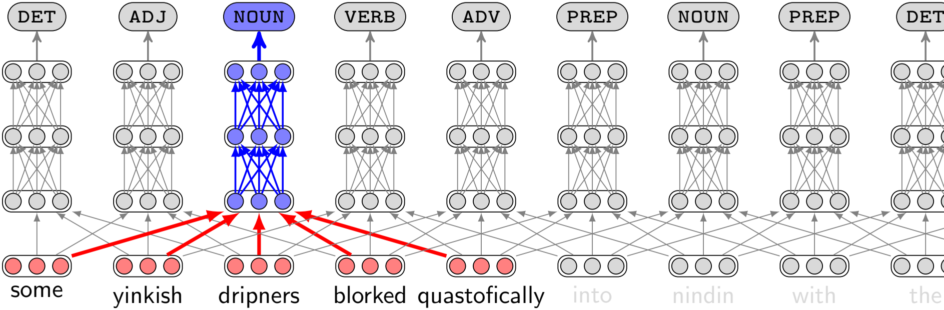


the first step: vectorizing words

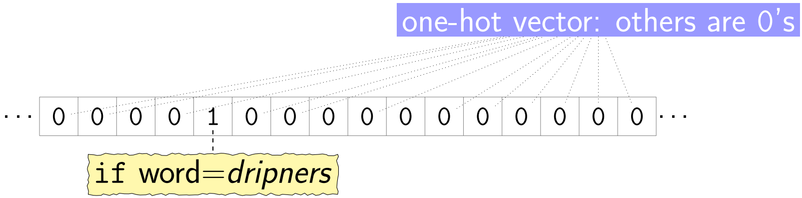
Lecture 7: Word Representations

1. Getting distributions from text
2. Count-based approaches
3. Prediction-based approaches
4. Dimension reduction

some slides are from Ann Copestake

Getting Distributions from Text

Naive representation



- The vast majority of rule-based, statistical and neural NLP systems regard words as atomic symbols:

  email, Cambridge, study

- This is a vector with one 1 and a lot of 0's

  $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$ in $\mathbb{R}^{|\text{vocabulary}|}$.

- Dimensionality is very large: 50K (Penn TreeBank), 13M (Google 1T)

The general intuition

it was authentic nindin, rather sharp and very strong

we could taste a famous local product — nindin

spending hours in the pub drinking nindin

The general intuition

it was authentic nindin, rather sharp and very strong

we could taste a famous local product — nindin

spending hours in the pub drinking nindin

- Use linguistic context to represent word and phrase meaning (partially).
- Meaning space with dimensions corresponding to elements in the context (*features*).
- Most computational techniques use vectors, or more generally tensors: aka *semantic space models*, *vector space models*, *embeddings*.

The general intuition

it was authentic nindin, rather sharp and very strong

we could taste a famous local product — nindin

spending hours in the pub drinking nindin

- Use linguistic context to represent word and phrase meaning (partially).
- Meaning space with dimensions corresponding to elements in the context (*features*).
- Most computational techniques use vectors, or more generally tensors: aka *semantic space models*, *vector space models*, *embeddings*.

Distributional representation

$$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0] \quad \triangleright\text{sparse}$$

$$\Downarrow$$

$$[0.456\ 0.193\ 5.39\ 1.235\ \text{-}93.0] \quad \triangleright\text{dense}$$

E.g. *nindin* [. . . , *pub* 0.8, *drink* 0.7, *strong* 0.4, *joke* 0.2, *mansion* 0.02, *zebra* 0.1, . . .]

The distributional hypothesis

> *You shall know a word by the company it keeps.*
>
> *the complete meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously.*
>
> *John Firth, (1957, A synopsis of linguistic theory)*

> *distributional statements can cover all of the material of a language without requiring support from other types of information.*
>
> *Zellig Harris (1954, Distributional structure)*

Distributional semantics: family of techniques for representing word meaning based on (linguistic) contexts of use.

Count-Based Approaches

Contexts

Word windows (unfiltered): n words on either side of the lexical item.

Example: $n = 2$ (5 words window)

The prime minister acknowledged the question.

minister [the 2, prime 1, acknowledged 1, question 0]

Contexts

Word windows (unfiltered): n words on either side of the lexical item.

Example: $n = 1$ (3 words window)

The prime minister acknowledged the question.

minister	[the 2, prime 1, acknowledged 1, question 0]
	[prime 1, acknowledged 1, question 0]

Contexts

Word windows (unfiltered): n words on either side of the lexical item.

Example: $n = 2$ (5 words window)

The prime minister acknowledged the question.

minister	[the 2, prime 1, acknowledged 1, question 0]
	[prime 1, acknowledged 1, question 0]
+stop list	[~~the 2~~, prime 1, acknowledged 1, question 0]
	the and *of* may be not informative

Contexts

Word windows (unfiltered): n words on either side of the lexical item.

Example: $n = 2$ (5 words window)

The prime minister acknowledged the question.

minister	[the 2, prime 1, acknowledged 1, question 0]
	[prime 1, acknowledged 1, question 0]
+stop list	[~~the 2~~, prime 1, acknowledged 1, question 0]
	the and *of* may be not informative
+lemmatization	[prime 1, acknowledge 1, question 0]

Contexts

Word windows (unfiltered): n words on either side of the lexical item.

Example: $n = 2$ (5 words window)

The prime minister acknowledged the question.

minister	[the 2, prime 1, acknowledged 1, question 0]
	[prime 1, acknowledged 1, question 0]
+stop list	[~~the 2~~, prime 1, acknowledged 1, question 0]
	the and *of* may be not informative
+lemmatization	[prime 1, acknowledge 1, question 0]

The size of windows depends on your goals

- Shorter windows ⇒ more syntactic representation
- Longer windows ⇒ more semantic representation

Problem with raw counts

Raw word frequency is not a great measure of association between words

the and *of* are very frequent, but maybe not the most discriminative

Pointwise mutual information

Information-theoretic measurement: Do events x and y co-occur more than if they were independent?

$$PMI(X,Y) = \log \frac{P(x,y)}{P(x) \cdot P(y)}$$

Positive PMI

It is not clear people are good at *unrelatedness*. So we just replace negative PMI values by 0

An example

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	
pineapple	0	0	1	0	1	
digital	2	1	0	1	0	
information	1	6	0	4	0	

An example

	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	
pineapple	0.00	0.00	0.05	0.00	0.05	
digital	0.11	0.05	0.00	0.05	0.00	
information	0.05	0.32	0.00	0.21	0.00	

An example

	computer	data	pinch	result	sugar	p(word)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58

An example

	computer	data	pinch	result	sugar	p(word)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

- Matrix: words $\times$ contexts
- f_{ij} is $\#$ of times w_i occurs in context c_j

An example

	computer	data	pinch	result	sugar	p(word)
apricot			2.25		2.25	0.11
pineapple			2.25		2.25	0.11
digital	1.66	0.00		0.00		0.21
information	0.00	0.57		0.00		0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

- Matrix: words $\times$ contexts
- f_{ij} is # of times w_i occurs in context c_j

Using syntax to define a word's context

> *The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.*
>
> *Zellig Harris (1968)*

- Two words are similar if they have similar syntactic contexts
- *duty* and *responsibility* have similar syntactic distribution:
  - **Modified by adjectives**: additional, administrative, assumed, collective, congressional, constitutional, ...
  - **Objects of verbs**: assert, assign, assume, attend to, avoid, become, breach, ...

Context based on dependency parsing (1)

`I have a brown dog`

(*have* subj *I*), (*I* subj-of *have*), (*dog* obj-of *have*), (*dog* adj-mod *brown*), (*brown* adj-mod-of *dog*), (*dog* det *a*), (*a* det-of *dog*)

The description of *cell*

COUNT(*cell*, subj-of, *absorb*)=1
COUNT(*cell*, subj-of, *adapt*)=1
COUNT(*cell*, subj-of, *behave*)=1
...
COUNT(*cell*, pobj-of, *in*)=159
COUNT(*cell*, pobj-of, *inside*)=16
COUNT(*cell*, pobj-of, *into*)=30
...

Cosine similarity

Given two target words/phrases/sentences/paragraphs/. . . , we'll need a way to measure their *similarity*.

Cosine similarity

Given two target words/phrases/sentences/paragraphs/..., we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.

Cosine similarity

Given two target words/phrases/sentences/paragraphs/. . . , we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.

Cosine similarity

Given two target words/phrases/sentences/paragraphs/. . . , we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.
- Take angle between vectors as measure of similarity.

Cosine similarity

Given two target words/phrases/sentences/paragraphs/. . . , we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.
- Take angle between vectors as measure of similarity.
    - similar angle = similar proportion of context words

Cosine similarity

Given two target words/phrases/sentences/paragraphs/. . . , we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.
- Take angle between vectors as measure of similarity.
  - similar angle = similar proportion of context words
- Cosine of angle is easy to compute.

$$\cos(u, v) = \frac{u^\top v}{||u|| \cdot ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} u_i \cdot u_i} \cdot \sqrt{\sum_{i=1}^{n} v_i \cdot v_i}}$$

Cosine similarity

Given two target words/phrases/sentences/paragraphs/..., we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.
- Take angle between vectors as measure of similarity.
  - similar angle = similar proportion of context words
- Cosine of angle is easy to compute.

$$\cos(u, v) = \frac{u^\top v}{||u|| \cdot ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} u_i \cdot u_i} \cdot \sqrt{\sum_{i=1}^{n} v_i \cdot v_i}}$$

  - $\cos = 1$ means angle is $0°$, i.e. very similar

Cosine similarity

Given two target words/phrases/sentences/paragraphs/. . . , we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.
- Take angle between vectors as measure of similarity.
  - similar angle = similar proportion of context words
- Cosine of angle is easy to compute.

$$\cos(u,v) = \frac{u^\top v}{||u|| \cdot ||v||} = \frac{\sum_{i=1}^n u_i \cdot v_i}{\sqrt{\sum_{i=1}^n u_i \cdot u_i} \cdot \sqrt{\sum_{i=1}^n v_i \cdot v_i}}$$

  - $\cos = 1$ means angle is 0°, i.e. very similar
  - $\cos = 0$ means angle is 90°, i.e. very dissimilar

Cosine similarity

Given two target words/phrases/sentences/paragraphs/..., we'll need a way to measure their *similarity*.

- Distributions are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The semantic space has dimensions which correspond to possible contexts.
- Take angle between vectors as measure of similarity.
  - similar angle = similar proportion of context words
- Cosine of angle is easy to compute.

$$\cos(u, v) = \frac{u^\top v}{||u|| \cdot ||v||} = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} u_i \cdot u_i} \cdot \sqrt{\sum_{i=1}^{n} v_i \cdot v_i}}$$

  - $\cos = 1$ means angle is 0°, i.e. very similar
  - $\cos = 0$ means angle is 90°, i.e. very dissimilar
- Many other methods to compute similarity

Context based on dependency parsing (2)

hope (N):
optimism 0.141, chance 0.137, expectation 0.136, prospect 0.126, dream 0.119, desire 0.118, fear 0.116, effort 0.111, confidence 0.109, promise 0.108

hope (V):
would like 0.158, wish 0.140, plan 0.139, say 0.137, believe 0.135, think 0.133, agree 0.130, wonder 0.130, try 0.127, decide 0.125

brief (N):
legal brief 0.139, affidavit 0.103, filing 0.098, petition 0.086, document 0.083, argument 0.083, letter 0.079, rebuttal 0.078, memo 0.077

brief (A):
lengthy 0.256, hour-long 0.191, short 0.173, extended 0.163, frequent 0.162, recent 0.158, short-lived 0.155, prolonged 0.149, week-long 0.149

Reference

Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words.

Problems

Similarity = synonymy?

- Antonyms are basically as distributionally similar as synonyms:
- Distributional similarity is not referential similarity.
- Distinguishing synonyms from antonyms is notoriously hard problem.

brief (A):
lengthy 0.256, hour-long 0.191, short 0.173, extended 0.163, frequent 0.162, recent 0.158, short-lived 0.155, prolonged 0.149, week-long 0.149, occasional 0.146

Prediction-Based Approaches

Prediction and *natural* annotations

To define a model that aims to predict between a center word w_t and context words in terms of word vectors $p(\text{context}|w_t)$ which has a loss function, e.g.,

$$J = 1 - \sum_t p(w_{t-1}|w_t)$$

We look at many positions t in a big language corpus, and try to minimize this loss.

Main idea of `word2vec`

A recent, even simpler and faster model: `word2vec`

Predict between every word and its context words!

Two algorithms

- Skip-grams (SG)

  Predict context words given target (position independent)
- Continuous Bag of Words (CBOW)

  Predict target word from bag-of-words context

Reference

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.

Skip-gram prediction (1)

Predict context words given target (position independent)

window size=2

`The course covers methods for trees, sequences and words.`

center word trees

context words *methods*, *for*, *sequences*, *and*

Predicting

$$
\begin{array}{ll}
p(w_{t-2}|w_t) & p(\text{methods}|\text{trees}) \\
p(w_{t-1}|w_t) & p(\text{for}|\text{trees}) \\
p(w_{t+1}|w_t) & p(\text{sequences}|\text{trees}) \\
p(w_{t+2}|w_t) & p(\text{and}|\text{trees})
\end{array}
$$

Skip-gram prediction (2)

Objective function: Maximize the probability of any context word given the current center word:

$$J'(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} p(w_{t+j} | w_t; \theta)$$

Negative log likelihood:

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log p(w_{t+j} | w_t; \theta)$$

Define $p(w_{t+j}|w_t)$ as:

$$p(o|c) = \frac{\exp(u_o^\top v_c)}{\sum_{w=1}^{|V|} \exp(u_w^\top v_c)}$$

Skip-gram prediction (3)

$p(w_{t+j}|w_t)$

$$p(o|c) = \frac{\exp(u_o^\top v_c)}{\sum_{w=1}^{|V|} \exp(u_w^\top v_c)}$$

Every word has **two vectors**! Makes modeling simpler!

- o is the output word index, c is the center word index
- v_c and u_o are *center* and *outside* vectors of indices c and o

Softmax function: Map from $\mathbb{R}^{|V|}$ to a probability distribution.

- $u_w^\top v_c$ is bigger if u_w and v_c are more similar
- Iterate over the vocabulary.

Skip-gram prediction (4)

All parameters in this model can be viewed as one long vector:

u_{a}, u_{aardvark}, ..., u_{zymurgy}, v_{a}, v_{aardvark}, ..., v_{zymurgy}

- u and v: d-dimensional vectors
- θ: $\mathbb{R}^{2d|V|}$

Optimize these parameters

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log \frac{\exp(u_{w_{t+j}}^\top v_{w_t})}{\sum_{w=1}^{|V|} \exp(u_w^\top v_{w_t})}$$

$|V|$ is too large $\rightarrow$ Negative sampling

Count-based vs predictive

Count-based approaches

- Sparse vector representations
- Fast training
- Efficient usage of statistics
- Primarily used to capture word similarity

Prediction-based approaches

- Dense vector representations
- Scales with corpus size
- Inefficient usage of statistics
- Generate improved performance on other tasks
- Can capture complex patterns beyond word similarity

Sparse vs dense vectors

PMI vectors are

- long (length $|V| = 20,000$ to $50,000$)
- sparse (most elements are zero)

Predictive: learn vectors which are

- short (length 200–1000)
- dense (most elements are non-zero)

Why dense vectors?

- Short vectors may be easier to use as features in machine learning
- Dense vectors may generalize better than storing explicit counts

Dimension Reduction

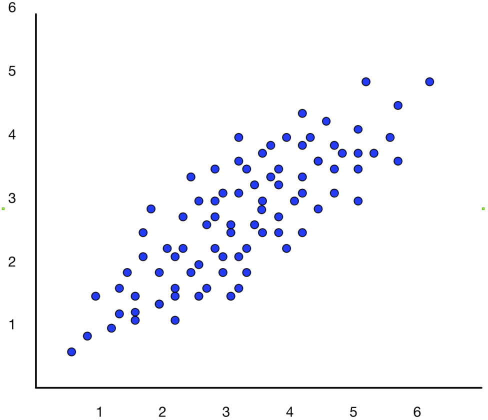
Dimension reduction

Idea

Approximate an N-dimensional dataset using fewer dimensions

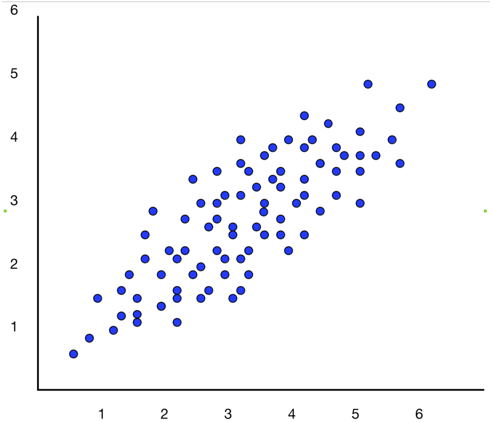
- By first rotating the axes into a new space
- In which the highest order dimension captures the most variance in the original dataset
- And the next dimension captures the next most variance, etc.
- Many such (related) methods: principle components analysis, Factor Analysis, SVD, etc.

Principal Component Analysis



Dimension reduction: vector $x \Rightarrow \text{FUNCTION} \Rightarrow$ vector z

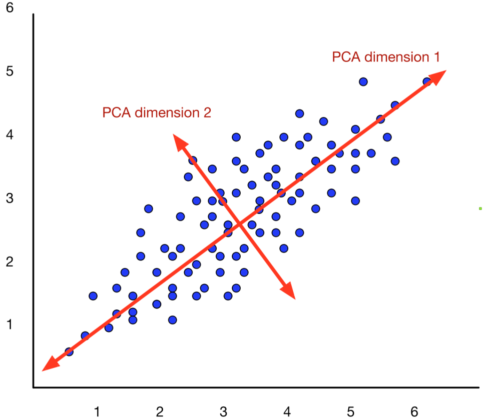
Principal Component Analysis



Dimension reduction: vector $x \Rightarrow \text{FUNCTION} \Rightarrow$ vector z

$$\boxed{\text{PCA: } z = Wx}$$

Principal Component Analysis



- Fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component.
- If some axis of the ellipsoid is small, then the variance along that axis is also small.

PCA (2)

$$\boxed{\text{PCA: } z = Wx}$$

Reduce to 1-D:

$$z_1 = w_1^\top x$$

We want the variance of z_1 as large as possible,

$$Var(z_1) = \frac{1}{N} \sum_{z_{1,i}} (z_{1,i} - \bar{z_1})^2$$

subject to $||w_1|| = 1$

PCA (3)

Reduce to 2-D:

$$z_1 = w_1^\top x$$
$$z_2 = w_2^\top x$$

We want the variance of z_1 as large as possible,

$$Var(z_1) = \frac{1}{N} \sum_{z_{1,i}} (z_{1,i} - \bar{z_1})^2$$

We also want the variance of z_2 as large as possible,

$$Var(z_2) = \frac{1}{N} \sum_{z_{2,i}} (z_{2,i} - \bar{z_2})^2$$

subject to $w_1^\top w_2 = 0$

PCA (4)

$$
\begin{aligned}
Var(z_1) &= \frac{1}{N}\sum_{z_{1,i}}(z_{1,i} - \bar{z_1})^2 \\
&= \frac{1}{N}\sum_{x_i}(w_1^\top x_i - w_1^\top \bar{x})^2 \\
&= \frac{1}{N}\sum_{x_i}(w_1^\top (x_i - \bar{x}))^2 \\
&= \frac{1}{N}\sum_{x_i}(w_1^\top (x_i - \bar{x})(x_i - \bar{x})^\top w_1) \\
&= w_1^\top (\frac{1}{N}\sum_{x_i}(x_i - \bar{x})(x_i - \bar{x})^\top) w_1 \\
&= w_1^\top S w_1
\end{aligned}
$$

PCA (5)

$$\begin{aligned} \text{max.} \quad & w_1^\top S w_1 \\ \text{s.t.} \quad & w_1^\top w_1 = 1 \end{aligned} \tag{1}$$

S is symmetric positive-semidefinite (non-negative eigenvalues)
Using Lagrange multiplier

$$\mathcal{L}(w_1, \alpha) = w_1^\top S w_1 - \alpha(w_1^\top w - 1)$$

We get

$$S w_1 - \alpha w_1 = 0$$

w_1: eigenvector

$$w_1^\top S w_1 = \alpha w^\top w_1$$

Choose the maximum largest eigenvalue λ_1

PCA (6)

$$
\begin{aligned}
\text{max.} \quad & w_2^\top S w_2 \\
\text{s.t.} \quad & w_2^\top w_2 = 1,\ w_2^\top w_1 = 0
\end{aligned} \tag{2}
$$

Using Lagrange multiplier

$$\mathcal{L}'(w_2, \alpha, \beta) = w_2^\top S w_2 - \alpha(w_2^\top w_2 - 1) - \beta(w_2^\top w_1)$$

calculate the gradient,

$$S w_2 - \alpha w_2 - \beta w_1 = 0$$

$$w_1^\top S w_2 - \alpha w_1^\top w_2 - \beta w_1^\top w_1 = 0$$

$$w_1^\top S w_2 = w_2^\top S^\top w_1 = w_2^\top S w_1 = w_2^\top \lambda_1 w_1 = 0$$

So $\beta = 0$. And again, we get $S w_2 = \alpha w_2$. w_2 is another eigenvector.

Dimensionality reduction

Why dense vectors?

- Short vectors may be easier to use as features in machine learning
- Dense vectors may generalize better than storing explicit counts

Dense embeddings sometimes work better than sparse PMI matrices at tasks like word similarity

- Denoising: low-order dimensions may represent unimportant information
- Truncation may help the models generalize better to unseen data.

Reading

- Ann's node
- D Jurafsky and J Martin. *Speech and Language Processing* Chapter 6. `web.stanford.edu/~jurafsky/slp3/6.pdf`

* Essence of linear algebra `www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab`