Natural Language Processing

WORKSHOP

24-28 August, 2022

TOPICS TO BE COVERED IN THIS WORKSHOP

1. Introduction
   - GitHub Page
   - Goal of NLP
2. Word2Vector
   - 2-D Vector Space
   - 3-D Vector Space
   - Vector Space Model for Words and Documents
   - Document Vector Space Model
   - Document-Term Matrix
   - Document Similarity
   - Demo - Cosine Similarity
   - Word Vector
   - One-Hot Vector
   - One-Hot- Vector - example
   - Relationship among terms
   - Is-A Vector
   - Information Extraction
3. Context
4. Co-occurrences
   - Contextual Understanding of Text
   - Co-occurrence Matrix
   - Unigram, Bigrams and Trigrams
   - N-grams
   - Collocations
5. Word Similarity

GITHUB PAGE

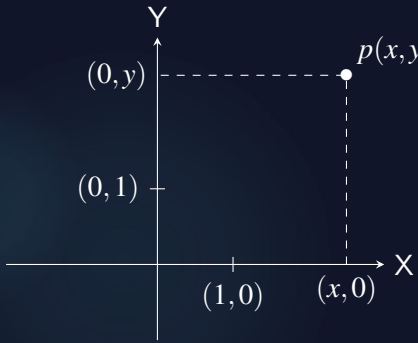
The python Notebook is available at https://github.com/Ramaseshanr/IITMDS

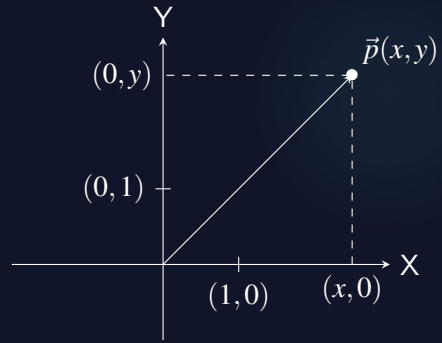
GOAL

Ability to process and harness information from a large corpus of text with no human intervention

2-D VECTOR SPACE

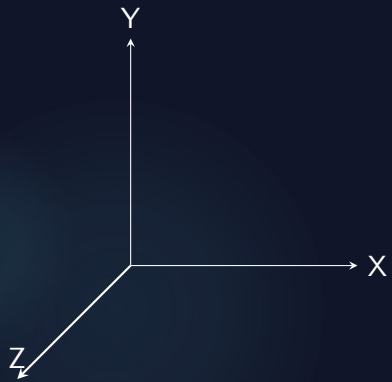
A 2-D vector-space is defined as a set of linearly independent basis vectors with 2 axes. Each axis corresponds to a dimension in the vector-space

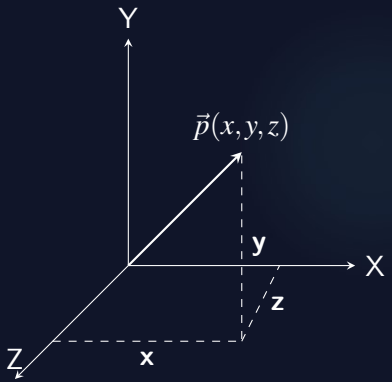




3-D VECTOR SPACE

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 $\mathcal{N}$ will result in $\mathcal{N}$-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.
If a corpus contains $|\mathscr{V}|$ words which are linearly independent, then every word represents an axis in the continuous vector space $\mathscr{R}$.
Each word takes an independent axis which is orthogonal to other words/axes.
Then $\mathscr{R}$ will contain $|\mathscr{V}|$ axes.

Examples

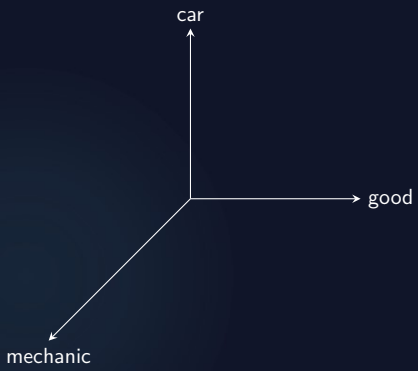
1. The vocabulary size of *emma corpus* is 7079. If we plot all the words in the real space $\mathscr{R}$, we get 7079 axes
2. The vocabulary size of *Google News Corpus corpus* is 3 million. If we plot all the words in the real space $\mathscr{R}$, we get 3 million axes

DOCUMENT VECTOR SPACE MODEL

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

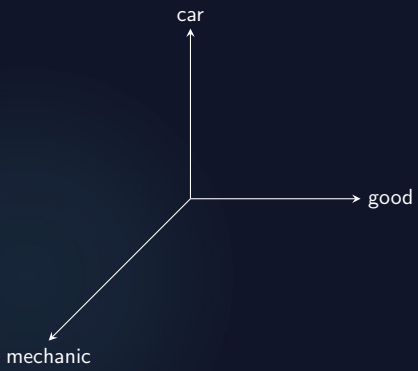
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

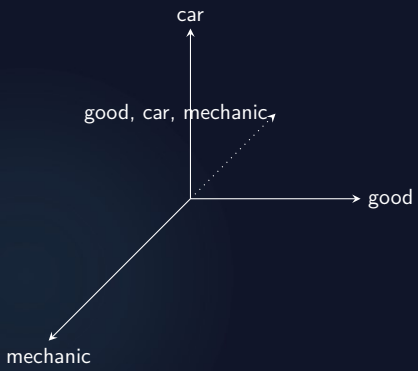
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

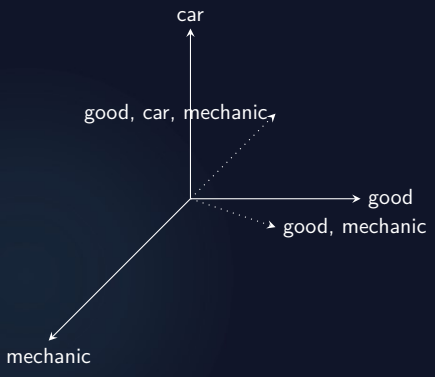
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

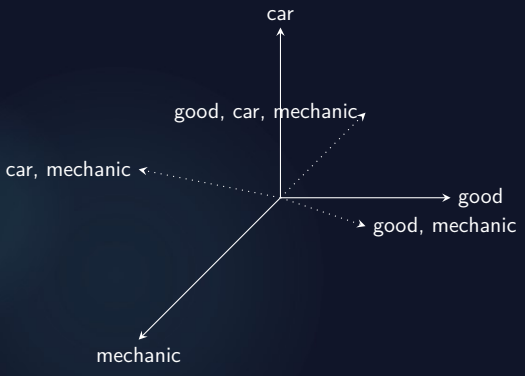
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

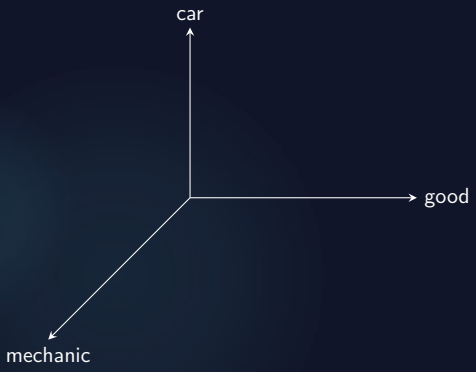
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - TF-IDF INCIDENCE MATRIX

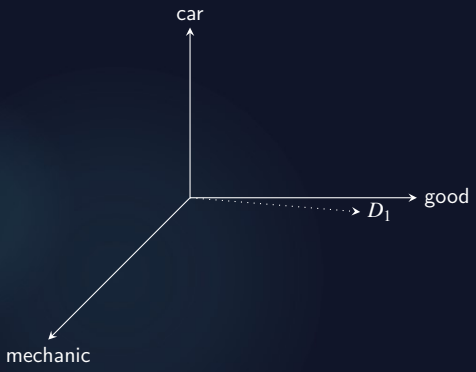
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

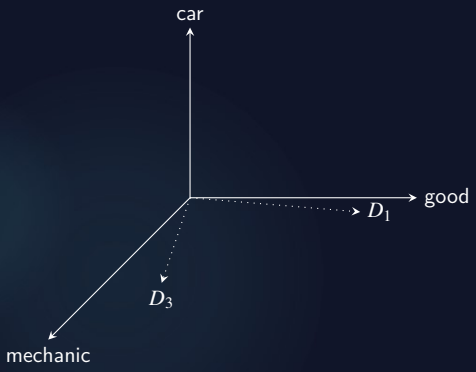
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

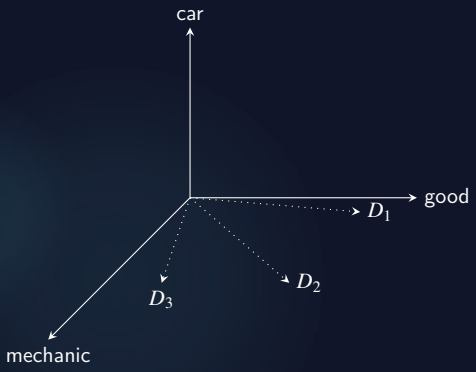
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

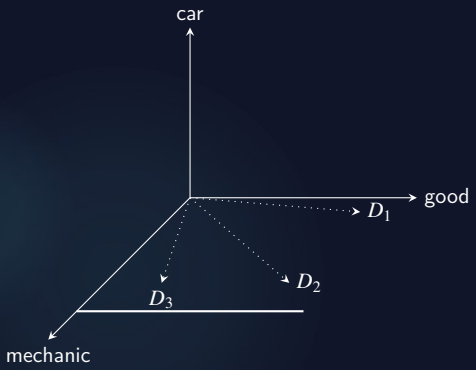
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

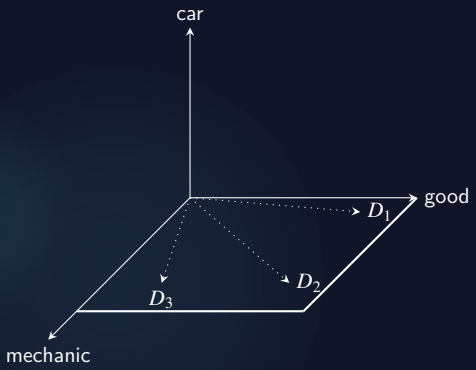
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

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



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

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

Every element in the matrix represent tf-idf either in the plain form or in some of the weighted forms as given below:

$$tf.idf = tf \times log_{10}\left(\frac{N}{df_t}\right) \textbf{ or}$$

$$= w_{t,d} \times \left(\frac{N}{df_t}\right)$$

$$\text{where } w_{t,d} = \begin{cases} (1 + log_{10} tf_t), & \text{if } tf_{t,d} > 0 \\ 0 & otherwise \end{cases}$$

DOCUMENT SIMILARITY

Earlier, using the binary incidence matrix, a query returned a set of documents whether the query keywords were found in documents or absent. It did not give any ranking for the retrieved documents. A similarity measure is a real-valued function that quantifies the similarity between two objects. Some of the methods are given below.

Euclidean Distance - $\mathscr{E}(\vec{d}_1, \vec{d}_2) = \sqrt{d_1^2 - d_2^2}$

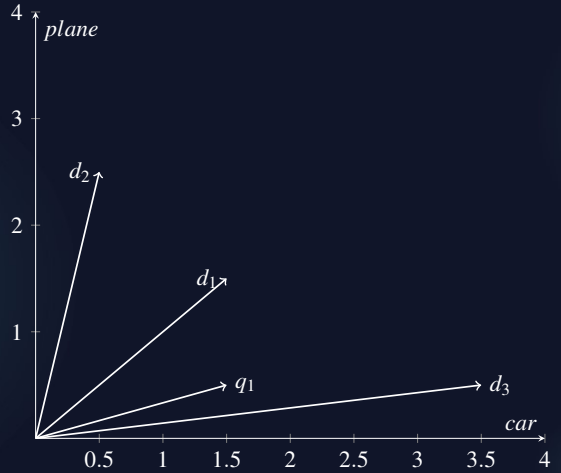
$$\textbf{Cosine Similarity} = \frac{\vec{d}_1 . \vec{d}_2}{\left\|\vec{d}_1\right\| \left\|\vec{d}_2\right\|} = \frac{\vec{d}_1}{\left\|\vec{d}_1\right\|} . \frac{\vec{d}_2}{\left\|\vec{d}_2\right\|}$$

$$\text{Cosine distance} = 1 - \frac{\vec{d}_1 . \vec{d}_2}{\left\|\vec{d}_1\right\| \left\|\vec{d}_2\right\|} = 1 - \frac{\vec{d}_1}{\left\|\vec{d}_1\right\|} . \frac{\vec{d}_2}{\left\|\vec{d}_2\right\|}$$

$$\textbf{Cluster similarity-}\mathscr{L}(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 . \vec{d}_2}{\left\|\vec{d}_1\right\|_1}$$

WHICH MEASURE?

Euclidean measure does not work well for unequal sized vectors as the vectors are not normalized. We often use normalized correlation coefficient or cosine distance for similarity measure



DEMO - COSINE SIMILARITY

Vector Representation of Words

VECTOR REPRESENTATION OF WORDS

Let V be the unique 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

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

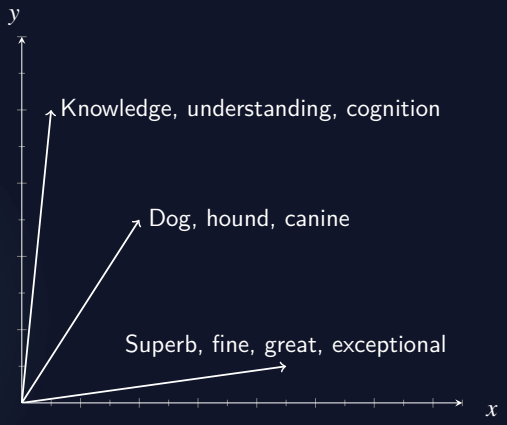
With one-Hot vector, there is no notion of similarity or synonyms.

The Goal of Word to Vector

- Reduce word-vector space into a smaller sub-space
- Encode the relationship among words

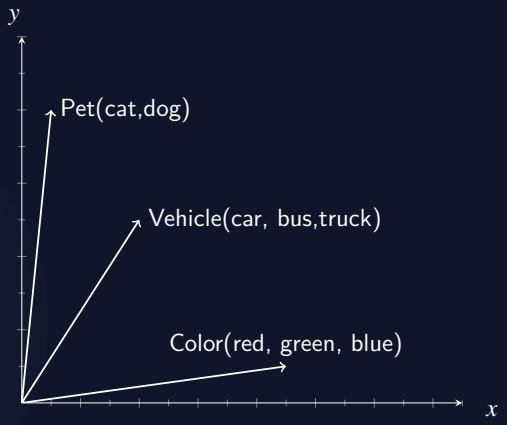
RELATIONSHIP AMONG TERMS - SYNONYMS

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

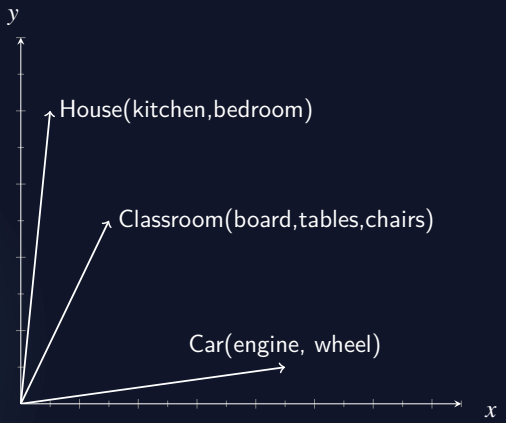


RELATIONSHIP AMONG TERMS - IS-A VECTOR

We could represent inheritance relationships of words as vectors.



RELATIONSHIP AMONG TERMS - HAS-A VECTOR - COMPOSITIONS



IS-A VECTOR

	Color	Animal	Fruit	Company Name
Apple	0	0	10	1850
Banana	0	0	165	0
Blackberry	0	0	156	190
Elephant	0	87	0	0
Fox	0	76	0	1
Goat	0	57	0	0
Green	145	0	0	0
Orange	454	0	213	134
Raspberry	0	0	197	74
Red	650	0	0	0
Sheep	0	132	0	0
Yellow	345	0	0	0

INFORMATION EXTRACTION USING IS-A RELATIONSHIP

A simple example of Named Entity Extraction

The Apple Watch has a completely new user interface, different from the iPhone, and the 'crown' on the Apple Watch is a dial called the 'digital crown.' A key quality attribute of apple is its peel or skin color, which affects consumer preferences. Immature fruits are green, and as the fruit ripens the green may fade partially or completely, resulting in very pale cream to green background colors.

The ⟨**org**⟩**Apple** Watch has a completely new user interface, different from the iPhone, and the 'crown' on the ⟨**org**⟩**Apple** Watch is a dial called the 'digital crown.' A key quality attribute of ⟨**org**⟩**apple** is its peel or skin color, which affects consumer preferences. Immature fruits are green, and as the fruit ripens the green may fade partially or completely, resulting in very pale cream to green background colors.

You shall know a word by the company it keeps[1]

[1] Firth, J. R. 1957

CONTEXTUAL UNDERSTANDING OF WORDS

- The study of *meaning* and *context* should be central to linguistics
- Exploiting the context-dependent nature of words
- Language patterns cannot be accounted for in terms of a single system
- The *collocation*, gives enough clue to understand a word and its meaning
- *No study of meaning apart from context can be taken seriously*[2]

[2] Firth, J. R. 1957

DISAMBIGUATION OF 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('bank.v.02')	enclose with a bank
Synset('bank.v.03')	do business with a bank or keep an account at a bank
Synset('bank.v.04')	act as the banker in a game or in gambling
Synset('bank.v.05')	be in the banking business
Synset('deposit.v.02')	put into a bank account
Synset('trust.v.01')	have confidence or faith in

DIFFERENT MEANINGS FOR THE 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 system
- 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	awesome breathtaking amazing stunning astounding astonishing awe-inspiring extraordinary incredible unbelievable magnificent wonderful spectacular remarkable

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.
Example:Consider the following short documents:
1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.
Example:Consider the following short documents:
1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0
hates	0	0	0	1	0	1	0	0	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0
hates	0	0	0	1	0	1	0	0	0
Maths	0	0	0	0	1	0	0	0	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0
hates	0	0	0	1	0	1	0	0	0
Maths	0	0	0	0	1	0	0	0	0
She	0	0	0	0	0	0	0	1	0

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0
hates	0	0	0	1	0	1	0	0	0
Maths	0	0	0	0	1	0	0	0	0
She	0	0	0	0	0	0	0	1	0
loves	0	0	0	0	0	0	1	0	1

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example:Consider the following short documents:

1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0
hates	0	0	0	1	0	1	0	0	0
Maths	0	0	0	0	1	0	0	0	0
She	0	0	0	0	0	0	0	1	0
loves	0	0	0	0	0	0	1	0	1
Biology	0	0	0	0	0	0	0	1	0

UNIGRAM, BIGRAMS, TRIGRAMS

- A sequence of two words is called a bigram
- A three-word sequence is called a trigram
- n-gram means a sequence of words of length n

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a
a	a peck	picked a peck

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a
a	a peck	picked a peck
peck	peck of	a peck of

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a
a	a peck	picked a peck
peck	peck of	a peck of
of	of pickled	peck of pickled

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a
a	a peck	picked a peck
peck	peck of	a peck of
of	of pickled	peck of pickled
pickled	pickled peppers	of pickled peppers

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
$< s >$	$< s >$Peter	$< s1 >< s2 >$Peter
Peter	Peter Piper	$< s2 >$Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a
a	a peck	picked a peck
peck	peck of	a peck of
of	of pickled	peck of pickled
pickled	pickled peppers	of pickled peppers
peppers	peppers	–

COLLOCATIONS

Collocations is a juxtaposition of two or more words that more often occur together than ny chance.

- Poverty is a ***major problem*** for many countries
- Ram has a ***powerful computer***
- I had a ***brief chat*** with Raj
- I could not see anything in the room, it was ***pitch dark*** inside
- The crime was committed in ***broad daylight*** - We don't use wide, large, big daylight
- I wish I had a ***strong tea*** - we don't use powerful, tough
- The ***heavy rain*** prevented us from playing outside - We don't use strong rain
- Someone ***knocked*** on the front ***door***

SEMANTIC UNDERSTANDING USING CO-OCCURRENCE - EXAMPLE

	awesome
	breathtaking
	amazing
	stunning
	astounding
	astonishing
The view from the top of the mountain was	awe-inspiring
The shot was	extraordinary
What a magnificent sight. It was	incredible
The photograph is	unbelievable
	magnificent
	wonderful
	spectacular
	remarkable

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
  - Consider these two documents (1) Automobile association (2) car driver
  - Connects the neighbor of Automobile and the neighbor of car
  - "Automobile association" with "car driver" - driver and association could be connected using the similar words ***Automobile and car***
- What is Xalapa?

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
  - Consider these two documents (1) Automobile association (2) car driver
  - Connects the neighbor of Automobile and the neighbor of car
  - "Automobile association" with "car driver" - driver and association could be connected using the similar words ***Automobile and car***
- What is Xalapa?
- Every one likes Xalapa

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
  - Consider these two documents (1) Automobile association (2) car driver
  - Connects the neighbor of Automobile and the neighbor of car
  - "Automobile association" with "car driver" - driver and association could be connected using the similar words ***Automobile and car***
- What is Xalapa?
- Every one likes Xalapa
- Xalapa is served at lunch

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
  - Consider these two documents (1) Automobile association (2) car driver
  - Connects the neighbor of Automobile and the neighbor of car
  - "Automobile association" with "car driver" - driver and association could be connected using the similar words ***Automobile and car***
- What is Xalapa?
- Every one likes Xalapa
- Xalapa is served at lunch
- Main Ingredients - black beans, avocado, tortilla, cumins, tomato puree

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
  - Consider these two documents (1) Automobile association (2) car driver
  - Connects the neighbor of Automobile and the neighbor of car
  - "Automobile association" with "car driver" - driver and association could be connected using the similar words ***Automobile and car***
- What is Xalapa?
- Every one likes Xalapa
- Xalapa is served at lunch
- Main Ingredients - black beans, avocado, tortilla, cumins, tomato puree

Intuition

Xalapa is a food and served at lunch
Xalapa is a lunch item like veg kati roll
Xalapa is vegetarian
Xalapa and kati roll are related

What is the context? food
Xalapa and kati roll are related as they both are served at lunch

You shall know a word by the company it keeps

- Firth, 1957

THANK YOU

Ramaseshan.nlp@gmail.com