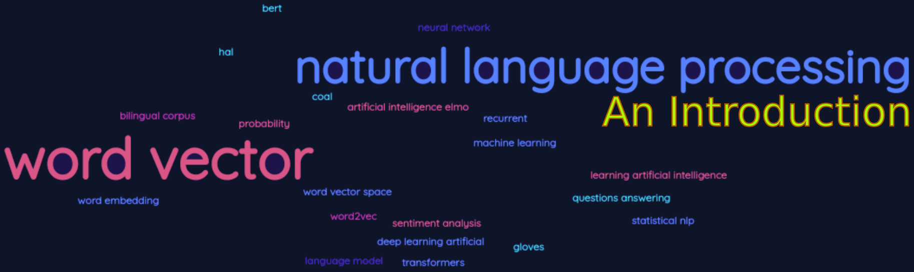
nenote



An Introduction

Ramaseshan Ramachandran

TOPICS TO BE COVERED IN THIS WORKSHOP

GITHUB PAGE

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

Ability to process and harness information from a large corpus of text with a very little human intervention

WHAT IS A CORPUS?

- Collection of a written text in a digital form
- Useful to verify a hypothesis about a language
  - To determine how the usage of a particular sound, word, or syntactic construction varies in different contexts
  - The boys play cricket on the river bank. The boys play cricket by the side of a national bank
- Contains most of the words of a language
- Changes as a function of time - regular increase of corpus size with addition of new text samples
- Corpus is huge - Several billions of words [**Dash2018**]
- Even distribution of texts from all domains of language use
- Represents all areas of coverage of texts of a language
- Access of language data in an easy and simplified manner

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.10') a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning) Synset('bank.v.01') tip laterally

Synset('trust.v.01') have confidence or faith in

IS NLP HARD?

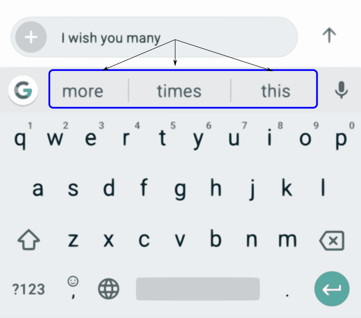
- What is added with 15 to get 45?
- Juvenile court to try shooting defendant
- Safety experts say school bus passengers should be belted
- The king saw a rabbit with his glasses
- Local high school dropouts cut in half

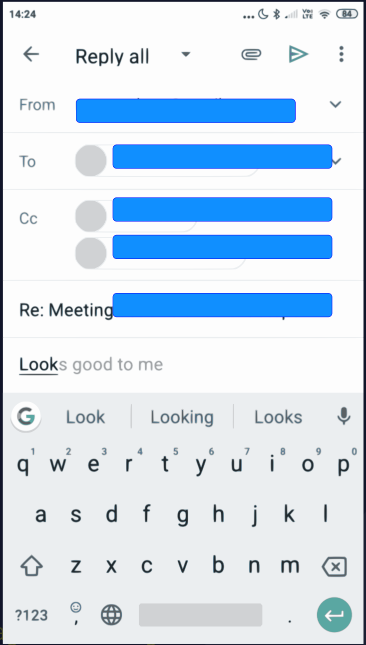
WHY IS NLP HARD?

- Multiple ways of representation of the same scenario
- Includes common sense and contextual representation
- Complex representation information (simple to hard vocabulary)
- Mixing of visual cues
- Ambiguous in nature
- Idioms, metaphors, sarcasm (Yeah! right), double negatives, etc. make it difficult for automatic processing
- Human language interpretation depends on real world, common sense, and contextual knowledge

TYPICAL NLP TASKS

Information Retrieval	Find documents based on keywords
Information Extraction	Identify and extract personal name, date, company name, city..
Language generation	Description based on a photograph Title for a photograph
Text clustering	Automatic grouping of documents
Text classification	Assigning predefined categorization to documents. Identify Spam emails and move them to a Spam folder
Machine Translation	Translate any language Text to another
Grammar checkers	Check the grammar for any language


I wish you many
more
times
this


14:24
Reply all
From
To
Cc
Re: Meeting
Looks good to me
Look
Looking
Looks

APPLICATIONS OF NLP

- Sentiment Analysis
- Search Engines
- Content or News curation
- Automatic Machine Translation
- Spam filtering
- Transcription of Text from Audio/Video
- Chatbots
- ...
- ...

LEXICAL RESOURCES

- A corpus is a collection of machine readable text collected according certain criteria
- Representative collection of text
- Used for statistical analysis and hypothesis testing
- Used for validating linguistic rules within a specific language

- *Brown Corpus* contains a collection of written American English
- *Sussane* is a subset of Brown, but is freely available
- A bi-lingual parallel corpus, *Canadian Hansards*, contains French and English transcripts of the parliament
- Penn-Treebank contains annotated text from the Wall Street journal
- Most NLP software platforms such as *NLTK*, *Spacy* include several corpora for learning purposes
- *HuggingFace and Kaggle* - Several corpora text and image for machine learning applications

OPERATIONS ON A TEXT CORPUS

The basic operation on text is *tokenization*. This is the process of dividing input text into tokens/words by identifying word boundary

- Identify paragraphs, sentences
- Extract tokens
- Count the number of tokens/words in the corpus
- Find the vocabulary count
- Find patterns of words
- Find co-occurrence of words

WORDS AND TERMS

In many applications in NLP the basic alphabet is a ***word***
The next logical step after the binary representation of words or terms t, is to assign weights to words

- 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 computation purposes
- Vocabulary of size $N = 1 \dots n$ is defined as $V = w_1, w_2, w_3, \dots, w_n$ is the vocabulary containing unique words of a language
- Some words found in V appear in documents $(D = D_1, D_2 D_3, \dots, D_m)$, once or several times or may not appear at all.

TERM FREQUENCY

Term Frequency

For the given document, ***term frequency*** is defined as the number of occurrences of a term, t_i, in a document d_i belonging to a corpus $(d_1, d_2, d_3, \ldots, d_m)$. This is denoted by $tf_{t,d}$

TERM FREQUENCY - DEMO

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

MULTIPLE WEIGHTING FACTORS TF

$$Boolean - 0, 1$$

$$RawCount - tf_{i,d}$$

$$\text{Adjusted to document length} - \frac{tf_{i,d}}{M}$$

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

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

BAG OF WORDS

The collection words 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

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

$$IDF_t = \log\left(\frac{N}{D_{f_t}}\right) \tag{0}$$

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

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_{t,d} \tag{0}$$

- 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

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

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

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

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

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

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

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

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

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 with a large corpus.

MANDELBROT APPROXIMATION

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

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

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

$$M \propto T^b$$
$$= kT^b$$

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. It emphasizes the importance of the compression of the dictionary.

DEMO - ZIPF'S AND HEAP'S EMPIRICAL LAWS

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

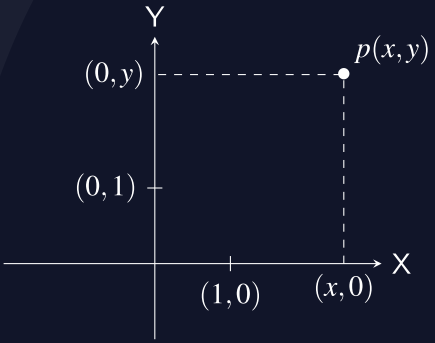
EXERCISES

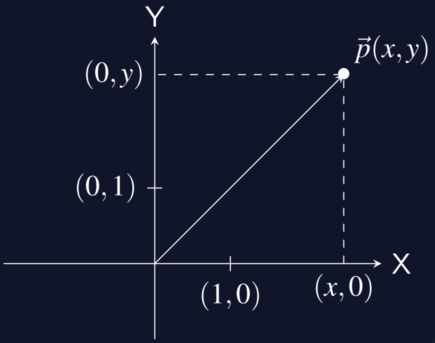
- Write a program to find out whether Mandelbrot's approximation provides a better fit than Zipf's law. Use the same corpus for Zipf and Mandelbrot approximation.
- Write a program for Heap's law and predict the vocabulary size in any corpus. Also, find out whether it is closer to the actual size of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

BREAK

2-D VECTOR SPACE

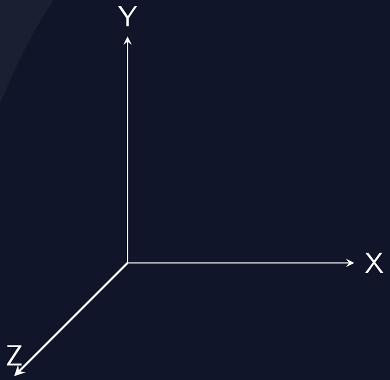
A 2-D vector-space is defined as a set of linearly independent basis vectors with 2 axes. Each axis corresponds to a dimension in the vector-space

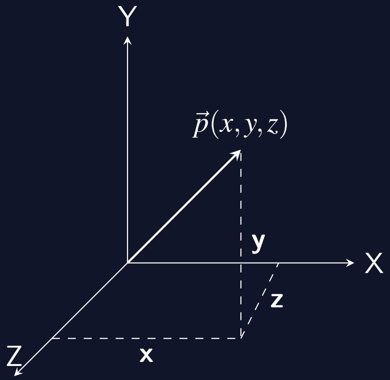




3-D VECTOR SPACE

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.
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

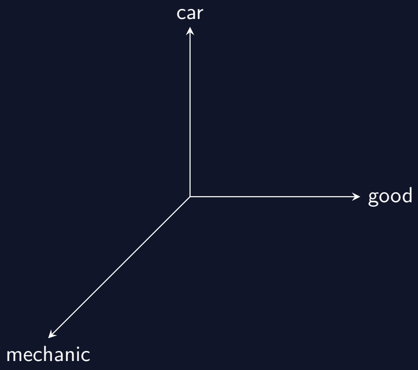
1. The vocabulary size of *emma corpus* is 7079. If we plot all the words in the real space $\mathscr{R}$, we get 7079 axes
2. The vocabulary size of *Google News Corpus corpus* is 3 million. If we plot all the words in the real space $\mathscr{R}$, we get 3 million axes

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

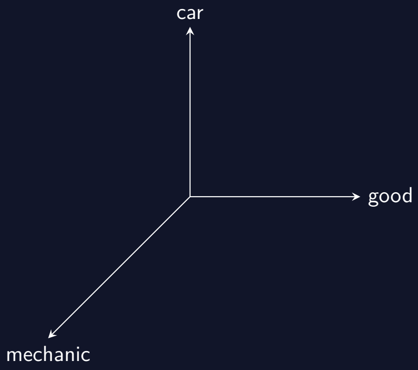
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



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

EXAMPLE - BINARY INCIDENCE MATRIX

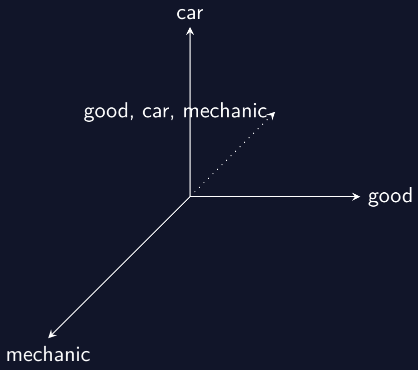
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



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

EXAMPLE - BINARY INCIDENCE MATRIX

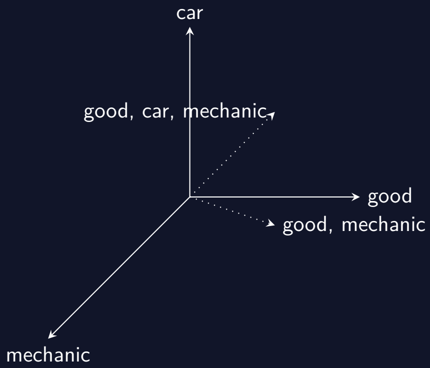
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



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

EXAMPLE - BINARY INCIDENCE MATRIX

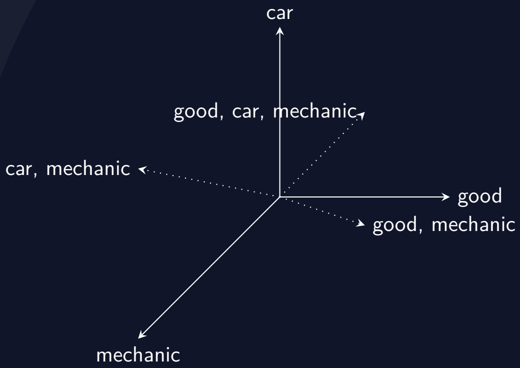
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



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

EXAMPLE - BINARY INCIDENCE MATRIX

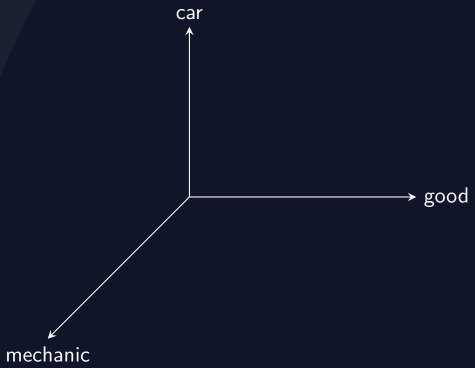
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



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

EXAMPLE - TF-IDF INCIDENCE MATRIX

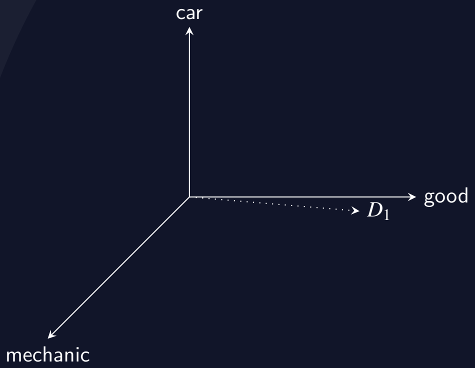
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	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

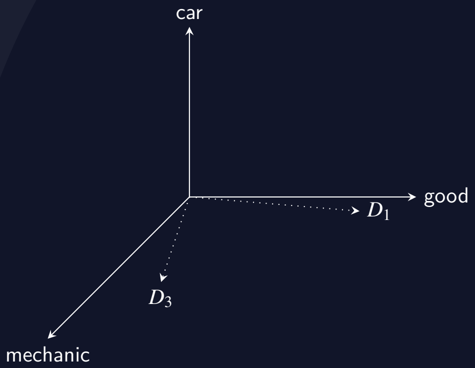
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	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

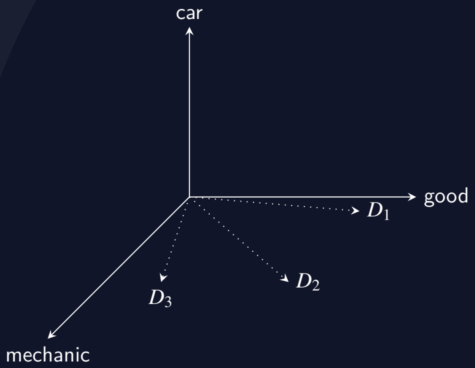
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	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

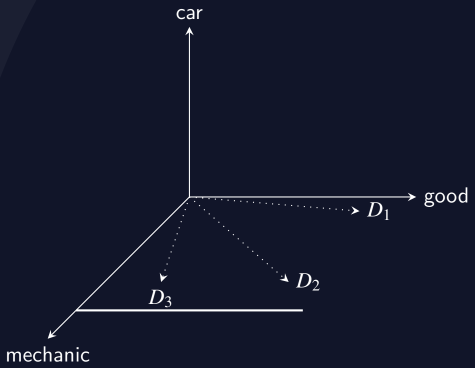
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	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

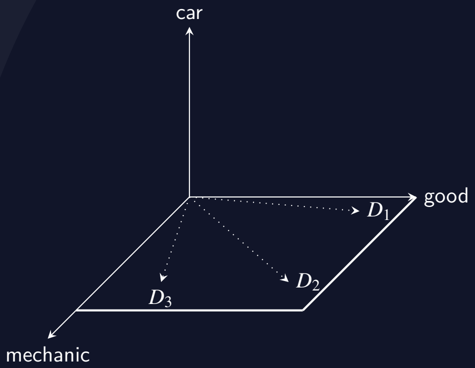
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	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

WEIGHTED-TF-IDF

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.

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

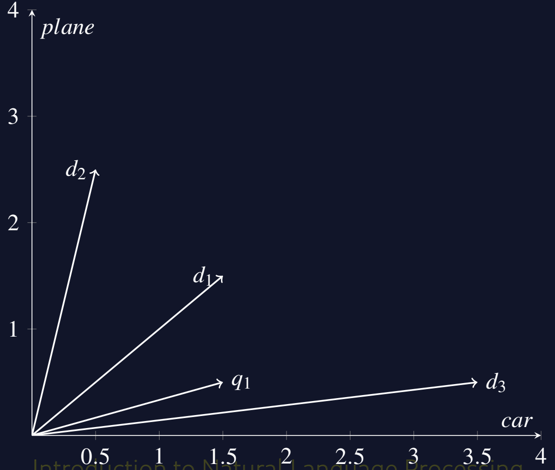
$$\textbf{Cosine Similarity} = \frac{\vec{d}_1 . \vec{d}_2}{\left\|\vec{d}_1\right\| \left\|\vec{d}_2\right\|} = \frac{\vec{d}_1}{\left\|\vec{d}_1\right\|} . \frac{\vec{d}_2}{\left\|\vec{d}_2\right\|}$$

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

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

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

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

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

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

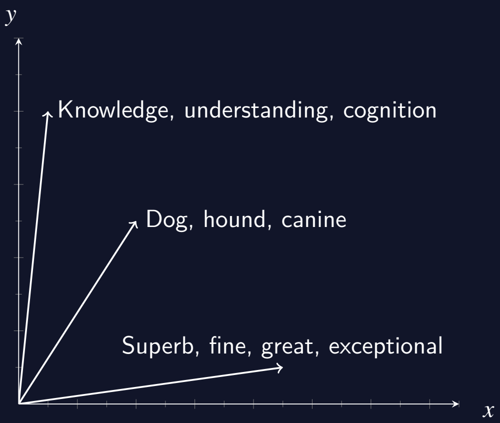
$$\left(t^{Home}\right)^T \cdot t^{House} = 0$$

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

The Goal of Word to Vector

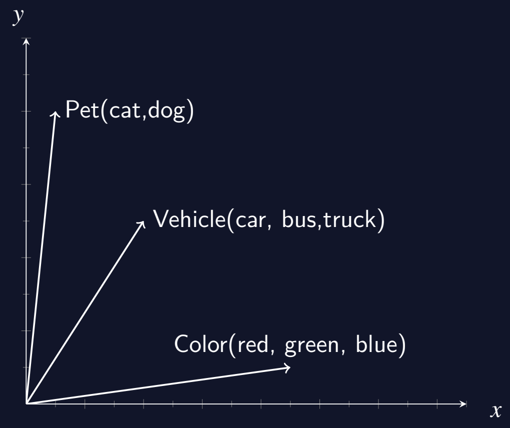
- Reduce word-vector space into a smaller sub-space
- Encode the relationship among words

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

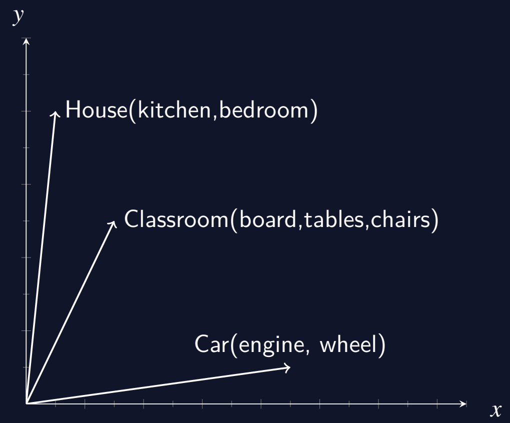


RELATIONSHIP AMONG TERMS - IS-A VECTOR

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('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 The view from the summit was La vue du sommet de la montagne *é*tait Mtazamo wa juu wa mlima huo ulikuwa	awesome breathtaking amazing stunning astounding astonishing awe-inspiring extraordinary incredible unbelievable magnificent wonderful spectacular remarkable

- Translation by analogy: Example based machine translation (EBMT) (lazy learning)

  This is my house - Hii ni nyumba yangu
  My dog loves to run - Mbwa wangu anapenda kukimbia
  I run with my dog - Mimi kukimbia na mbwa wangu
  My house is blue in color - Nyumba yangu ni rangi ya bluu
  This is my dog -

MT FROM EXAMPLE SENTENCES

- Translation by analogy: Example based machine translation (EBMT) (lazy learning)

  This is my house - Hii ni nyumba yangu
  My dog loves to run - Mbwa wangu anapenda kukimbia
  I run with my dog - Mimi kukimbia na mbwa wangu
  My house is blue in color - Nyumba yangu ni rangi ya bluu
  This is my dog - Hii ni mbwa wangu
- Learn MT models from data: Statistical Machine Learning
  - Translation models with language-specific parameters
  - Train model parameters & apply to unseen data

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
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 in the morning

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 in the morning
- 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 in the morning
- Main Ingredients - black beans, avocado, tortilla, cumins, tomato puree

Intuition

Xalapa is food
Xalapa is served as breakfast
Xalapa is a breakfast item like chapathi roll

Xalapa and chapathi roll are related as the context is breakfast
Xalapa and chapathi roll are related as they both are vegetarian

You shall know a word by the company it keeps

- Firth, 1957

LEXICAL SEMANTIC MODELS

- HAL - Hyperspace Analogue to Language[3]
- COALS - Correlated Occurrence Analogue to Lexical Semantic[4]
- GloVe - Global Vectors[5]
- Word2Vec

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

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

[5] Pennington et al, "Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, 1532-1543

CONSTRUCTING SEMANTIC MODELS

- Semantic spaces are constructed by selecting an axis
- Use human judgment to place words in each axis
- To place a set of desirable words, one must choose the axis and find a set of words that must be confined to the chosen axis
- In a *size* axis, placing *ant* and *mountain*
- Can we use lexical co-occurrence to construct semantic spaces?
- Is it possible to construct high-dimensional distributed semantic spaces automatically?

HYPERSPACE ANALOGUE TO LANGUAGE - HAL

- A Window size n representing a span of words is used
- Words within the window (or ramped window), are recorded
- The strength of co-occurrence is computed using a inverse relationship with respect to the word in question
- The co-occurring word strengths are distance and direction sensitive
- A term-term matrix is constructed with every cell representing summed co-occurrence counts for a single word pair
- If the words have similar values in the same dimensions, they will be closer together in the space,meaning they share similar contexts
- The word vectors closest to a given word are considered its neighbors.
- Does this method mimic the actual cognitive process of identifying words belonging to a semantic space?

SEMANTIC SPACE CONSTRUCTION

- High dimensional Semantic space is constructed by using global count of co-occurrences of words
- If the vocabulary size V_s, then a $V_s \times V_s$ co-occurrence matrix is constructed
- A ramped window of length K is moved across the corpus to capture the co-occurrence count
- The strength of co-occurrence $\propto \frac{1}{M}$
- The weighting for a term t_0 with respect to another term term t is given by

$$Weight(t|t_0) = \sum_{m=1}^{M} w(m)n(w_0,m,w)$$

where $n(w_0,m,w)$ is the number of times term w co-occurs with w_0, and $w(m) = M - m + 1$ denotes the strength of relationship between the two terms given m.

EXAMPLE

Example Matrix for the sentence *The Horse Raced Past the Barn Fell.*[6]
(Computed for Window Width of Five Words)

	barn	fell	horse	past	raced	the
barn	0	0	2	4	3	6
fell	5	0	1	3	2	4
horse	0	0	0	0	0	5
past	0	0	4	0	5	3
raced	0	0	5	0	0	4
the	0	0	3	5	4	2

Rows - Count from right to left
Columns - Count from Left to right

- Pick up a text that contains conversational text, variety of topics to cover all type of co-occurrences

[6] All tables and figures mentioned in the presentation were taken from the respective papers as mentioned earlier

EXPERIMENT 1

- 160 million words from Usenet news groups
- Window size = 10
- A word appearing with a frequency of 50 or more is considered as a vocabulary item
- 20 target words selected at random from middle frequency words (Using Zipf's law) - to eliminate most common and rare words
- For each target word, a normalized Euclidean distance was computed from the word to each vocabulary item
- The neighbors with the smallest distances is shown in Table
- These relationships appear to be both semantic and associative
- The high-dimensional neighborhood surrounding each word is similar to a semantic field

EXAMPLE FROM LUND AND BURGESS (1996)

Table 2
Five Nearest Neighbors for Target Words From Experiment 1 (*n*1 . . . *n*5)

Target	*n*1	*n*2	*n*3	*n*4	*n*5
jugs	juice	butter	vinegar	bottles	cans
leningrad	rome	iran	dresden	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

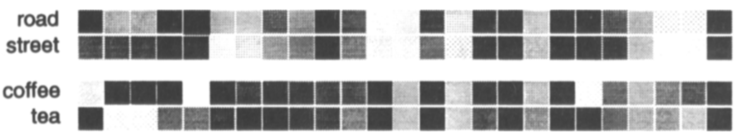
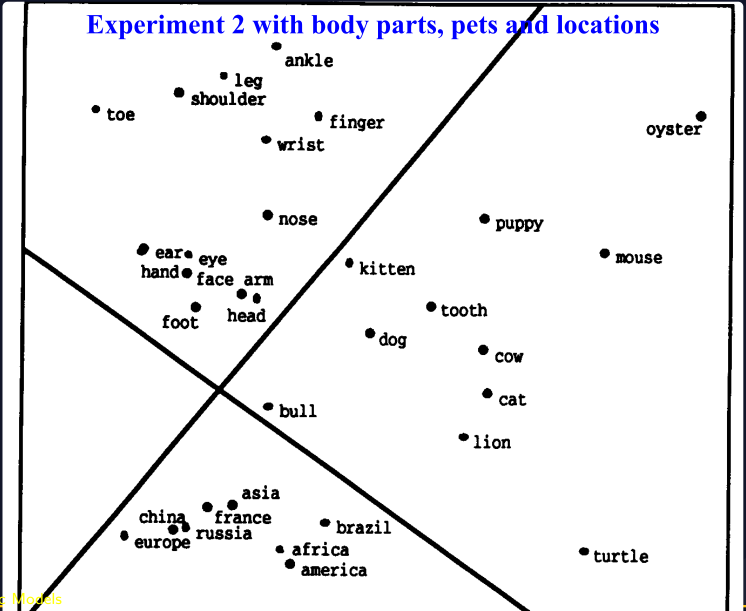


Figure 1. Gray-scaled 25-element co-occurrence vectors.

EXAMPLE FROM LUND BURGESS (1996)



SUMMARY

- HAL captures information about word meanings through the unsupervised analysis of text
- This produces word vectors that are more semantic (similar words) than associative in nature
- HAL acquires word meanings as a function of keeping track of how words are used in context
- The term-term co-occurrence matrix carries the history of the contextual experience by using a moving window and weighting of co-occurring words based on the distance
- HAL exploits the regularities of language such that conceptual generalizations can be captured in a data matrix

HAL DEMO

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

CORRELATED OCCURRENCE ANALOGUE TO LEXICAL SEMANTIC - COALS

1. Gather co-occurrence counts, typically ignoring closed-class neighbors and using a ramped, size 4 window
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 - Instead of using the raw frequency score, correlation score is used to analyze the relationship between pair of words
4. Set negative values to 0, and take square roots of positive ones.
5. The semantic similarity between two words is given by the correlation of their vectors. The correlation coefficient values with this normalization will be in the range of [-1,1]
6. The matrix constructed using this correlation would be semantic space

In HAL, high frequency neighbors have undue influence on the scores. COALS method employs a normalization strategy that largely factors out lexical frequency.
Columns representing low-frequency words are removed

COALS-STEP 1

Consider the corpus

How much wood would a woodchuck chuck,
if a woodchuck could chuck wood?
As much wood as a woodchuck would,
if a woodchuck could chuck wood.

Table 5
Step 1 of the COALS method: The initial co-occurrence table with a ramped, 4-word window.

	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?
a	0	5	9	6	1	10	4	8	18	9	10	0	0
as	5	4	2	1	0	0	7	10	3	2	1	0	5
chuck	9	2	0	8	0	5	1	9	11	2	4	3	3
could	6	1	8	0	0	4	0	6	8	0	2	2	2
how	1	0	0	0	0	0	4	3	0	2	0	0	0
if	10	0	5	4	0	0	0	0	10	3	8	0	0
much	4	7	1	0	4	0	0	10	2	3	0	0	3
wood	8	10	9	6	3	0	10	2	8	5	0	4	6
woodch.	18	3	11	8	0	10	2	8	0	8	10	1	1
would	9	2	2	0	2	3	3	5	8	0	5	0	0
,	10	1	4	2	0	8	0	0	10	5	0	0	0
.	0	0	3	2	0	0	0	4	1	0	0	0	0
?	0	5	3	2	0	0	3	6	1	0	0	0	0

Table 6

Step 2 of the COALS method: Raw counts are converted to correlations.

	a	**as**	**chuck**	**could**	**how**	**if**	**much**	**wood**	**woodch.**	**would**	**,**	**.**	**?**
a	-0.167	-0.014	0.014	0.009	-0.017	0.085	-0.018	-0.033	0.096	0.069	0.085	-0.055	-0.079
as	-0.014	0.031	-0.048	-0.049	-0.037	-0.077	0.133	0.103	-0.054	-0.021	-0.050	-0.037	0.133
chuck	0.014	-0.048	-0.113	0.094	-0.045	0.021	-0.061	0.031	0.048	-0.046	-0.002	0.088	0.031
could	0.009	-0.049	0.094	-0.075	-0.037	0.033	-0.070	0.022	0.049	-0.075	-0.021	0.069	0.023
how	-0.017	-0.037	-0.045	-0.037	-0.018	-0.037	0.192	0.070	-0.055	0.069	-0.037	-0.018	-0.026
if	0.085	-0.077	0.021	0.033	-0.037	-0.077	-0.071	-0.106	0.085	0.006	0.138	-0.037	-0.053
much	-0.018	0.133	-0.061	-0.070	0.192	-0.071	-0.065	0.128	-0.061	0.019	-0.071	-0.034	0.072
wood	-0.033	0.103	0.031	0.022	0.070	-0.106	0.128	-0.113	-0.033	0.001	-0.106	0.111	0.100
woodch.	0.096	-0.054	0.048	0.049	-0.055	0.085	-0.061	-0.033	-0.167	0.049	0.085	-0.017	-0.051
would	0.069	-0.021	-0.046	-0.075	0.069	0.006	0.019	0.001	0.049	-0.075	0.060	-0.037	-0.053
,	0.085	-0.050	-0.002	-0.021	-0.037	0.138	-0.071	-0.106	0.085	0.060	-0.077	-0.037	-0.053
.	-0.055	-0.037	0.088	0.069	-0.018	-0.037	-0.034	0.111	-0.017	-0.037	-0.037	-0.018	-0.026
?	-0.079	0.133	0.031	0.023	-0.026	-0.053	0.072	0.100	-0.051	-0.053	-0.053	-0.026	-0.037

Pearson's correlation coefficient

$$r = \frac{T w_{a,b} - \sum_j w_{a,j} \sum_i w_{b,i}}{\sqrt{\sum_j w_{a,j}(T - \sum_j w_{a,j}) \sum_i w_{b,i}(T - \sum_i w_{b,i})}} \quad (0)$$

where,

$$T = \sum_i \sum_j w_{i,j}$$

COALS-STEP 3

Table 7
Step 3 of the COALS method: Negative values discarded and the positive values square rooted.

	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?
a	0	0	0.120	0.093	0	0.291	0	0	0.310	0.262	0.291	0	0
as	0	0.175	0	0	0	0	0.364	0.320	0	0	0	0	0.365
chuck	0.120	0	0	0.306	0	0.146	0	0.177	0.220	0	0	0.297	0.175
could	0.093	0	0.306	0	0	0.182	0	0.149	0.221	0	0	0.263	0.151
how	0	0	0	0	0	0	0.438	0.265	0	0.263	0	0	0
if	0.291	0	0.146	0.182	0	0	0	0	0.291	0.076	0.372	0	0
much	0	0.364	0	0	0.438	0	0	0.358	0	0.136	0	0	0.268
wood	0	0.320	0.177	0.149	0.265	0	0.358	0	0	0.034	0	0.333	0.317
woodch.	0.310	0	0.220	0.221	0	0.291	0	0	0	0.221	0.291	0	0
would	0.262	0	0	0	0.263	0.076	0.136	0.034	0.221	0	0.246	0	0
,	0.291	0	0	0	0	0.372	0	0	0.291	0.246	0	0	0
.	0	0	0.297	0.263	0	0	0	0.333	0	0	0	0	0
?	0	0.365	0.175	0.151	0	0	0.268	0.317	0	0	0	0	0

COALS - RESULTS

Table 10

The 10 nearest neighbors and their percent correlation similarities for a set of nouns, under the COALS-14K model.

	gun	point	mind	monopoly	cardboard	lipstick	leningrad	feet
1)	46.4 handgun	32.4 points	33.5 minds	39.9 monopolies	47.4 plastic	42.9 shimmery	24.0 moscow	59.5 inches
2)	41.1 f rearms	29.2 argument	24.9 consciousness	27.8 monopolistic	37.2 foam	40.8 eyeliner	22.7 sevastopol	57.7 foot
3)	41.0 f rearm	25.4 question	23.2 thoughts	26.5 corporations	36.7 plywood	38.8 clinique	22.7 petersburg	52.0 metres
4)	35.3 handguns	22.3 arguments	22.4 senses	25.0 government	35.6 paper	38.4 mascara	20.7 novosibirsk	45.7 legs
5)	35.0 guns	21.5 idea	22.2 subconscious	23.2 ownership	34.8 corrugated	37.2 revlon	20.3 russia	45.4 centimeters
6)	32.7 pistol	20.1 assertion	20.8 thinking	22.2 property	32.3 boxes	35.4 lipsticks	19.6 oblast	44.4 meters
7)	26.3 weapon	19.5 premise	20.6 perception	22.2 capitalism	31.3 wooden	35.3 gloss	19.5 minsk	40.2 inch
8)	24.4 rif es	19.3 moot	20.4 emotions	21.8 capitalist	31.0 glass	34.1 shimmer	19.2 stalingrad	38.4 shoulders
9)	24.2 shotgun	18.9 distinction	20.1 brain	21.6 authority	30.7 fabric	33.6 blush	19.1 ussr	37.8 knees
10)	23.6 weapons	18.7 statement	19.9 psyche	21.3 subsidies	30.5 aluminum	33.5 nars	19.0 soviet	36.9 toes

Table 11

The 10 nearest neighbors for a set of verbs, according to the COALS-14K model.

	need	buy	play	change	send	understand	explain	create
1)	50.4 want	53.5 buying	63.5 playing	56.9 changing	55.0 sending	56.3 comprehend	53.0 understand	58.2 creating
2)	50.2 needed	52.5 sell	55.5 played	55.3 changes	42.0 email	53.0 explain	46.3 describe	50.6 creates
3)	42.1 needing	49.1 bought	47.6 plays	48.9 changed	40.2 e-mail	49.5 understood	40.0 explaining	45.1 develop
4)	41.2 needs	41.8 purchase	37.2 players	32.2 adjust	39.8 unsubscribe	44.8 realize	39.8 comprehend	43.3 created
5)	41.1 can	40.3 purchased	35.4 player	30.2 affect	37.3 mail	40.9 grasp	39.7 explained	42.6 generate
6)	39.5 able	39.7 selling	33.8 game	29.5 modify	35.7 please	39.1 know	39.0 prove	37.8 build
7)	36.3 try	38.2 sells	32.3 games	28.3 different	33.3 subscribe	38.8 believe	38.2 clarify	36.4 maintain
8)	35.4 should	36.3 buys	29.0 listen	27.1 alter	33.1 receive	38.5 recognize	37.1 argue	36.4 produce
9)	35.3 do	34.0 sale	26.8 playable	25.6 shift	32.7 submit	38.0 misunderstand	37.0 refute	35.4 integrate
10)	34.7 necessary	31.5 cheap	25.0 beat	25.1 altering	31.5 address	37.9 understands	35.9 tell	35.2 implement

Table 12

The 10 nearest neighbors for a set of adjectives, according to the COALS-14K model.

	high	frightened	red	correct	similar	fast	evil	christian
1)	57.5 low	45.6 scared	53.7 blue	59.0 incorrect	44.9 similiar	43.1 faster	24.3 sinful	48.5 catholic
2)	51.9 higher	37.2 terrif ed	47.8 yellow	37.7 accurate	43.2 different	41.2 slow	23.4 wicked	48.1 protestant
3)	43.4 lower	33.7 confused	45.1 purple	37.5 proper	40.8 same	37.8 slower	23.2 vile	47.9 christians
4)	43.2 highest	33.3 frustrated	44.9 green	36.3 wrong	40.6 such	28.2 rapidly	22.5 demons	47.2 orthodox
5)	35.9 lowest	32.6 worried	43.2 white	34.1 precise	37.7 specif c	27.3 quicker	22.3 satan	47.1 religious
6)	31.5 increases	32.4 embarrassed	42.8 black	32.9 exact	35.6 identical	26.8 quick	22.3 god	46.4 christianity
7)	30.7 increase	32.3 angry	36.8 colored	30.7 erroneous	34.6 these	25.9 speeds	22.3 sinister	43.8 fundamentalist
8)	29.2 increasing	31.6 afraid	35.6 orange	30.6 valid	34.4 unusual	25.8 quickly	22.0 immoral	43.5 jewish
9)	28.7 increased	30.4 upset	33.5 grey	30.6 inaccurate	34.1 certain	25.5 speed	21.5 hateful	43.2 evangelical
10)	[illegible] lowering	30.3 annoyed	32.4 reddish	29.8 acceptable	32.7 various	24.3 easy	21.3 sadistic	41.2 mormon

COALS - SOME INSIGHTS

- The majority of the correlations are negative[7]
- Word 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, ...

[7] Based on a Usenet corpus of size 1.2 billion words

COALS DEMO

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

COMPUTING DENSE WORD VECTORS

- Identify a model that enumerates the relationships between terms and documents
- Identify a model that tries to put similar items closer to each other in some space or structure
- 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

MODELS TO CREATE DENSE VECTORS

- Latent Semantic Analysis or Latent Semantic Indexing
- 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
- 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
- Global Vectors - GloVe

SINGULAR VALUE DECOMPOSITION

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

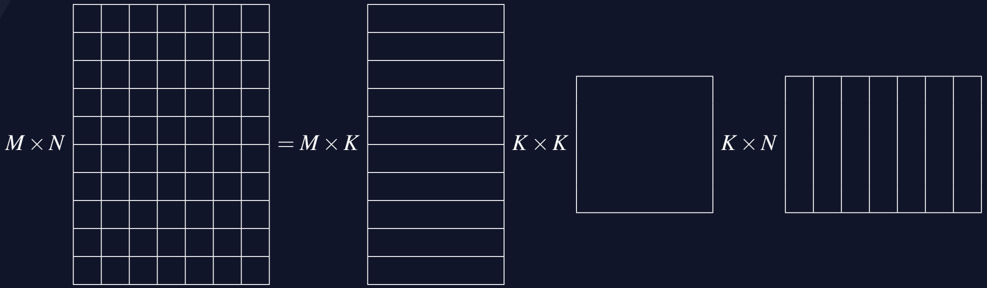
$$A = U\Sigma V^T \tag{0}$$

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

DEMO - IMAGE PROCESSING EXAMPLE

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

DIMENSIONALITY REDUCTION

- SVD is better suited for measuring the similarity between documents, by exploiting the similarity patterns that exist in the word co-occurrence[**Manning1999**]
- The co-occurring terms are mapped into the same dimension thereby reducing the dimensions
- Increases the similarity representation of the semantically similar documents

SVD takes the original Matrix A in the m-dimensional space and transforms it as $\hat{A}$ in the reduced dimensional space $k \leq m$

$$\Delta = \left\| A - \hat{A} \right\|_2 \tag{0}$$

where $\|.\|$ is the L_2 norm for the matrices.

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[8]

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

DEMO - DENSE VECTORS

Cartoon Credit: Ed Himelblau

OMG, we seem to accept cookies more than Broccoli

THANK YOU

Ramaseshan.nlp@gmail.com