Word Embedding

Ramaseshan Ramachandran

WHAT IS CONTEXT?

- All words within a window or ideally within a sentence
- All content words within a window or sentence that fall in a certain frequency range
- All content words which stand in closest proximity to the word in question in the grammatical schema of each window or sentence

CONTEXT

- Context influences the word meaning
- Small boy, small car, small house, small island
- Words that occur in similar contexts will tend to have similar meanings
- Semantic similarity beween two words (w_x, w_y) is a function of how frequently they appeared in similar linguistic contexts
  - $\vec{w_x} \approx \vec{w_y}$ when the frequency of the context $(f_{C_{xy}(k)})$ with a window of size k in which both words w_x and w_y appeared is higher
- If $f_{C_{xy}}(k)$ is higher, then the semantic relationship of (w_x, w_y) is stronger
- Extending to multiple similar words for w_x:

  $\vec{w_x} \approx \vec{w_{y_i}}$ when the frequency of $C_{xy_i}(k)$ is higher, where $i = 1 \ldots n$

  Note: Here $approx$ represents similarity

SEMANTIC SPACE

- A space where the similar words (synonyms, hyponyms, hypernyms) are classified and arranged in various axes
  - Colour (hypernym) - $\underbrace{\text{Red}, \text{Green}, \text{Orange}}_{\text{co-hyponyms}}$ (hyponym) - Attributional Similarity
- A space where the similar words (synonyms, hyponyms, hypernyms) are classified and arranged in various axes
- A model or models that automatically find similar words are known as Distributed Semantic Models (DSM)
- Semantically similar words are found automatically using co-occurrences/co-locations/context

  OR
- Words connected by similar patterns are probably semantically similar

DISTRIBUTED SEMANTIC MODELS

- Extract the meaning of the words using distributed linguistic properties
- Compute lexical co-occurrence of every word (co-locates with certain distance) with every other word in the Vocabulary
  - Linear proximity of words within a window is considered
  - They need not represent any relations

  Example He drove the car through a red bridge. The verb drove relates to red and bridge only through the proximity, but carries no relations with red and bridge in terms of semantics
- Build a co-occurrence matrix using co-occurrence statistics
- Rows/columns in the matrix represent distributed semantic information of words

DISTRIBUTED SEMANTIC MODELS

I cook dinner every Sunday

...

I cooked dinner last Sunday

...

I am cooking dinner today

...

My son cooks dinner every Sunday

...

- The words in this corpus are related by association
- The verb cook, cooked, cooks and cooking are related due to its co-occurrence statistics - semantic relationship
- The words dinner and Sunday are similar due to associative relationship and due to co-occurrence

- In the COVID19 corpus it iis difficult to search and find *needle in a haystack*
- You will find needle related to $\underbrace{\text{pain}, \text{illness}, \text{blood}, \text{drugs}, \text{syringe}}_{\text{Associative relationship}}$

and not to thread, knitting, cloth

You shall know a word by the company it keeps

- Firth, 1957

VECTOR REPRESENTATION OF WORDS

Let V be the unique set of terms and $|V|$ be the size of the vocabulary. Then every vector representing the word $\mathscr{R}^{|V|x1}$ would point to a vector in the V-dimensional space

ONE-HOT VECTOR - 1

Consider all the ≈39000 words (estimated tokens in English is ≈ 13M) in the Oxford Learner's pocket dictionary. We can represent each word as an independent vector quantity as follows in the real space $\mathscr{R}^{|V|X1}$

$$t^{a} = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \quad t^{aback} = \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \quad \dots \; t^{zoom} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 1 \\ 0 \end{pmatrix} \quad t^{zucchini} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \\ 1 \end{pmatrix}$$

This is a very simple codification scheme to represent words independently in the vector space. This is known as ***one-hot vector***.

ONE-HOT VECTOR - 2

In one-hot vector, every word is represented independently. The terms, *home, house, apartments, flats* are independently coded. With one-hot vector based model, the dot product

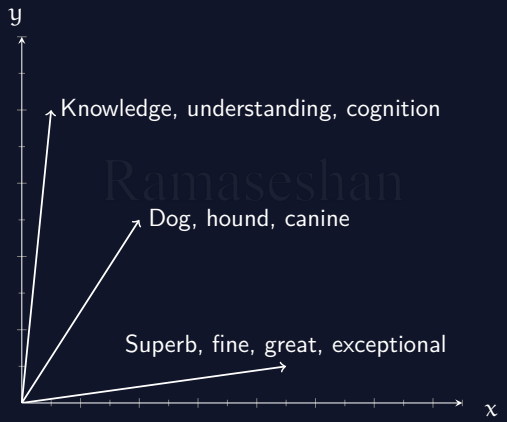
$$\left(t^{\text{House}}\right)^{T} \cdot t^{\text{Apartment}} = 0 \tag{1}$$

$$\left(t^{\text{Home}}\right)^{T} \cdot t^{\text{House}} = 0 \tag{2}$$

With one-Hot vector, there is no notion of similarity or synonyms.

RELATIONSHIP AMONG TERMS - SYNONYMS

We could represent all the synonyms of a word in one axis



POLYSEMOUS WORD - BANK

Synset('bank.n.01')	sloping land (especially the slope beside a body of water)
Synset('depository-financial-institution.n.01')	a financial institution that accepts deposits and channels the money into lending activities
Synset('bank.n.03')	a long ridge or pile
Synset('bank.n.10')	a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)
Synset('trust.v.01')	have confidence or faith in

Bank appears in different word senses - or the meaning of the word is determined by the context in which appears

POLYSEMOUS WORD - PROGRAM

Synset('plan.n.01')	a series of steps to be carried out or goals to be accomplished
Synset('program.n.02')	a system of projects or services intended to meet a public need
Synset('broadcast.n.02')	a radio or television show
Synset('platform.n.02')	a document stating the aims and principles of a political party
Synset('program.n.05')	an announcement of the events that will occur as part of a theatrical or sporting event
Synset('course_of_study.n.01')	an integrated course of academic studies
Synset('program.n.07')	(computer science) a sequence of instructions that a computer can interpret and execute
Synset('program.n.08')	a performance (or series of performances) at a public presentation
Synset('program.v.01')	arrange a program of or for
Synset('program.v.02')	write a computer program

SYNONYMS

small.a.01	['small', 'little']
minor.s.10	['minor', 'modest', 'small', 'small-scale', 'pocket-size', 'pocket-sized']
humble.s.01	['humble', 'low', 'lowly', 'modest', 'small']
little.s.07	['little', 'minuscule', 'small']
belittled.s.01	['belittled', 'diminished', 'small']
potent.a.03	['potent', 'strong', 'stiff']
impregnable.s.01	['impregnable', 'inviolable', 'secure', 'strong', 'unassailable', 'hard']
	He has such an impregnable defense (Cricket-Very hard to find the gap between the bat and the pad)
solid.s.07	['solid', 'strong', 'substantial']
strong.s.09	['strong', 'warm']
firm.s.03	['firm', 'strong'] - firm grasp of fundamentals

CONTEXTUAL UNDERSTANDING OF TEXT

You shall know a word by the company it keeps - (Firth, J. R. 1957)

- In order to understand the word and its meaning, it not enough if we consider only the individual word
- The *meaning* and *context* should be central in understanding word/text
- Exploit the context-dependent nature of words
- Language patterns cannot be accounted for in terms of a single entity
- The *collocation*, a particular word consistently co-occurs with the other words, gives enough clue to understand a word and its meaning

UNDERSTANDING A WORD FROM ITS CONTEXT

The view from the top of the mountain was
The view from the summit was
La vue du sommet de la montagne était
Mtazamo wa juu wa mlima huo ulikuwa

awesome/(*impressionnante*, *impressionnant*)
breathtaking
amazing, அற்புதமான/అద్భుతకరమాయ/
stunning/(*superbe*)/అద్భుతమైన
astounding अद्भुत/চমকপ্রদ
astonishing
awe-inspiring
extraordinary
incredible/(*incroyable*)
unbelievable
magnificent शानदार/ಗಂಭೀರಮಾಯ/ભવ્ય
wonderful/(*ajabu*)
spectacular
remarkable/(*yakuvutia*)

SEMANTICALLY CONNECTED VECTORS

- Identify a model that enumerates the relationships between terms
- Identify a model that tries to put similar items closer to each other in some space or structure
- Build a model that discovers/uncovers the semantic similarity between words and documents in the latent semantic domain
- Develop a distributed word vectors or dense vectors that captures the linear combination of word vectors in the transformed domain
- Transform the term-document space into a synonymy and a semantic space

METHODS TO CREATE WORD VECTORS

- Brown clustering - statistical algorithms for assigning words to classes based on the frequency of their co-occurrence with other words
- Hyperspace Analogue to Language - HAL
- Correlated Occurrence Analogue to Lexical Semantic - COALS
- Latent Semantic Analysis or Latent Semantic Indexing
- Global Vectors - GloVe
- Neural networks using skip grams and CBOW
  - CBOW - uses surrounding words to predict the center of words
  - Skip grams use center of words to predict the surrounding words

WORD SIMILARITY

- Sparse vectors are too long and not very convenient as features machine learning
- Abstracts more than just frequency counts
- It captures neighborhood words that are connected by synonyms

You shall know a word by the company it keeps

- Firth, 1957

ATTRIBUTIONAL SIMILARITY

Attributional Similarity

- Two words are similar if they shared similar attributes - cat and kitten, dog and puppy
- Refers to the degree of similarity between two words or phrases in terms of their shared attributes
- Words that share many collocates denote concepts that share many attributes

Relational Similarity

- Related by concepts/roles - King and queen are related by the roles in the monarchy
- Blood relationships - siblings, aunts, uncles, parents, etc.

Techniques - Distributional Semantic Models

VECTOR BASED MODELS

Assumption Context words within a certain distance from the target word are semantically relevant

- Ability to represent word meaning simply by using distributional statistics
- The context surrounding a given word provides important information about its meaning
  - Small number of words surrounding the target word is known as context
- Distributional patterns of co-occurrence with their neighboring words provide semantic properties of words

TECHNIQUES TO CAPTURE CO-OCCURRENCE INFORMATION

- Create a frequency matrix, where each row corresponds to a unique/target word, and each column represents a context
- Use a semantic measure, Euclidean or Cosine distance to find the semantic similarity between any two words
- Segmented corpus of written or spoken text - language independent

A SAMPLE CO-OCCURRENCE MATRIX

	w_0	w_1	w_2	...	w_{n-3}	w_{n-2}	w_{n-1}	w_n
w_0	0	33	29	...	33	37	39	39
w_1	33	1	45	...	0	27	21	10
w_2	29	45	0	...	37	40	19	23
...	49	17	43	...	7	32	18	37
w_{n-3}	33	0	37	...	0	24	26	49
w_{n-2}	37	27	40	...	24	0	22	31
w_{n-1}	39	21	19	...	26	22	1	38
w_n	39	10	23	...	49	31	38	0

A word vector can be defined as

$$\vec{v} = \langle A(f(t, b_1)), A(f(t, b_2)), ..., A(f(t, b_n)) \rangle \quad (3)$$

where A is an association function. A is an identity matrix, if raw frequencies are used. t is the target word and b is the basis element[1]

[1] Sebastian Padó and Mirella Lapata. 2007. Dependency-Based Construction of Semantic Space Models. Computational Linguistics, 33(2):161-199.

SIMILARITY MEASURES

A similarity measure[2] is a real-valued function that quantifies the similarity between two objects - in this case words. Some of the similarity measures are given below.

$$\textbf{Euclidean Distance - } \mathbb{E}(\vec{w_1}, \vec{w_2}) = \sqrt{w_1^2 - w_2^2} \tag{4}$$

$$\textbf{Cosine Similarity} = \frac{\vec{w_1}.\vec{w_2}}{\|\vec{w_1}\| \, \|\vec{w_2}\|} \tag{5}$$

$$\textbf{Cosine distance} = 1 - \textbf{Cosine Similarity} \tag{6}$$

$$\textbf{Cluster similarity-} \mathscr{L}(\vec{w_1}, \vec{w_2}) = \frac{\vec{w_1}.\vec{w_2}}{\|\vec{w_1}\|} \tag{7}$$

[2] Lillian Lee. 1999. Measures of Distributional Similarity. In Proceedings of the 37^{th} Annual Meeting of the Association for Computational Linguistics, pages 25-32, College Park, Maryland, USA. Association for Computational Linguistics.

WORD VECTOR EXAMPLES

Similar words for apple

'apple', 0
'iphone', 0.266
'ipad', 0.287
'apples', 0.356
'blackberry', 0.361
'ipod', 0.365
'macbook', 0.383
'mac', 0.391
'android', 0.391
'google', 0.395
'microsoft', 0.418
'ios', 0.433
'iphones', 0.445
'touch', 0.446
'sony', 0.447

Similar words for - american

'american', 0
'america', 0.255
'americans', 0.312
'u.s.', 0.320
'british', 0.323
'canadian', 0.329
'history', 0.356
'national', 0.364
'african', 0.374
'society', 0.375
'states', 0.386
'european', 0.387
'world', 0.394
'nation', 0.399

VECTOR DIFFERENCE BETWEEN TWO WORDS

$$vec(apple) - vec(iphone)$$

```
('raisin', 0.5744591153088133)
('pecan', 0.5760617374141159)
('cranberry', 0.5840016172254104)
('butternut', 0.5882322018694753)
('cider', 0.5910795032086132)
('apricot', 0.6036644437522422)
('tomato', 0.6073715970323961)
('rosemary', 0.6150986936477657)
('rhubarb', 0.6157884153793192)
('feta', 0.6183016129045151)
('apples', 0.6226003361980218)
('avocado', 0.6235366677962004)
('fennel', 0.6306016018912576)
```

VECTOR ARITHMETIC ON WORD VECTORS...

840B words and 300 elements word vectors used for this computation

$\vec{apple}$	$\vec{apple} - \vec{iphone}$	$\vec{apple} - \vec{fruit}$
('apple', 0)	('apples', 0.39)	('ipad', 0.412)
('apples', 0.25)	('fruit', 0.43)	('iphone', 0.433)
('blackberry', 0.31)	('grape', 0.44)	('macbook', 0.435)
('Apple', 0.35)	('tomato', 0.44)	('ipod', 0.445))
('iphone', 0.37)	('pecan', 0.45)	('imac', 0.465
('fruit', 0.37)	('rhubarb', 0.45)	('3gs', 0.473)
('blueberry', 0.38)	('pears', 0.45)	('Ipad', 0.490)
('strawberry', 0.38)	('cranberry', 0.452)	('itouch', 0.512)
('ipad', 0.39)	('raisin', 0.453)	('ipad2', 0.514)
('pineapple', 0.39)	('apricot', 0.459)	('Iphone', 0.514)
('pear', 0.39)	('carrot', 0.461)	('ios', 0.520)
('cider', 0.39)	('candied', 0.462)	('Macbook', 0.524)
('mango', 0.40)	('blueberry', 0.463)	('ibook', 0.534)
('ipod', 0.40)	('apricots', 0.466)	('IPhone', 0.541)
('raspberry', 0.40)	('tomatoes', 0.466)	('32gb', 0.545)

Hyperspace Analogue To Language[3] (HAL)

[3] Lund, K., Burgess, C. Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments, & Computers 28, 203-208 (1996).

HAL ALGORITHM

Require a big corpus >5 GB for a reasonable similarity measures

1. Preprocess to limit the vocabulary size
2. Perform two scans using a ramping window of size 11 - first $\rightarrow$ direction and later in the $\leftarrow$ direction
3. Use the first word as the key word and the rest as context words
4. Use the last word as the key and rest as the context words, during the $\leftarrow$ scanning
5. The nearest neighbor of the key gets the weight 10 and the 10th word gets the weight 1
6. Construct an incidence matrix using the co-occurrence values
7. Concatenate two word vectors found for every word (row and column) in the matrix Concatenate them to get the word vectors for all the words in the vocabulary.
8. The number of elements in the word vecord will be $2\|V|$

HAL SCANNING

the horse raced past the barn fell

Left2Right Scanning

the	horse	raced	past	the	barn	fell
K	5	4	3	2	1	

	horse	raced	past	the	barn	fell
	K	5	4	3	2	1

Right2Left Scanning

fell	barn	the	past	raced	horse	the
K	5	4	3	2	1	0

	barn	the	past	raced	horse	the
	K	5	4	3	2	1

Incidence Matrix

	the	horse	raced	past	barn	fell
the	1	3			5	
horse	5				1	
raced	4	5			3	
past	3	4			4	
barn	2	2				
fell	0+4	1+1	3	4	5	

HAL EXPERIMENT

160 million words from Usenet news groups

Window size $= 10$

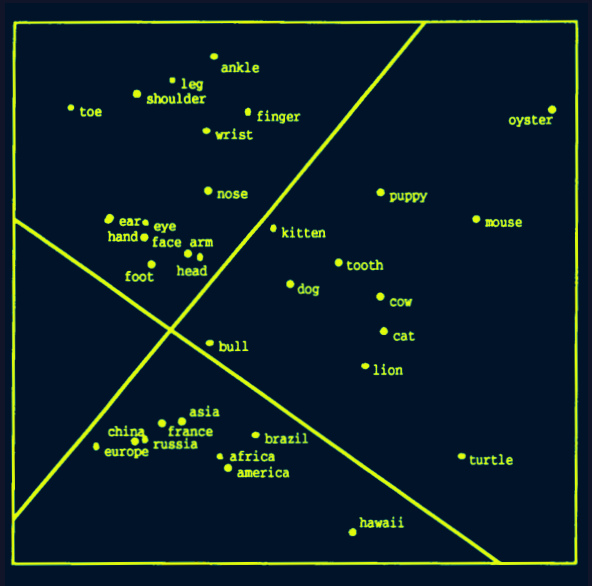
- Vocabulary - Words with a frequency > 50
- Zipf's law is used to eliminate most common and rare words
- Two word vectors are obtained for every word
- Minkowski distance measure is used for computing word similarities

$$d_{x_i y_j} = \sqrt[r]{|x_i - y_i|^r}$$

- The word vectors produce high dimensional semantic space - associative

Target	n1	n2	n3	n4	n5
jugs	juice	butter	vinegar	bottles	cans
leningrad	rome	iran	dresdan	azerbaijan	tibet
lipstick	lace	pink	cream	purple	soft
triumph	beauty	prime	grand	former	rolling
cardboard	plastic	rubber	glass	thin	tiny
monopoly	threat	huge	moral	gun	large

HAL WORD VECTORS - SIMILARITY CHART



Correlated Occurrence Analogue to Lexical Semantic[4] (COALS)

[4] Rhode et al, "An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence", CACM, 2006, 8, 627-633

1. Gather co-occurrence counts, typically ignoring closed-class neighbors and using a ramped, size 4 window:

   1 2 3 4 [0] 4 3 2 1

2. Discard all but the m (14,000, in this case) columns reflecting the most common open-class words.
3. Convert counts to word pair correlations, set negative values to 0, and take square roots of positive ones.
4. The semantic similarity between two words is given by the correlation of their vectors.

COALS RESULTS

Nearest neighbours and their percent correlation similarities for a set of nouns

	gun	point	mind	monopoly
1)	46.4 handgun	32.4 points	33.5 minds	39.9 monopolies
2)	41.1 firearms	29.2 argument	24.9 consciousness	27.8 monopolistic
3)	41.0 firearm	25.4 question	23.2 thoughts	26.5 corporations
4)	35.3 handguns	22.3 arguments	22.4 senses	25.0 government
5)	35.0 guns	21.5 idea	22.2 subconscious	23.2 ownership
6)	32.7 pistol	20.1 assertion	20.8 thinking	22.2 property
7)	26.3 weapon	19.5 premise	20.6 perception	22.2 capitalism
8)	24.4 rifles	19.3 moot	20.4 emotions	21.8 capitalist
9)	24.2 shotgun	18.9 distinction	20.1 brain	21.6 authority
10)	23.6 weapons	18.7 statement	19.9 psyche	21.3 subsidies

SOME INSIGHTS1/2

- The majority of the correlations are negative
- Words with negative correlations do not contribute well to finding similarity than the ones with positive correlation
- Closed-class words (147) convey syntactic information than semantic - could be removed from the correlation table punctuation marks, she, he, where, after, ...

SOME INSIGHTS 2/2

- **Positive Correlation** means that two words often appear together. In other words, their contexts are similar.
  - CMI and Prodigy have a strong correlation
- **Zero correlation** means the pair of words are statistically independent. Hence no influence
  - CMI and Engineering_Drawing/blueprint have no inherent or direct relationship in terms or context.
- **Negative correlation** indicates an inverse relationship. For every word pair in this set, the second word in each pair is less likely to appear, and vice versa.
  - When talking about one of the specializations of CMI as the first word in a pair, some words, such as art, literature, philosophy, and religion, are less likely to appear together (although they may appear a few times).

DENSE VECTORS

- Not sparse
- Shorter than sparse word vectors
- Real-valued and continuous
- Captures fine-grained semantic information
- Can be used to represent the entire sentences or paragraphs
- Word2Vec, GloVe, and FastText use dense embeddings
- Typical sizes are 50, 100, or 300 elements

SINGULAR VALUE DECOMPOSITION

Singular value decomposition is a method to factorize a rectangular/square matrix into three matrices.

$$A = U\Sigma V^T \tag{8}$$

where A is an MXN matrix

- U is the $M \times K$ matrix
- Σ is a diagonal matrix of size $K \times K$
- V^T is the $K \times N$ matrix
- The row vectors of U are called as the left-singular vectors
- Row vectors of U form an orthogonal set

- The columns of V^T are called as the right singular matrix
- The rows of V^T form an orthonormal set
- The Σ is the singular matrix. It is a diagonal matrix and i1ts values are arranged in the descending order.

SVD

SINGULAR VALUES

- It is a diagonal matrix
- Singular values are arranged in the descending order
- Highest order dimension captures the most variance in the original dataset or most of the information related to term-document matrix
- The next higher dimension captures the next higher variance in the original data set
- Singular values reflect the major associative patterns in the data, and ignore the smaller, less important influences

SUMMARY OF SVD

- Find a new set of dimensions or attributes that capture the variability of the data
- Identify the strongest pattern in the data
- Most variability is captured by a small fraction of the total set of dimensions
- Patterns among the terms are captured by the left-singular matrix
- Patterns among the documents are captured by the right-singular matrix
- The eigen vectors associated with the largest eigen value indicates the direction of largest variance[5]

[5] Pang-Ning Tan et al, "Introduction to Data Mining"2007

SVD DEMO

NEED FOR BETTER MODELS

- HAL, COALS and their derivatives employ $\vec{v} = \langle A(f(t, b_1)), A(f(t, b_2)), ..., A(f(t, b_n)) \rangle$, where A is an association function
- Is it possible to generalize and learn semantic relationships and analogies?
- Understanding of the structure of sentences is not enough to get the semantics of a sentence

SUMMARY OF WORD EMBEDDINGS

- Maps words to vectors of real numbers
- Learned from a corpus of text
- Capture the semantic meaning of words
- Used as input to many downstream applications, such as text classification, sentiment analysis, and machine translation, context embedding