

Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP**
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

Sparse Vector Representations

One Hot

The cat sat on the mat

The: [0 1 0 0 0 0 0]

cat: [0 0 1 0 0 0 0]

sat: [0 0 0 1 0 0 0]

on: [0 0 0 0 1 0 0]

the: [0 0 0 0 0 1 0]

mat: [0 0 0 0 0 0 1]

Term-document matrix

Documents

Documents
We study the complexity of influencing elections through bribery: How computationally complex is it for an external actor to determine whether by a certain amount of bribing voters a specified candidate can be made the election's winner? We study this problem for election systems as varied as scoring ...

Vector-space representation

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

sugar, a sliced lemon, a tablespoonful of their enjoyment. Cautiously she sampled her first well suited to programming on the digital for the purpose of gathering data and

apricot
pineapple
computer.
information

preserve or jam, a pinch each of, and another fruit whose taste she likened In finding the optimal R-stage policy from necessary for the study authorized in the

Word-word matrix

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

Sparse Vector Representations (PMI)

Raw word frequency is skewed and non-discriminative

$$PMI(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)} \text{ where } word_1 \neq word_2$$

Positive Pointwise Mutual Information (PPMI) is preferred for its informative context-target word association

Positive PMI (PPMI) between $word_1$ and $word_2$

$$PPMI(word_1, word_2) = \max(\log_2 \frac{p(word_1, word_2)}{p(word_1)p(word_2)}, 0)$$

Sparse Vector Representations (PMI) Cont'd..

	aardvark	computer	data	pinch	result	sugar
apricot	0	0	0	1	0	1
pineapple	0	0	0	1	0	1
digital	0	2	1	0	1	0
information	0	1	6	0	4	0

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$

$$p_{i*} = \frac{\sum_{j=1}^C f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$

$$p_{*j} = \frac{\sum_{i=1}^W f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*} p_{*j}}$$

$$ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Sparse Vector Representations (PMI) Cont'd..

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$

$$p(w_i) = \frac{\sum_{j=1}^C f_{ij}}{N} \quad p(c_j) = \frac{\sum_{i=1}^W f_{ij}}{N}$$

$$p(w=\text{information}, c=\text{data}) = 6/19 = .32$$

$$p(w=\text{information}) = 11/19 = .58$$

$$p(c=\text{data}) = 7/19 = .37$$

	Count(w,context)					
	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	
pineapple	0	0	1	0	1	
digital	2	1	0	1	0	
information	1	6	0	4	0	

	p(w,context)					p(w)
	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

Sparse Vector Representations (PMI) Cont'd..

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_i * p_j}$$

		p(w,context)					p(w)
		computer	data	pinch	result	sugar	
	apricot	0.00	0.00	0.05	0.00	0.05	0.11
	pineapple	0.00	0.00	0.05	0.00	0.05	0.11
	digital	0.11	0.05	0.00	0.05	0.00	0.21
	information	0.05	0.32	0.00	0.21	0.00	0.58
	p(context)	0.16	0.37	0.11	0.26	0.11	

$$pmi(\text{information}, \text{data}) = \log_2 (.32 / (.37 * .58)) = .58$$

	PPMI(w,context)				
	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Distributional Representation - Knowing Words by their Context

C1: A bottle of ____ is on the table.

C2: Everybody likes ____.

C3: Don't have ____ before you drive.

C4: We make ____ out of corn.

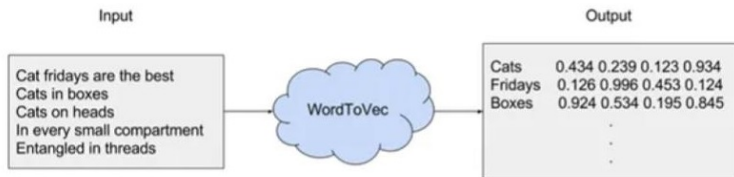
	C1	C2	C3	C4
tejuino	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0

“words that occur in similar contexts tend to have similar meanings”

Dense Vector Representations

- Dense vectors, being shorter and requiring fewer adjustments in machine learning, offer ease in feature utilization.
- Additionally, they excel in capturing nuanced relationships between words like synonyms, unlike sparse representations, allowing better generalization and similarity capture in language models.

Word Embedding - Overview of Word2Vec



```
[ 0.50451, 0.68607, -0.59517, -0.022801, 0.60046, -0.13498, -0.08813, 0.47377, -0.61798, -0.31012,
-0.076666, 1.493, -0.034189, -0.98173, 0.68229, 0.81722, -0.51874, -0.31503, -0.55809, 0.66421, 0.1961
, -0.13495, -0.11476, -0.30344, 0.41177, -2.223, -1.0756, -1.0783, -0.34354, 0.33505, 1.9927,
-0.04234, -0.64319, 0.71125, 0.49159, 0.16754, 0.34344, -0.25663, -0.8523, 0.1661, 0.40102, 1.1685,
-1.0137, -0.21585, -0.15155, 0.78321, -0.91241, -1.6106, -0.64426, -0.51042 ]
```

"king"



"Man"

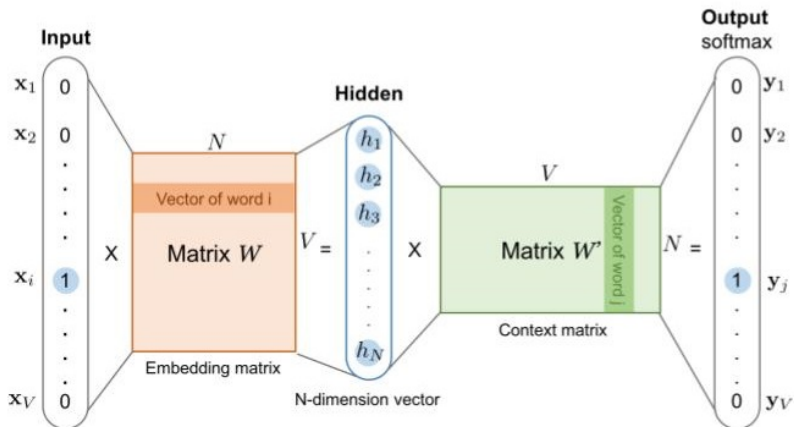


"Woman"

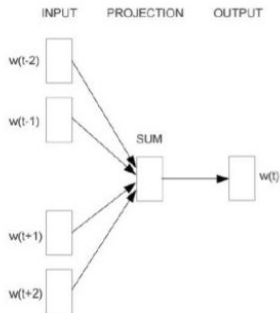


The vector representations are effective in capturing substantial information, meaning, and associations of these words.

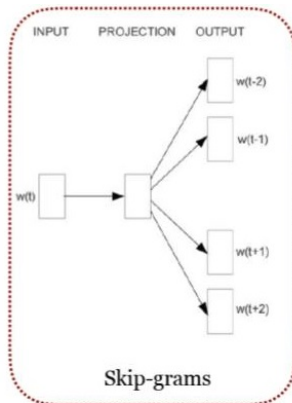
Overall Skipgram Architecture



Dense Vector Representations - Skipgram



Continuous Bag of Words (CBOW)

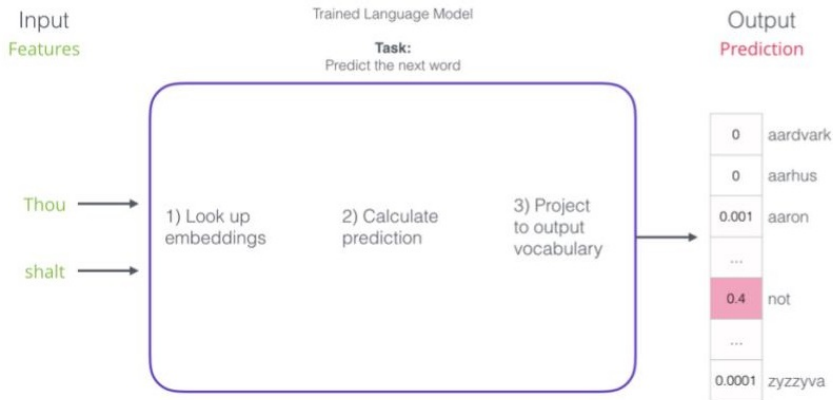


Skip-grams

Dense Vector Representations - Language Modeling



Dense Vector Representations - Language Modeling Cont'd



Language Modeling - Training Dataset Generation with CBOW

Thou shalt not make a machine in the likeness of a human mind

thou	shalt	not	make	a	machine	in	the	...
------	-------	-----	------	---	---------	----	-----	-----

input word	target word
not	thou
not	shalt
not	make
not	a

Language Modeling - Training Skipgram

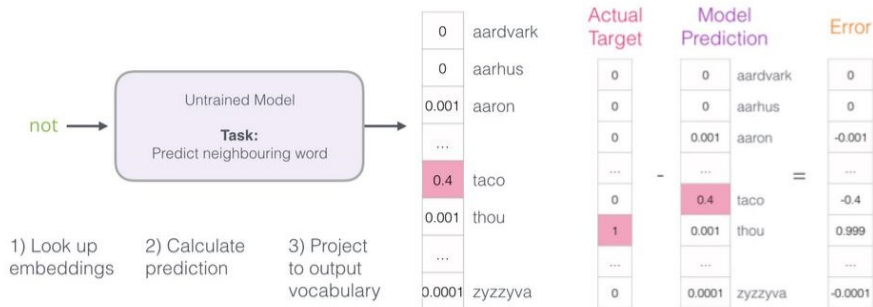
input word	target word
not	thou
not	shalt
not	make
not	a
make	shalt
make	not
make	a
make	machine
a	not
a	make
a	machine
a	in
machine	make
machine	a
machine	in
machine	the
in	a
in	machine
in	the
in	likeness

not →

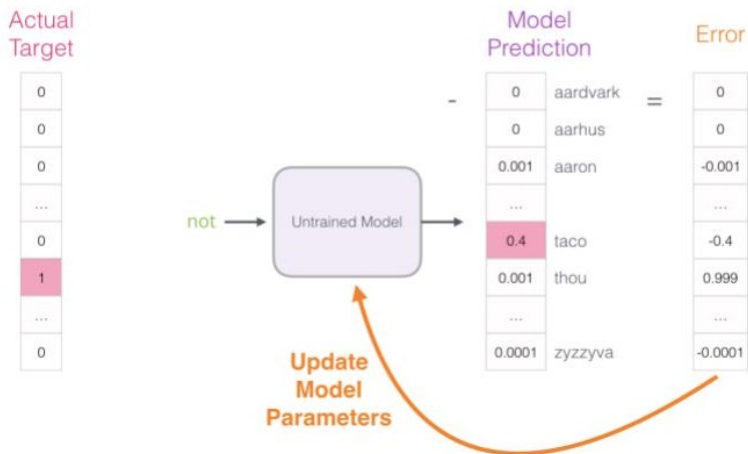
Untrained Model

Task:
Predict neighbouring word

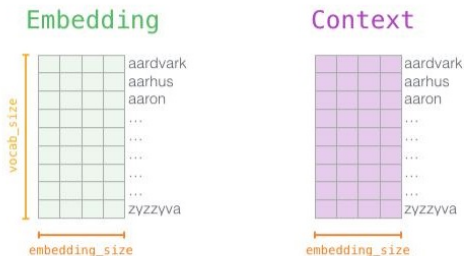
Language Modeling - Training Skipgram Cont'd



Language Modeling - Training Skipgram Cont'd



Word2Vec Training - Skipgram with Negative Sampling

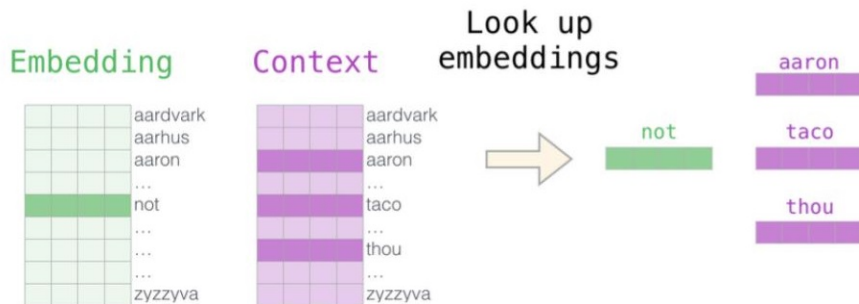


Initialized with random values

dataset

input word	output word	target
not	thou	1
not	aaron	0
not	taco	0
not	shalt	1
not	mango	0
not	finglonger	0
not	make	1
not	plumbus	0
...

Word2Vec Training - Skipgram with Negative Sampling Cont'd



Analogy

$\text{vector}('king') - \text{vector}('man') + \text{vector}('woman') \approx \text{vector}('queen')$

$\text{vector}('Paris') - \text{vector}('France') + \text{vector}('Italy') \approx \text{vector}('Rome')$

