Distributional Semantics

How do we represent the meaning in NLP?

- The idea that is represented by a word, phrase, etc.
- The connection between signifier (symbol) and signified (idea or concept).

How do we have usable meaning in a computer?

Common Solution: Use WordNet

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: duck [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **duck** (small wild or domesticated web-footed broad-billed swimming bird usually having a depressed body and short legs)
- S: (n) **duck**, duck's egg ((cricket) a score of nothing by a batsman)
- S: (n) **duck** (flesh of a duck (domestic or wild))
- S: (n) **duck** (a heavy cotton fabric of plain weave; used for clothing and tents)

Verb

- S: (v) **duck** (to move (the head or body) quickly downwards or away) *"Before he could duck, another stone struck him"*
- S: (v) **duck** (submerge or plunge suddenly)
- S: (v) dip, douse, **duck** (dip into a liquid) *"He dipped into the pool"*
- S: (v) hedge, fudge, evade, put off, circumvent, parry, elude, skirt, dodge, **duck**, sidestep (avoid or try to avoid fulfilling, answering, or performing (duties, questions, or issues)) *"He dodged the issue"; "she skirted the problem"; "They tend to evade their responsibilities"; "he evaded the questions skillfully"*

Problems: Lot of manual efforts, still can never be up to date! How to compute word similarity?

In traditional NLP / IR, words are treated as discrete symbols.

One-hot representation

Words are represented as one-hot vectors: one 1, the rest 0s

Vector dimension = number of words in vocabulary (e.g., 500,000)

Example: In web search, if user searches for “Baltimore motel”, we would like to match documents containing “Baltimore hotel”. But

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

The vectors are orthogonal, and there is no natural notion of similarity between one-hot vectors!

Solution: Can we learn to encode similarity in the vectors themselves?

Distributional Hypothesis

Distributional Hypothesis

Distributional Hypothesis: Basic Intuition

"The meaning of a word is its use in language." (Wittgenstein, 1953)

"You know a word by the company it keeps." (Firth, 1957)

$\rightarrow$ Word meaning (whatever it might be) is reflected in linguistic distributions.

"Words that occur in the same contexts tend to have similar meanings." (Zellig Harris, 1968)

$\rightarrow$ Semantically similar words tend to have similar distributional patterns.

Distributional Semantics: a cognitive perspective

Contextual representation

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

Distributional Semantics: a cognitive perspective

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

Distributional Semantics: a cognitive perspective

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.

Distributional Semantics: a cognitive perspective

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.

Distributional Similarity Based Representations

Distributional Semantics: A word's meaning is given by the words that frequently appear close-by

You know a word by the company it keeps

One of the most successful ideas of modern statistical NLP!

Distributional Similarity Based Representations

Distributional Semantics: A word's meaning is given by the words that frequently appear close-by

You know a word by the company it keeps

One of the most successful ideas of modern statistical NLP!

- The context of a word is the set of words that appear nearby within a fixed size window
- Use the many contexts of a word to build up its representation

government debt problems turning into	banking	crises as has happened in
saying that Europe needs unified	banking	regulation to replace the hodgepodge

These context words will represent banking

Building a DSM step-by-step

The "linguistic" steps

Pre-process a corpus (to define targets and contexts)

$\Downarrow$

Select the targets and the contexts

The "mathematical" steps

Count the target-context co-occurrences

$\Downarrow$

Weight the contexts (optional)

$\Downarrow$

Build the distributional matrix

$\Downarrow$

Reduce the matrix dimensions (optional)

$\Downarrow$

Compute the vector distances on the (reduced) matrix

Word Space

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⟩

Constructing Word spaces

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**
- Build vectors out of (a function of) these co-occurrence counts

Constructing Word spaces: distributional vectors

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

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

Many design choices

Matrix type		Weighting		Dimensionality reduction		Vector comparison
word × document		probabilities		LSA		Euclidean
word × word		length normalization		PLSA		Cosine
word × search proximity	×	TF-IDF	×	LDA	×	Dice
adj. × modified noun		PMI		PCA		Jaccard
word × dependency rel.		Positive PMI		IS		KL
verb × arguments		PPMI with discounting		DCA		KL with skew
⋮		⋮		⋮		⋮

Context weighting: words as context

While constructing the vector for a target word, we gave equal weightage to each context word.

But some word associations (e.g., target - context) are more significant, or more informative, than other word associations.

Context weighting: words as context

basic intuition

word1	word2	freq(1,2)	freq(1)	freq(2)
dog	small	855	33,338	490,580
dog	domesticated	29	33,338	918

Association measures are used to give more weight to contexts that are more significantly associated with a target word.

- The less frequent the target and context element are, the higher the weight given to their co-occurrence count should be.
  $\Rightarrow$ Co-occurrence with frequent context element *small* is less informative than co-occurrence with rarer *domesticated*.

Context weighting: words as context

basic intuition

word1	word2	freq(1,2)	freq(1)	freq(2)
dog	small	855	33,338	490,580
dog	domesticated	29	33,338	918

Association measures are used to give more weight to contexts that are more significantly associated with a target word.

- The less frequent the target and context element are, the higher the weight given to their co-occurrence count should be.
  $\Rightarrow$ Co-occurrence with frequent context element *small* is less informative than co-occurrence with rarer *domesticated*.
- different measures - e.g., Mutual information, Log-likelihood ratio

Pointwise Mutual Information (PMI)

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{ind}(w_1, w_2)}$$

Pointwise Mutual Information (PMI)

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{ind}(w_1, w_2)}$$

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{corpus}(w_1)P_{corpus}(w_2)}$$

Pointwise Mutual Information (PMI)

$$PMI(w_1,w_2) = log_2 \frac{P_{corpus}(w_1,w_2)}{P_{ind}(w_1,w_2)}$$

$$PMI(w_1,w_2) = log_2 \frac{P_{corpus}(w_1,w_2)}{P_{corpus}(w_1)P_{corpus}(w_2)}$$

$$P_{corpus}(w_1,w_2) = \frac{freq(w_1,w_2)}{N}$$

$$P_{corpus}(w) = \frac{freq(w)}{N}$$

Distributional Vectors: Example

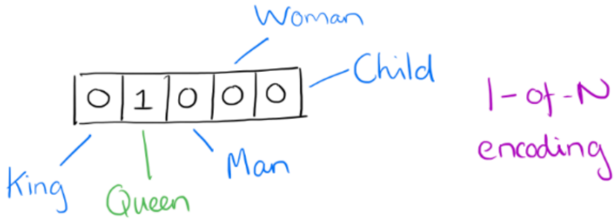
Normalized Distributional Vectors using Pointwise Mutual Information

petroleum	oil:0.032 gas:0.029 crude:0.029 barrels:0.028 exploration:0.027 barrel:0.026 opec:0.026 refining:0.026 gasoline:0.026 fuel:0.025 natural:0.025 exporting:0.025
drug	trafficking:0.029 cocaine:0.028 narcotics:0.027 fda:0.026 police:0.026 abuse:0.026 marijuana:0.025 crime:0.025 colombian:0.025 arrested:0.025 addicts:0.024
insurance	insurers:0.028 premiums:0.028 lloyds:0.026 reinsurance:0.026 underwriting:0.025 pension:0.025 mortgage:0.025 credit:0.025 investors:0.024 claims:0.024 benefits:0.024
forest	timber:0.028 trees:0.027 land:0.027 forestry:0.026 environmental:0.026 species:0.026 wildlife:0.026 habitat:0.025 tree:0.025 mountain:0.025 river:0.025 lake:0.025
robotics	robots:0.032 automation:0.029 technology:0.028 engineering:0.026 systems:0.026 sensors:0.025 welding:0.025 computer:0.025 manufacturing:0.025 automated:0.025

Word embeddings using Word2vec

Word Vectors - One-hot Encoding

- Suppose our vocabulary has only five words: King, Queen, Man, Woman, and Child.
- We could encode the word 'Queen' as:



Word2Vec – A distributed representation

Distributional representation – word embedding?

Any word w_i in the corpus is given a distributional representation by an embedding

$$w_i \in R^d$$

i.e., a $d-$dimensional vector, which is mostly learnt!

$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Distributional Representation

- Take a vector with several hundred dimensions (say 1000).
- Each word is represented by a distribution of weights across those elements.
- So instead of a one-to-one mapping between an element in the vector and a word, the representation of a word is spread across all of the elements in the vector, and
- Each element in the vector contributes to the definition of many words.

Distributional Representation: Illustration

If we label the dimensions in a hypothetical word vector (there are no such pre-assigned labels in the algorithm of course), it might look a bit like this:

Such a vector comes to represent in some abstract way the 'meaning' of a word

Word Embeddings

- d typically in the range 50 to 1000
- Similar words should have similar embeddings

Reasoning with Word Vectors

- It has been found that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way.
- Specifically, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship.

Reasoning with Word Vectors

- It has been found that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way.
- Specifically, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship.

Case of Singular-Plural Relations

If we denote the vector for word i as x_i, and focus on the singular/plural relation, we observe that

$$x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families} \approx x_{car} - x_{cars}$$

and so on.

Perhaps more surprisingly, we find that this is also the case for a variety of semantic relations.

Good at answering analogy questions

a is to b, as c is to ?

man is to *woman* as *uncle* is to ? (*aunt*)

Reasoning with Word Vectors

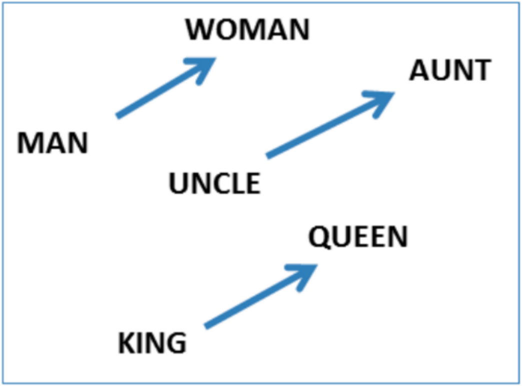
Perhaps more surprisingly, we find that this is also the case for a variety of semantic relations.

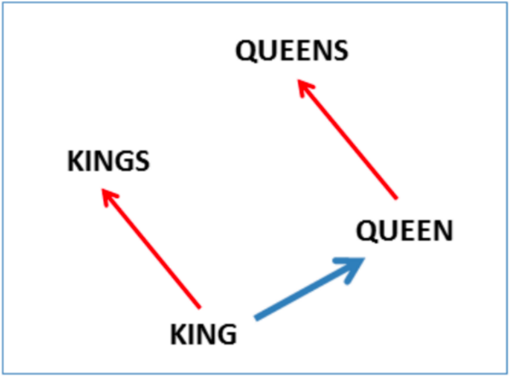
Good at answering analogy questions

a is to b, as c is to ?
man is to *woman* as *uncle* is to ? (*aunt*)

A simple vector offset method based on cosine distance shows the relation.

Vcctor Offset for Gender Relation




QUEENS
KINGS
QUEEN
KING

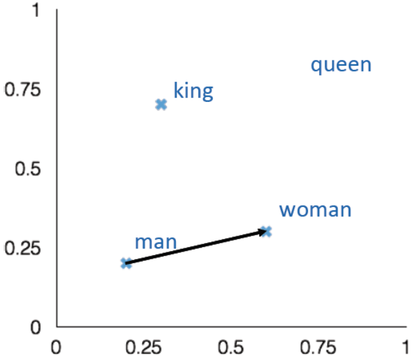
Analogy Testing

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]



More Analogy Questions

Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Table 2: Examples of the analogical reasoning task for phrases (the full test set has 3218 examples). The goal is to compute the fourth phrase using the first three. Our best model achieved an accuracy of 72% on this dataset.

Element Wise Addition

We can also use element-wise addition of vector elements to ask questions such as 'German + airlines' and by looking at the closest tokens to the composite vector come up with impressive answers:

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

Okay, so word vectors seem very useful

But how do we learn such vectors?

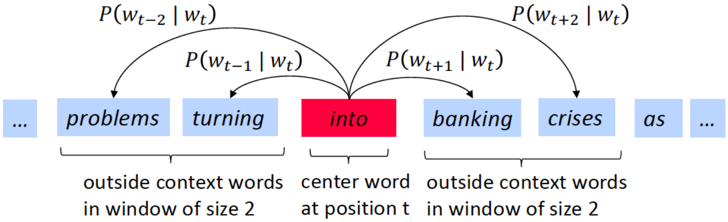
Learning Word Vectors: Overview

Basic Idea

- We have a large corpus of text
- Every word in a fixed vocabulary is represented by a *vector*
- Go through each position t in the text, which has a center word c and context ("outside") words o
- Use the similarity of the word vectors for c and o to calculate the probability of o given c (or vice versa)
- Keep adjusting the word vectors to maximize this probability

Word2Vec (Skip-gram) Overview

Example windows and process for computing $P(w_{t+j}|w_t)$



Example windows and process for computing $P(w_{t+j}|w_t)$