Distributional Semantics

Introduction

So far

- Lexical level → tokenization, POS tag
- Syntax → Parsing, CFG, PCFG
- Dependency parsing, Transition based parsing

Next:

Semantics
Distributional semantics
Word embedding

Semantics

- Study of meaning
- What is meaning?
- How come words and sentences have meaning?
- What is the meaning of words and sentences?
- How can the meanings of words combine to form the meaning of sentences?
- Do two people mean the same thing when they utter the word ‘cat’?
- How do we communicate? Etc, etc.

Computational Semantics

- The study of how to automate the process of constructing and reasoning of meaning representations of Natural Language expressions

Approaches:

- Formal representation: using predicate logic
  - Eg: John eat mango → $\exists x[\text{mango}(x) \wedge \text{eat}(\text{John}, x)]$
- Distributional semantics: using statistical pattern learning

Two types of semantics

- Lexical Semantics
- **Distributional Semantics**

Lexical Semantics

- Lexical semantics is the study of word meaning
- Composing the sentence meaning from word meaning
- Issues:
  - Representation of meaning
  - Acquiring broad-domain knowledge
  - Polysemy: same word, multiple meaning
  - Multi-word expressions
- Example:

  What is the good way to remove wine stains?

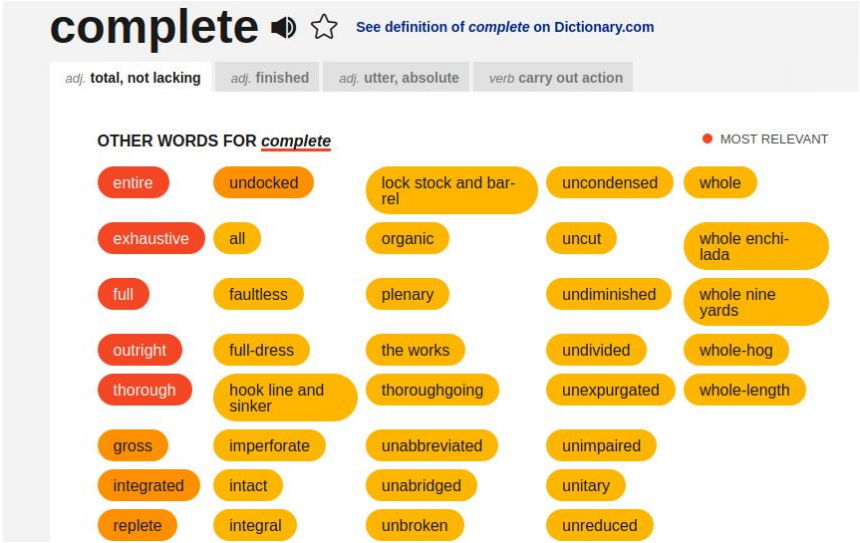
  Salt is a great way to eliminate wine stains

Lexical Semantics

Approaches

- **Wordnet**: dictionary of synonyms
- Basically a large machine-readable thesaurus
- Also capture complex network of relationships between words
  - Hyponymy (IS-A relation)
  - Meronymy (part-of relation)
  - Synset

- https://wordnet.princeton.edu/



Distributional Semantics

Motivation

- 'In most cases, the meaning of a word is its use', (Wittgenstein, 1953) (meaning comes from its usage)
- You know a word by the company it keep (Firth, 1957)
- Words that occur in similar context have similar meaning (i.e Word meaning is reflected in linguistic distribution)
- Semantically similar words have similar distribution pattern

So we need a distributional pattern

Distributional semantics

- We can derive a model of meaning from observable uses of language.
- A typical way to get an approximation of the meaning of words in distributional semantics is to look at their linguistic context in a very large corpus, for instance Wikipedia.
- The linguistic context of a word is simply the other words that appear next to it.

Example:
Let’s now assume that we collect every single instance of the word ‘coconut’ in Wikipedia, and start counting how many times it appears next to ‘tropic’, ‘versatility’, ‘the’, ‘subatomic’, etc.

- We would probably find that coconuts appear many more times next to ‘tropic’ than next to ‘subatomic’:
- this is a good indication of what coconuts are about.

Contextual representation

A word's contextual representation is an abstract cognitive structure that accumulates from encounters with the word in various linguistic contexts.

We learn new words based on contextual cues

He filled the **wampimuk** with the substance, passed it around and we all drunk some.

We found a little **wampimuk** sleeping behind the tree.

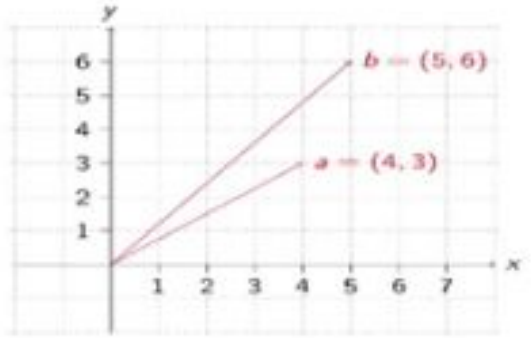
A **Wampimuk** is a fictitious concept proposed by Lazaridou et al. (2014) , to convey how context can shape our perception of a concept, even if we have never heard of it before.

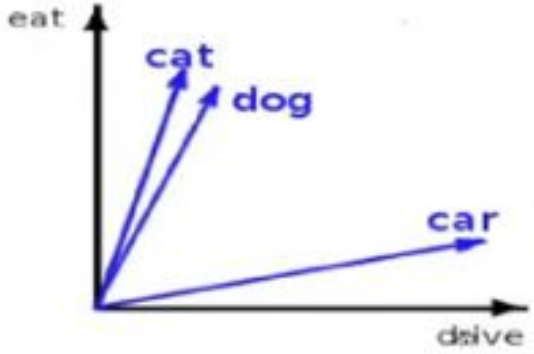
Distributional Semantic Models

- Computational models that builds that builds semantic representations from corpus data
- These models dynamically build semantic representations -- in the form of high-dimensional vector spaces -- through a statistical analysis of the contexts in which words occur.
- Also known as:
  - Corpus based semantics
  - Statistical semantics
  - Word space model
  - Vector space model

Distributional Semantic Models

- Distributions are vectors in a multidimensional semantic space
- Semantic space have dimension which corresponds to the possible context gathered from a given corpus





Example

Small Dataset

An automobile is a wheeled motor vehicle used for transporting passengers .
A car is a form of transport , usually with four wheels and the capacity to carry around five passengers .
Transport for the London games is limited , with spectators strongly advised to avoid the use of cars .
The London 2012 soccer tournament began yesterday , with plenty of goals in the opening matches .
Giggs scored the first goal of the football tournament at Wembley , North London .
Bellamy was largely a passenger in the football match , playing no part in either goal .

Target words: ⟨automobile, car, soccer, football⟩
Term vocabulary: ⟨wheel, transport, passenger, tournament, London, goal, match⟩

Informal algorithm for constructing word spaces

- Pick the words you are interested in: **target words**
- Define a **context window**, number of words surrounding target word
  - The context can in general be defined in terms of documents, paragraphs or sentences.
- Count number of times the target word co-occurs with the context words: **co-occurrence matrix**

distributional matrix = targets X contexts

	wheel	transport	passenger	tournament	London	goal	match
automobile	1	1	1	0	0	0	0
car	1	2	1	0	1	0	0
soccer	0	0	0	1	1	1	1
football	0	0	1	1	1	2	1

Word distribution

Computing similarity

	wheel	transport	passenger	tournament	London	goal	match
automobile	1	1	1	0	0	0	0
car	1	2	1	0	1	0	0
soccer	0	0	0	1	1	1	1
football	0	0	1	1	1	2	1

Using simple vector product

automobile . car = 4
automobile . soccer = 0
automobile . football = 1
car . soccer = 1
car . football = 2
soccer . football = 5

Word Embedding

- Word embedding is one of the most popular representation of document vocabulary.
- It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc.
- **vector representations of a particular word**
- How do we generate?
- How do we capture context?

Why do we need?

Have a good day and ***Have a great day.*** They hardly have different meaning.

If we construct an exhaustive vocabulary (let's call it V), it would have V = {Have, a, good, great, day}.

One-hot encoding

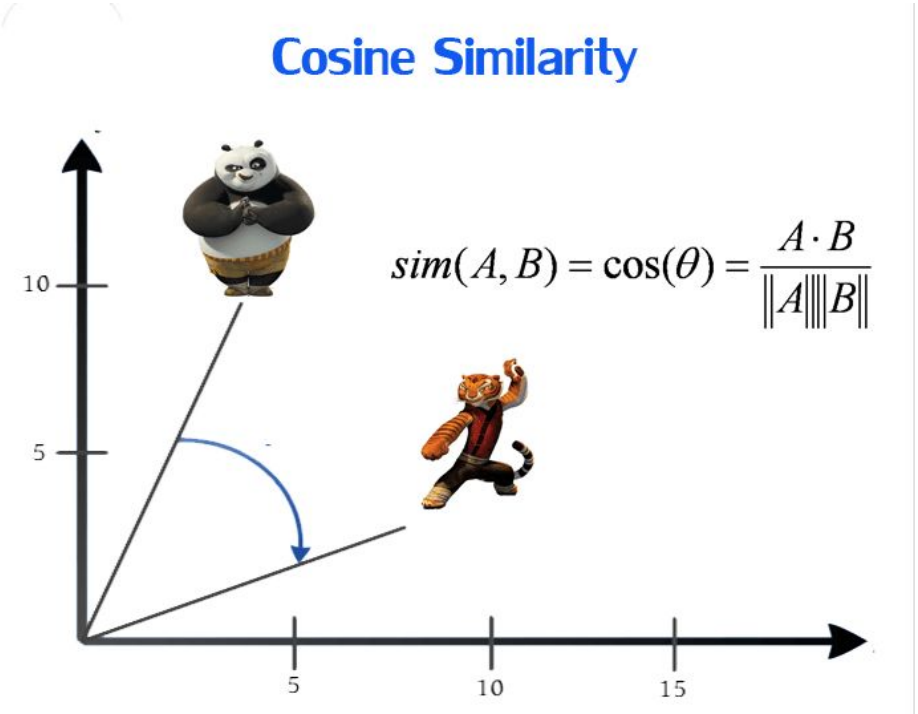
Have = [1,0,0,0,0]`; a=[0,1,0,0,0]` ; good=[0,0,1,0,0]` ; great=
[0,0,0,1,0]` ; day=[0,0,0,0,1]` (` represents transpose)

If we project this vectors in 5D space, we can see good and great have difference.. Which is not True

What we can do?

Objective: words with similar context occupy close spatial positions

the idea of generating **distributed representations**.



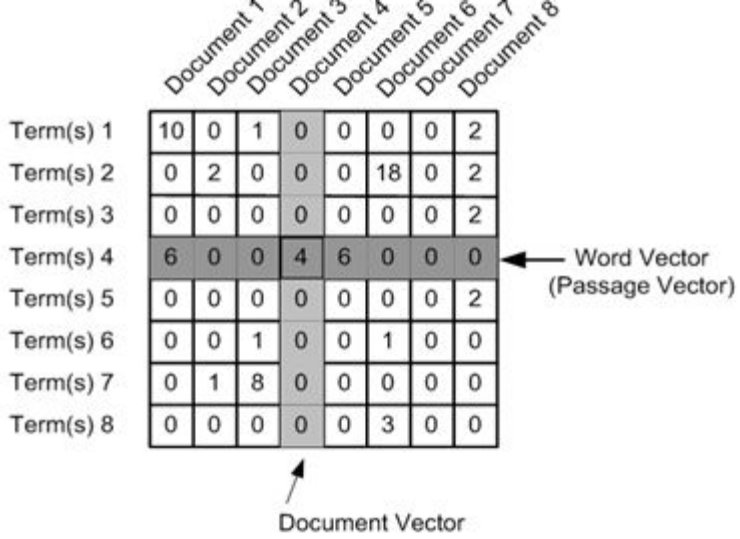
Different Types of word embedding

The different types of word embeddings can be broadly classified into two categories-

Frequency based Embedding: count vector, TF-IDF vector, co-occurrence

Prediction based Embedding: (Word2vec): CBOW model, Skip-gram model

Count Vector



Let us understand this using a simple example.

D1: He is a lazy boy. She is also lazy.

D2: Neeraj is a lazy person.

The dictionary created may be a list of unique tokens(words) in the corpus =['He','She','lazy','boy','Neeraj',

Here, D=2, N=6

The count matrix M of size 2 X 6 will be represented as –

	He	She	lazy	boy	Neeraj	person
D1	1	1	2	1	0	0
D2	0	0	1	0	1	1

TF-IDF vector

TF = (Number of times term t appears in a document)/(Number of terms in the document)

So, TF(This,Document1) = 1/8

TF(This, Document2)=1/5

It denotes the contribution of the word to the document i.e words relevant to the document should be frequent.
eg: A document about Messi should contain the word 'Messi' in large number.

IDF = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.

where N is the number of documents and n is the number of documents a term t has appeared in.

So, IDF(This) = log(2/2) = 0.

Co-occurrence Matrix

Similar words tend to occur together and will have similar context

Apple is a fruit. Mango is a fruit.
Apple and mango tend to have a similar context i.e fruit.

Use term-document matrix and SVD (Latent semantic Indexing)

$$\underset{m \times n}{\hat{X}} \approx \underset{m \times r}{U} \; \underset{r \times r}{S} \; \underset{r \times n}{V^{\mathsf{T}}}$$

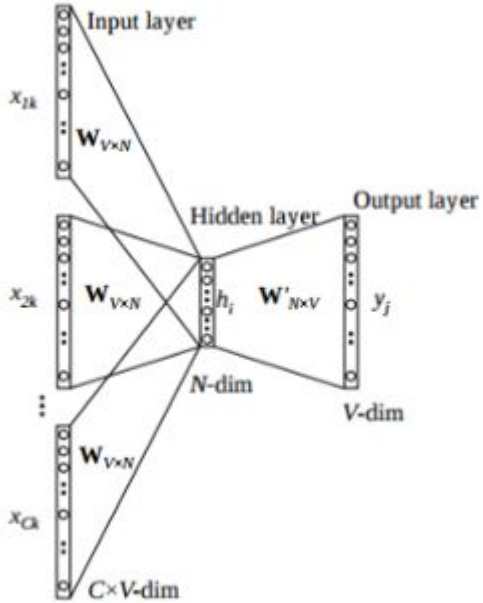
CBOW- Methods

predict the probability of a word given a context

A context may be single word or multiple word for a given target words.

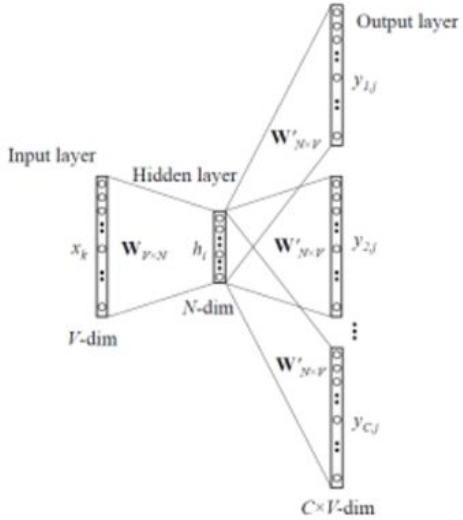
lets see this by an example “The cat jumped over the puddle.”

So one approach is to treat {“The”, “cat”, ’over”, “the’, “puddle”} as a context and from these words, be able to predict or generate the center word “jumped”. This type of model we call a Continuous Bag of Words (CBOW) Model.

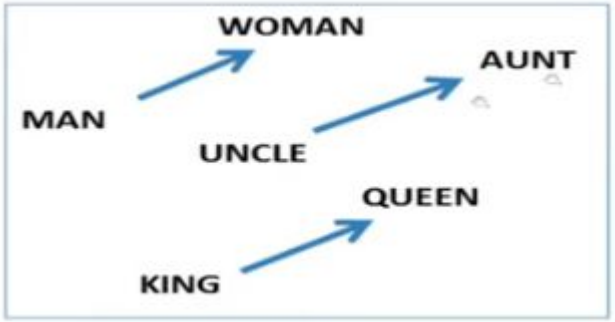


Skip-gram model

given the center word “jumped”, the model will be able to predict or generate the surrounding words “The”, “cat”, “over”, “the”, “puddle”.



Learned vectors can be used for reasoning



Analogy Testing

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

a:b :: c:?

$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

man:woman :: king:?

+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]