COMP9444: Neural Networks and Deep Learning

Week 5b. Word Vectors

Sonit Singh
School of Computer Science and Engineering

June 25, 2024

Outline

- Statistical Language Processing
- *n*-gram models
- co-occurence matrix
- word representation
- Word2Vec
- GloVe
- word relationships

Word Meaning - Synonyms and Taxonomy?

What is the meaning of meaning?

- dictionary definitions
- synonyms and antonyms
- taxonomy
  - penguin is-a bird is-a mammal is-a vertebrae

Statistical Language Processing

Synonyms for “elegant”

> *stylish, graceful, tasteful, discerning, refined, sophisticated, dignified, cultivated, distinguished, classic, smart, fashionable, modish, decorous, beautiful, artistic, aesthetic, lovely; charming, polished, suave, urbane, cultured, dashing, debonair; luxurious, sumptuous, opulent, grand, plush, high-class, exquisite*

Synonyms, antonyms and taxonomy require human effort, may be incomplete and require discrete choices. Nuances are lost. Words like “king”, “queen” can be similar in some attributes but opposite in others.

Could we instead extract some statistical properties automatically, without human involvement?

There was a Crooked Man

There was a crooked man,
who walked a crooked mile
And found a crooked sixpence
upon a crooked stile.
He bought a crooked cat,
who caught a crooked mouse
And they all lived together
in a little crooked house.

www.kearley.co.uk/images/uploads/JohnPatiencePJ03.gif

Counting Frequencies

word	frequency
a	7
all	1
and	2
bought	1
cat	1
caught	1
crooked	7
found	1
he	1
house	1
in	1
little	1
lived	1
man	1
mile	1
mouse	1
sixpence	1
stile	1
there	1
they	1
together	1
upon	1
walked	1
was	1
who	2

- some words occur frequently in all (or most) documents
- some words occur frequently in a particular document, but not generally
- this information can be useful for document classification

Document Classification

word	doc 1	doc 2	doc X
a	.	.	7
all	.	.	1
and	.	.	2
bought	.	.	1
cat	.	.	1
caught	.	.	1
crooked	.	.	7
found	.	.	1
he	.	.	1
house	.	.	1
in	.	.	1
little	.	.	1
lived	.	.	1
man	.	.	1
mile	.	.	1
mouse	.	.	1
sixpence	.	.	1
stile	.	.	1
there	.	.	1
they	.	.	1
together	.	.	1
upon	.	.	1
walked	.	.	1
was	.	.	1
who	.	.	2

Document Classification

- each column of the matrix becomes a vector representing the corresponding document
- words like “cat”, “mouse”, “house” tend to occur in children’s books or rhymes
- other groups of words may be characteristic of legal documents, political news, sporting results, etc.
- words occurring many times in one document may skew the vector – might be better to just have a “1” or “0” indicating whether the word occurs at all

Counting Consecutive Word Pairs

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a							6					1													
all													1												
and								1												1					
bought	1																								
cat																									1
caught	1																								
crooked					1					1				1	1	1	1								
found	1																								
he				1																					
house																									
in	1																								
little							1																		
lived																					1				
man																									1
mile			1																						
mouse			1																						
sixpence																						1			
stile									1																
there																								1	
they		1																							
together											1														
upon	1																								
walked	1																								
was	1																								
who						1																	1		

Predictive 1-Gram Word Model

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a							$\frac{1}{7}$					$\frac{1}{7}$													
all													1												
and								$\frac{1}{2}$													$\frac{1}{2}$				
bought	1																								
cat																									1
caught	1																								
crooked					$\frac{1}{7}$						$\frac{1}{7}$			$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$							
found	1																								
he			1																						
house																									
in	1																								
little							1																		
lived																					1				
man																									1
mile		1																							
mouse		1																							
sixpence																						1			
stile									1																
there																								1	
they		1																							
together											1														
upon	1																								
walked	1																								
was	1																								
who						$\frac{1}{2}$																	$\frac{1}{2}$		

N-Gram Model

- by normalizing each row (to sum to 1) we can estimate the probability $\mathrm{prob}(w_j|w_i)$ of word w_j occurring after w_i
- need to aggregrate over a large corpus, so that unusual words like “crooked” will not dominate
- the model captures some common combinations like “there was”, “man who”, “and found”, “he bought”, “who caught”, “and they”, “they all”, “lived together”, etc.
- this ***unigram*** model can be generalized to a bi-gram, tri-gram, . . . , *n*-gram model by considering the *n* preceding words
- if the vocabulary is large, we need some tricks to avoid exponential use of memory

1-Gram Text Generator

“Rashly – Good night is very liberal – it is easily said there is – gyved to a sore distraction in wrath and with my king may choose but none of shapes and editing by this , and shows a sea And what this is miching malhecho ; And gins to me a pass , Transports his wit , Hamlet , my arms against the mind impatient , by the conditions that would fain know ; which , the wicked deed to get from a deed to your tutor .”

Co-occurrence Matrix

- sometimes, we don't necessarily predict the next , but simply a "nearby word" (e.g. a word occurring within an *n*-word window centered on that word)
- we can build a matrix in which each row represents a word, and each column a nearby word
- each row of this matrix could be considered as a vector representation for the corresponding word, but the number of dimensions is equal to the size of the vocabulary, which could be very large ($\sim 10^5$)
  - is there a way to reduce the dimensionality while still preserving the relationships between words?

Co-occurrence Matrix (2-word window)

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a				1		1	6	1			1	1										1	1	1	
all													1							1					
and								1							1	1				1					
bought	1								1																
cat							1																		1
caught	1																								1
crooked	6				1					1		1		1	1	1	1	1							
found	1		1																						
he				1														1							
house							1																		
in	1																				1				
little	1						1																		
lived		1																			1				
man							1																		1
mile			1				1																		
mouse			1				1																		
sixpence							1															1			
stile							1		1																
there																								1	
they		1	1																						
together											1		1												
upon	1																1								
walked	1																								1
was	1																		1						
who					1	1								1										1	

Co-occurrence Matrix (10-word window)

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a	10	2	3	2	2	2	13	3	2	1	1	1	1	2	2	1	2	2	1	2	1	2	2	1	4
all	2		1				1				1	1	1			1				1	1				
and	3	1				1	3	1			1		1		1	1	1			1	1	1	1		2
bought	2				1	1	2		1									1				1			1
cat	2			1		1	2		1							1		1							1
caught	2		1	1	1		2									1				1					1
crooked	13	1	3	2	2	2	10	2	2	1	1	1	2	2	2	1	2	3	1	1	1	2	2	1	4
found	3		1				2								1		1					1	1		
he	2			1	1		2										1	1				1			1
house	1						1				1	1									1				
in	1	1	1				1			1		1	1							1	1				
little	1	1					1			1	1		1								1				
lived	1	1	1				2				1	1				1				1	1				
man	2						2								1				1				1	1	1
mile	2		1				2	1						1			1						1		1
mouse	1	1	1		1	1	1						1							1	1				1
sixpence	2		1				2	1	1						1			1				1			
stile	2			1	1		3		1								1					1			
there	1						1							1										1	1
they	2	1	1			1	1				1		1			1					1				
together	1	1	1				1			1	1	1	1			1				1					
upon	2		1	1			2	1	1								1	1							
walked	2		1				2	1						1	1									1	1
was	1						1							1					1				1		1
who	4		2	1	1	1	4		1					1	1	1			1				1	1	

Co-occurrence Matrix

- by aggregating over many documents, pairs (or groups) of words emerge which tend to occur near each other (but not necessarily consecutively)
  - “cat”, “caught”, “mouse”
  - “walked”, “mile”
  - “little”, “house”
- common words tend to dominate the matrix
  - could we sample common words less often, in order to reveal the relationships of less common words?

Word Embeddings

"Words that are used and occur in the same contexts tend to purport similar meanings."

Z. Harris (1954)

"You shall know a word by the company it keeps."

J.R. Firth (1957)

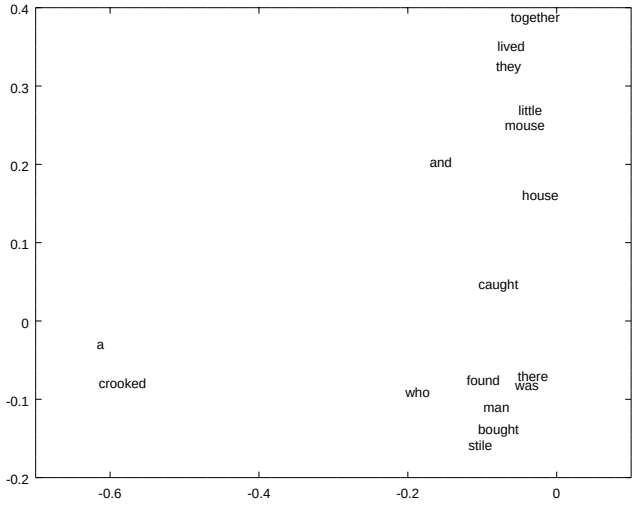
Aim of Word Embeddings:

Find a vector representation of each word, such that words with nearby representations are likely to occur in similar contexts.

History of Word Embeddings

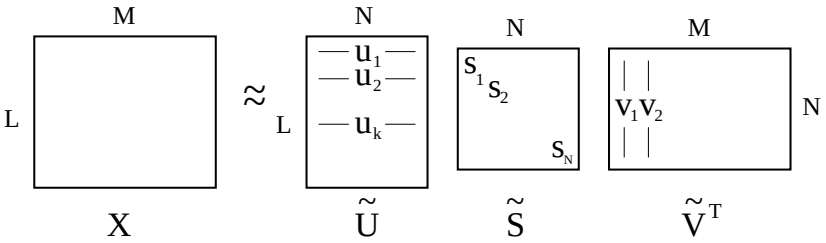
- Structuralist Linguistics (Firth, 1957)
- Recurrent Networks (Rumelhart, Hinton & Williams, 1986)
- Latent Semantic Analysis (Deerwester et al., 1990)
- Hyperspace Analogue to Language (Lund, Burgess & Atchley, 1995)
- Neural Probabilistic Language Models (Bengio, 2000)
- NLP (almost) from Scratch (Collobert et al., 2008)
- word2vec (Mikolov et al., 2013)
- GloVe (Pennington, Socher & Manning, 2014)

Word Embeddings



Singular Value Decomposition

Co-occurrence matrix $X_{(L\times M)}$ can be decomposed as $\mathrm{X} = \mathrm{U}\,\mathrm{S}\,\mathrm{V}^{\mathrm{T}}$ where $U_{(L\times L)}$, $V_{(M\times M)}$ are unitary (all columns have unit length) and $S_{(L\times M)}$ is diagonal, with diagonal entries $s_1 \geq s_2 \geq \ldots \geq s_M \geq 0$



We can obtain an approximation for X of rank $N < M$ by truncating U to $\tilde{\mathrm{U}}_{(L\times N)}$, S to $\tilde{\mathrm{S}}_{(N\times N)}$ and V to $\tilde{\mathrm{V}}_{(N\times M)}$. The kth row of $\tilde{\mathrm{U}}$ then provides an N-dimensional vector representing the k^{th} word in the vocabulary.

Word2Vec and GloVe

For language processing tasks, typically, L is the number of words in the vocabulary (about $60,000$) and M is either equal to L or, in the case of document classification, the number of documents in the collection. SVD is computationally expensive, proportional to $L \times M^2$ if $L \geq M$. Can we generate word vectors in a similar way but with less computation, and incrementally?

- Word2Vec
  - predictive model
  - maximize the probability of a word based on surrounding words
- GloVe
  - count-based model
  - reconstruct a close approximation to the co-occurrence matrix X

Eigenvalue vs. Singular Value Decomposition

Eigenvalue Decomposition:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \Omega \, \mathrm{D} \, \Omega^{-1}, \quad \text{where} \quad \Omega = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathrm{D} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \Omega \, \mathrm{D} \, \Omega^{-1}, \quad \text{where} \quad \Omega = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}, \quad \mathrm{D} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}$$

Singular Value Decomposition:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \mathrm{USV}^{\mathrm{T}}, \quad \mathrm{U} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathrm{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathrm{V} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \mathrm{USV}^{\mathrm{T}}, \quad \mathrm{U} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathrm{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathrm{V} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Eigenvalue vs. Singular Value Decomposition

- if X is symmetric and positive semi-definite, eigenvalue and singular value decompositions are the same.
- in general, eigenvalues can be negative or even complex, but singular values are always real and non-negative.
- even if X is a square matrix, singular value decompositon treats the source and target as two entirely different spaces.
- the word co-occurrence matrix is symmetric but not positive semi-definite; for example, if the text consisted entirely of two alternating letters ..ABABABABABABAB.. then A would be the context for B, and vice-versa.

Summary

- *Word vectors*, also sometimes called *word embeddings* or *word representations* are distributed representations of words.
- Two kinds of embeddings
  - Sparse vectors: Words are represented by a simple function of the counts of nearby words.
  - Dense vectors: Representation is created by training a classifier to distinguish nearby and far-away words
- The ***contexts*** in which a word appears tells us a lot about what it means. Distributional similarities use the set of contexts in which words appear to measure their similarity
- Word2Vec and GloVe are two important dense representations of words.
- Various choice:
  - dimensionality of embeddings (50, 100, 200, 300, 500)
  - scale, quality and type of text to get word embeddings
  - size of the context window