CS5560 Knowledge Discovery and Management

Problem Set 4

June 26 (T), 2017

Name: Megha Nagabhushan

Class ID: 15

I. N-Gram

Consider a mini-corpus of three sentences

<s> I am Sam </s>

<s> Sam I am </s>

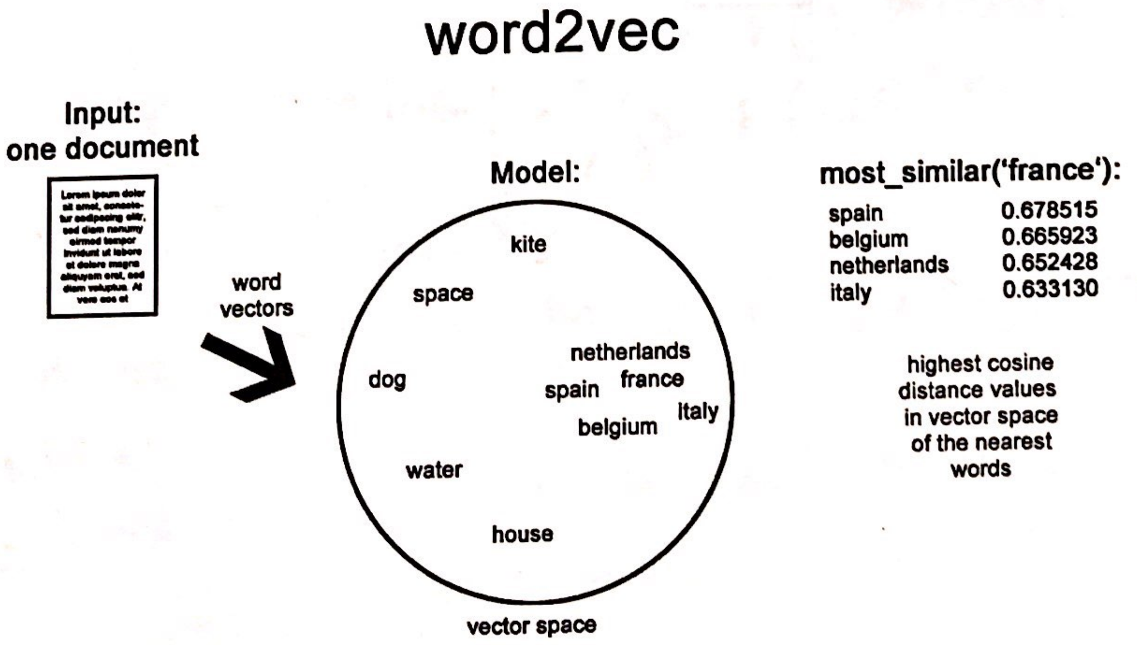
<s> I like green eggs and ham </s>

1) Compute the probability of sentence "I like green eggs and ham" using the appropriate bigram probabilities.
2) Compute the probability of sentence "I like green eggs and ham" using the appropriate trigram probabilities.

II. Word2Vec

Word2Vec reference: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

Consider the following figure showing the Word2Vec model.



word2vec
Input: one document
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et
word vectors
Model:
kite
space
dog
netherlands
spain
france
italy
belgium
water
house
vector space
most_similar('france'):
spain 0.678515
belgium 0.665923
netherlands 0.652428
italy 0.633130
highest cosine distance values in vector space of the nearest words

a. Describe the word2vec model

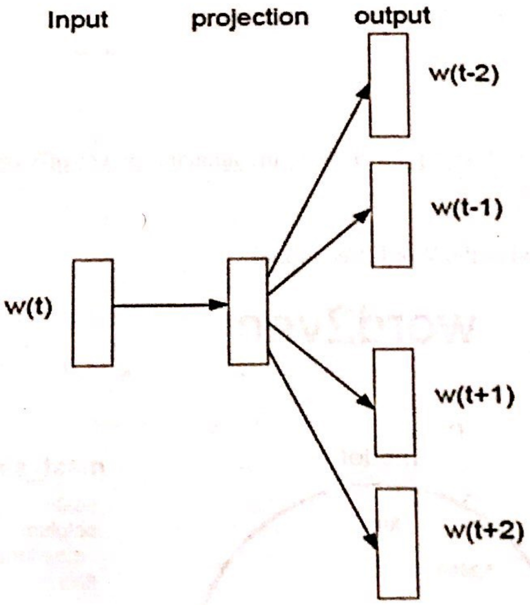
b. Describe How to extend this model for multiple documents. Also draw a similar diagram for the extended model.

Describe the differences of the following approaches

- Continuous Bag-of-Words model,
- Continuous Skip-gram model

For the sentence “morning fog, afternoon light rain,”

- Place the words on the skip-gram Word2Vec model below.
- Draw a CBOW model using the same words.



① a) Calculations of bigram probabilities from this corpus.

$$P(W_i \mid W_{i-1}) = \frac{C(W_{i-1}, W_i)}{C(W_{i-1})}$$

<u>Training corpus</u>:

$\langle s \rangle \; \overset{i-2}{I} \; \overset{i-1}{am} \; \overset{i}{Sam} \; \langle /s \rangle$

$\langle s \rangle$ Sam I am $\langle /s \rangle$

$\langle s \rangle$ I like green eggs and ham $\langle /s \rangle$

$P(I \mid \langle s \rangle) = \frac{2}{3} = 0.67$ ~~P(I|~~

$P(\overset{i}{am} \mid \overset{i-1}{I}) = \frac{2}{3} = 0.67$

$P(Sam \mid am) = \frac{1}{2} = 0.5$

$P(\langle /s \rangle \mid Sam) = \frac{1}{2} = 0.5$

$P(\langle /s \rangle \mid am) = \frac{1}{2} = 0.5$

$P(I \mid Sam) = \frac{1}{2} = 0.5$

$P(\overset{i}{Sam} \mid \overset{i-1}{\langle s \rangle}) = \frac{1}{3} = 0.33$

$P(like \mid I) = \frac{1}{1} = 1$

$P(green \mid like) = \frac{1}{1} = 1$

$P(eggs \mid green) = \frac{1}{1} = 1$

$P(and \mid eggs) = \frac{1}{1} = 1$

$P(ham \mid and) = \frac{1}{1} = 1$

$P(\langle /s \rangle \mid ham) = \frac{1}{1} = 1$.

Bi-gram probability for "I like green eggs and ham" will be

$P(I \mid \langle s \rangle) \times P(like \mid I) \times P(green \mid like) \times P(eggs \mid green) \times$ ~~P(and | eggs)~~ $\times P(ham \mid and) \times$
$P(and \mid eggs) \times$
$P(\langle /s \rangle \mid ham)$.

$0.67 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 = \underline{\underline{0.67}}$

b) Trigram probability.

$$P(Wi / Wi_{-1}, Wi_{-2}) = \frac{count\ (Wi, Wi_{-1}, Wi_{-2})}{count\ (Wi_{-1}, Wi_{-2})}$$

$$P(Sam \mid I\ am) = \frac{count\ (Sam\ I\ am)}{count\ (I\ am)} = \frac{1}{2}$$

$P(\langle s\rangle \mid am\ Sam) = \frac{1}{1} = 1.$

$P(am \mid \langle s\rangle\ I) = 1/2$ $\quad P(am \mid sam\ I) = \frac{1}{1} = 1$ $\quad P(\langle /s\rangle \mid I\ am) = 1/2$

$P(I \mid \langle s\rangle\ Sam) = 1/1 = 1$ $\quad P(eggs \mid like\ green) = 1$

$P(like \mid \langle s\rangle I) = 1/2$ $\quad P(green \mid I\ like) = 1$ $\quad P(\langle /s\rangle \mid and\ ham) = 1.$

$P(and \mid green\ eggs) = 1$ $\quad P(ham \mid eggs\ and) = 1$

Tri-gram probability for $\langle s\rangle$ "I like green eggs and ham" $\langle /s\rangle$ will be

$P(like \mid \langle s\rangle\ I) \times P(green \mid I\ like) \times P(eggs \mid like\ green) \times P(\langle /s\rangle \mid and\ ham$

$P(and \mid green\ eggs) \times P(ham \mid eggs\ and) \times$

$= 1/2 \times 1 \times 1 \times 1 \times 1 \times 1 = 1/2 = \underline{\underline{0.5}}.$

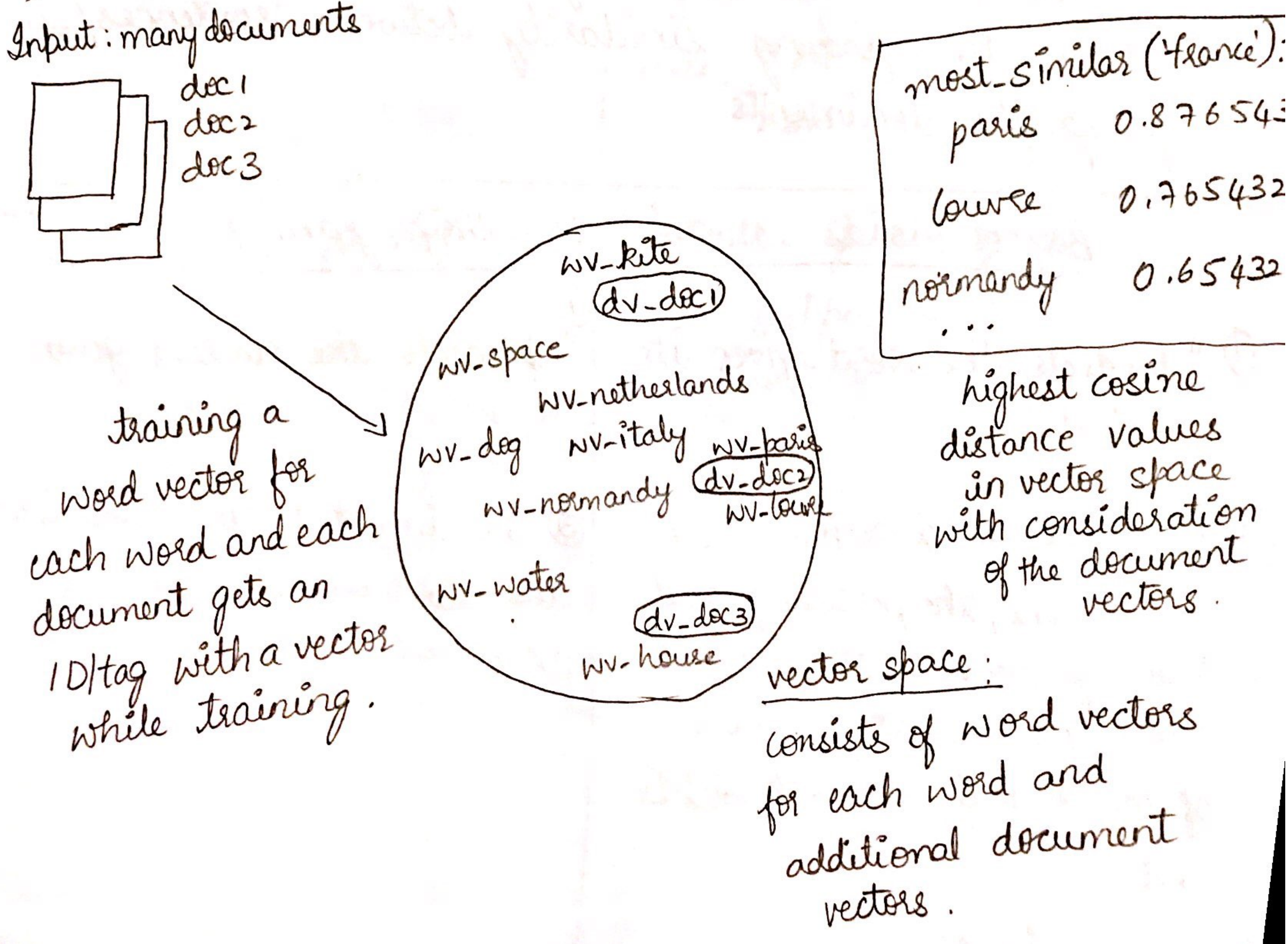
II.

a) In the Word2Vec model, the input is a large document and for each word in the document, a vector is built. With all the word vectors you have vector space which is the model of word2vec. By calculating the cosine distance (similarity) between word vectors you get the most similar words you looked for a word.

b)

DOC 2 VEC.

b)



most_similar ('france'):

paris	0.876543
louvre	0.765432
normandy	0.65432
...	

training a word vector for each word and each document gets an ID/tag with a vector while training.

highest cosine distance values in vector space with consideration of the document vectors.

vector space:

consists of word vectors for each word and additional document vectors.

Word vectors generated by the neural net have nice semantic and syntactic behaviors. We need a clear way to combine them into high quality document. vectors.

So here we can use Doc2vec, an unsupervised algorithm to generate vectors for sentence / paragraphs/ documents. The algorithm is an adaptation of word2vec which can generate vectors for words.

The vectors generated by doc2vec can be used for tasks like finding similarity between sentences/ paragraphs / documents.

Bag of Words model vs Skip gram model.

① "predicts the word given its content"	① "predicts the context given a word".
② The input could be W_{i-2}, W_{i-1}, W_{i+1}, W_{i+2}, the preceding and following words of the current word we are at. The output of the neural network will be W_i.	② The input to the model is W_i and the output could be W_{i-1}, W_{i-2}, W_{i+1}, W_{i+2}.
③ ~~several times~~	

③ several times faster to train than the skip-gram and better accuracy for the frequent words.

③ words well with small amount of training data, represents well even rare words or phrases.

④ Ex: "Hi fred, how was the pizza?"

Bag of words - 3 grams.

"hi fred how" "fred how was" "how was the" - - - - -

Skip-gram - 3grams

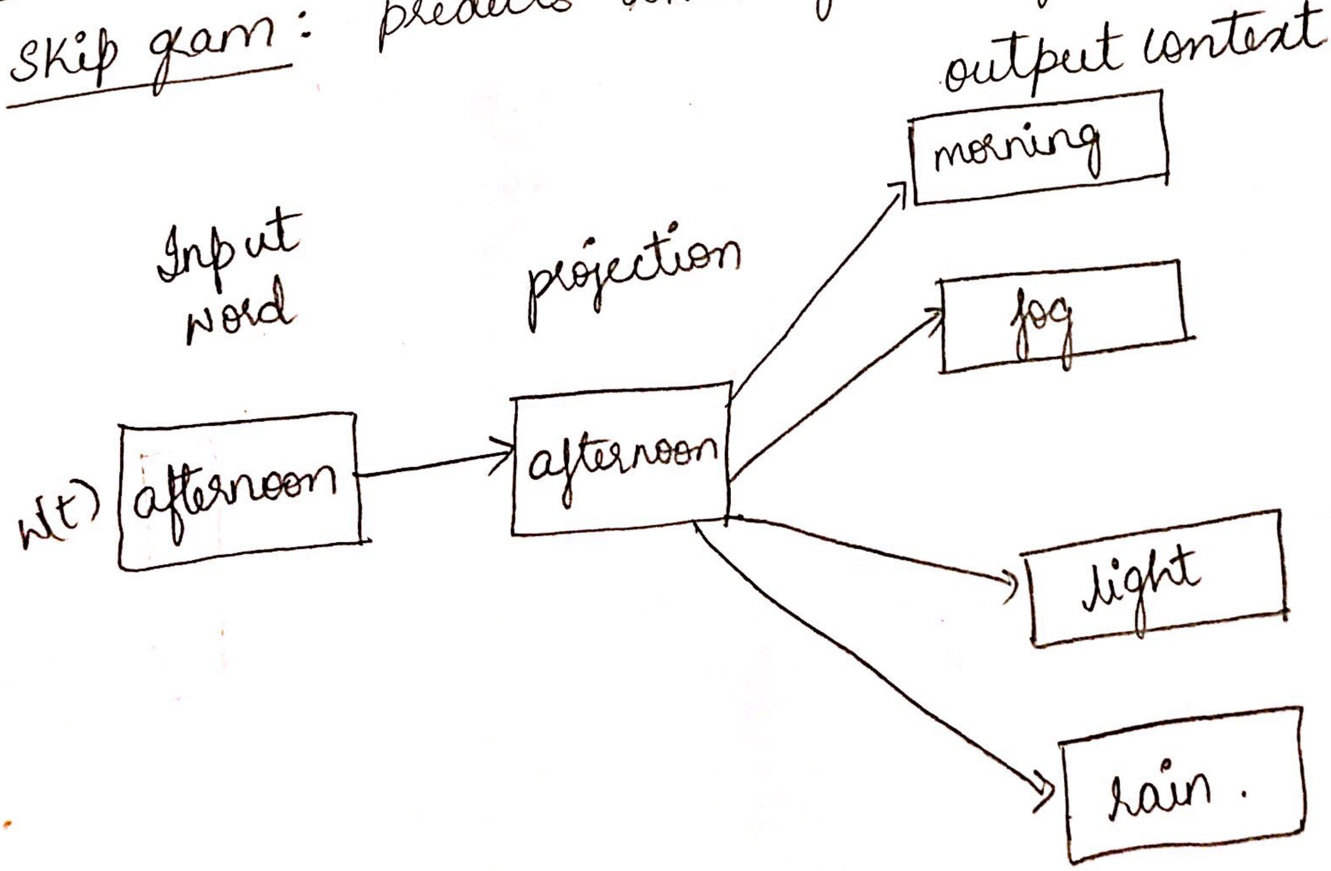
"Hi fred how", "Hi fred was

"fred how was", "fred low t

Here, "Hi fred was" skips

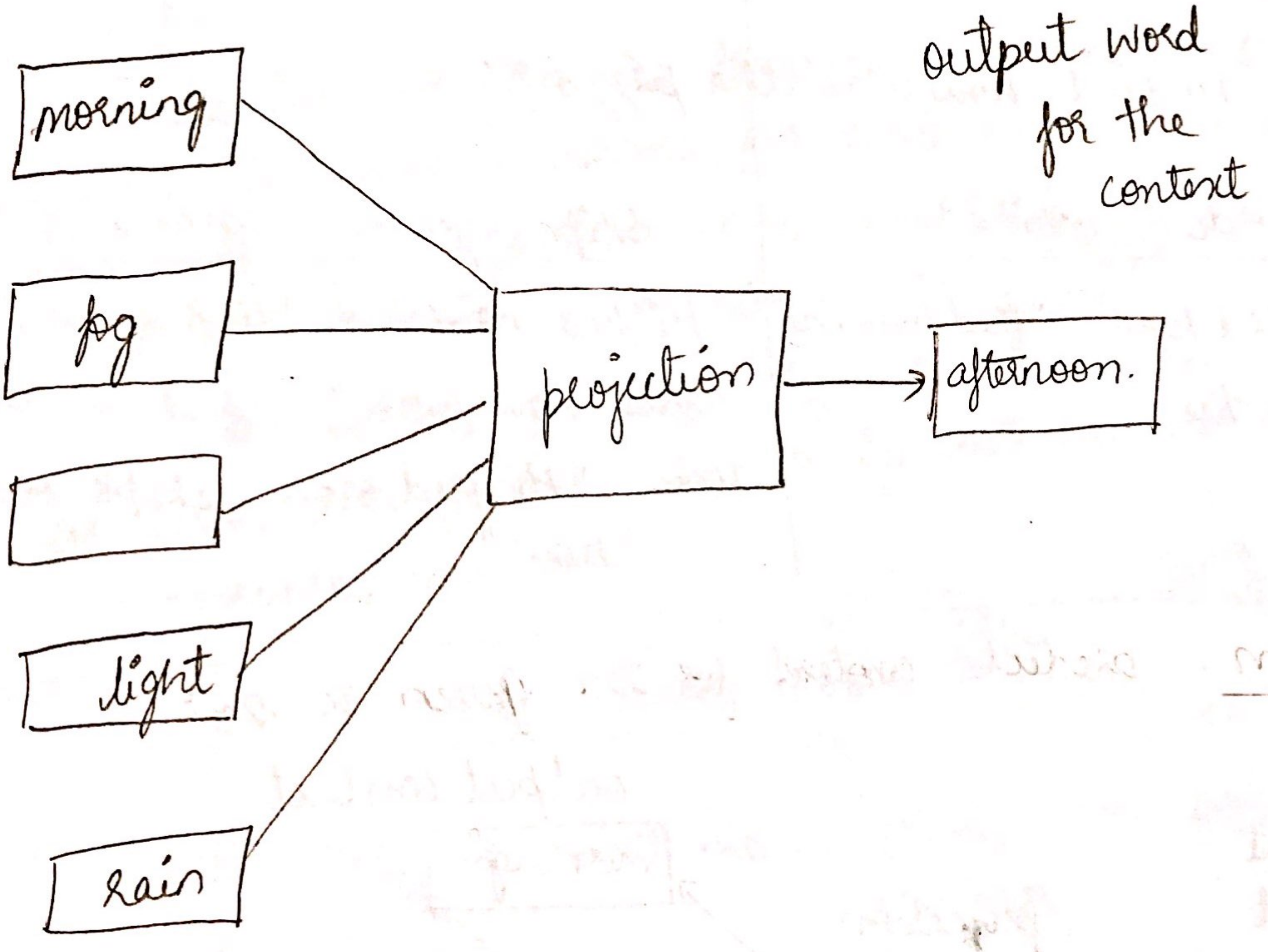
"how".

Skip gram: predicts context for the given word.



Continuous Bag of Words : predicts word for a given context.

Input context


output word for the context
morning
fog
light
rain
projection
afternoon.