# 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?

# *Word Representation*

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

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

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

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

# Problems with words as discrete symbols

**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: 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)*

# 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.

# 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)*

# *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)*

→ 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)*

→ 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 |

# 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!

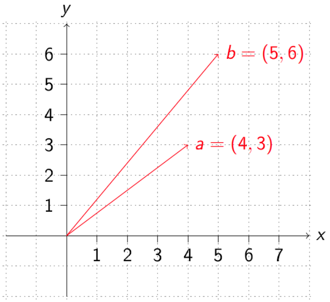
- The context of a word is the set of words that appear nearby within a fixed size window
- Use the many contexts of a word to build up its representation

government debt problems turning into banking crises as has happened in

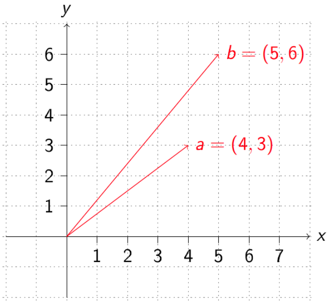
saying that Europe needs unified banking regulation to replace the hodgepodge

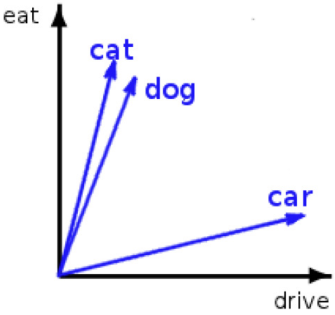
*These context words will represent banking*

# Representation Framework: Vector Space Model?



# Representation Framework: Vector Space Model?




In practice, many more dimensions are used.

$cat$ = [...dog 0.8, eat 0.7, joke 0.01, mansion 0.2,...]

# *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

# 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

$\Downarrow$

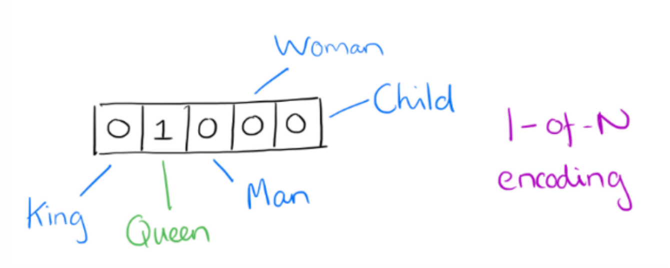
Reduce the matrix dimensions (optional)

$\Downarrow$

Compute the vector distances on the (reduced) matrix

# Word 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!

## *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*

# What do word vectors denote

Within the word embedding, various features of syntax and semantics may be included, e.g.,

- Element 1 might be more positive for nouns
- Element 2 might be positive for animate objects
- Element 3 might have no intuitive meaning whatsoever

# *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.

# 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

# 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.

# *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*)

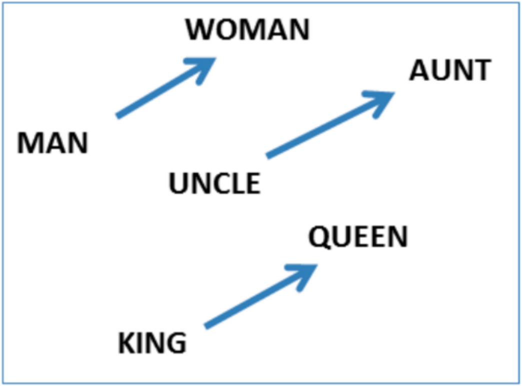
Perhaps more surprisingly, we find that this is also the case for a variety of semantic relations.

### *Good at answering analogy questions*

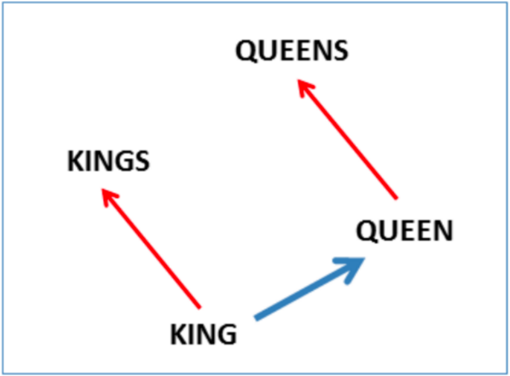
a is to b, as c is to ?
*man* is to *woman* as *uncle* is to ? (*aunt*)

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

# Vcctor Offset for Gender Relation



# Vcctor Offset for Singular-Plural Relation



# Encoding Other Dimensions of Similarity

## 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 |

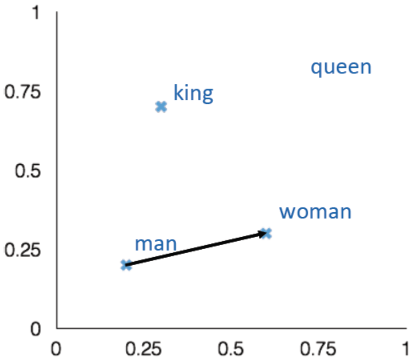
# Analogy Testing

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 ] |



# *Country-capital city relationships*

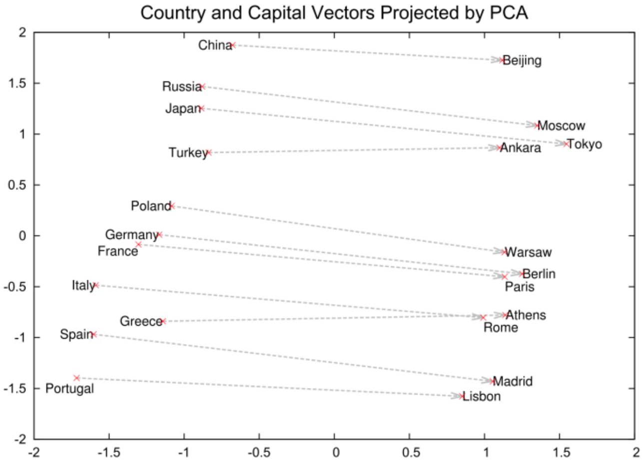



Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

# *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.