01-10–2024

Word embeddings: vector semantics

Word embeddings: vector semantics

- Word meaning
  - Mouse (N) - Oxford
    - a small rodent that typically has a pointed snout, relatively large ears and eyes, and a long tail.
    - a small handheld device which is moved across a mat or flat surface to move the cursor on a computer screen.
- Synonyms
  - relationship between word senses
    - couch/sofa
    - car/automobile
    - Cat/dog ?
- Similar
  - Cat/dog

Word embeddings: vector semantics (cont.)

- Semantic field
  - Hospitals
    - surgeon, scalpel, nurse, anesthetic
  - restaurants
    - waiter, menu, plate, food, chef
- Topics
- Vector semantics
  - define the meaning of a word by its distribution in language use, meaning its neighbouring words
  - idea is that two words that occur in very similar distributions (whose neighboring words are similar) have similar meanings

Word embeddings: vector semantics (cont.)

- Word embeddings

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Word embeddings

- Vectors and documents:
  - Term-document matrix

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Word embeddings (cont.)

- Word-word matrix:

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Co-occurrence vectors for four words in the Wikipedia corpus, showing six of the dimensions (hand-picked for pedagogical purposes)

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Word embeddings: similarity

- Dot-product ?
- Cosine similarity ?
- How to consider the entries of the embedding?
  - Row values?
  - Scaling?

Term frequency - inverse document frequency (TF-IDF)

- IDF: $idf_t = \log_{10}\left[\frac{N}{df_t}\right]$
- $w_{t,d} = tf_{t,d} \times idf_t$

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Term frequency - inverse document frequency (TF-IDF)

- IDF: $idf_t = \log_{10}\left[\frac{N}{df_t}\right]$
- $w_{t,d} = tf_{t,d} \times idf_t$

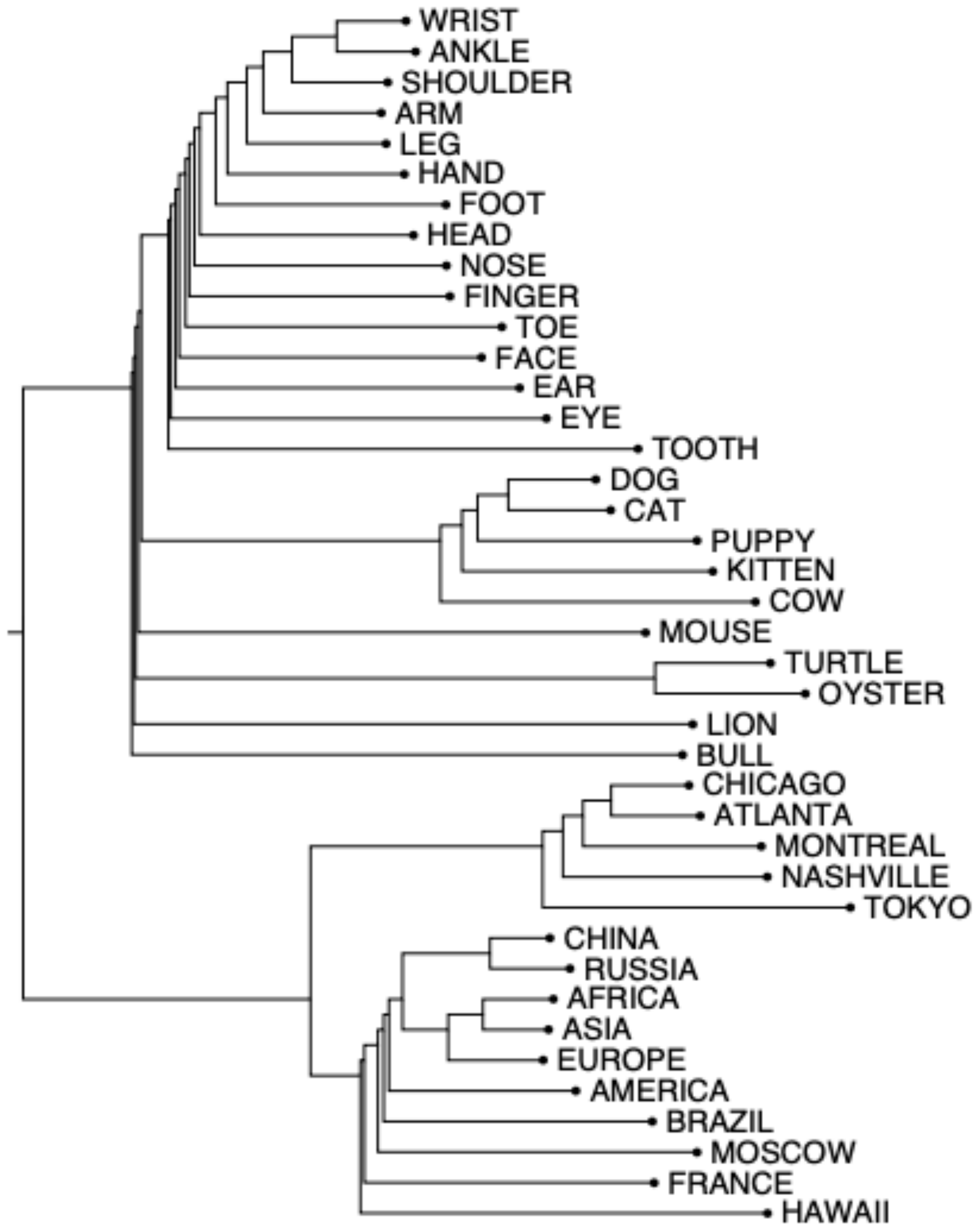
	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Visualising words

- Project the word embedding in 2-dimension
  - PCA
  - t-SNE
  - UMAP
  - …
- Dendrogram



Jurafsky & Martin “Speech and Language Processing, 3rd ed., 2023

Word vectors

- One-hot-vector: $w \in R^{|V|}$
  - What about the similarity between two words?
  - Orthogonal
  - Very sparse
- SVD on Word-word matrix:
  - Example: I enjoy flying. I like NLP. I like deep learning.

$$w^{a} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots \; w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$|V| \overset{|V|}{\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}} = |V| \overset{|V|}{\begin{bmatrix} | & | & \\ u_1 & u_2 & \cdots \\ | & | & \end{bmatrix}} |V| \overset{|V|}{\begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}} |V| \overset{|V|}{\begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ & \vdots & \end{bmatrix}}$$

$$|V| \overset{|V|}{\begin{bmatrix} & & \\ & \hat{X} & \\ & & \end{bmatrix}} = |V| \overset{k}{\begin{bmatrix} | & | & \\ u_1 & u_2 & \cdots \\ | & | & \end{bmatrix}} k \overset{k}{\begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}} k \overset{|V|}{\begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ & \vdots & \end{bmatrix}}$$

Example: Chaubard et. al. note

Neural word embeddings

Word embeddings: vector semantics (cont.)

- Word embeddings

Jurafsky & Martin "Speech and Language Processing, 3rd ed., 2023

Efficient Estimation of Word Representations in Vector Space: Word2vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

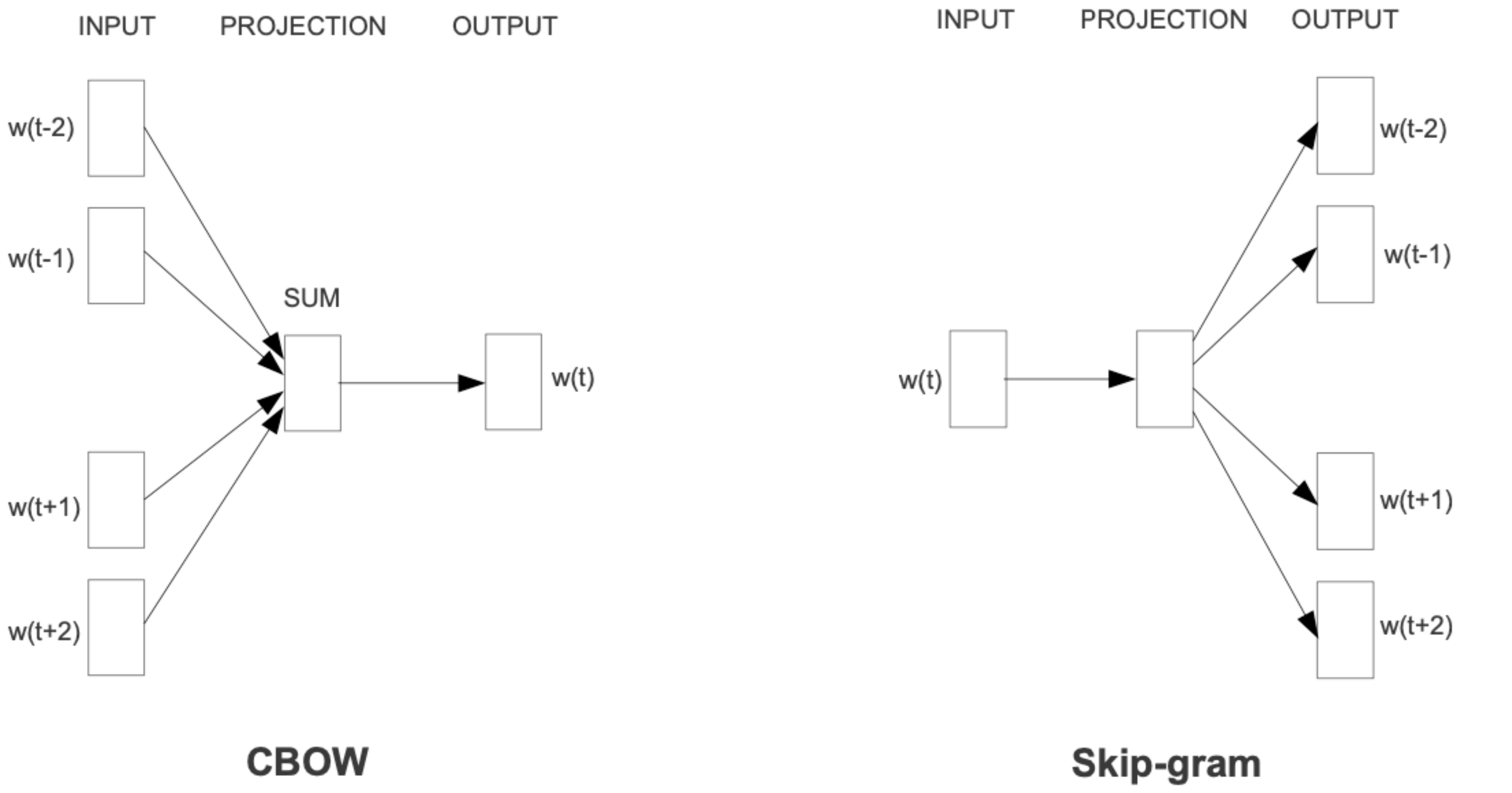
Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

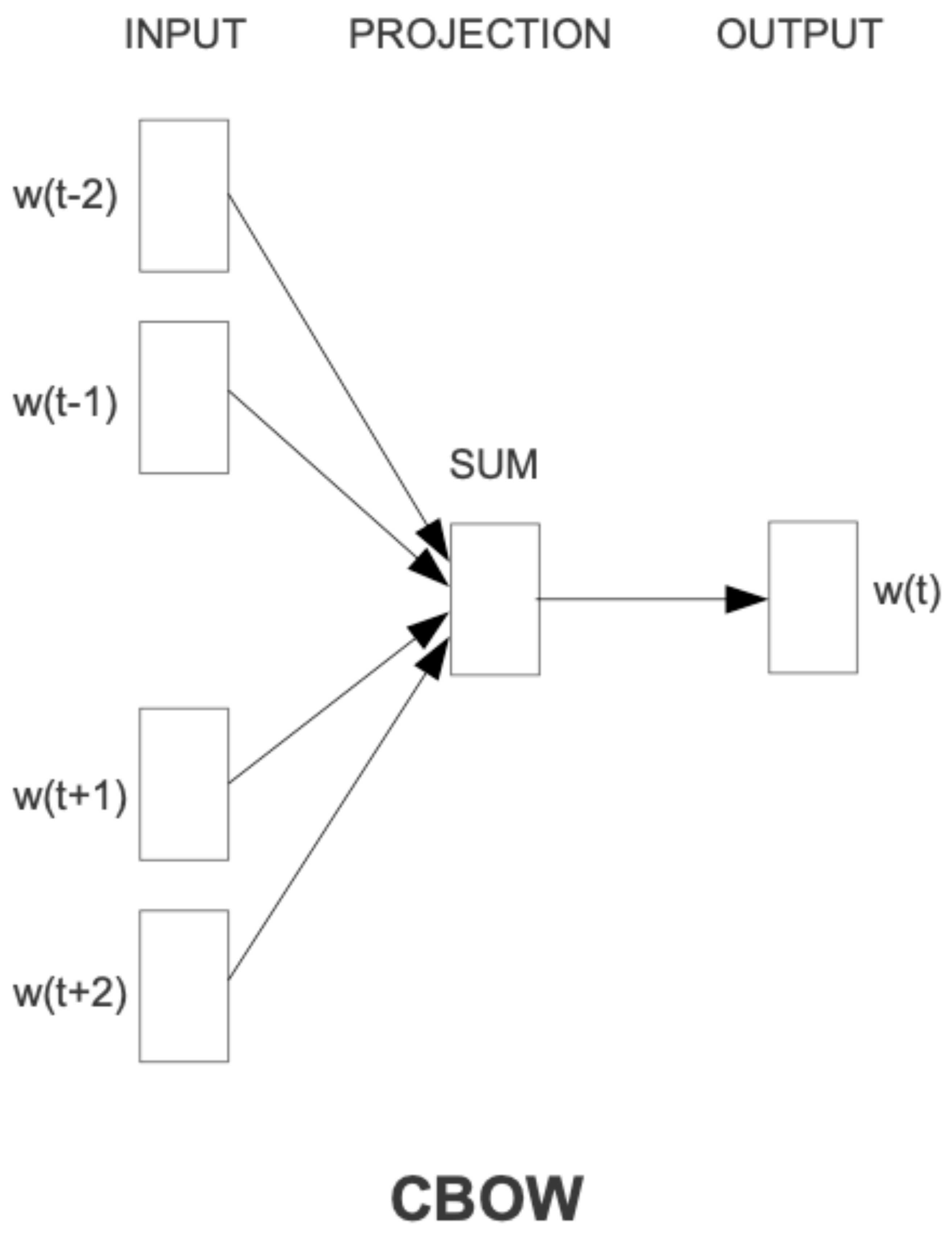
Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space. ICLR (Workshop), 2013

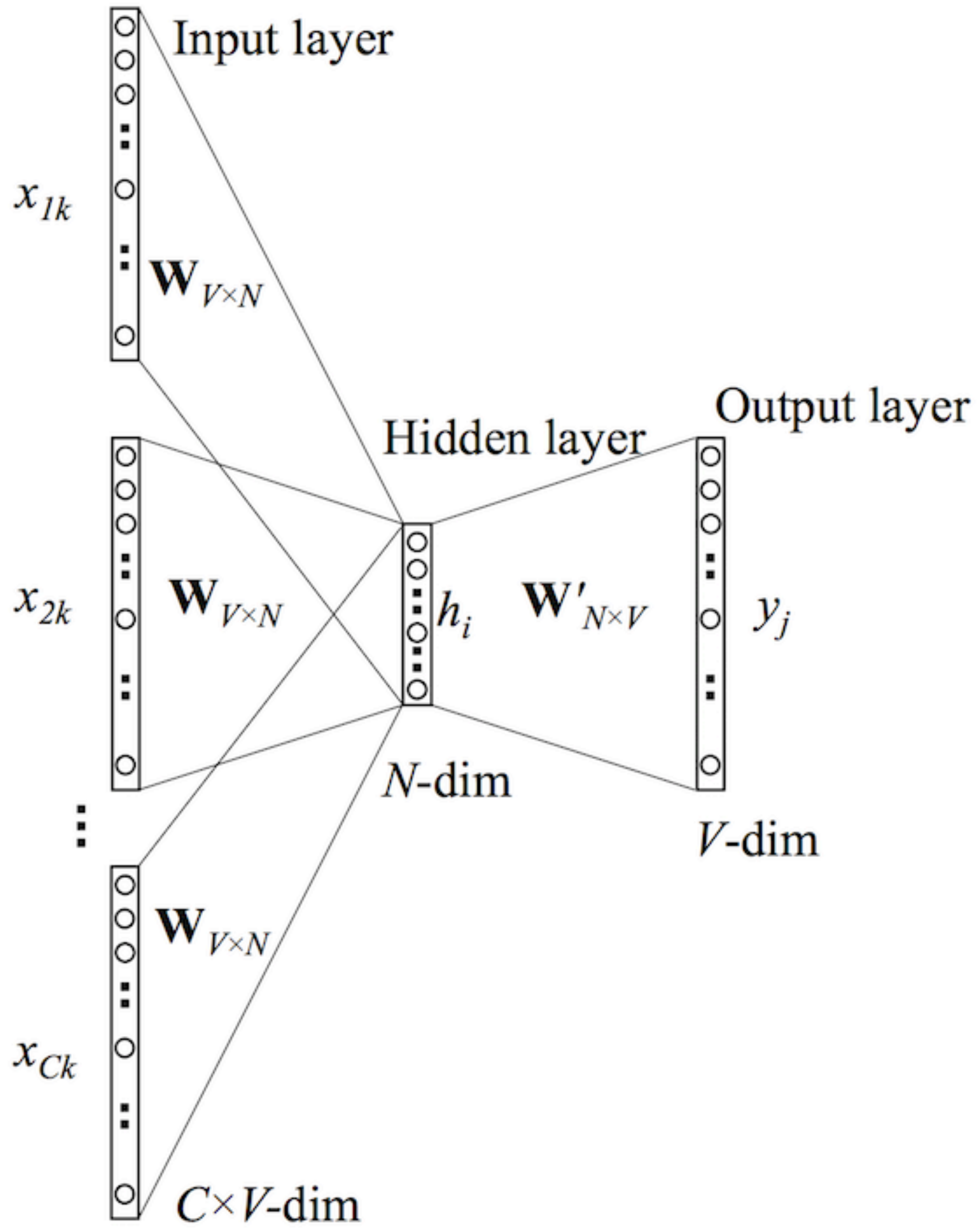
Efficient Estimation of Word Representations in Vector Space: Word2vec



Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space. ICLR (Workshop), 2013

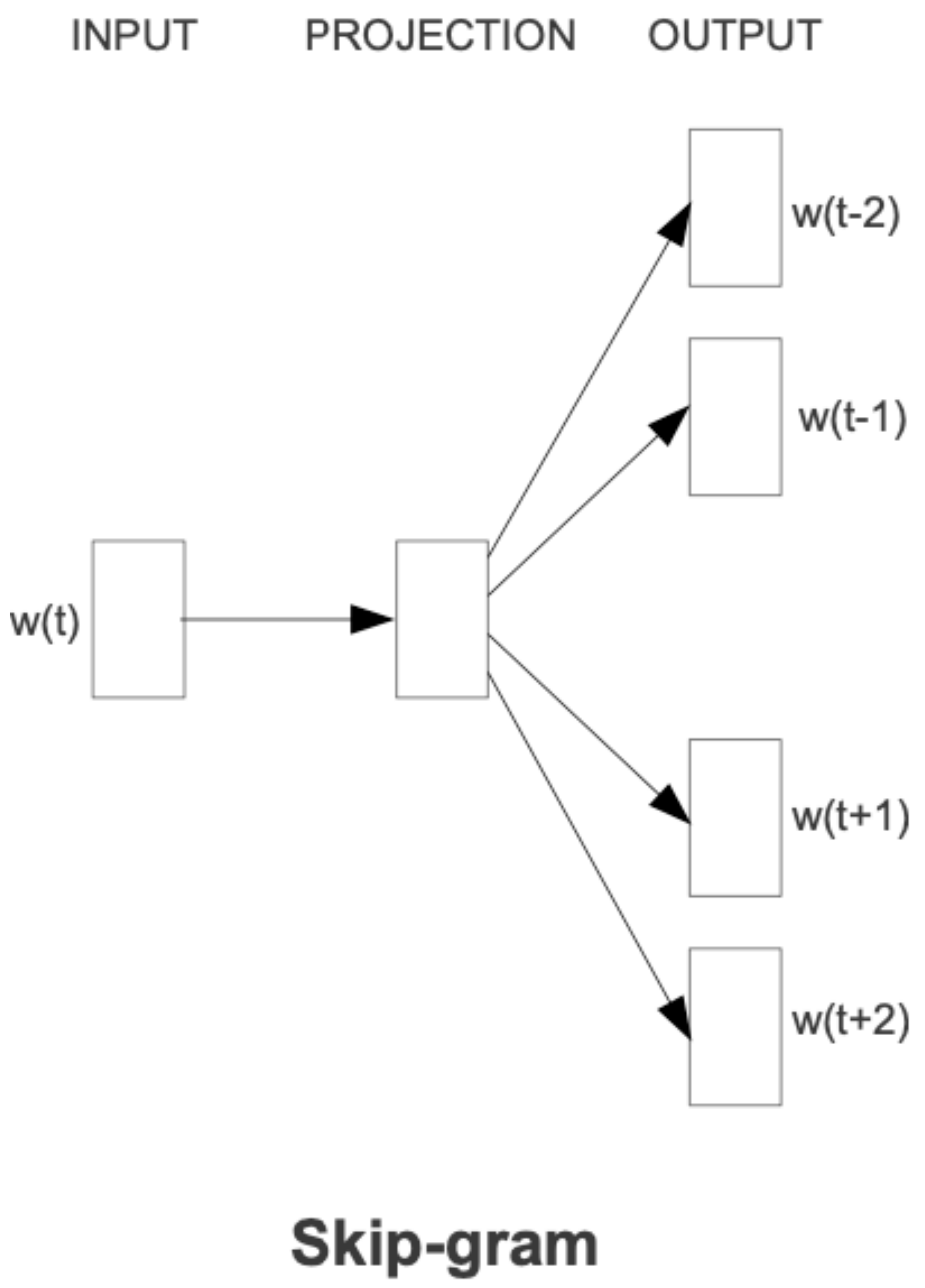
Efficient Estimation of Word Representations in Vector Space: Word2vec

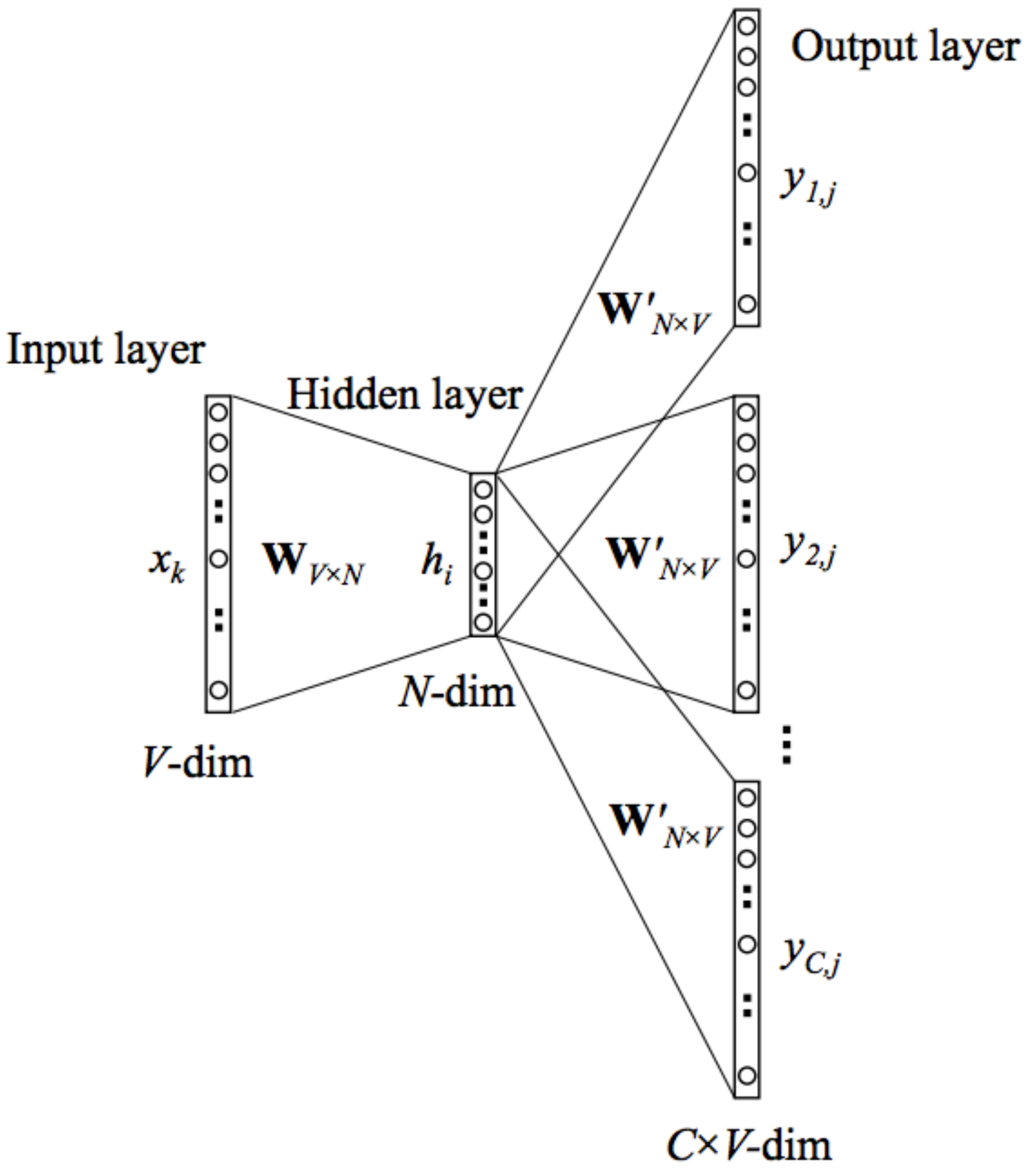




Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space. ICLR (Workshop), 2013

Efficient Estimation of Word Representations in Vector Space: Word2vec





Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space. ICLR (Workshop), 2013

What semantic properties are embedding?

- Parallelogram model
  - Man:Woman::King:Queen
  - India:France::New Delhi:Paris
  - Man:Women::Computer programmer:?
  - Man:Women::Computer programmer:Homemaker
- Embedding bias
  - Man:Women::Computer programmer:Homemaker
  - Father:Mother::Doctor:Nurse

Evaluation of word embeddings

- How can we evaluate different type of embedding?
  - Similarity
  - Datasets
    - WordSim-353 (Finkelstein et al., 2002)
    - Stanford Contextual Word Similarity (SCWS) dataset (Huang et al., 2012)
    - Word-in-Context (WiC) dataset (Pilehvar and Camacho-Collados, 2019)
    - ...
- All embedding algorithms has some issues
  - Word2Vec
    - randomness in negative samples
    - Context window size