Word Representation

Ramaseshan Ramachandran

1. Context
2. Semantic Space
3. Representation of words
4. Empirical Laws
5. Exercises
6. Vector Representation of Words
7. Semantically connected Word Vectors
8. Word Vectors
9. Similarity
10. References

WHAT IS CONTEXT?

The main challenge in understanding how words relate to real-world situations has increasingly become a matter of 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 - THE INFLUENCER

Phrase	Meaning
Spill the tea	Share the gossip or details
On the same page	Everyone understanding or agreeing on something.
I'm so over it	Tired of something
That's so cool	Approval or exclamation
It's raining cats and dogs	It's raining heavily

- 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 between 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 OR CONCEPT SPACE

Hyponyms and hypernyms are linguistic terms that describe the relationship between words based on their meaning.

- A space where the similar words (synonyms, hyponyms, hypernyms) are classified and arranged in various axes
  - Colour (hypernym) - $\underbrace{red, Green, Orange}_{co-hyponyms}$ (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

Side-effect: Lexical co-occurrences associate semantically dis-similar words

LAWS OF ASSOCIATION

- **The law of similarity**: If two things are similar, the thought of one will tend to trigger the thought of the other
- **The law of frequency**: The more often two things or events are linked, the more powerful will be that association.
- **The law of contiguity**: Things or events that occur close to each other in space or time tend to get linked together in the mind.
- **The law of contrast**: Seeing or recalling something may also trigger the recollection of something completely opposite.

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

Context is essential in the mapping the relationship between words and concepts.

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 research corpus, one may not find the phrase $\underline{\textit{ needle in a haystack}}$
- You will find the word needle will be lexically associated with $\underbrace{pain, illness, blood, drugs, syringe}_{Associative\ relationship}$, but not to thread, knitting, cloth, etc.

WORD SIMILARITY

You shall know a word by the company it keeps

- Firth, 1957

WORDS AND TERMS

The atomic unit is a ***word***

- The atomic unit for constructing a word in a language is its alphabet
- We use ***Term*** (co-located/co-occurring words) and ***word*** as atomic.
- It is necessary to consider the numerical representation of the word for computational purposes
- Vocabulary of size $N = 1 \ldots n$ defined as $V = w_1, w_2, w_3, \ldots, w_n$ is the vocabulary containing unique words of a corpus
- Some words found in V appear in documents $(D = D_1, D_2 D_3, \ldots, D_m)$, once or several times or may not appear at all.

TERM FREQUENCY

Term Frequency

Term frequency is defined as the number of times the term, t_i, occurs in a document d_j, belonging to a corpus $(d_1, d_2, d_3, \ldots, d_m)$. This is denoted by $tf_{t,d}$

TERM FREQUENCY - DEMO

```
import nltk
from nltk.probability import  FreqDist
from nltk.corpus import stopwords
import pandas as pd
stop_words = set(stopwords.words(’english’))
#read the corpus
words = nltk.Text(nltk.corpus.gutenberg.words(’bryant-stories.txt’))
#convert to small letters
words=[word.lower() for word in words if word.isalpha() ]
words=[word.lower() for word in words if word not in stop_words ]
#Get the frequency distribution of words
freq_dist = FreqDist(words)
#Heading for the results table
heading = [’Word’,’Frequency’]
tf_list = []
for x,v in freq_dist.most_common(10):
    tf_list.append((x,v))
print(pd.DataFrame(tf_list,columns=heading))
#Weighted term freequency
heading = [’Word’,’Weighted Frequency’];tf_list = []
for x,v in freq_dist.most_common(10):
    tf_list.append((x,v/len(freq_dist)))
print(pd.DataFrame(tf_list,columns=heading))
```

RAW AND ADJUSTED TERM FREQUENCY

word	Raw Frequency	Adjusted Frequency*
little	597	0.169
said	453	0.126
came	191	0.052
one	183	0.049
could	158	0.047
king	141	0.039

*The term frequency is adjusted to the document length

MULTIPLE WEIGHTING FACTORS TF

$$\mathrm{Boolean} - 0, 1 \tag{1}$$

$$\mathrm{RawCount} - \mathrm{tf}_{i,d} \tag{2}$$

$$\text{Adjusted to document length} - \frac{\mathrm{tf}_{i,d}}{M} \tag{3}$$

$$\text{Log weighting} - \begin{cases} f_{t,d} - 1 + \log \mathrm{tf}_{i,d} & \text{if } \mathrm{tf}_{i,d} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

DISADVANTAGES OF RAW FREQUENCY

- All terms are given equal importance
- The Common term ***the*** has no relevance to the document, but gets high relevancy score
- May not be suitable for classification when common words appear in documents

INVERSE DOCUMENT FREQUENCY

In order to attenuate the effect of frequently occurring terms, it is important to scale it down and at the same time it is necessary to increase the weight of terms that occur rarely.

Inverse document frequency (IDF) is defined as

$$\mathrm{IDF}_t = \log\left(\frac{N}{D_{f_t}}\right) \tag{5}$$

where N is the total number of documents in a collection, and D_{f_t} is the count of documents containing the term t

- Rare documents gets a significantly higher value
- Commonly occurring terms are attenuated
- It is a measure of informativeness
- Reduce the tf weight of a term by a factor that grows with its collection frequency.
- If a term appears in all the documents, then IDF is zero. This implies that the term is not important

TF-IDF

Composition of TF and IDF produces a composite scaling for each term in the document

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_{t,d} \tag{6}$$

- The value is high when t occurs many times within a few documents
- The value is very low when a term appears in all documents

INVERSE DOCUMENT FREQUENCY-IDF

IDF is the inverse frequency of the word 't' appearing in the corpus. It is computed as

$$\text{IDF of a term t} = \log_{10}\left(\frac{\text{Total number of documents in a corpus}}{\text{Count of documents with term t}}\right)$$

IDF is the measure of ***informativeness***

Example:

Consider a corpus with 100K documents. The word ***moon*** occurs in some documents (say, 100) with the following frequency:
$TF_{d_1} = \frac{20}{427}, TF_{d_2} = \frac{30}{250}, TF_{d_3} = \frac{20}{250}, TF_{d_9} = \frac{5}{125}$ and $TF_{d_{1000}} = \frac{20}{1000}$
The total number of words in the corpus = 100000

$$\therefore \mathrm{IDF}_{d_1} = \log_{10}\left(\frac{100000}{100}\right)$$

$$TF_{d_1} * \mathrm{IDF} = 0.141$$

If the word ***Andromeda*** appears only once d_1, then $TF_{d_1} * \mathrm{IDF} = 0.0117$. If the word ***the*** appeared in every document and 45 times in d1, then $TF * \mathrm{IDF} = 0.210$

DOCUMENT RANKING USING TF-IDF

Using the TF-IDF, the rank order for the documents can be determined for the documents for the term *moon*.

Document Name	tf-idf	Rank
d1	0.14	3
d2	0.36	1
d3	0.24	2
d9	0.12	4
d1000	0.06	5

BAG OF WORDS

The collection $[\mathrm{tf}_1, \mathrm{tf}_2, \mathrm{tf}_5, \mathrm{tf}_{15}, \ldots, \ldots, \mathrm{tf}_n]$ is known as *bag of words*

- The ordering of the terms is not important
- Two documents with similar bag of words are similar in content
- It refers to the quantitative representation of the document

ZIPF'S LAW

Zipf's law states that for a given some corpus, the frequency of any word is inversely proportional to its rank in the term frequency table

$$f(r) \propto \frac{1}{r^{\alpha}} \tag{7}$$

where $\alpha \approx 1$, r is the *frequency rank* of a word and $f(r)$ is the frequency in the corpus. The most frequent word will have the value 1, the word ranked second in the frequency will have $\frac{1}{2^{\alpha}}$, the word ranked third in the frequency will have $\frac{1}{3^{\alpha}}$, etc

Distribution of terms/words

This empirical law models the frequency distribution of words in languages. This distribution is observed across several languages when a large corpus is used.

MANDELBROT APPROXIMATION

Mandelbrot derived a more generalized law to closely fit the frequency distribution in language by adding an offset to the rank

$$f(r) \propto \frac{1}{(r+\beta)^{\alpha}} \tag{8}$$

where $\alpha \approx 1$ and $\beta \approx 2.7$

HEAPS' LAW

This is used to estimate the number of unique terms M in a corpus given the total number of tokens

$$\begin{aligned} M &\propto T^b \\ &= kT^b \end{aligned} \tag{9}$$

where $30 \leq k \leq 100$ and $b \approx 0.49$
According to this empirical law, the dictionary or the vocabulary size increases linearly with the total number of tokens/words in the corpus.

APPLICATIONS OF TTR

- Monitor the vocabulary usage
- Monitor child vocabulary development
- Estimate the vocabulary variation in the text

EXERCISES

- Write a program to find out whether Mandelbrot's approximation provides a better fit than Zipf's law. Use the same corpus for Zipf and Mandelbrot approximation.
- Find out the percentage of tokens of the stop words.
- Find out the top to words by frequency
- Write a program for Heap's law and predict the vocabulary size in any corpus. Also, find out whether it is closer to the actual size of the vocabulary of the same corpus.

OPEN-CLASS AND STOP WORDS

Open-class words: Carry the main semantic content of a sentence

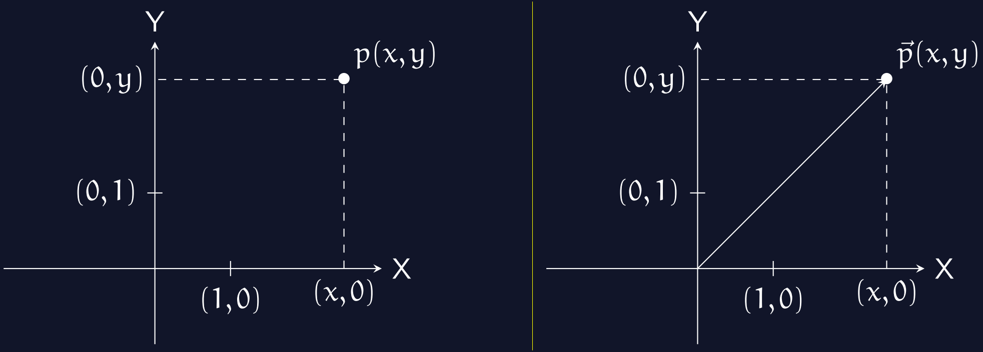
- Contribute to the overall meaning of a sentence
- Include nouns, verbs, adjectives, and adverbs
- **Expansive**: New words can be added to this category over time (e.g., "vlog," "emoji", “Mulligatawny”)
- **Adaptability**: New terms to express emerging concepts, technologies, and innovation

Stop words: Serve grammatical purposes

- Include
  - prepositions (in, on, with, ...)
  - conjunctions (and, but, or..)
  - articles (a, an, the)
  - pronouns(he, she, they...)
- Essential for the grammatical integrity of sentences
- Contribute less to the overall meaning of the sentence

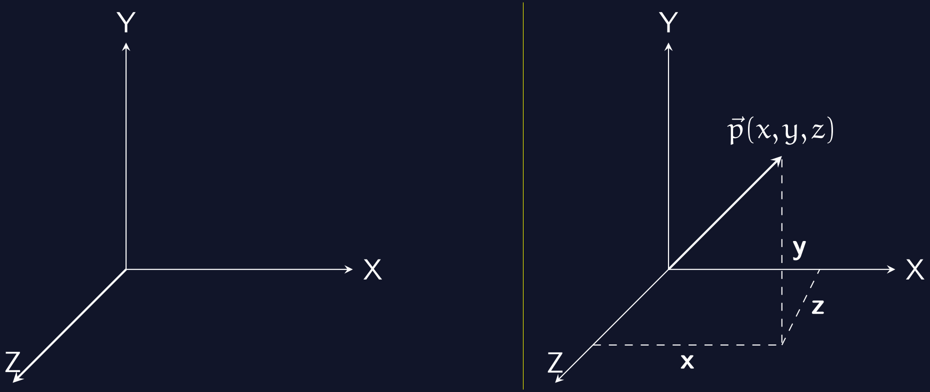
2-D VECTOR SPACE

A 2-D vector-space is defined as a set of linearly independent basis vectors with 2 axes. Each axis corresponds to a dimension in the vector-space



3-D VECTOR SPACE

A 3-D vector-space is defined as a set of linearly independent basis vectors with 3 axes. Each axis corresponds to a dimension in the vector-space



Linearly independent vectors of size $\mathscr{N}$ will result in $\mathscr{N}$-dimensional axes which are mutually orthogonal to each other

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.

If a corpus contains $|\mathbb{V}|$ words which are linearly independent, then every word represents an axis in the continuous vector space $\mathbb{R}$.

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.
If a corpus contains $|\mathbb{V}|$ words which are linearly independent, then every word represents an axis in the continuous vector space $\mathbb{R}$.
Each word takes an independent axis which is orthogonal to other words/axes.

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.
If a corpus contains $|\mathbb{V}|$ words which are linearly independent, then every word represents an axis in the continuous vector space $\mathbb{R}$.
Each word takes an independent axis which is orthogonal to other words/axes.
Then $\mathbb{R}$ will contain $|\mathbb{V}|$ axes.

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.
If a corpus contains $|\mathbb{V}|$ words which are linearly independent, then every word represents an axis in the continuous vector space $\mathbb{R}$.
Each word takes an independent axis which is orthogonal to other words/axes.
Then $\mathbb{R}$ will contain $|\mathbb{V}|$ axes.

Examples

1. The vocabulary size of *emma corpus* is 7079. If we plot all the words in the real space $\mathscr{R}$, we get 7079 axes
2. The vocabulary size of *Google News Corpus corpus* is 3 million. If we plot all the words in the real space $\mathscr{R}$, we get 3 million axes

DOCUMENT VECTOR SPACE MODEL

- Vector space models are used to represent words in a continuous vector space $\mathscr{R}$

DOCUMENT VECTOR SPACE MODEL

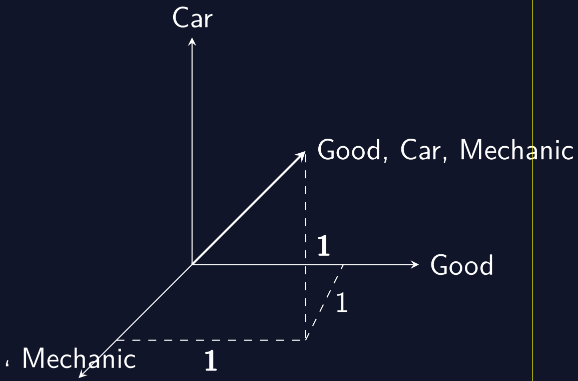
- Vector space models are used to represent words in a continuous vector space $\mathscr{R}$
- Combination of Terms represent a document vector in the word vector space

DOCUMENT VECTOR SPACE MODEL

- Vector space models are used to represent words in a continuous vector space $\mathscr{R}$
- Combination of Terms represent a document vector in the word vector space
- Very high dimensional space - several million axes, representing terms and several million documents containing several terms

EXAMPLE

Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	Good	Car	Mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

DOCUMENT-TERM MATRIX

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12
t1	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t2	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t3	0.0	0.9	0.0	0.2	0.3	0.1	0.7	0.0	0.2	0.7	0.5	0.5
t4	0.0	0.9	0.3	0.9	0.5	0.1	0.9	0.3	0.8	0.4	0.1	0.4
t5	0.4	0.0	0.3	0.2	0.5	0.9	0.3	0.7	0.4	0.6	0.0	0.3
t6	0.6	0.0	0.4	0.7	0.3	0.3	0.9	0.1	0.9	0.0	0.0	0.3
t7	0.0	0.8	0.5	0.6	0.6	0.6	0.0	0.1	0.4	0.9	0.3	0.1
t8	0.4	0.0	0.6	0.5	0.5	0.1	0.7	0.1	0.5	0.3	0.8	0.1
t9	0.3	0.0	0.7	0.9	0.8	0.7	0.7	0.8	0.6	0.6	0.8	0.0
t10	0.0	0.5	0.5	0.0	0.2	0.0	0.0	0.1	0.3	0.4	0.5	0.3

The columns of the matrix represent the document as vectors. A document vector is represented by the terms present in the document

TERM-TERM MATRIX

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
t1	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t2	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t3	0.0	0.9	0.0	0.2	0.3	0.1	0.7	0.0	0.2	0.7	0.5	0.5
t4	0.0	0.9	0.3	0.9	0.5	0.1	0.9	0.3	0.8	0.4	0.1	0.4
t5	0.4	0.0	0.3	0.2	0.5	0.9	0.3	0.7	0.4	0.6	0.0	0.3
t6	0.6	0.0	0.4	0.7	0.3	0.3	0.9	0.1	0.9	0.0	0.0	0.3
t7	0.0	0.8	0.5	0.6	0.6	0.6	0.0	0.1	0.4	0.9	0.3	0.1
t8	0.4	0.0	0.6	0.5	0.5	0.1	0.7	0.1	0.5	0.3	0.8	0.1
t9	0.3	0.0	0.7	0.9	0.8	0.7	0.7	0.8	0.6	0.6	0.8	0.0
t10	0.0	0.5	0.5	0.0	0.2	0.0	0.0	0.1	0.3	0.4	0.5	0.3
t11	0.01	0.2	0.4	0.1	0.2	0.2	0.0	0.0	0.0	0.1	0.2	0.0
t12	0.1	0.12	0.54	0.01	0.02	0.0	0.0	0.0	0.0	0.6	0.7	0.0

The columns and rows of the matrix represent the words as vectors.

Do they represent the contextual relationship among words?

WORD SIMILARITY

A similarity measure is a real-valued function that quantifies the similarity between two objects - in this case words [1]. Some of the similarity measures are given below.

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

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

$$\textbf{Cosine distance} = 1 - \frac{\vec{w_1}.\vec{w_2}}{\|\vec{w_1}\|\,\|\vec{w_2}\|} = \frac{\vec{w_1}}{\|\vec{w_1}\|}.\frac{\vec{w_2}}{\|\vec{w_2}\|}$$

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

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 $\mathscr{R}^{|\mathrm{V}|\mathrm{x}1}$ 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 $\mathscr{R}^{|V|X1}$

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

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

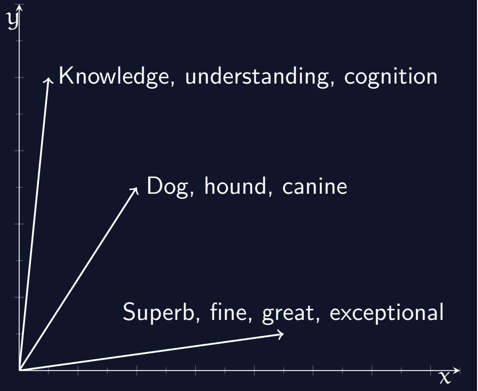
$$\left(t^{\mathrm{House}}\right)^{\mathsf{T}} \cdot t^{\mathrm{Apartment}} = 0 \tag{10}$$

$$\left(t^{\mathrm{Home}}\right)^{\mathsf{T}} \cdot t^{\mathrm{House}} = 0 \tag{11}$$

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

RELATIONSHIP AMONG TERMS - SYNONYMS

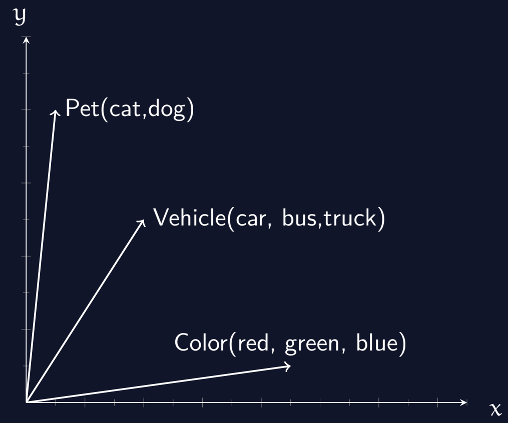
We could represent all the synonyms of a word in one axis


y
Knowledge, understanding, cognition
Dog, hound, canine
Superb, fine, great, exceptional
x

- Can we assume that similar words be situated/placed in the respective concept space?
- Can we assume that the properties of the word that inherit the properties of that space?

RELATIONSHIP AMONG TERMS - IS-A VECTOR

We could represent inheritance relationships of words as vectors.



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

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

How do we place theses polysemous word in an axis? Using multiple vectors?

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 *é*tait
Mtazamo wa juu wa mlima huo ulikuwa

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

AJI AMARILLO

- Heard of Aji Amarillo?

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale
- It is not as hot as bhüt jolokia which measures 1 million on the same scale

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale
- It is not as hot as bhüt jolokia which measures 1 million on the same scale
- If you visit Peru, do not miss dishes made from this

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale
- It is not as hot as bhüt jolokia which measures 1 million on the same scale
- If you visit Peru, do not miss dishes made from this
- They are yellow in color and have a balanced, fruity flavor close to mango and even passion fruit

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale
- It is not as hot as bhüt jolokia which measures 1 million on the same scale
- If you visit Peru, do not miss dishes made from this
- They are yellow in color and have a balanced, fruity flavor close to mango and even passion fruit
- It is a member of the capsicum baccatum

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale
- It is not as hot as bhüt jolokia which measures 1 million on the same scale
- If you visit Peru, do not miss dishes made from this
- They are yellow in color and have a balanced, fruity flavor close to mango and even passion fruit
- It is a member of the capsicum baccatum
- Star ingredient in many Peru's dishes

AJI AMARILLO

- Heard of Aji Amarillo?
- It measures 30K/50K on the Scoville Heat scale
- It is not as hot as bhüt jolokia which measures 1 million on the same scale
- If you visit Peru, do not miss dishes made from this
- They are yellow in color and have a balanced, fruity flavor close to mango and even passion fruit
- It is a member of the capsicum baccatum
- Star ingredient in many Peru's dishes

Intuition

Aji Amarillo is a hot food item
Aji Amarillo is grown in Peru

Aji Amarillo is a member of pepper family

SEMANTICALLY CONNECTED VECTORS

- Identify a model that enumerates the relationships between terms
- Identify a model that tries to place 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
- Similar to the term-document space identify a semantic space for similar 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

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

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 to extract similarities - 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

A SAMPLE CO-OCCURRENCE MATRIX WITH RAW FREQUENCY COUNT

	w_0	w_1	w_2	$\ldots$	w_{n-3}	w_{n-2}	w_{n-1}	w_n
w_0	0	33	29	$\ldots$	33	37	39	39
w_1	33	1	45	$\ldots$	0	27	21	10
w_2	29	45	0	$\ldots$	37	40	19	23
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
w_{n-3}	33	0	37	$\ldots$	0	24	26	49
w_{n-2}	37	27	40	$\ldots$	24	0	22	31
w_{n-1}	39	21	19	$\ldots$	26	22	1	38
w_n	39	10	23	$\ldots$	49	31	38	0

- This matrix captures the lexical space of the corpus
- Statistical knowledge about words and their relationship in terms of co-occurrence are static in nature

TECHNIQUES TO CAPTURE CO-OCCURRENCE INFORMATION

Let A denote the word-word co-occurrence matrix where each row corresponds to a unique/target word, and each column represents a context.
a_{ij} denotes every element of the A

a_i denotes the number of times the word i co-occurring with the word j, $a_i = \sum_j a_{ij}$

Now, we can fill the co-occurrence using:

1. **Raw Frequency Count:** Each element, a_{ij} denotes the raw frequency count of word i co-occurring with the word j
2. **Probability:** $p_{ij} = P(w_j \mid w_i) = \dfrac{a_{ij}}{a_i}$
3. **Positive Point-wise Mutual Information(PMI)**:
$$\mathrm{PPMI}(w_i, w_j) = \max\left(\log_2\left(\frac{p_{ij}}{p_i * p_j}\right), 0\right)$$

The co-occurrence statistics are captured by scaning the entire corpus once using a windowing approach

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.

$$\textbf{Euclidean Distance - } \mathbb{E}(\vec{w}_1, \vec{w}_2) = \sqrt{w_1^2 - w_2^2} \tag{12}$$

$$\textbf{Cosine Similarity} = \frac{\vec{w}_1 . \vec{w}_2}{\|\vec{w}_1\| \, \|\vec{w}_2\|} \tag{13}$$

$$\textbf{Cosine distance} = 1 - \textbf{Cosine Similarity} \tag{14}$$

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
'us', 0.399
```

VECTOR DIFFERENCE BETWEEN TWO WORDS

$$\overrightarrow{apple} - \overrightarrow{iphone}$$

```
('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)
('chutney', 0.6312524337590703)
('spiced', 0.6327632200841328)
```

VECTOR ARITHMETIC ON WORD VECTORS...

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

$\overrightarrow{apple}$	$\overrightarrow{apple} - \overrightarrow{iphone}$	$\overrightarrow{apple} - \overrightarrow{fruit}$
('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)

REFERENCES

[1] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *An Introduction to Information Retrieval*. Cambridge UP, 2009. Chap. 6, pp. 109–133.

[2] Lillian Lee. “Measures of Distributional Similarity”. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. College Park, Maryland, USA: Association for Computational Linguistics, June 1999, pp. 25–32. DOI: 10.3115/1034678.1034693. URL: https://aclanthology.org/P99-1004.