Meanings and Distributional Representations

Goals

1. Consolidating the concept of ‘Distributional Hypothesis’
2. Using the ‘Distributional Hypothesis’ to infer the semantic similarity between words

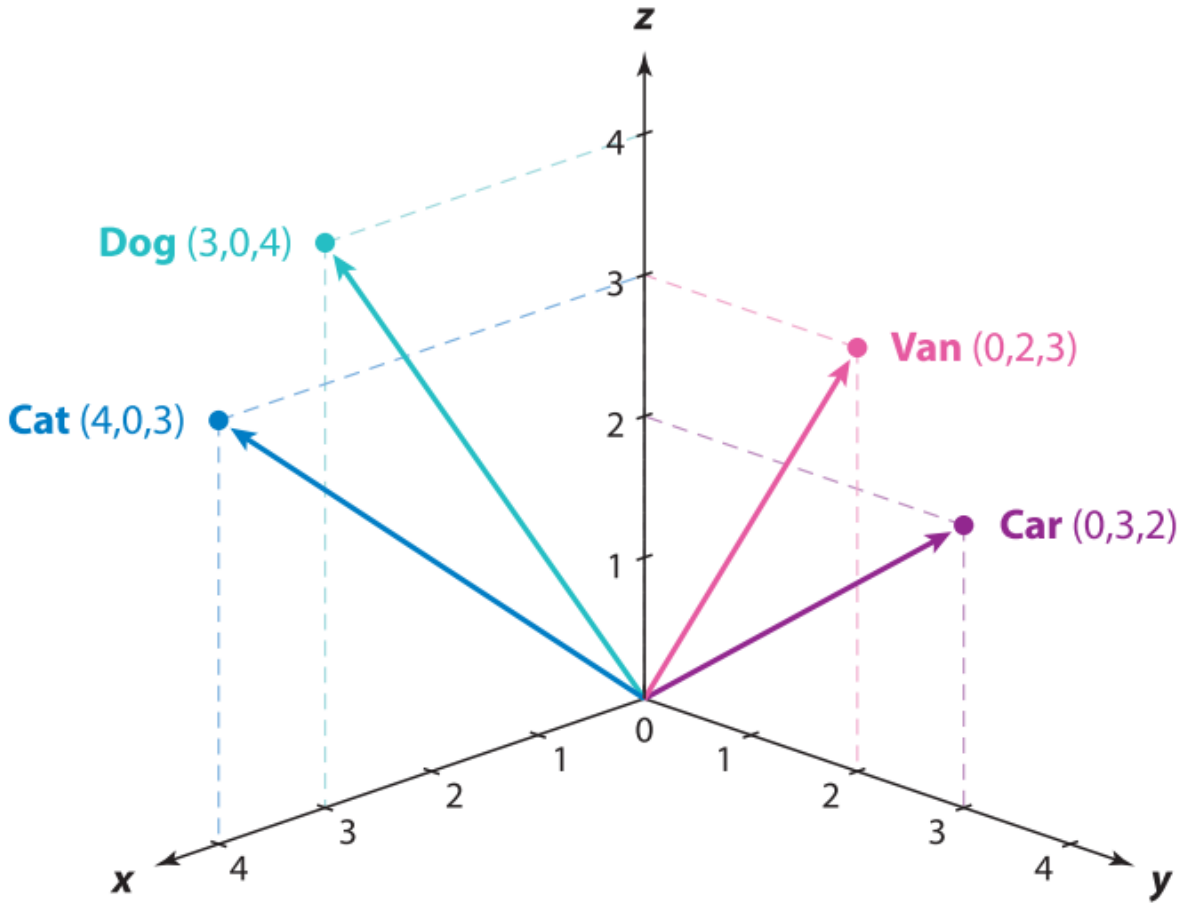
Recalling DH

The Distributional Hypothesis (DH)

“Difference of meaning correlates with [a] difference of distribution’’ (Harris, 1954)

”Semantic similarity is a function of the contexts in which words are used”. (Miller & Charles, 1951)

“DS is not only a method for lexical analysis but also a theoretical framework to build computational models of semantic memory’’ (Lenci, 2018)



From DH to distributional representations

The distributional representation of a lexical item is typically a distributional vector representing its co-occurrences with linguistic contexts — hence the name vector space semantics.

The kind of co-occurrence relation between target and context lexemes determines different types of collocates and distributional representations.

Context types (Firth (1957): (You shall know a word) by the company it keeps!

Linguistic contexts and o-occurrences

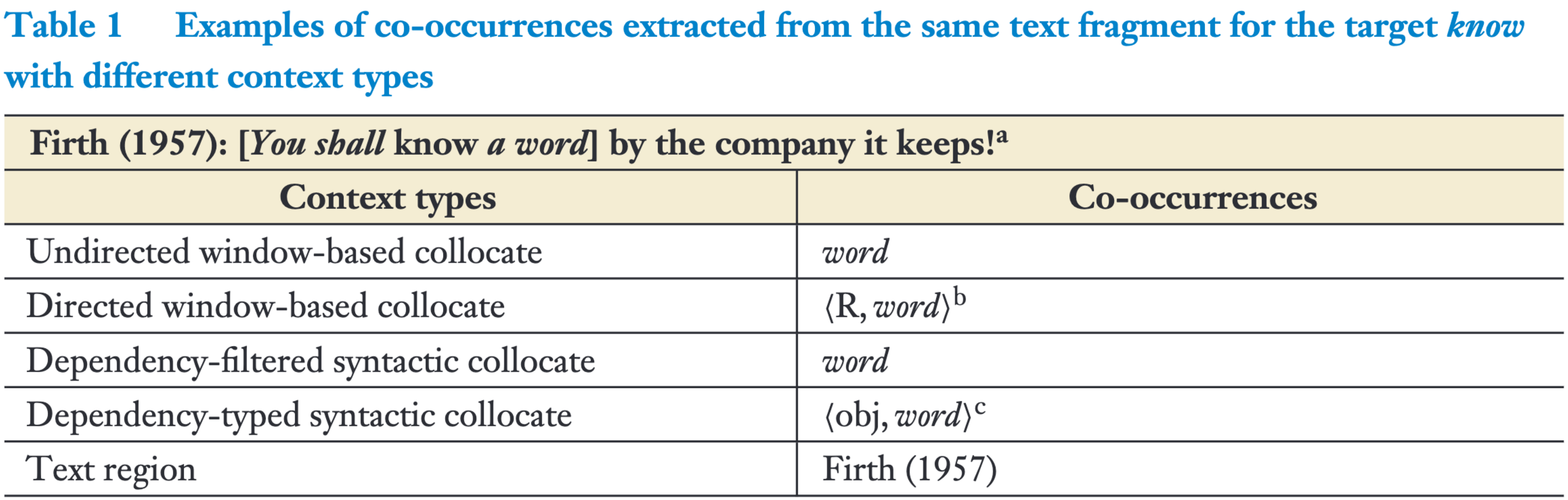


Table 1 Examples of co-occurrences extracted from the same text fragment for the target *know* with different context types

Firth (1957): [*You shall* know *a word*] by the company it keeps![a]	
Context types	**Co-occurrences**
Undirected window-based collocate	*word*
Directed window-based collocate	⟨R, *word*⟩[b]
Dependency-filtered syntactic collocate	*word*
Dependency-typed syntactic collocate	⟨obj, *word*⟩[c]
Text region	Firth (1957)

Source: Lenci, 2018

Procedure to build distributional vectors

Procedure to build distributional vectors

The primary method of building distributional vectors consists of the following procedure:

1. Co-occurrences between lexical items and linguistic contexts are extracted from a corpus and counted. Then, the distribution of lexical items is represented with a co-occurrence matrix, whose rows correspond to target lexical items, columns to contexts, and the entries to their co-occurrence frequency

2. Raw frequencies are then usually transformed into significance weights to reflect the importance of the contexts

3. The semantic similarity between lexemes is measured with the similarity between their row vectors in the co-occurrence matrix.

Setting

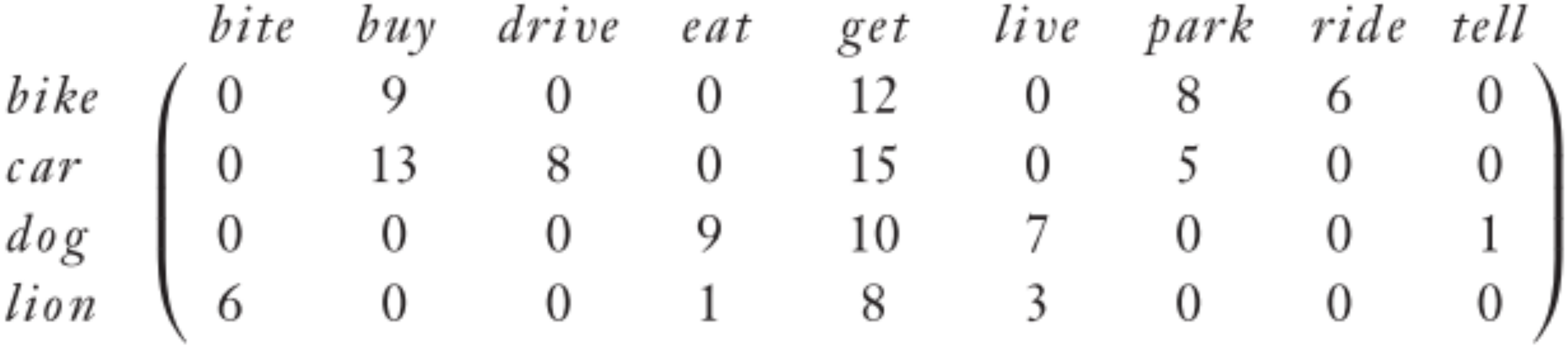
Suppose we have a corpus of text from which we extracted and counted the co-occurrences of the targets

$\mathrm{T} = \{\mathrm{bike}, \mathrm{car}, \mathrm{dog}, \mathrm{lion}\}$

with the context lexemes

$\mathrm{C} = \{\mathrm{bite}, \mathrm{buy}, \mathrm{drive}, \mathrm{eat}, \mathrm{get}, \mathrm{live}, \mathrm{park}, \mathrm{ride}, \mathrm{tell}\}$

Step 1: Input data $\mathrm{M}[\mathrm{T},\mathrm{C}]$



$$\begin{array}{c|ccccccccc}
 & bite & buy & drive & eat & get & live & park & ride & tell \\
\hline
bike & 0 & 9 & 0 & 0 & 12 & 0 & 8 & 6 & 0 \\
car & 0 & 13 & 8 & 0 & 15 & 0 & 5 & 0 & 0 \\
dog & 0 & 0 & 0 & 9 & 10 & 7 & 0 & 0 & 1 \\
lion & 6 & 0 & 0 & 1 & 8 & 3 & 0 & 0 & 0
\end{array}$$

Their distribution is represented with the co-occurrence matrix $\mathrm{M}[\mathrm{T},\mathrm{C}]$, in which the t, c element is the co-occurrence frequency of t with c.

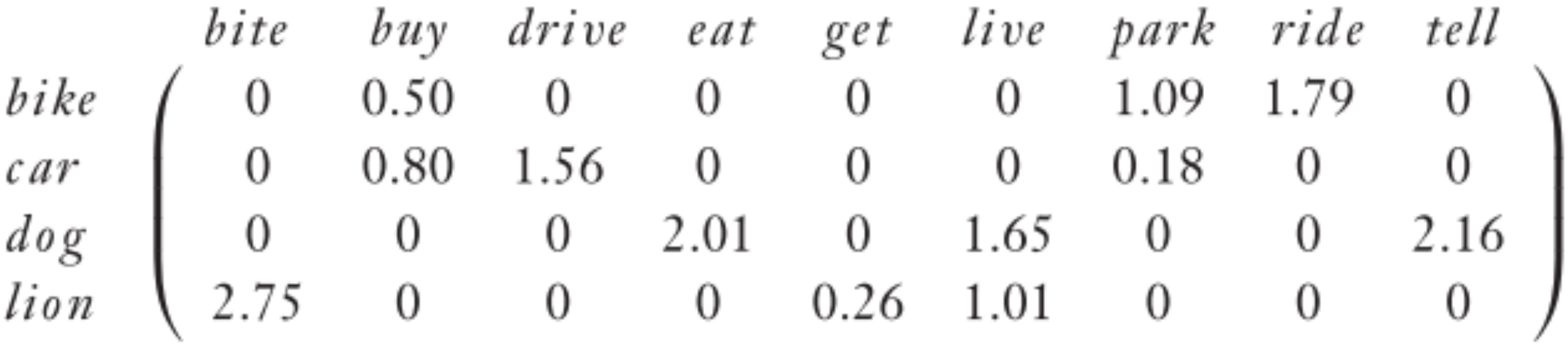
Step 2: getting significance weights

PPMI measures how much the probability of a target–context pair estimated in the training corpus is higher than the probability we should expect if the target and the context occurred independently of one another:

$$PPMI(t,c) = max\left(0, log_2 \frac{p(t,c)}{p(t)p(c)}\right)$$

Step 2: getting significance weights (cont’d)

The below-displayed matrix contains the PPMI weights computed from the raw co-occurrence presented in Step 1.

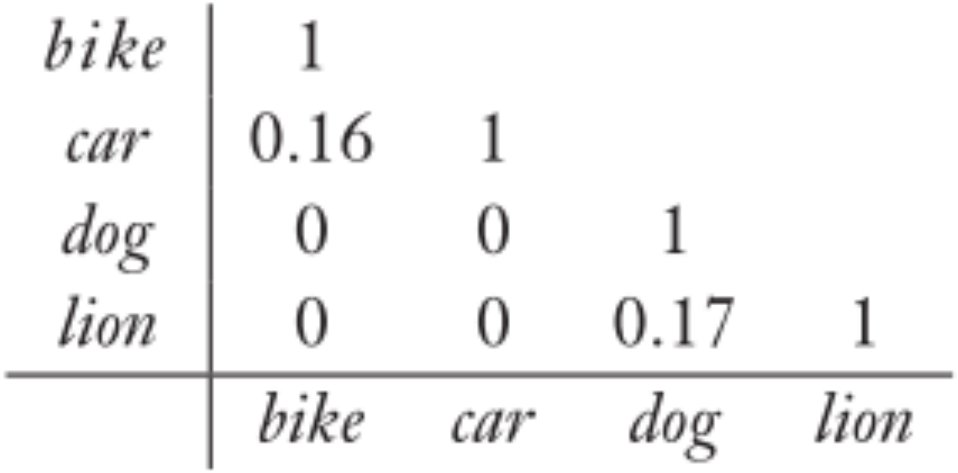


$$\begin{array}{l} \\ bike \\ car \\ dog \\ lion \end{array}
\begin{array}{c} \begin{array}{ccccccccc} bite & buy & drive & eat & get & live & park & ride & tell \end{array} \\
\begin{pmatrix} 0 & 0.50 & 0 & 0 & 0 & 0 & 1.09 & 1.79 & 0 \\ 0 & 0.80 & 1.56 & 0 & 0 & 0 & 0.18 & 0 & 0 \\ 0 & 0 & 0 & 2.01 & 0 & 1.65 & 0 & 0 & 2.16 \\ 2.75 & 0 & 0 & 0 & 0.26 & 1.01 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Step 3: assessing the distributional similarities between lexemes

The distributional similarity between two lexemes u and v is measured with the similarity between their distributional vectors u and v.

Once we have computed the pairwise distributional similarity between the targets, we can identify the k nearest neighbours of each target t, the k lexical items with the highest similarity score with t.



bike	1			
car	0.16	1		
dog	0	0	1	
lion	0	0	0.17	1
	bike	*car*	*dog*	*lion*

Step 3: assessing the distributional similarities between lexemes (cont’d)

Which similarity measure to use?

Cosine Similarity (CS) is the most popular measure of vector similarity in distributional semantics.

$$cos(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$$

CS scores belong to the interval $[-1,\ 1]$ ($[0{,}1]$ for vectors in the positive space).

CS takes value -1 for opposite vectors; 0 for orthogonal vectors; 1 for proportional vectors.

Popular distributional representations in NLP

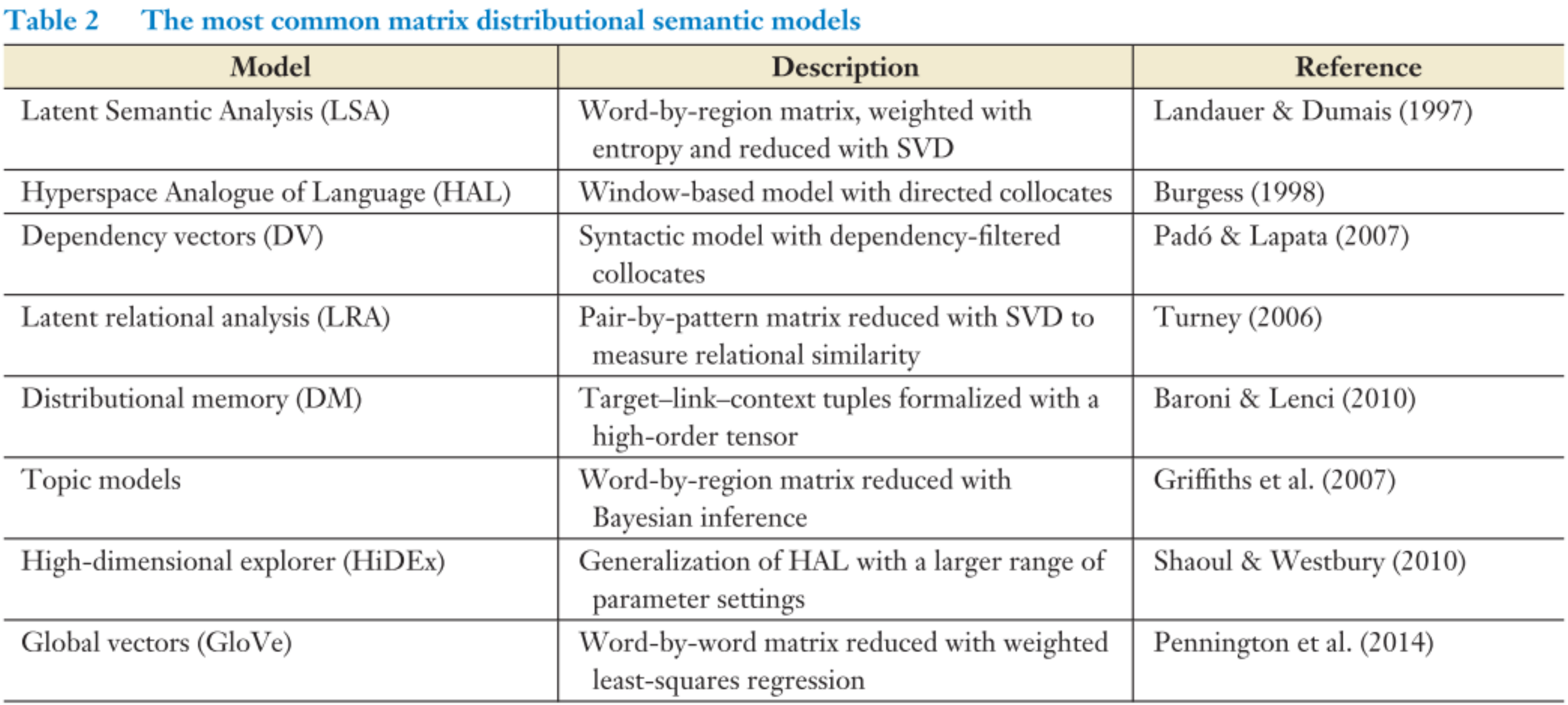


Table 2 The most common matrix distributional semantic models

Model	Description	Reference
Latent Semantic Analysis (LSA)	Word-by-region matrix, weighted with entropy and reduced with SVD	Landauer & Dumais (1997)
Hyperspace Analogue of Language (HAL)	Window-based model with directed collocates	Burgess (1998)
Dependency vectors (DV)	Syntactic model with dependency-filtered collocates	Padó & Lapata (2007)
Latent relational analysis (LRA)	Pair-by-pattern matrix reduced with SVD to measure relational similarity	Turney (2006)
Distributional memory (DM)	Target–link–context tuples formalized with a high-order tensor	Baroni & Lenci (2010)
Topic models	Word-by-region matrix reduced with Bayesian inference	Griffiths et al. (2007)
High-dimensional explorer (HiDEx)	Generalization of HAL with a larger range of parameter settings	Shaoul & Westbury (2010)
Global vectors (GloVe)	Word-by-word matrix reduced with weighted least-squares regression	Pennington et al. (2014)

Source: Lenci, 2018

Wrap-up

Main points

- Computational linguistics offers an established three-step procedure to assess the semantic similarity of two lexemes; that is, to build distributional representations.
- The simplest form of distributional representations — shown in this presentation — builds on 101 stats.
- Although more sophisticated, the alternative distributional representations such as Latent Semantic Analysis of Topic Modeling still build on the three-step procedure we have just gone through.