

Recitation 3: Distributional Semantics

From “meaning as logic” to “meaning as distribution”

Bohan Jiang

Fall 2026

1. **Why distributional semantics?**

From logic to context; the distributional hypothesis

2. **Building word vectors from context**

Co-occurrence matrices; what the window size measures

3. **Weighting and comparing**

Why raw counts fail; PMI and PPMI; cosine similarity

4. **What distribution cannot see**

Antonyms, and the answer to our opening question

Why do we switch to counting words?

Recap: logic, written as linear algebra

Fix a domain and enumerate it: $D_e = \{\text{Alice}, \text{Bob}, \text{Carol}, \text{Fido}\}$.

Type	λ -term	Vector object	Example
e	a	one-hot $\mathbf{e}_i$	Alice = $(1, 0, 0, 0)$
$\langle e, t \rangle$	$\lambda x. \text{sleep}(x)$	indicator vector	sleep = $(0, 0, 1, 1)$
t	$\text{sleep}(a)$	scalar in $\{0, 1\}$	$\mathbf{p} \cdot \mathbf{e}_a$

- Application is an inner product: $\mathbf{sleep} \cdot \mathbf{e}_{\text{Carol}} = 1$, so *Carol sleeps* is true.
- Everything is **exact**: the numbers are read off a model $M = \langle D_e, I \rangle$ that someone stipulated.

Practice: how similar is a dog to a cat?

Practice

Extend the domain: $D_e = \{\text{Alice}, \text{Bob}, \text{Carol}, \text{Fido}, \text{Tom}\}$, where Fido is a dog and Tom is a cat.

$student = \{A, B, C\}$, $dog = \{\text{Fido}\}$, $cat = \{\text{Tom}\}$.

1. Compute $\|\mathbf{e}_{\text{Alice}} - \mathbf{e}_{\text{Bob}}\|$ and $\|\mathbf{e}_{\text{Alice}} - \mathbf{e}_{\text{Fido}}\|$.
2. Compute $\cos(\mathbf{dog}, \mathbf{cat})$ and $\cos(\mathbf{dog}, \mathbf{student})$ from the indicator vectors.
3. What do (1) and (2) say about “a dog is more like a cat than like a student”?

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

Answer: similarity is absent by construction

Answer

1. Both are $\sqrt{1^2 + (-1)^2} = \sqrt{2}$. **Every pair of individuals is equivalent.**
2. **dog** = (0, 0, 0, 1, 0), **cat** = (0, 0, 0, 0, 1), **student** = (1, 1, 1, 0, 0).
Both dot products are 0, so **both cosines are 0**.
3. The model cannot express it. dog-cat looks exactly like dog-student.

Answer: similarity is absent by construction

Answer

1. Both are $\sqrt{1^2 + (-1)^2} = \sqrt{2}$. **Every pair of individuals is equivalent.**
 2. **dog** = (0, 0, 0, 1, 0), **cat** = (0, 0, 0, 0, 1), **student** = (1, 1, 1, 0, 0).
Both dot products are 0, so **both cosines are 0**.
 3. The model cannot express it. dog-cat looks exactly like dog-student.
- One-hot vectors are orthogonal: they carry *identity* and nothing else.
 - Predicate vectors only measure *extension overlap*, and dogs and cats share no members.

This is not a bug. It is a geometric fact about the encoding.

The move: keep the machinery, change the index set

A vector is a function from an *index set* to numbers.

	Logic	Distribution
A vector is indexed by	entities	contexts
A vector represents	a predicate	a word
The numbers come from	a stipulated model M	counting a corpus
$\mathbf{v} \cdot \mathbf{w}$ means	membership, exactly	similarity, approximately
Two words are	orthogonal unless identical	at any angle

The move: keep the machinery, change the index set

A vector is a function from an *index set* to numbers.

	Logic	Distribution
A vector is indexed by	entities	contexts
A vector represents	a predicate	a word
The numbers come from	a stipulated model M	counting a corpus
$\mathbf{v} \cdot \mathbf{w}$ means	membership, exactly	similarity, approximately
Two words are	orthogonal unless identical	at any angle

Take the model away. You only have a corpus.

Where do the numbers come from?

The distributional hypothesis

The distributional hypothesis

- **Harris (1954):** “Difference of meaning *correlates* with difference of distribution.”
A correlation, not an identity. That hedge matters later.
- **Firth (1957):** “You shall know a word by the company it keeps.”

Working hypothesis: three separate claims

1. **Contexts are informative.** The environments of a word carry signal about it.
2. **The signal can be summarized.** One fixed-length vector per word is enough.
3. **Similarity transfers.** Geometric closeness tracks closeness in meaning.

(1) is close to uncontroversial. (3) is where the trouble is, and we return to it at the end.

The classic demonstration: *tezgüino*

1. A bottle of *tezgüino* is on the table.
2. Everyone likes *tezgüino*.
3. *Tezgüino* makes you drunk.
4. We make *tezgüino* out of corn.
5. Don't have any more *tezgüino* before you drive.
6. The *tezgüino* had gone sour.

The classic demonstration: *tezgüino*

1. A bottle of *tezgüino* is on the table.
2. Everyone likes *tezgüino*.
3. *Tezgüino* makes you drunk.
4. We make *tezgüino* out of corn.
5. Don't have any more *tezgüino* before you drive.
6. The *tezgüino* had gone sour.

Cue	Inference
a bottle of	liquid, mass noun
makes you drunk	alcoholic
out of corn	made from maize
had gone sour	fermentable

The classic demonstration: *tezgüino*

1. A bottle of *tezgüino* is on the table.
2. Everyone likes *tezgüino*.
3. *Tezgüino* makes you drunk.
4. We make *tezgüino* out of corn.
5. Don't have any more *tezgüino* before you drive.
6. The *tezgüino* had gone sour.

Cue	Inference
a bottle of	liquid, mass noun
makes you drunk	alcoholic
out of corn	made from maize
had gone sour	fermentable

No single sentence is decisive: sentence 1 alone fits *ink*. Meaning comes from the **overlap of many weak cues**.

Practice: two unknown words

Practice

Guess each word, and name the cues that gave it away.

X: a slice of X on the burger · grate some X over the pasta · aged X from the cellar · the X had melted

Y: Y the dough for ten minutes · she Y-ed it until smooth · don't over-Y bread dough · Y with the heel of your hand

Answer: two unknown words

Answer

	X	Y
Type	mass noun, food	transitive verb, manual action
Cues	<i>slice, grate, aged, melted</i>	<i>dough, until smooth, heel of your hand</i>
Neighbours	cheese: cheddar, parmesan, gouda	knead, work, fold

Answer: two unknown words

Answer

	X	Y
Type	mass noun, food	transitive verb, manual action
Cues	<i>slice, grate, aged, melted</i>	<i>dough, until smooth, heel of your hand</i>
Neighbours	cheese: cheddar, parmesan, gouda	knead, work, fold

- The method works for verbs as well as nouns.
- What you **could not** recover: cheddar vs. parmesan, *knead* vs. *work*.
Near-identical distributions give near-identical vectors.

Building a co-occurrence matrix

Two kinds of matrix

Term-document

cell (w, d) = count of w in document d .

	D1	D2	D3	D4
ship	14	0	21	1
sail	9	0	17	0
proof	0	23	0	19

Columns $\rightarrow$ retrieval.

Rows $\rightarrow$ lexical semantics.

Term-term

cell (w, c) = how often c appears within $\pm k$ tokens of w .

they drank the hot coffee in the morning
└──────────────────┘
window $k = 2$

Row *coffee*: the +2, hot +1, in +1.

Slide the window over every token and add up the counts. **That is the entire algorithm.**

Practice: build a row by hand

Practice

Target word **dog**, window $k = 2$, stopwords kept.

1. we feed the **dog** every morning
2. the old **dog** has soft fur
3. wash the **dog** in the tub

- (a) Write the row for **dog**.
- (b) Which entry is largest? Is it informative?
- (c) Redo it with $k = 3$. What gets added?

Answer: build a row by hand

Answer

	Left 2	Right 2	Increments
1	feed, the	every, morning	feed, the, every, morning +1
2	the, old	has, soft	the, old, has, soft +1
3	wash, the	in, the	wash, in +1; the +2

- (a) **the** = 4, and 1 each for feed, every, morning, old, has, soft, wash, in.
- (b) **the**, which tells us nothing about dogs. **(Remember this for later.)**
- (c) $k = 3$ adds **we, fur, tub**.

Answer: build a row by hand

Answer

	Left 2	Right 2	Increments
1	feed, the	every, morning	feed, the, every, morning +1
2	the, old	has, soft	the, old, has, soft +1
3	wash, the	in, the	wash, in +1; the +2

- (a) **the** = 4, and 1 each for feed, every, morning, old, has, soft, wash, in.
- (b) **the**, which tells us nothing about dogs. **(Remember this for later.)**
- (c) $k = 3$ adds **we, fur, tub**.

fur, arguably the best cue, was 3 tokens away. **The window size decides what you see.**

Context choice decides the kind of similarity

Window size changes what you measure

Window (k)	Similarity you get	Neighbours of <i>dog</i>
1 to 2	paradigmatic (can substitute)	cat, horse, rabbit, puppy
5 to 10	mixed	cat, pet, bark, leash
whole document	topical	vet, kennel, walk, owner

Similarity

Same slot: the words can substitute for each other.

car / automobile, doctor / nurse

Relatedness

Same scene: the words show up together.

car / tire, doctor / hospital

A narrow window gives you similarity, a wide window gives you relatedness. **There is no neutral choice.**

Practice: which similarity?

Practice

(a) Label each pair *similar*, *related only*, or *both*:

surgeon / dentist surgeon / scalpel coffee / tea

coffee / mug teacher / student dog / wolf

(b) Predict the top neighbours of **guitar** with a $k = 2$ window and with whole-document contexts.

Answer: which similarity?

Answer

(a) **Similar (and related):** surgeon / dentist, coffee / tea, dog / wolf.

Related only: surgeon / scalpel, coffee / mug, teacher / student.

(b) ± 2 : **violin, piano, bass**, the things you can “play the ___”.

Document: **chord, concert, band, amp**.

Answer: which similarity?

Answer

(a) **Similar (and related)**: surgeon / dentist, coffee / tea, dog / wolf.

Related only: surgeon / scalpel, coffee / mug, teacher / student.

(b) ± 2 : **violin, piano, bass**, the things you can “play the ___”.

Document: **chord, concert, band, amp**.

- *teacher / student* is the trap: they appear in the same sentences but play different roles.
- Neither answer to (b) is wrong. **The context type chooses which relation you measure.**

Why raw counts fail

Our running example

A term-term matrix, window ± 4 , $N = 1000$.

	the	fur	vet	engine	road	wash	row sum
dog	200	25	15	0	2	8	250
cat	160	25	10	0	0	5	200
car	240	0	0	30	20	10	300
truck	200	0	0	20	28	2	250
col sum	800	50	25	50	50	25	1000

By eye there are two clusters: {dog, cat} with {fur, vet}, and {car, truck} with {engine, road}.

Our running example

A term-term matrix, window ± 4 , $N = 1000$.

	the	fur	vet	engine	road	wash	row sum
dog	200	25	15	0	2	8	250
cat	160	25	10	0	0	5	200
car	240	0	0	30	20	10	300
truck	200	0	0	20	28	2	250
col sum	800	50	25	50	50	25	1000

By eye there are two clusters: {dog, cat} with {fur, vet}, and {car, truck} with {engine, road}.

Cosine on the raw rows:

pair	dog/cat	car/truck	dog/car	dog/truck	cat/car	cat/truck
raw cosine	0.999	0.998	0.979	0.976	0.975	0.972

Practice: find the culprit

Practice

The ranking is right: the two correct pairs score highest.

But **every** pair is between 0.97 and 1.00, so the margin is only 0.02.

Which column causes this, and why?

	the	fur	vet	engine	road	wash
col sum	800	50	25	50	50	25

Answer: find the culprit

Answer

the: 800 of the 1000 counts (80%).

Every row has a huge component along the same axis, so the angle between any two rows is small.

The principle

Frequency is not informativeness. A context that occurs with everything distinguishes nothing.

We want to know not *how often* w and c co-occur, but *how much more often than chance*.

PMI and PPMI

Pointwise mutual information

$$\text{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)} \quad P(w, c) = \frac{\#(w, c)}{N}, \quad P(w) = \frac{\text{row sum}}{N}, \quad P(c) = \frac{\text{col sum}}{N}$$

Read it as **observed / expected if independent**.

$\text{PMI} > 0$	observed > expected	c is attracted to w
$\text{PMI} = 0$	observed = expected	c is independent of w
$\text{PMI} < 0$	observed < expected	c avoids w

Shortcut for hand computation:

$$\frac{P(w, c)}{P(w)P(c)} = \frac{\#(w, c) \cdot N}{\text{row sum} \cdot \text{col sum}}$$

Units are bits: $\text{PMI} = 1$ means c appears with w twice as often as chance would predict.

$$\text{PPMI}(w, c) = \max(\text{PMI}(w, c), 0)$$

Why clip? Negative PMI is the troublesome half:

- **Unbounded:** a zero count gives $\log_2 0 = -\infty$.
- **Fragile:** to show that a pair occurs *less* than chance you need huge amounts of data.
- **Dense:** keeping the negatives fills the matrix; clipping keeps it sparse.

What it costs

“independent” and “strongly repelled” now look the same.

What it buys

A sparse, non-negative, finite matrix, and empirically better neighbours.

Practice: PMI by hand

Practice

Using the running matrix ($N = 1000$), compute PMI and PPMI for

- (a) (dog, fur)
- (b) (dog, the)
- (c) (car, fur)
- (d) (dog, road)

$$\text{Tip: obs/exp} = \frac{\#(w, c) \cdot N}{\text{row} \cdot \text{col}}$$

	the	fur	...	road	row
dog	200	25		2	250
cat	160	25		0	200
car	240	0		20	300
truck	200	0		28	250
col	800	50		50	1000

Answer: PMI by hand

Answer

cell	count	obs/exp	PMI	PPMI
(dog, fur)	25	$\frac{25 \cdot 1000}{250 \cdot 50} = 2$	1	1
(dog, the)	200	$\frac{200 \cdot 1000}{250 \cdot 800} = 1$	0	0
(car, fur)	0	0	$-\infty$	0
(dog, road)	2	$\frac{2 \cdot 1000}{250 \cdot 50} = 0.16$	-2.64	0

Answer: PMI by hand

Answer

cell	count	obs/exp	PMI	PPMI
(dog, fur)	25	$\frac{25 \cdot 1000}{250 \cdot 50} = 2$	1	1
(dog, the)	200	$\frac{200 \cdot 1000}{250 \cdot 800} = 1$	0	0
(car, fur)	0	0	$-\infty$	0
(dog, road)	2	$\frac{2 \cdot 1000}{250 \cdot 50} = 0.16$	-2.64	0

- Look at (b): **the largest count in the matrix gets PMI = 0**. *the* occurs with *dog* exactly as often as it occurs with everything else.
- Stopword removal falls out of the formula. It is not a preprocessing hack.

The full PPMI matrix

	the	fur	vet	engine	road	wash
dog	0	1.00	1.26	0	0	0.36
cat	0	1.32	1.00	0	0	0
car	0	0	0	1.00	0.42	0.42
truck	0	0	0	0.68	1.16	0

- The *the* column is gone, with no stopwords list.
- The two clusters now sit on (almost) **separate coordinates**.
- Only *wash* is shared across the clusters (you wash dogs *and* cars).
- No supervision, no tuning, one formula.

PMI's own bias: rare contexts

Add one more column, **groomer**: dog 2, cat 1, others 0 ($N = 1003$).

Context of <i>dog</i>	count	PMI	PMI, $\alpha = 0.75$
groomer	2	1.41 (top context!)	$-0.31 \rightarrow 0$
vet	15	1.26	0.30
fur	25	0.99	0.29

- $P(c)$ is in the denominator, so a **rare context inflates the ratio**. Two tokens beat 25.
- Fix: **context distribution smoothing**. Raise the context counts to a power $\alpha < 1$:

$$P_{\alpha}(c) = \frac{\#(c)^{\alpha}}{\sum_{c'} \#(c')^{\alpha}}, \quad \alpha = 0.75$$

- This raises $P(c)$ for rare contexts, so their PMI shrinks and the spurious winner is clipped away.

Cosine, not Euclidean

How to compare two rows

Measure	Formula	Sensitive to
Euclidean	$\ \mathbf{v} - \mathbf{w}\ $	length and direction
Dot product	$\mathbf{v} \cdot \mathbf{w}$	length and direction
Cosine	$\frac{\mathbf{v} \cdot \mathbf{w}}{\ \mathbf{v}\ \ \mathbf{w}\ }$	direction only

The principle

In a co-occurrence matrix, **vector length encodes corpus frequency**, and **direction encodes the contextual profile**.

We want the profile, so we divide out the length.

Practice

- (a) Suppose we had twice as much dog text, so **dog** $\rightarrow$ 2 **dog**. Give the Euclidean distance and the cosine between **dog** and 2 **dog** (raw counts, $\|\mathbf{dog}\| \approx 202.3$).
- (b) Compute $\cos(\mathbf{dog}, \mathbf{cat})$ on the PPMI rows:

$$\mathbf{dog} = (0, 1, 1.263, 0, 0, 0.356), \quad \mathbf{cat} = (0, 1.322, 1, 0, 0, 0)$$

- (c) The *margin* is the lowest correct-pair cosine minus the highest wrong-pair cosine. Compare raw counts with PPMI.

Answer: cosine (a) and (b)

Answer

(a) $\|\mathbf{v} - 2\mathbf{v}\| = \|\mathbf{v}\| \approx \mathbf{202.3}$ (huge), but $\cos(\mathbf{v}, 2\mathbf{v}) = \frac{2\mathbf{v} \cdot \mathbf{v}}{\|\mathbf{v}\| \cdot 2\|\mathbf{v}\|} = \mathbf{1}$ (identical).

(b) Only two coordinates are nonzero in both:

$$\mathbf{dog} \cdot \mathbf{cat} = (1)(1.322) + (1.263)(1) = 2.585$$

$$\|\mathbf{dog}\| = \sqrt{1 + 1.595 + 0.127} = 1.650, \quad \|\mathbf{cat}\| = \sqrt{1.748 + 1} = 1.658$$

$$\cos(\mathbf{dog}, \mathbf{cat}) = \frac{2.585}{1.650 \times 1.658} = \frac{2.585}{2.735} = \mathbf{0.945}$$

Euclidean distance measures *how often* a word was seen. Sparsity makes cosine cheap: two products, not six.

Answer: cosine (c), before and after

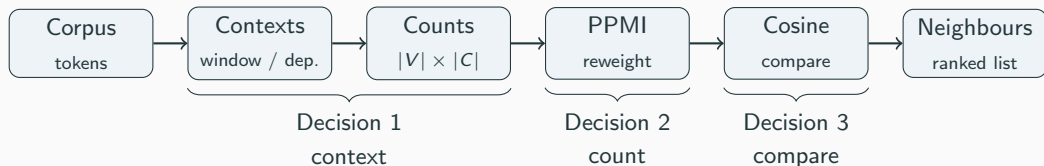
Answer

Pair	Raw counts	PPMI
dog / cat	0.999	0.945
car / truck	0.998	0.743
dog / car	0.979	0.077
other wrong pairs	0.97–0.98	0.000
Margin	0.02	0.67

- Same corpus, same contexts, same similarity function. **Only the weighting changed.**
- Why is dog / car 0.077 and not 0? They share *wash*.

In a count-based model, the weighting is not preprocessing. It is the model.

The method, end to end



- **Decision 1** decides *which* similarity you get.
- **Decision 2** decides whether you get *any* signal at all.
- **Decision 3** has a clear right answer: cosine.

Principled failures

Where the hypothesis breaks: antonyms

Context	Fits
the ___ water	hot, cold
too ___ to drink	hot, cold
the ___ tap	hot, cold
it got ___ overnight	hot, cold

Antonyms substitute each other perfectly (high similarity), but the meaning is far away (low meaning)

Why this is not a bug

1. Antonyms agree on *everything*: the same nouns, the same syntax, the same topics.
2. They differ along **one** dimension only: the value on the scale.
3. Context counts measure exactly the shared properties in (1).

So antonyms are distributionally *maximally* similar. **The method is working correctly**; the difference is simply not in the data.

Same for *buy / sell, rise / fall, above / below, always / never*.

Practice

- (a) Show that $\text{PMI}(w, c) = \text{PMI}(c, w)$. What kind of meaning relation does this symmetry rule out?
- (b) Predict the nearest distributional neighbour of **buy**, of **always**, and of **rise**. Is each one a synonym?

Answer: principled failures

Answer

(a) $P(w, c) = P(c, w)$, and the denominator $P(w)P(c)$ is a product, which commutes. So PMI is symmetric, and so is cosine.

Entailment is directional: *poodle* $\Rightarrow$ *dog*, but not *dog* $\Rightarrow$ *poodle*. No symmetric score can model it.

(b) *sell*, *never*, *fall*: all antonyms. **None of them is a synonym.**

Back to the opening question

Why are we switching from logic to counting words?

	Logic	Distribution
Similarity	absent	graded ✓
Learned from text	no	yes ✓
Covers the whole lexicon	no	yes ✓
Truth conditions	yes ✓	no
Entailment	yes ✓	no (cosine is symmetric)
Negation, quantifiers	yes ✓	invisible

The columns are close to complementary. **We are not moving away from logic. We are adding a second source of information, one that fails in different places.**

What's next

- **Dependency contexts**

Use typed relations as contexts, e.g. $\langle \text{brew}, \text{dobj}^{-1} \rangle$, instead of raw adjacency. This gives sharper paradigmatic neighbours, at the cost of needing a parser.

- **Other weightings**

tf-idf for term-document matrices; **shifted PPMI** $= \max(\text{PMI} - \log k, 0)$ turns out to be what word2vec implicitly factorizes.

- **The matrix is enormous**

$|V| = 50,000$ gives 2.5×10^9 cells. PPMI makes it sparse, but sparsity is also blindness: *espresso* and *cappuccino* may share no context, so their cosine is exactly 0.

Next session: fifty thousand dimensions down to three hundred, by counting, factorizing, and predicting.

Questions?