NLP Sheet 4

1) . $N = 10{,}000$ documents
. One of them has 250 words where 'sheet' occurs 20 times (call it D)
. 'sheet' also occurs in 2500 documents (overall)
→ what is tf-idf for 'sheet' in a bag of words representation of D.

Recall.

$$tf_{t,d} = \frac{Count(t,d)}{len(d)}$$

$$\rightarrow tf_{sheet,D} = \frac{20}{250} = \frac{2}{25}$$

$$Idf_t = \log_{10}(N/df_t)$$

df_t: No. of docs where t appears

$$\rightarrow idf_{sheet} = \log_{10}\left(\frac{10{,}000}{2500}\right) = 0.6$$

. $tf\text{-}idf = tf * idf = 0.048$

2) Given word vectors

$w_1 = (1, 0.4, 0.3, 0.8, 0.04)$
$w_2 = (1, 0.2, 0.5, 0.7, 0.6)$

. Are they similar or not?

$$\cos(w_1, w_2) = \frac{1.1 + 0.4 \cdot 0.2 + 0.3 \cdot 0.5 + 0.8 \cdot 0.7 + 0.04 \cdot 0.6}{\sqrt{1^2 + 0.4^2 + 0.3^2 + 0.8^2 + 0.04^2}\ \sqrt{1 + 0.2^2 + 0.5^2 + 0.7^2 + 0.6^2}}$$

$$= 0.9$$

- They are similar (slightly below max similarity 1.0)
- Make one change to reduce similarity
  → let's try inverting the biggest term (causes largest reduction in cos θ)

→ Now $\cos(w_1, w_2) = -0.09$

- $-0.09 \ll 1$ and we can no longer say the two words are similar.

3. Given are 3 sentences

→ Perform text normalization (stopword removal & lemmatization)

D1: natural language ~~processing~~ Process is ~~becoming~~ become important ~~so~~ soon we will begin ~~talking~~ talk to our ~~computers~~ computer.

D2: If ~~computers~~ computer understand natural language ~~they will become much simpler to use~~ simple

D3: Speech recognition is ~~the~~ first step to build ~~computers~~ computer like us

Vocab: natural, language, process, become, important, begin, talk, computer, understand, simple, use, speech, recognition, first, step, build, like

b) What is document vector for D3 → use Freq. Counts ①
→ then tf-idf ②

→ has |V| elements (for each word in vocab.)

Vocab.	D_3	tf_{t,D_3}	idf_t	tf-idf
natural	0			0
language	0			0
Process	0			0
become	0			0
important	0			0
begin	0			0
talk	0			0
computer	1	1/7	$\log_{10}(\frac{3}{3})$ ← N_{docs} / ← docs with computer	0
understand	0			0
simple	0			0
use	0			0
speech	1	1/7	$\log_{10}(3/1)$	0.068
recognition	1	1/7	$\log_{10}(3/1)$	0.068
first	1	1/7	$\log_{10}(3/1)$	0.068
step	1	1/7	$\log_{10}(3/1)$	0.068
build	1	1/7	$\log_{10}(3/1)$	0.068
like	1	1/7	$\log_{10}(3/1)$	0.068

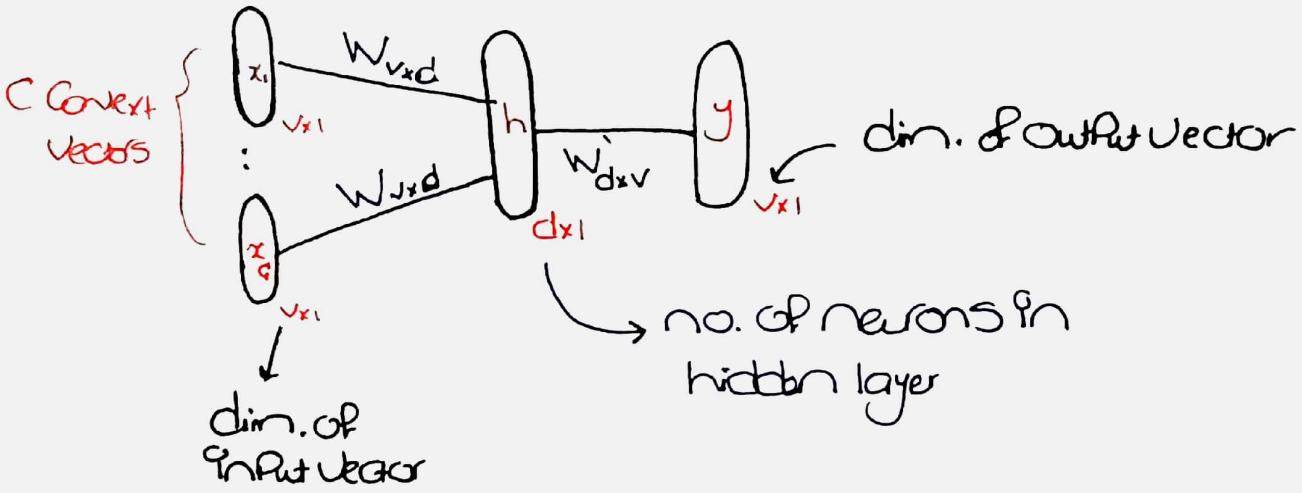
- Document vector (Frequency counts)
- normalized frequency counts (D_3 has 7 words)
- tf-idf vector

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

4. Consider CBOW model

→ V words in the vocab.
→ d is embedding dimension

• Draw the architecture



• For Skip-gram each of the dimensions is the same $(V, V \times d, d, d \times V, V)$
→ but only 1 input vector (target word)
→ C or 1 output vectors depending on formulation

5. Find big-O runtime of computing a single prob. $P(\text{context} = c \mid \text{word} = w)$ for Skip-gram
→ write it in terms of embedding dim in d and vocab size V

• To multiply $A_{k \times m} \times B_{m \times n}$ it takes
→ we do kn dot products (for each elem. in the result)
→ each dot product involves two m vectors
  • m multiplications
  • m − 1 additions

- Hence, overall the matrix multiplication corresponds to Knm multiplications
  → will disregard the $Kn(m-1)$ additions since the Knm multiplications dominate the complexity.

- Skip-gram: (Big-Oh)

$$h = x_{1\times V} W_{V\times d} \rightarrow d.(V) \text{ multiplications}$$

$$u_c = h_{1\times d} W'_{d\times V} \rightarrow V.(d) \text{ multiplications}$$

$$y_c = \text{Softmax}(u_c) \rightarrow V \text{ multiplications}$$ ②

- Hence, overall complexity is $O(dV)$

6. Users A & B have used word2vec on a specific vocabulary

- Each of them has obtained two word vectors u_w, v_w for each word in the vocabulary. ↳ For w as context → For w as center (target)

- If for every two words w, w' in V (vocab.) it holds that

$$u_w^A \cdot v_{w'}^A = u_w^B \cdot v_{w'}^B \quad ①$$

- Can we claim $v_w^A = v_w^B \ \forall w \in V$

No.

- It's obvious that if $a.b = c.d$ then that doesn't imply $c = d$

- hence, can make the following counterexample

Let vocab:

	by A		by B	
	u_w^A	v_w^A	u_w^B	v_w^B
Malt	(1 1)	(0.5 0.5)	(2 2)	$(\frac{1}{3}\ \frac{1}{3})$
Thresh	(2 2)	(1 0)	(3 3)	$(\frac{1}{2}\ 0)$

- There's only two pairs of words

(Malt, thresh):

$$(1\ 1).(1\ 0) = (2\ 2)(\tfrac{1}{2}\ 0)$$

(thresh, malt):

$$(2\ 2).(\tfrac{1}{2}\ \tfrac{1}{2}) = (3\ 3).(\tfrac{1}{3}\ \tfrac{1}{3})$$

* Hence, ① from last page holds for every pair of words

- however, clearly for each word

→ $v_w^A \neq v_w^B$ (Thresh, Malt)

Q.E.D

7. Compare word2vec to co-occurence based count methods

	Word2vec	Co-occurence
disadv.	• Only takes local statistics into account	• Global & local* statistics (e.g. word-word matrix)
adv.	• Dense, short vector	• Sparse, long vector