CS5560 Knowledge Discovery and Management

Problem Set 4

June 26 (T), 2017

Name: SUJITHA PUTHANA

Class ID: 24

I. N-Gram

Consider a mini-corpus of three sentences

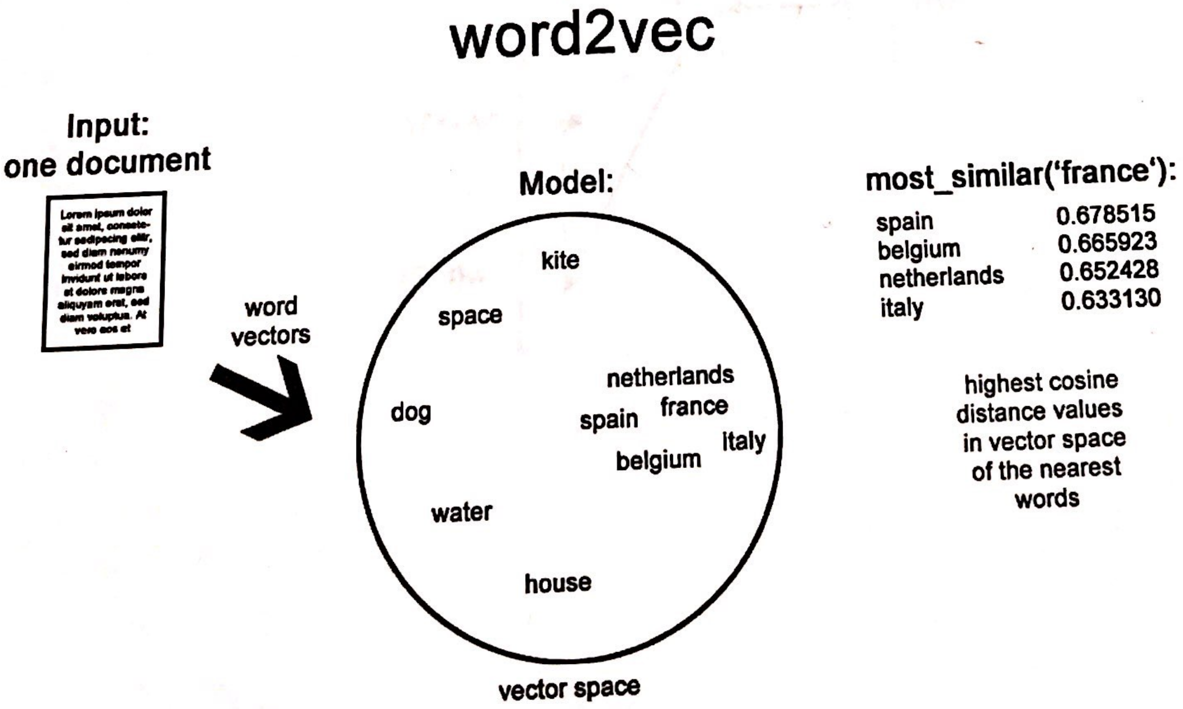
<s> I am Sam </s>
<s> Sam I am </s>
<s> I like green eggs and ham </s>

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
france
spain
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

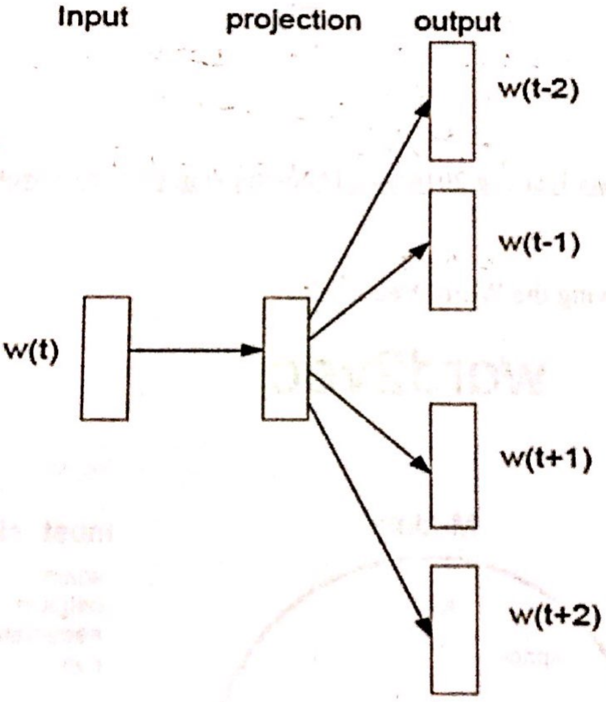
b. Describe How to extend this model for multiple documents. Also draw a similar diagram for the extended model.

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.



Given Sentences

① <s> I am Sam </s>

② <s> Sam I am </s>

③ <s> I like green eggs and ham </s>

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

Completes one round calculation of <s>①

$P(I / \langle s \rangle) = \frac{2}{3} = 0.67$ [Here the (I | <s>) exist is ① & ③ sentence and, total available sentence are 3
I → ①, ②, ③]

$P(am | I) = \frac{2}{3} = 0.67$ [①, ②
w (I) = 3]

$P(sam | am) = \frac{1}{2} = 0.5$ [①] [w (am) = 2]

$P(\langle /s \rangle | Sam) = \frac{1}{2} = 0.5$ [①] [w (sam) = 2]

<s> ②

$P(Sam | \langle s \rangle) = \frac{1}{3} = 0.33$ [②] [w (<s>) = 3]

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

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

<s> ③

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

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

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

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

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

$P(\langle s \rangle | ham) = \frac{1}{1} = 1$

④

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

$$P(I \mid \langle s\rangle) * P(\text{like} \mid I) * P(\text{green} \mid \text{like}) * P(\text{eggs} \mid \text{green}) * P(\text{and} \mid \text{eggs}) * P(\text{ham} \mid \text{and}) * P(\langle /s\rangle \mid \text{ham})$$

$$= 0.67 * 1 * 1 * 1 * 1 * 1 * 1 = \boxed{0.67}$$

$$\boxed{\text{Bigram Probability} = 0.67}$$

b) Trigram Probability :-

$$\boxed{P(w_i \mid w_{i-1}, w_{i-2}) = \frac{C(w_i, w_{i-1}, w_{i-2})}{C(w_{i-1}, w_{i-2})}}$$

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

$$P(\text{like} \mid \langle s\rangle\, I) * P(\text{green} \mid I\ \text{like}) * P(\text{eggs} \mid \text{like green}) * P(\text{and} \mid \text{green eggs}) * P(\text{ham} \mid \text{eggs and}) * P(\langle /s\rangle \mid \text{and ham})$$

$$P(\text{like} \mid \langle s\rangle\, I) = \frac{1}{2} = 0.5$$

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

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

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

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

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

$\Rightarrow 0.5 * 1 * 1 * 1 * 1 * 1$

$= 0.5$

$$\boxed{\text{Trigram Probability} = 0.5}$$

II)

a) **Word2Vec Model**

-> The input of the word2vec model is a large documend and for each word in the document, a vector is built.

-> With all the word vectors, we have vector space which is the model of word2vec.

-> We get the most similar words by calculating the cosine distinance i.e., similarity.

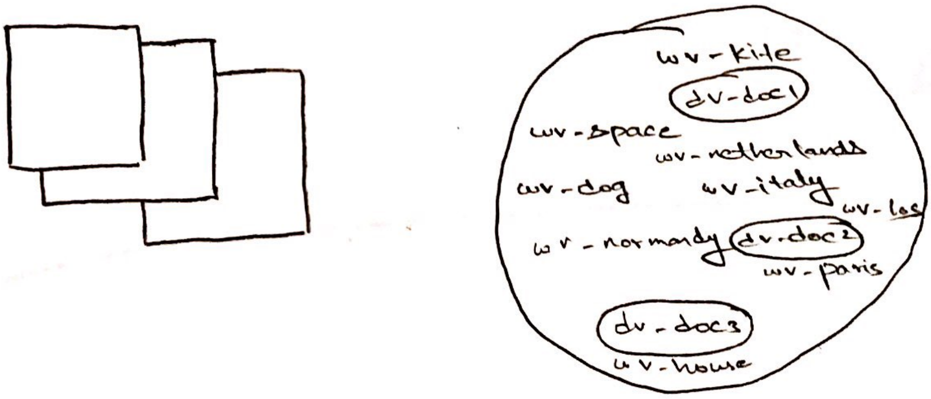
b) **Doc2Vec**

-> Doc2Vec, an unsupervised algorithm to generate vectors for sentence, paragraphs, documents.

-> This algorithm is an adaptation of word2vec which can generate vector of words.

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

Input
Multiple
documents



Vector space

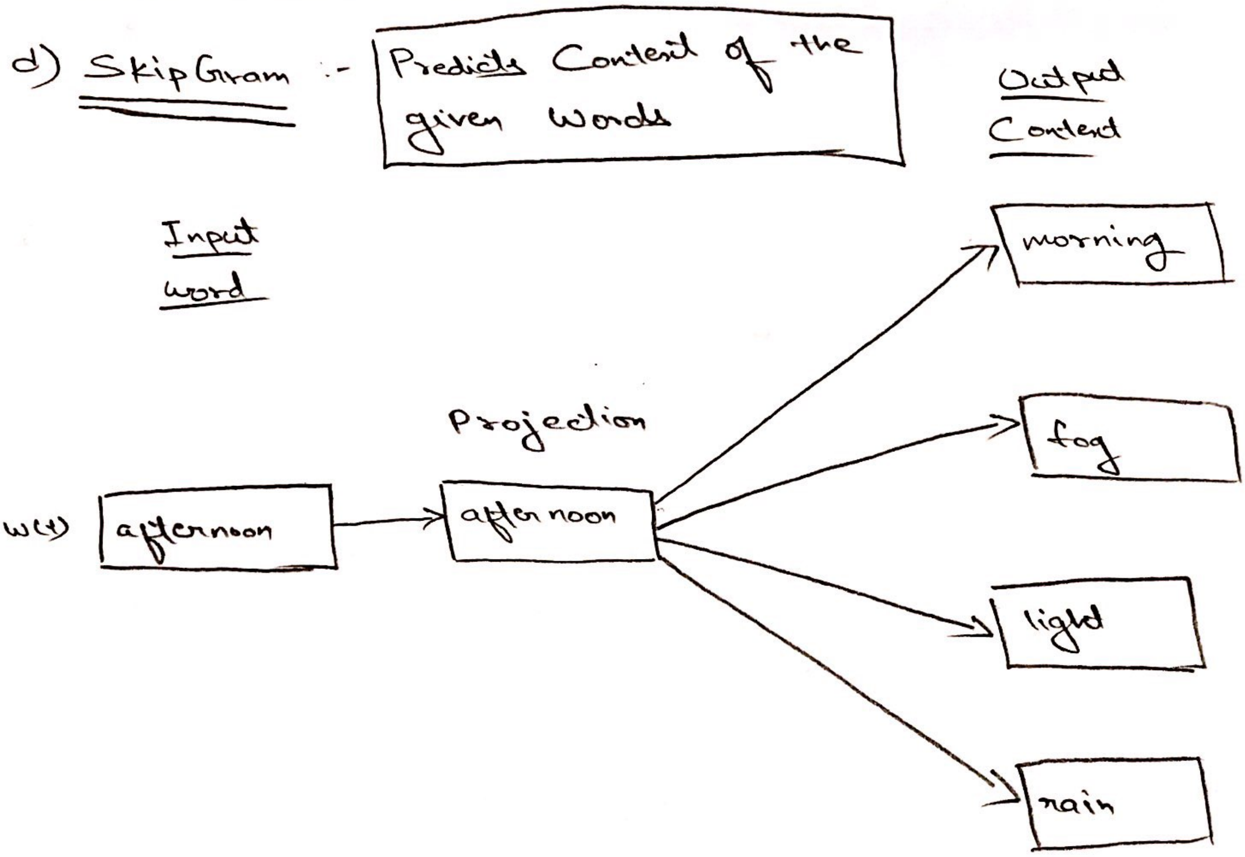
most_similar ('france')	
paris	0.87654
kite	0.7654
dog	0.65

→ Vector space consits of word vectors for each word and additional document vectors.

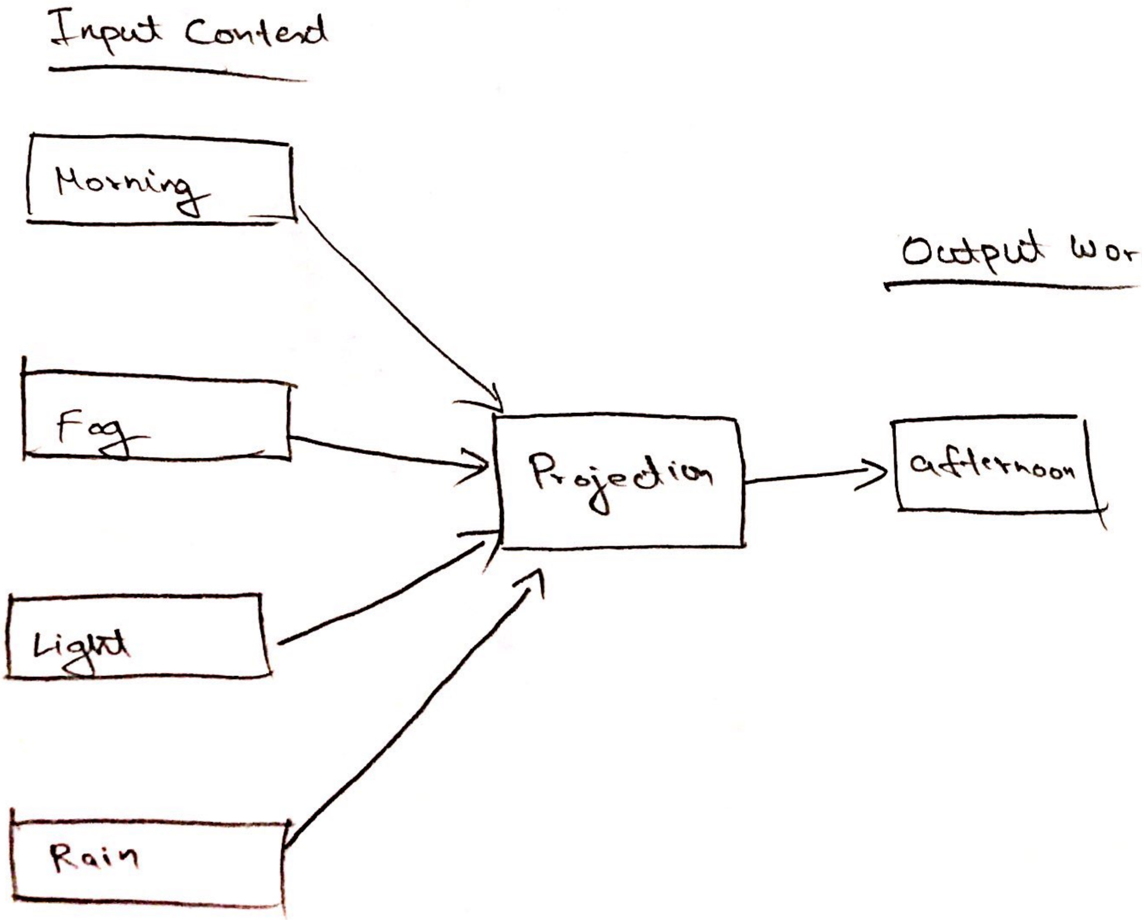
ⓒ

Bag of words	Skip gram models
① Input - $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$ Output - w_i	① Input - w_i Output - $w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}$
② Predicts the word given its context	② Predicts the context given a word
③ Faster to train than skip-gram. Better accuracy for the frequent words.	③ For even small amount of training data, it gives well even rare words & phrases.

d) <u>SkipGram</u> :- Predicts Context of the given Words



<u>Continuous Bag of Words</u> :- Predicts words for given context


Input Context
Morning
Fog
Light
Rain
Projection
Output word
afternoon