IN5550: Neural Methods in Natural Language Processing Sub-lecture 5.2 *Count-based (explicit) vector semantic models*

Andrey Kutuzov

University of Oslo

21 February 2023

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.
- $\boldsymbol{a}$ components are mapped to all other words in V, its contexts $(b, c, d...z)$.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.
- $\boldsymbol{a}$ components are mapped to all other words in V, its contexts $(b, c, d...z)$.
- Values of components are frequencies of words co-occurrences: ab, ac, ad, etc, resulting in a square 'co-occurence matrix'.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.
- $\boldsymbol{a}$ components are mapped to all other words in V, its contexts $(b, c, d...z)$.
- Values of components are frequencies of words co-occurrences: ab, ac, ad, etc, resulting in a square 'co-occurence matrix'.
- Words are vectors or points in a multi-dimensional 'semantic space'.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.
- $\boldsymbol{a}$ components are mapped to all other words in V, its contexts $(b, c, d...z)$.
- Values of components are frequencies of words co-occurrences: ab, ac, ad, etc, resulting in a square 'co-occurence matrix'.
- Words are vectors or points in a multi-dimensional 'semantic space'.
- Contexts are axes (dimensions) in this space.

Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.
- $\boldsymbol{a}$ components are mapped to all other words in V, its contexts $(b, c, d...z)$.
- Values of components are frequencies of words co-occurrences: ab, ac, ad, etc, resulting in a square 'co-occurence matrix'.
- Words are vectors or points in a multi-dimensional 'semantic space'.
- Contexts are axes (dimensions) in this space.
- Dimensions of a word vector are interpretable: they are associated with particular context words...
- ...or other types of contexts: documents, sentences, even characters.

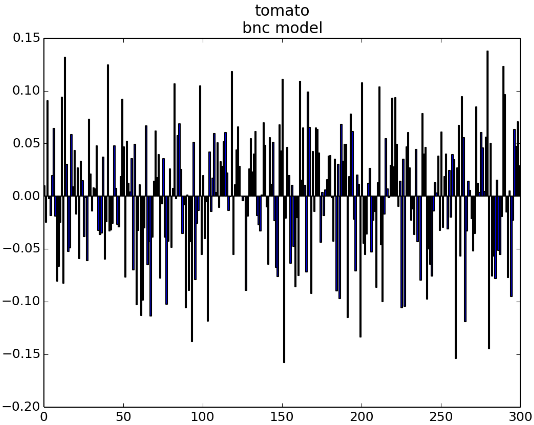
Count-based ('explicit') vector space models

Meaning is represented with vectors derived from frequency of word co-occurrences in some corpus.

- Corpus vocabulary is V.
- Each word a is represented with a vector $\boldsymbol{a} \in \mathbb{R}^{|V|}$.
- $\boldsymbol{a}$ components are mapped to all other words in V, its contexts $(b, c, d...z)$.
- Values of components are frequencies of words co-occurrences: ab, ac, ad, etc, resulting in a square 'co-occurence matrix'.
- Words are vectors or points in a multi-dimensional 'semantic space'.
- Contexts are axes (dimensions) in this space.
- Dimensions of a word vector are interpretable: they are associated with particular context words...
- ...or other types of contexts: documents, sentences, even characters.
- Interpretability is an important property of sparse representations (could be employed in the Obligatory 1!).

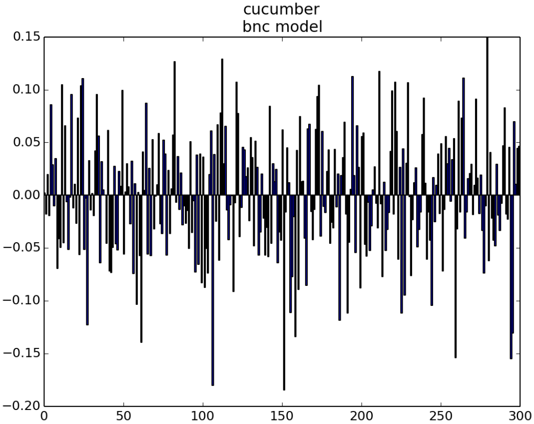
Count-based ('explicit') vector space models

300-D vector of 'tomato'



In this toy example, $|V| = 300$. Co-occurrence frequencies are normalized.

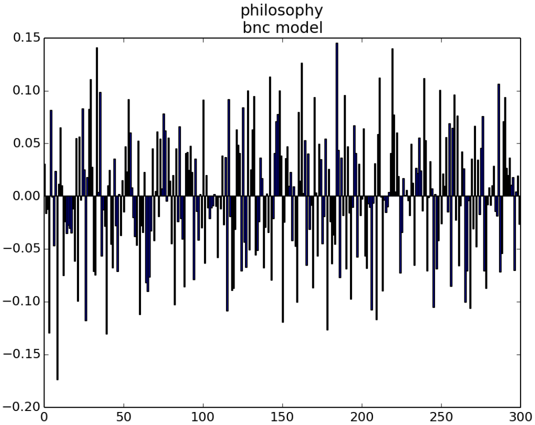
300-D vector of 'cucumber'



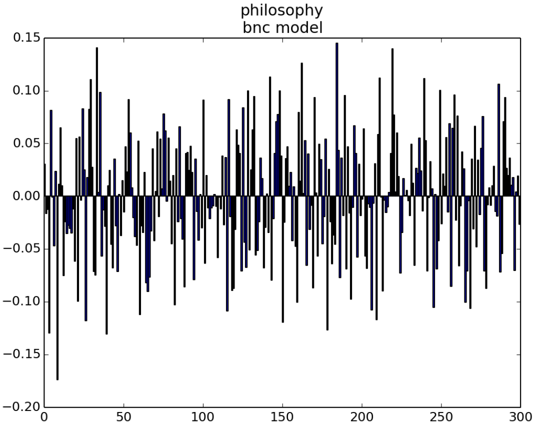
In this toy example, $|V| = 300$. Co-occurrence frequencies are normalized.

Count-based ('explicit') vector space models

300-D vector of 'philosophy'



300-D vector of 'philosophy'



Can we prove that tomatoes are more similar to cucumbers than to philosophy?

Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \tag{1}$$

(dot product of unit-normalized vectors)

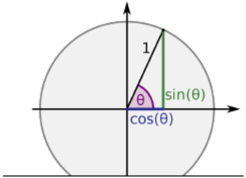
Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \tag{1}$$

(dot product of unit-normalized vectors)

- Similarity lowers as the angle between word vectors grows.



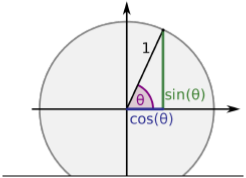
Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \tag{1}$$

(dot product of unit-normalized vectors)

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

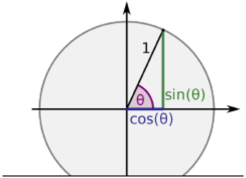


Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \tag{1}$$

(dot product of unit-normalized vectors)



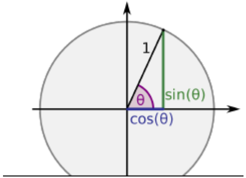
- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.
- Vectors point at the same direction: $cos = 1$
- Vectors are orthogonal: $cos = 0$

Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \tag{1}$$

(dot product of unit-normalized vectors)



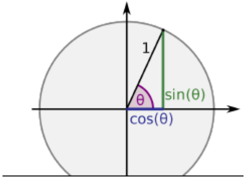
- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.
- Vectors point at the same direction: $cos = 1$
- Vectors are orthogonal: $cos = 0$
- Vectors point at the opposite directions: $cos = -1$

Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \quad (1)$$

(dot product of unit-normalized vectors)



- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.
- Vectors point at the same direction: $cos = 1$
- Vectors are orthogonal: $cos = 0$
- Vectors point at the opposite directions: $cos = -1$

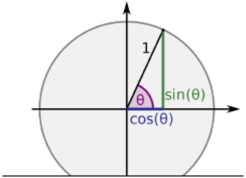
$cos(tomato, philosophy) = 0.09$

Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \tag{1}$$

(dot product of unit-normalized vectors)



- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.
- Vectors point at the same direction: $cos = 1$
- Vectors are orthogonal: $cos = 0$
- Vectors point at the opposite directions: $cos = -1$

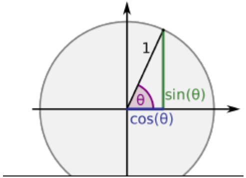
$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$

Vector similarity

Semantic similarity between words is measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

$$cos(\boldsymbol{w1}, \boldsymbol{w2}) = \frac{\boldsymbol{w1} \cdot \boldsymbol{w2}}{|\boldsymbol{w1}||\boldsymbol{w2}|} \qquad (1)$$

(dot product of unit-normalized vectors)



- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.
- Vectors point at the same direction: $cos = 1$
- Vectors are orthogonal: $cos = 0$
- Vectors point at the opposite directions: $cos = -1$

$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$
$cos(tomato, cucumber) = 0.66$

Question: why not simply use dot product?

Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Vector similarity

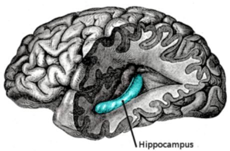
If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors

Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors



Hippocampus
(English Wikipedia co-occurrences):

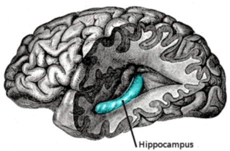
Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors

Hippocampus
(English Wikipedia co-occurrences):

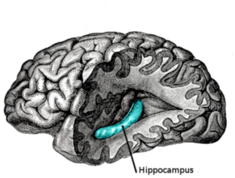
1. *cortex* 0.83



Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors



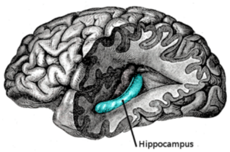
Hippocampus
(English Wikipedia co-occurrences):

1. *cortex* 0.83
2. *amygdala* 0.82

Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors



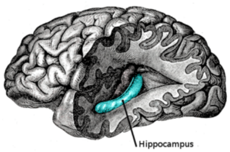
Hippocampus
(English Wikipedia co-occurrences):

1. *cortex* 0.83
2. *amygdala* 0.82
3. *cerebellum* 0.78

Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors



Hippocampus

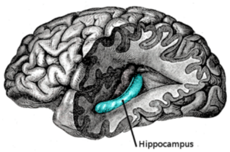
(English Wikipedia co-occurrences):

1. *cortex* 0.83
2. *amygdala* 0.82
3. *cerebellum* 0.78
4. *neuron* 0.76

Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors

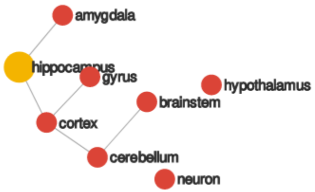


Hippocampus

(English Wikipedia co-occurrences):

1. *cortex* 0.83
2. *amygdala* 0.82
3. *cerebellum* 0.78
4. *neuron* 0.76
5. *gyrus* 0.75
6. ...

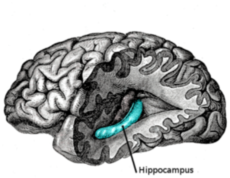
(these words have the same co-occurrences as '*hippocampus*')



Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

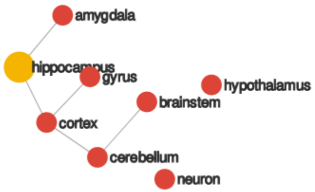
Nearest semantic associates/neighbors



Hippocampus
(English Wikipedia co-occurrences):

1. *cortex* 0.83
2. *amygdala* 0.82
3. *cerebellum* 0.78
4. *neuron* 0.76
5. *gyrus* 0.75
6. ...

(these words have the same co-occurrences as ‘*hippocampus*’)

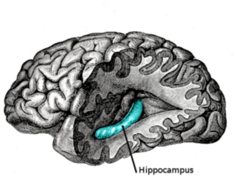


These lists themselves describe the ‘*hippocampus*’ meaning to a large extent.

Vector similarity

If one can measure similarity between words, one can rank words by similarity to a target word!

Nearest semantic associates/neighbors

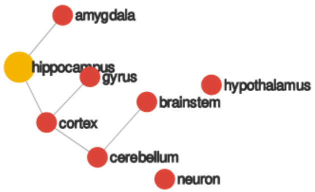


Hippocampus

(English Wikipedia co-occurrences):

1. *cortex* 0.83
2. *amygdala* 0.82
3. *cerebellum* 0.78
4. *neuron* 0.76
5. *gyrus* 0.75
6. ...

(these words have the same co-occurrences as '*hippocampus*')



These lists themselves describe the '*hippocampus*' meaning to a large extent.

Question: what do the edges in the graph denote?

Curse of dimensionality

Curse of dimensionality

- In explicit count-based models, we can end up with very high-dimensional vectors (the size of vocabulary).

Curse of dimensionality

- In explicit count-based models, we can end up with very high-dimensional vectors (the size of vocabulary).
- These vectors are very sparse.

Curse of dimensionality

- In explicit count-based models, we can end up with very high-dimensional vectors (the size of vocabulary).
- These vectors are very sparse.
- One can reduce vector sizes to some reasonable values, and still preserve meaningful relations between them.

Curse of dimensionality

- In explicit count-based models, we can end up with very high-dimensional vectors (the size of vocabulary).
- These vectors are very sparse.
- One can reduce vector sizes to some reasonable values, and still preserve meaningful relations between them.
  - e.g., by factorizing the co-occurrence matrix, using PCA or other dimensionality reduction techniques.

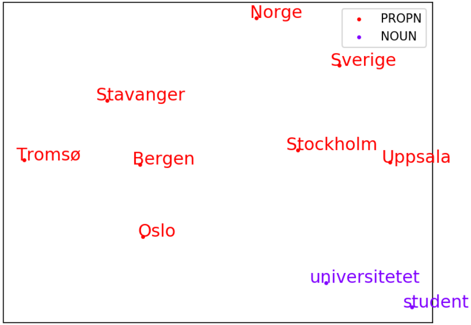
Word embeddings

Curse of dimensionality

- In explicit count-based models, we can end up with very high-dimensional vectors (the size of vocabulary).
- These vectors are very sparse.
- One can reduce vector sizes to some reasonable values, and still preserve meaningful relations between them.
  - e.g., by factorizing the co-occurrence matrix, using PCA or other dimensionality reduction techniques.
- Can even reduce to the dimensionality of 2 or 1.
- Such reduced 'implicit' vectors are usually dense and have much more rights to be called 'word embeddings'.
- NB: still nothing 'neural' or 'deep' here!

Word embeddings

An extreme case: 2-dimensional word embeddings:

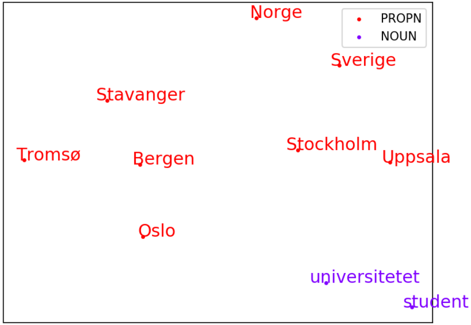


High-dimensional word vectors reduced to 2 dimensions by the t-SNE algorithm

[Van der Maaten and Hinton, 2008]

Word embeddings

An extreme case: 2-dimensional word embeddings:



High-dimensional word vectors reduced to 2 dimensions by the t-SNE algorithm

[Van der Maaten and Hinton, 2008]

Vector components (x and y) are not directly interpretable any more, of course.
An 'explicit' model turned to an 'implicit' one. Semantic information is distributed across the remaining dimensions.

Word embeddings

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;
2. scale absolute frequencies with positive point-wise mutual information (PPMI) association measure;

Word embeddings

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;
2. scale absolute frequencies with positive point-wise mutual information (PPMI) association measure;
3. factorize the matrix with singular value decomposition (SVD) or Principal Components Analysis (PCA) to reduce dimensionality to $d \ll |V|$.

Word embeddings

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;
2. scale absolute frequencies with positive point-wise mutual information (PPMI) association measure;
3. factorize the matrix with singular value decomposition (SVD) or Principal Components Analysis (PCA) to reduce dimensionality to $d \ll |V|$.
4. Semantically similar words are still represented with similar vectors...

Word embeddings

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;
2. scale absolute frequencies with positive point-wise mutual information (PPMI) association measure;
3. factorize the matrix with singular value decomposition (SVD) or Principal Components Analysis (PCA) to reduce dimensionality to $d \ll |V|$.
4. Semantically similar words are still represented with similar vectors...
5. ...but the matrix is no longer square, the number of columns is d and each row $\boldsymbol{a} \in \mathbb{R}^d$.
6. The word vectors are now dense and small: embedded in the d-dimensional space.

Word embeddings

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;
2. scale absolute frequencies with positive point-wise mutual information (PPMI) association measure;
3. factorize the matrix with singular value decomposition (SVD) or Principal Components Analysis (PCA) to reduce dimensionality to $d \ll |V|$.
4. Semantically similar words are still represented with similar vectors...
5. ...but the matrix is no longer square, the number of columns is d and each row $\boldsymbol{a} \in \mathbb{R}^d$.
6. The word vectors are now dense and small: embedded in the d-dimensional space.

For more details, see [Bullinaria and Levy, 2007] and [Goldberg, 2017].

Word embeddings

Distributional models of this kind are known as count-based: Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), etc.

How to construct a count-based model

1. compile full co-occurrence matrix on the corpus;
2. scale absolute frequencies with positive point-wise mutual information (PPMI) association measure;
3. factorize the matrix with singular value decomposition (SVD) or Principal Components Analysis (PCA) to reduce dimensionality to $d \ll |V|$.
4. Semantically similar words are still represented with similar vectors...
5. ...but the matrix is no longer square, the number of columns is d and each row $\boldsymbol{a} \in \mathbb{R}^d$.
6. The word vectors are now dense and small: embedded in the d-dimensional space.

For more details, see [Bullinaria and Levy, 2007] and [Goldberg, 2017].

But where is machine learning and neural networks? See sub-lecture 5.3!

Bullinaria, J. A. and Levy, J. P. (2007).
Extracting semantic representations from word co-occurrence statistics: A computational study.
Behavior research methods, 39(3):510–526.

Goldberg, Y. (2017).
Neural network methods for natural language processing.
Synthesis Lectures on Human Language Technologies, 10(1):1–309.

Van der Maaten, L. and Hinton, G. (2008).
Visualizing data using t-SNE.
Journal of Machine Learning Research, 9(2579-2605):85.