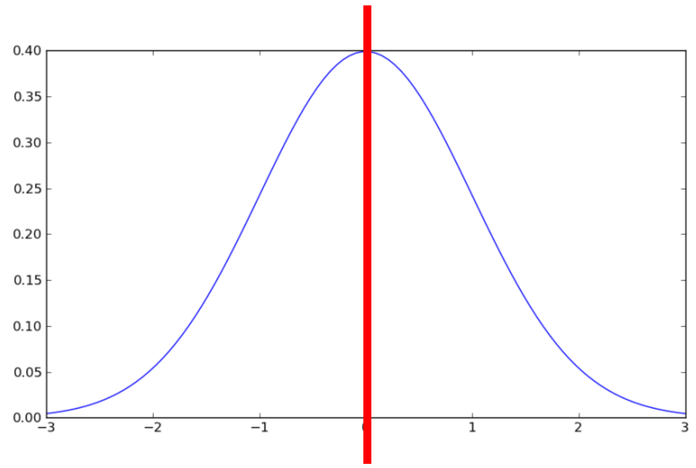
Likelihood Estimation

- Assume students' height is normal distribution

Student ID	1	2	3	4	5
Height (cm)	162	164	170	168	166

Intuitively:

mean = (162+164+170+168+166)/5 = 166



But why……?

Maximum Likelihood Estimation

- **Maximum likelihood estimation (MLE)** is a technique used for estimating the parameters of a given distribution, using some observed data

$Given\ probability\ distribution\ f\ that\ have\ some\ unkown\ parameters\ \theta$

$x1, x2, \ldots, xn\ are\ observation\ from\ f$

$$Likelihood\ function: f(x1, x2, \ldots, xn|\ \theta) = f(x1|\ \theta) * f(x2|\ \theta) * \cdots * f(xn|\ \theta)$$

Maximum Likelihood Estimation

Usually, we use Maximum **log** likelihood because Maximum likelihood is hard to calculate

$$Likelihood\ function: f(x1, x2, \ldots, xn|\ \theta) = f(x1|\ \theta) * f(x2|\ \theta) * \cdots * f(xn|\ \theta)$$

$$Log\ Likelihood\ function: f(x1, x2, \ldots, xn|\ \theta) = \sum_{i=1}^{n} \log(f(x_i|\ \theta))$$

Likelihood Estimation

Assume apple weights is normal distribution

We got three apples and their weights are 9, 9.5, 11 respectively

Our goal is to estimate the mean/std of total apples on the tree

$$P(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Likelihood Estimation

We want to maximize likelihood(the following equation)

$$P(9, 9.5, 11; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(9-\mu)^2}{2\sigma^2}\right) \times \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(9.5-\mu)^2}{2\sigma^2}\right) \times \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(11-\mu)^2}{2\sigma^2}\right)$$

But it is hard to derivative on this equation !

Likelihood Estimation

maximize likelihood equivalent to maximize log likelihood

$$P(9, 9.5, 11; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(9-\mu)^2}{2\sigma^2}\right) \times \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(9.5-\mu)^2}{2\sigma^2}\right) \times \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(11-\mu)^2}{2\sigma^2}\right)$$

$$\ln(P(x; \mu, \sigma)) = \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{(9-\mu)^2}{2\sigma^2} + \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{(9.5-\mu)^2}{2\sigma^2} + \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{(11-\mu)^2}{2\sigma^2}$$

Likelihood Estimation

$$\frac{\partial \ln(P(x;\mu,\sigma))}{\partial \mu} = \frac{1}{\sigma^2}\left[9 + 9.5 + 11 - 3\mu\right].$$

$$\mu = \frac{9 + 9.5 + 11}{3} = 9.833$$

Likelihood Estimation Table

Distribution	Estimated parameters
$\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x-\pi)^2}{2\sigma^2}}$	$\mu = \frac{\sum_{i=1}^{n} x_i}{n} = \bar{x}$ $\sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$
$\lambda e^{-\lambda x}$	$\lambda = \frac{1}{\bar{x}}$
$\frac{e^{-\lambda}\lambda^k}{k!}$	$\lambda = \bar{x}$
⋮	⋮

- Assume we have m word sequence, the probability of this sentence is

$$P(w_1, w_2, \cdots, w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \cdots P(w_m|w_1, \cdots, w_{m-1})$$

Assume Markov Property

$$P(w_i|w_1, \cdots, w_{i-1}) = P(w_i|w_{i-n+1}, \cdots, w_{i-1})$$

$$P(w_i|w_1,\cdots,w_{i-1}) = P(w_i|w_{i-n+1},\cdots,w_{i-1})$$

unigram model (n=1)

$$P(w_1,w_2,\cdots,w_m) = \prod_{i=1}^{m} P(w_i)$$

$$P(w_i) = \frac{C(w_i)}{M}$$

bigram model (n=2)

$$P(w_1,w_2,\cdots,w_m) = \prod_{i=1}^{m} P(w_i|w_{i-1})$$

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$$

trigram model (n=3)

$$P(w_1,w_2,\cdots,w_m) = \prod_{i=1}^{m} P(w_i|w_{i-2}w_{i-1})$$

$$P(w_i|w_{i-n-1},\cdots,w_{i-1}) = \frac{C(w_{i-n-1},\cdots,w_i)}{C(w_{i-n-1},\cdots,w_{i-1})}$$

Example

- Assume we have a corpus

Sentence	Content
1	<s1> <s2> yes no no no no yes </s1> </s2>
2	<s1> <s2> no no no yes yes yes no </s1> </s2>

We want to calculate probability of the following sentence:

<s1> <s2> yes no no yes </s1> </s2>

Example

Use trigram as example

$$P(yes| < s1 >< s2 >) = \frac{1}{2}, \qquad P(no| < s2 > yes) = 1$$

$$P(no|yes \quad no) = \frac{1}{2}, \qquad P(yes|no \quad no) = \frac{2}{5}$$

$$P(< /s2 > |no \quad yes) = \frac{1}{2}, \qquad P(< /s1 > |yes < /s2 >) = 1$$

<s1> <s2> yes no no yes </s1> </s2>

$$\frac{1}{2} \times 1 \times \frac{1}{2} \times \frac{2}{5} \times \frac{1}{2} \times 1 = 0.05$$

search engine

text generation

Skip-gram and CBOW

Skip-gram and CBOW

- NN-based algorithm
- Core concept
  - the meaning of a word can be inferred by the company it keeps

I like math.

I like programming.

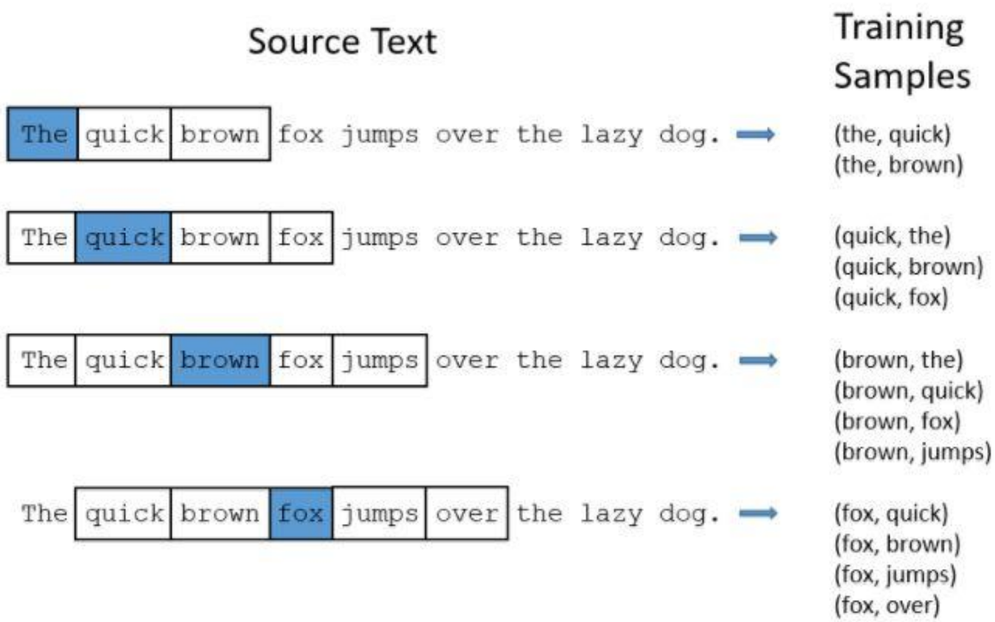
Today is Friday.

Today is a good day.

- *like* is the context of target *I*
- *math* is the context of target *like*
- *programming* is also the context of target *like*

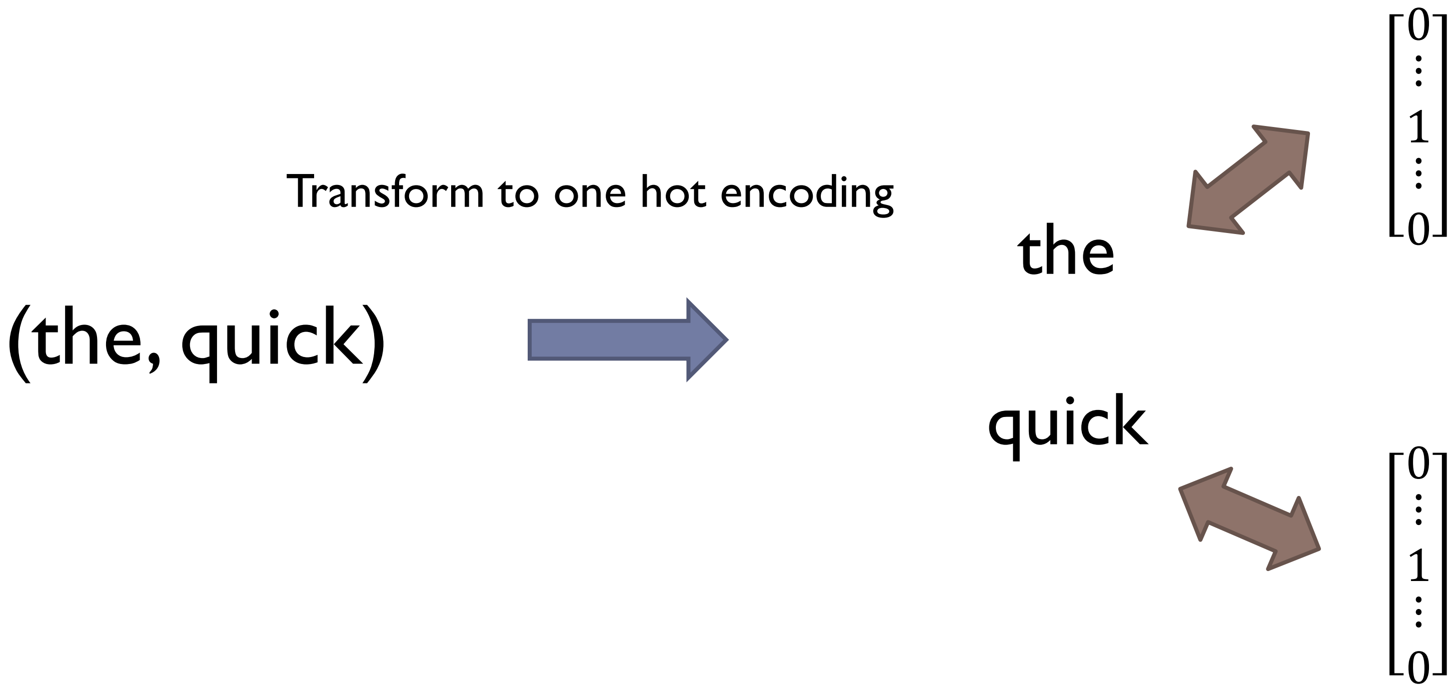
Skip-Gram

Transform many sentences into training pairs

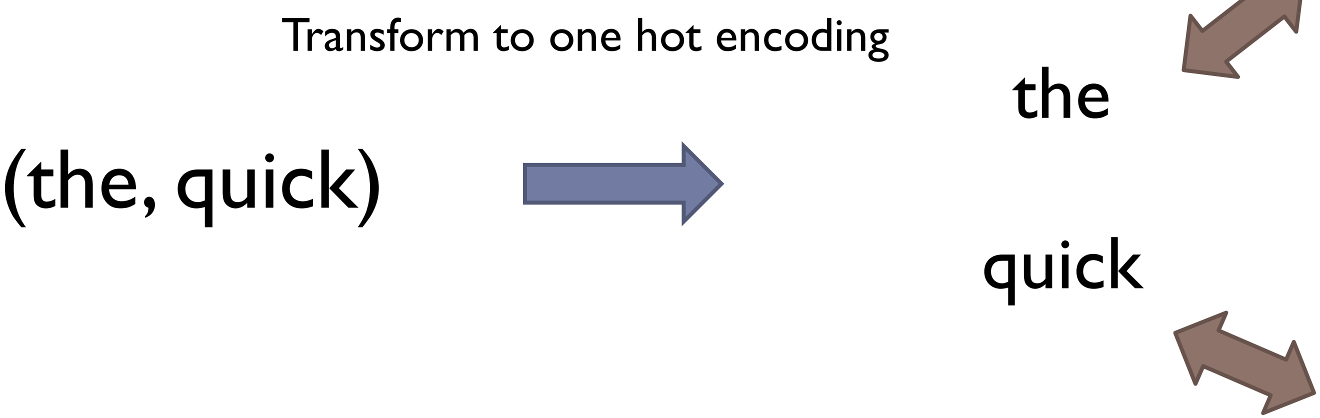


Window size = 2

Skip-Gram


Transform to one hot encoding
(the, quick)
the
quick
$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

Skip-Gram



$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Skip-Gram

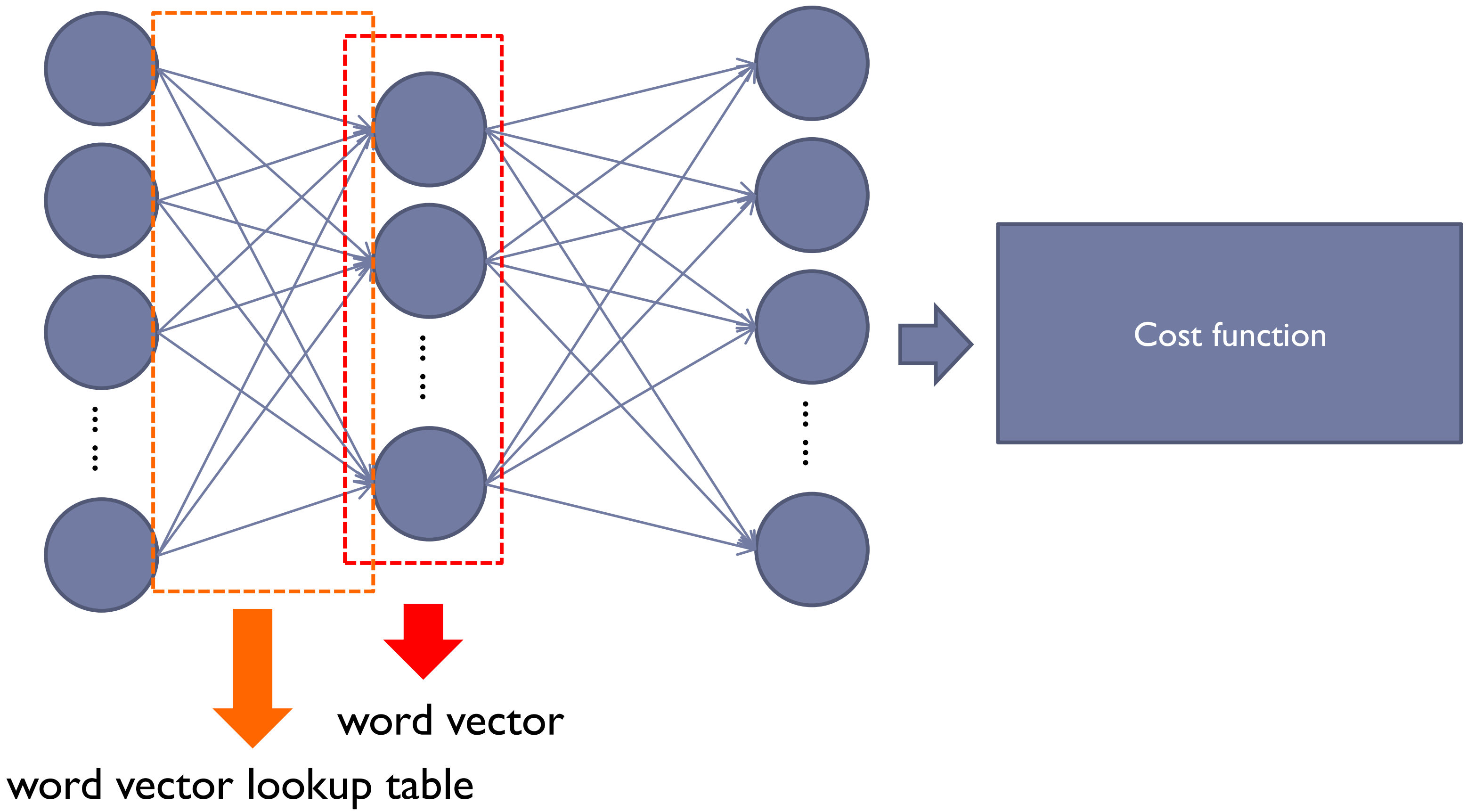
Training pairs

(the, quick)
(the, brown)

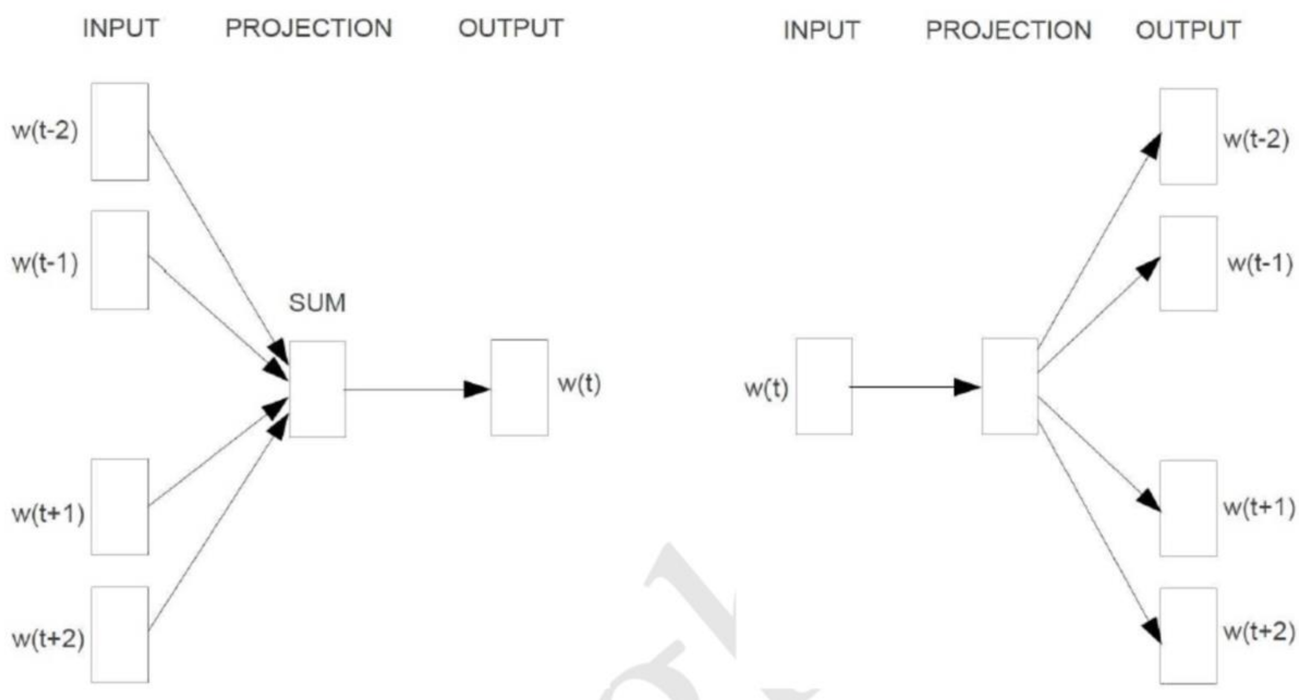
(quick, the)
(quick, brown)
(quick, fox)

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)

(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)



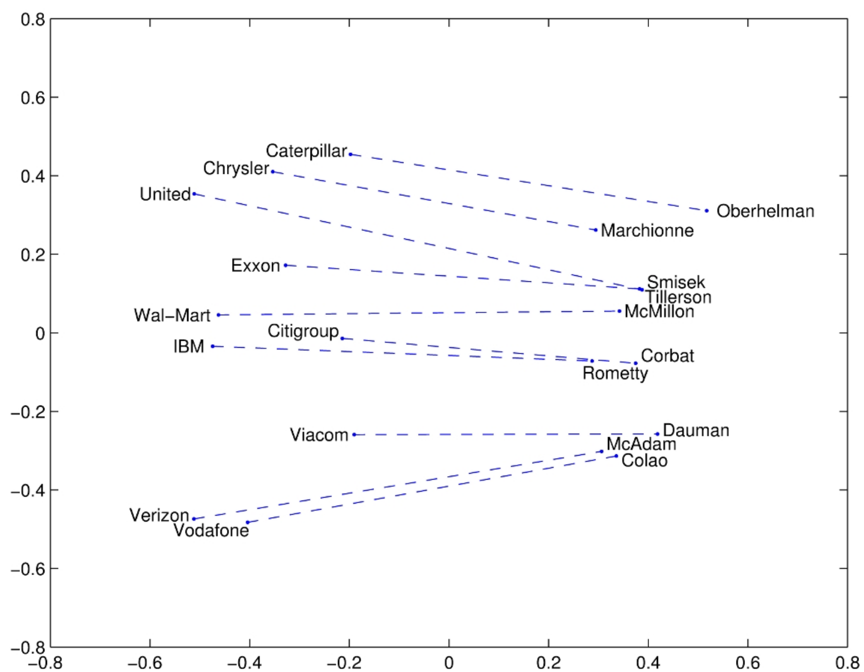
Skip-Gram V.S. CBOW



GloVe

What's GloVe

- Glove(Global Vectors for Word Representation) is a count-based and overall statistics word representation method



Construct Co-ocurrence Matrix

$$X = \begin{array}{c|ccccc} & \text{the} & \text{cat} & \text{sat} & \text{on} & \text{mat} \\ \hline \text{the} & 0 & 1 & 0 & 1 & 1 \\ \text{cat} & 1 & 0 & 1 & 0 & 0 \\ \text{sat} & 0 & 1 & 0 & 1 & 0 \\ \text{on} & 1 & 0 & 1 & 0 & 0 \\ \text{mat} & 1 & 0 & 0 & 0 & 0 \end{array}$$

$$w_i^T \tilde{w}_j + b_i + \tilde{b_j} = \log(X_{ij})$$

w_i, w_j: $word\ vector$

b_i, b_j: $bias\ term$

X_{ij}: $entity\ i, j\ in\ co - ocurrance\ matrix\ i$

relationship between word vector and co-ocurrence matrix

Define Cost

$$J = \sum_{i,j=1}^{V} \boxed{f(X_{ij})}(w_i^T \tilde{w}_j + b_i + \tilde{b_j} - \log(X_{ij}))^2$$

avoid Xij is zero

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases}$$

Define Cost

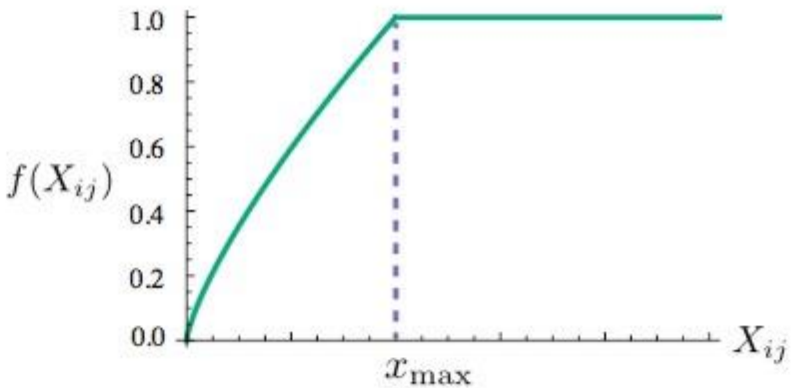


Figure 1: Weighting function f with $\alpha = 3/4$.

$$f(x) = \begin{cases} \left(\frac{x}{xmax}\right)^{\alpha}, & if\ x < xmax \\ 1 & , if\ x \geq xmax \end{cases}$$

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

If word k is very similar to ice but irrelevant to steam (e.g. k=solid)
$\rightarrow \mathrm{P}(k|ice)/\mathrm{P}(k|steam)$will be very high (>1)

If word k is very similar to steam but irrelevant to ice (e.g. k=gas)
$\rightarrow \mathrm{P}(k|ice)/\mathrm{P}(k|steam)$will be very small (<1)

If word k is related or unrelated to either words
$\rightarrow \mathrm{P}(k|ice)/\mathrm{P}(k|steam)$will be close to 1

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

$\Downarrow$

Word vectors are linear systems
vec(king)-vec(male)+vec(queen) = vec(female)

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

$\Downarrow$

Make LHS scalar

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

$$F((w_i - w_j)^T \tilde{w}_k)$$

homomorphism property
F(A-B) = F(A)/F(B)

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

assume F is exp

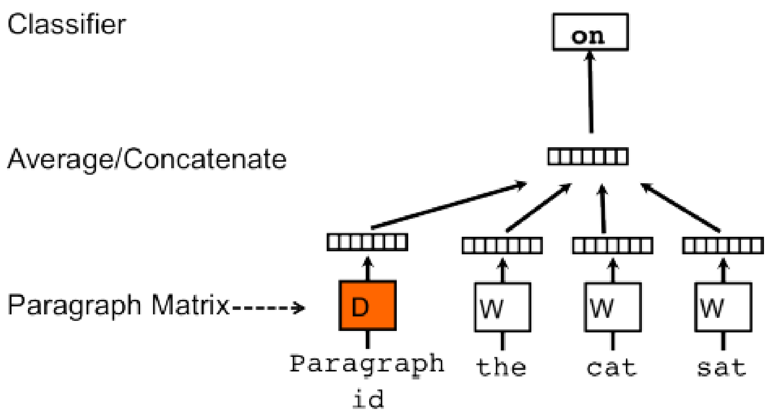
$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

assume bi + bj = Xi

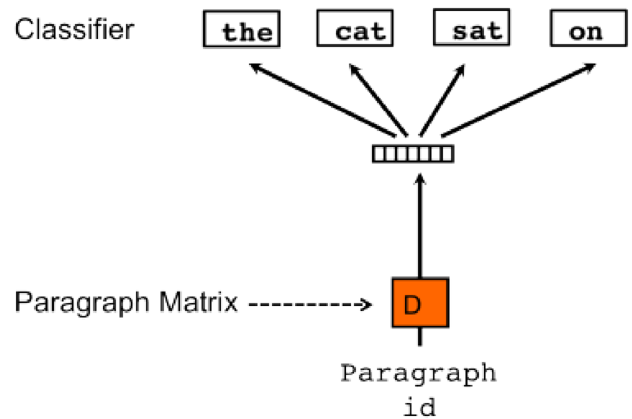
$$w_i^T \bar{w}_j + b_i + \bar{b_j} = \log(X_{ij})$$

Doc2vect

Doc2vect



Distributed Memory version of Paragraph Vector
(PV-DM)



Words version of Paragraph Vector
(PV-DBOW)