

The Distributional Hypothesis

Co-occurrence, Weighting, and What Counting Can and Cannot Buy

James Pustejovsky

Brandeis University
Department of Computer Science

CS 135: Computational Semantics

Fall 2026

Where Part I Left Us

Six sessions of model-theoretic semantics produced a machine that does something remarkable and something disappointing.

What it does

- Assigns truth conditions, not paraphrases
- Composes: the meaning of the whole from the parts and the rule
- Licenses inference, with proof
- Predicts syntactic distribution from semantic properties (NPIs)

What it does not do

- Know that dog and cat are more alike than dog and theorem
- Say anything at all about a word it has no lexical entry for
- Degrade gracefully
- Get learned from text

Session 6B made the second column precise: under the exact encoding, entities are *one-hot*, hence pairwise orthogonal, hence equidistant. Similarity is not hard to compute in that model — it is **absent by construction**.

Part II: The Plan

Session	Topic	The question it answers
7 (today)	Distributional hypothesis	Where do the numbers come from?
8	Reduction and prediction	Why 300 dimensions instead of 50,000?
9	Contextual embeddings	Why should a word have <i>one</i> vector?
10	The composition problem	What is the vector for a <i>phrase</i> ?

Session 10 is the hinge of the course. Everything in Part II is built so that when we get there, the difficulty is visible rather than asserted.

Today is the foundation: a hypothesis from 1954, an algorithm with three free parameters, and a very careful account of what the resulting geometry does and does not encode.

The Change of Question

Part I asked:

What are the truth conditions of this sentence?

Part II asks something that is not a weaker version of that. It is a *different* question:

What distinguishes this word from every other word?

These are related but not reducible to one another. The first is about the world; the second is about the lexicon. A theory can answer either without answering the other.

Keep this distinction live

Much confusion in the literature — and in press coverage of language models — comes from treating a good answer to the second question as though it were an answer to the first.

Distributional Structure, WORD 10(2–3)

“Difference of meaning correlates with difference of distribution.”

Harris was not doing semantics. He was a structural linguist pursuing a methodological program: describe a language using only **observable** properties of utterances, with no appeal to speaker intuition about meaning.

The *distribution* of an element, for Harris, is the set of environments in which it occurs — phonemes in the environment of other phonemes, morphemes in the environment of other morphemes, words in the environment of other words.

His claim is a **correlation** claim, and he states it as such. He does not say distribution *is* meaning. Read the paper — the hedging is careful, and it has mostly been dropped by those who cite it.

What Harris Actually Claimed

Harris says	Often reported as
Difference in meaning <i>correlates with</i> difference in distribution	Meaning <i>is</i> distribution
Distribution is a <i>discovery procedure</i> for structure	Distribution is a <i>definition</i> of content
The correlation is graded and imperfect	The correlation is the theory
Applies within a fixed corpus and register	Applies universally

The strong reading is what got implemented. The weak reading is what was argued for.

This matters for us practically, not just historically: several of the well-known failure modes of distributional models (we will see them at the end of today) are failures of the **strong** reading only. The weak reading predicts them.

A Synopsis of Linguistic Theory 1930–1955

“You shall know a word by the company it keeps.”

The most-quoted sentence in computational lexical semantics, and worth reading in context: Firth’s project was **collocation** — the observation that words have habitual partners that are not predictable from their meanings alone.

Attested	Odd, though synonymous
<i>strong tea</i>	? <i>powerful tea</i>
<i>powerful engine</i>	? <i>strong engine</i>
<i>heavy rain</i>	? <i>weighty rain</i>

Collocation is distributional information that is **not** semantic in the truth-conditional sense. It is a real fact about the language that Part I’s machinery has no place to put.

And Wittgenstein, With a Caveat

Philosophical Investigations (1953), §43

“For a large class of cases . . . the meaning of a word is its use in the language.”

This gets cited as a philosophical warrant for distributional semantics. Two cautions:

- ① “Use” for Wittgenstein means **use in a form of life** — the whole practice of asking, commanding, joking, promising. It does not mean “co-occurrence in a text corpus.”
- ② The qualification “for a large class of cases” is doing real work in the original, and is almost always dropped in the citation.

The honest statement: text co-occurrence is a *narrow, mechanizable proxy* for use. That it works as well as it does is a striking empirical fact, not a consequence of Wittgenstein.

The Distributional Hypothesis, Stated

The working hypothesis for Part II

Words that occur in similar contexts tend to have similar meanings. Therefore: represent a word by a summary of the contexts it occurs in, and measure meaning similarity by similarity of those summaries.

Three things are being asserted, and they are independent:

- ① **Contexts are informative.** The environments of a word carry signal about it.
- ② **The signal is summarizable.** A single fixed-length object suffices; we need not keep the contexts themselves.
- ③ **Similarity transfers.** Geometric closeness of summaries tracks semantic closeness of words.

(1) is close to uncontroversial. (2) is an engineering bet — Session 9 revisits it. (3) is where the trouble is, and we will spend the last quarter of today on it.

The Classic Demonstration

Here are six sentences containing a word you do not know.

- 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

Here are six sentences containing a word you do not know.

- 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.

You now know, with some confidence, that it is an alcoholic beverage, probably fermented rather than distilled, made from maize, and drinkable in quantity.

No definition was given. No entry was consulted. The contexts did all of it.

The example is Nida's (1975), made famous in this form by Lin (1998).

Why That Worked

Look at what each sentence actually contributed.

Sentence	Context element	Inference licensed
1	<i>a bottle of ____</i>	mass noun, liquid, containerizable
2	<i>likes ____</i>	consumable or enjoyable
3	<i>____ makes you drunk</i>	alcoholic
4	<i>make ____ out of corn</i>	manufactured, from maize
5	<i>don't have more ____</i>	dosed, consumed repeatedly
6	<i>____ had gone sour</i>	perishable, fermentable

Crucially, no single context is decisive. Sentence 1 alone is consistent with ink. The inference comes from the **intersection** of many weak constraints.

That is the shape of the whole method: many individually uninformative observations, aggregated.

Practice: Two Unknown Words

Word A: *The A was flush against the rafter. · Drive the A in at an angle. · a box of galvanized As · bent the A with a bad swing*

Word B: *She was elected B in March. · the outgoing B · B of the finance committee · report to the B by Friday*

Practice: Two Unknown Words

Word A: *The A was flush against the rafter. · Drive the A in at an angle. · a box of galvanized As · bent the A with a bad swing*

Word B: *She was elected B in March. · the outgoing B · B of the finance committee · report to the B by Friday*

What you inferred, and how:

	A	B
Type	concrete count noun, small artifact	human role
Cue	pluralizes, boxed, driven, bent	elected, outgoing, reports
Nearest neighbors	nail, screw, brad, spike	chair, president, director

Note what you did *not* recover: whether A is a nail or a screw. Those two have near-identical distributions in this sample. **Hold that thought** — it returns at the end of the lecture as the central limitation.

What the Hypothesis Does Not Say

Not claimed	Why it matters
That distribution determines <i>reference</i>	Nothing in a corpus fixes which individual Alice denotes
That distribution determines <i>truth</i>	Corpora contain false sentences, at scale
That similar means <i>synonymous</i>	hot/cold are distributionally near-identical
That the relation is <i>symmetric in strength</i>	Entailment is directional; cosine is not
That it works for <i>function words</i>	the, of, not occur everywhere

Every row here is a place where Part I's machinery is strictly better. That is why this is a course about both, and why Part IV asks whether a single algebra can host both.

We are not replacing the logic. We are adding a second source of information, with different strengths and a different failure profile.

Three Decisions

To turn the hypothesis into something you can run, you must fix three things. Every distributional model — including, in a disguised form, every language model — makes these three choices.

Decision	Options we will consider
1 What is a context ?	document, paragraph, $\pm k$ window, dependency relation
2 What is a count ?	raw frequency, PMI, PPMI, tf-idf, smoothed variants
3 What is a comparison ?	Euclidean, Jaccard, cosine, Kullback–Leibler

These are not tuning knobs of equal weight. Decision 1 determines *what kind* of similarity you get. Decision 2 determines whether you get any signal at all. Decision 3 is the most constrained — there is a clear right answer, and a clear reason for it.

Decision 1, First Option: The Term–Document Matrix

The oldest form, from information retrieval. Rows are words, columns are documents, cell (w, d) is the count of w in d .

	Doc 1	Doc 2	Doc 3	Doc 4
ship	14	0	21	1
sail	9	0	17	0
proof	0	23	0	19
lemma	0	18	0	24

Two ways to read this matrix, and both are used:

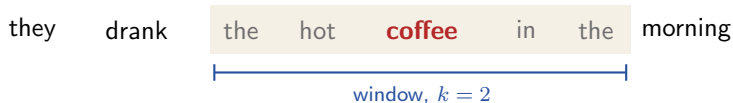
- **By column** — each document is a vector over words. This is retrieval: match a query vector against document vectors.
- **By row** — each word is a vector over documents. This is lexical semantics: ship and sail are similar because they share documents.

The row reading, with dimensionality reduction applied, is Latent Semantic Analysis (Deerwester et al. 1990); we return to it in Session 8.

Decision 1, Second Option: The Term–Term Matrix

Rows are words; columns are also words. Cell (w, c) counts how often c occurred **in the context of** w .

This requires saying what “in the context of” means. The standard choice: a symmetric window of $\pm k$ tokens.



This single position contributes four increments to row coffee: to columns the (twice), hot, and in.

Slide the window over every token of the corpus and accumulate. That is the entire algorithm. It is one pass, and it parallelizes trivially — which is exactly why it scaled in the 1990s on hardware that could not have trained anything.

Building a Row by Hand

Corpus fragment (three sentences), target coffee, window $k = 2$, stopwords kept:

- 1 they drank the hot **coffee** in the morning
- 2 pour the **coffee** into a cup
- 3 she brews **coffee** every morning

Occurrence	Left 2	Right 2	Increments
1	the, hot	in, the	the+2, hot+1, in+1
2	pour, the	into, a	pour+1, the+1, into+1, a+1
3	she, brews	every, morning	she+1, brews+1, every+1, morning+1

Row coffee so far: the=3, hot=1, in=1, pour=1, into=1, a=1, she=1, brews=1, every=1, morning=1.

Already the largest entry is the, and it tells us nothing. That observation is the whole of Decision 2.

Window Size Changes What You Measure

This is the single most consequential knob, and it is routinely set without comment.

Window	Similarity you get	Top neighbors of dog
± 1 to ± 2	paradigmatic / substitutional	cat, horse, rabbit, puppy
± 5 to ± 10	mixed	cat, pet, bark, leash
whole document	topical / associative	veterinarian, kennel, walk, owner

Why: a narrow window sees mostly *syntactic* neighbors, so words that can occupy the same slot look alike. A wide window sees the *topic*, so words that belong to the same activity look alike.

The terms are Saussure's, and they are worth having

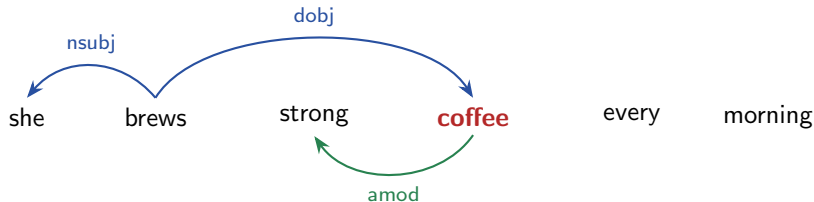
Paradigmatic: relation between items that can substitute for one another (dog/cat).

Syntagmatic: relation between items that combine (dog/bark).

Window size is the dial between them. Choose it according to the task, not the default.

Beyond Windows: Syntactic Contexts

A window is a crude proxy for “related to.” With a parser, you can use the dependency relation itself as the context.



Contexts for coffee become *typed*: $\langle \text{brew}, \text{dobj}^{-1} \rangle$ and $\langle \text{strong}, \text{amod} \rangle$, instead of untyped adjacency.

	Window	Dependency
Range	bounded by k	unbounded, but structurally local
Precision	includes accidental adjacency	only genuine relations
Neighbors produced	topical	sharply paradigmatic
Cost	one pass	requires a parser, and inherits its errors

Decision 1: Summary

Context type	Dimensions	Captures	Cost
Document	$ D $	topic	one pass
Paragraph	$ D $ large	topic, narrower	one pass
Window $\pm k$, large k	$ V $	topic + association	one pass
Window $\pm k$, small k	$ V $	substitutability	one pass
Dependency	$ V \times R $	typed relations	parsing

There is no neutral choice here. “Similar” is not one relation, and the context definition picks which one you are measuring.

Keep this in mind when you read a paper reporting that some model “captures semantic similarity.” The first question is always: *which* similarity, and what was the context window?

For the rest of today we fix the choice: term–term, ± 4 window, so that the arithmetic is concrete.

The Counts Are Wrong

Here is a small term–term matrix, raw counts, from a corpus of technology and beverage text. Window ± 4 .

	the	cup	brew	screen	battery	morning
coffee	300	48	32	2	1	38
tea	240	44	28	1	0	24
laptop	260	3	0	52	41	6
tablet	220	1	0	44	39	4

Your intuition says: two clusters, {coffee, tea} and {laptop, tablet}, and the columns cup/brew versus screen/battery separate them cleanly.

Your intuition is reading the small numbers. The arithmetic will read the large ones.

Raw Counts: The Disaster

Compute cosine similarity on the raw rows. (Cosine is defined properly in a few slides; for now, the standard angle-based similarity, 1 for identical direction.)

Pair	Cosine on raw counts
coffee / tea	0.999
laptop / tablet	1.000
coffee / laptop	0.950
coffee / tablet	0.946
tea / laptop	0.947

Everything is similar to everything, at around 0.95. The measure has almost no discriminative power.

And note: the two *correct* pairs do score highest. The problem is not the ranking — it is that the **margin** is 0.05, which will not survive noise.

The Diagnosis

One column is responsible.

Column	Total count	Share of all counts
the	1020	71.4%
screen	99	6.9%
cup	96	6.7%
battery	81	5.7%
morning	72	5.0%
brew	60	4.2%

Every row has a large the entry, so every pair of rows shares a large component in the the direction, and the angle between any two rows is small.

The principle

Frequency is not informativeness. A context occurring with *everything* distinguishes *nothing*. We want not how often w and c co-occur, but how much **more often** than chance.

Pointwise Mutual Information

Definition

$$\text{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w) P(c)}$$

where, with $\#(w, c)$ the cell count and N the sum of all cells:

$$P(w, c) = \frac{\#(w, c)}{N}, \quad P(w) = \frac{\sum_{c'} \#(w, c')}{N}, \quad P(c) = \frac{\sum_{w'} \#(w', c)}{N}.$$

Read the formula as a ratio of observed to expected. The denominator $P(w)P(c)$ is what the joint probability *would* be if w and c were independent; the numerator is what it is.

$\text{PMI} > 0$	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

Where PMI Comes From

It is not an arbitrary formula. Mutual information between two random variables is

$$I(W; C) = \sum_w \sum_c P(w, c) \log_2 \frac{P(w, c)}{P(w)P(c)},$$

the expected number of bits that observing one variable saves you when coding the other.

PMI is the **pointwise** version: the summand, before taking the expectation. It is the number of bits of information that this particular c gives you about this particular w .

$$I(W; C) = \mathbb{E}_{(w,c)} [\text{PMI}(w, c)]$$

So the units are meaningful: $\text{PMI}(w, c) = 1$ means seeing c halves your uncertainty about w relative to chance — one bit.

Church & Hanks (1990) introduced PMI to computational lexicography for exactly Firth's problem: finding collocations.

PMI Worked: coffee and cup

Using the six-column matrix. $N = 1428$.

$$P(\text{coffee, cup}) = \frac{48}{1428} = 0.0336$$

$$P(\text{coffee}) = \frac{300+48+32+2+1+38}{1428} = \frac{421}{1428} = 0.2948$$

$$P(\text{cup}) = \frac{48+44+3+1}{1428} = \frac{96}{1428} = 0.0672$$

$$\frac{P(w, c)}{P(w)P(c)} = \frac{0.0336}{0.2948 \times 0.0672} = \frac{0.0336}{0.01981} = 1.697$$

$$\text{PMI}(\text{coffee, cup}) = \log_2 1.697 = \mathbf{0.76}$$

cup occurs with coffee about 1.7 times more often than chance would predict — worth about three quarters of a bit.

PMI Worked: Three Contrasting Cells

Cell		Count	obs/exp	PMI
(coffee, cup)	attracted	48	1.70	+0.76
(coffee, brew)	attracted	32	1.80	+0.85
(laptop, cup)	repelled	3	0.12	-3.02
(coffee, screen)	repelled	2	0.07	-3.87
(coffee, the)	indifferent	300	0.99	-0.01

Look at the last row. The largest count in the entire matrix — 300 — receives a PMI of essentially zero.

That is not a bug being papered over. It is the formula doing precisely what it was designed to do: the occurs with coffee exactly as often as it occurs with everything, so it carries **zero bits** about coffee.

Stopword removal, which you have probably done as a preprocessing hack, falls out of PMI as a theorem rather than a heuristic.

The Full PMI Matrix

	the	cup	brew	screen	battery	morning
coffee	-0.00	0.76	0.85	-3.87	-4.58	0.83
tea	-0.00	0.96	0.98	-4.55	$-\infty$	0.50
laptop	0.01	-3.02	$-\infty$	1.05	1.00	-1.60
tablet	0.00	-4.37	$-\infty$	1.04	1.16	-1.95

The structure is now visible to the eye: a positive block at upper-left, a positive block at lower-right, negatives off the diagonal blocks, and a column of zeros where the used to dominate.

Two cells are $-\infty$. Those are the zero counts: $\log_2 0$ is undefined. This is not a numerical inconvenience — it is the first symptom of a real problem.

The Trouble Below Zero

Negative PMI is a claim that two words co-occur *less* than chance. Estimating that reliably is much harder than estimating attraction.

- ① **Unbounded below.** A count of zero gives $-\infty$. The value depends entirely on whether you happened to observe the pair.
- ② **Statistically fragile.** To establish that $P(w, c)$ is, say, 10^{-6} rather than 10^{-5} — one bit of negative PMI — you need on the order of 10^6 observations of w . For most of the vocabulary you do not have them.
- ③ **Psychologically dubious.** Do speakers represent the fact that coffee and battery are mutually *avoidant*? There are $|V|^2$ such facts and almost all of them are uninformative.
- ④ **Computationally costly.** A matrix with meaningful negatives is dense. A matrix with negatives clipped is sparse, and sparsity is what makes $50,000 \times 50,000$ tractable.

All four arguments point the same way, and the fix is the crudest possible one.

Positive PMI

Definition

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

Clip everything negative to zero. Undefined cells become zero. Done.

What this costs you: the distinction between “independent” and “strongly repelled” is erased. coffee/battery and coffee/the now look the same.

What this buys you: a sparse, non-negative matrix with no infinities, on which every similarity measure is well behaved — and, empirically, better neighbors than PMI on essentially every benchmark.

This is your first example of a pattern that recurs for the rest of the course: a theoretically motivated quantity, modified by a theoretically unmotivated clip, because the clipped version works better. Be neither scandalized nor complacent about it — but do notice when it happens.

The PPMI Matrix

	the	cup	brew	screen	battery	morning
coffee	0	0.76	0.85	0	0	0.84
tea	0	0.96	0.98	0	0	0.50
laptop	0.01	0	0	1.05	1.00	0
tablet	0	0	0	1.04	1.16	0

Fourteen of twenty-four cells are zero. The two clusters now occupy **disjoint sets of coordinates**: {cup, brew, morning} and {screen, battery}.

We went from a matrix in which one uninformative column carried 71% of the mass, to a matrix in which that column is gone and the remaining structure is exactly the structure we wanted. No supervision, no tuning, one formula.

PPMI's Own Bias: Rare Contexts

Add one more column. barista occurs 3 times with coffee, once with tea, never with the others.

Context of coffee	Count	PMI	Evidence
barista	3	1.34	three observations
brew	32	0.85	thirty-two observations
morning	38	0.83	thirty-eight observations
cup	48	0.76	forty-eight observations

barista is now the *strongest* context for coffee, on the strength of three tokens.

The reason is structural: $P(c)$ appears in the denominator, so a very rare context inflates the ratio. With $\#(c) = 4$ out of $N = 1432$, almost any co-occurrence looks like a miracle.

PMI overweights the tail. This is well known, was known to Church & Hanks, and is still the most common way to get garbage neighbors out of a PPMI model.

Fixing It: Context Distribution Smoothing

Raise the context probabilities to a power $\alpha < 1$ before dividing:

$$P_\alpha(c) = \frac{\#(c)^\alpha}{\sum_{c'} \#(c')^\alpha}, \quad \text{PPMI}_\alpha(w, c) = \max\left(\log_2 \frac{P(w, c)}{P(w) P_\alpha(c)}, 0\right)$$

Exponentiating by $\alpha = 0.75$ flattens the distribution: it raises the estimated probability of rare contexts, which shrinks their PMI.

Context of coffee	Count	PMI	PMI _{$\alpha=0.75$}
barista	3	1.34	-0.33
cup	48	0.76	0.23
brew	32	0.85	0.16
morning	38	0.83	0.21

The spurious winner is demoted below zero and clipped away. The value 0.75 is not principled: it was found by search, in word2vec's negative-sampling distribution, and shown by Levy, Goldberg & Dagan (2015) to help PPMI for the same reason.

Other Weightings You Will Meet

Scheme	Form	Notes
tf-idf	$\#(w, d) \cdot \log \frac{ D }{df_w}$	Term–document only; the log is a frequency damper, the idf an informativeness weight
PPMI	$\max(\text{PMI}, 0)$	The default for term–term; what we use
Smoothed PPMI	$\alpha = 0.75$	Corrects the rare-context bias
Shifted PPMI	$\max(\text{PMI} - \log k, 0)$	Levy & Goldberg: this is what word2vec implicitly factorizes
Local MI	$\#(w, c) \cdot \text{PMI}(w, c)$	Reweights by evidence; suppresses the tail directly
t-test	$\frac{P(w, c) - P(w)P(c)}{\sqrt{P(w)P(c)}}$	A significance test rather than an information measure

The fourth row is the one to remember. It is the bridge to Session 8: the neural method that appeared to replace counting turns out, under analysis, to be counting with a particular weighting and a particular factorization.

How to Compare Two Rows

We now have a vector per word. The last decision is how to measure closeness.

Measure	Formula	Sensitive to
Euclidean	$\ \mathbf{v} - \mathbf{w}\ $	magnitude <i>and</i> direction
Dot product	$\mathbf{v} \cdot \mathbf{w}$	magnitude <i>and</i> direction
Cosine	$\frac{\mathbf{v} \cdot \mathbf{w}}{\ \mathbf{v}\ \ \mathbf{w}\ }$	direction only
Jaccard	$\frac{\sum_i \min(v_i, w_i)}{\sum_i \max(v_i, w_i)}$	overlap of support

The right choice follows from one observation about what magnitude *means* in this matrix.

Why Not Euclidean

Take the coffee vector and suppose we had simply seen twice as much coffee text: every count doubles, so (approximately) the vector doubles.

v versus 2v	
Euclidean distance	$\ \mathbf{v} - 2\mathbf{v}\ = \ \mathbf{v}\ = 1.42$ (large)
Cosine similarity	$\frac{\mathbf{v} \cdot 2\mathbf{v}}{\ \mathbf{v}\ \cdot 2\ \mathbf{v}\ } = 1$ (identical)

The principle

In a co-occurrence matrix, **vector length encodes corpus frequency; vector direction encodes contextual profile**. We want the profile. So we discard the length — which is exactly what normalizing by $\|\mathbf{v}\|$ does.

Corollary: on length-normalized vectors, $\|\hat{\mathbf{v}} - \hat{\mathbf{w}}\|^2 = 2 - 2\cos\theta$ — the choice only matters if you skip the normalization.

Cosine, Worked

$\mathbf{v} = \text{PPMI}(\text{coffee})$, $\mathbf{w} = \text{PPMI}(\text{tea})$, over (the, cup, brew, screen, battery, morning):

$$\mathbf{v} = (0, 0.762, 0.855, 0, 0, 0.840), \quad \mathbf{w} = (0, 0.958, 0.984, 0, 0, 0.498)$$

Numerator. Only three coordinates are nonzero in both:

$$\mathbf{v} \cdot \mathbf{w} = (0.762)(0.958) + (0.855)(0.984) + (0.840)(0.498) = 0.730 + 0.841 + 0.418 = 1.990$$

Denominator.

$$\|\mathbf{v}\| = \sqrt{0.581 + 0.731 + 0.706} = 1.420, \quad \|\mathbf{w}\| = \sqrt{0.918 + 0.968 + 0.248} = 1.461$$

$$\cos(\mathbf{v}, \mathbf{w}) = \frac{1.990}{1.420 \times 1.461} = \frac{1.990}{2.075} = \mathbf{0.959}$$

Notice how little arithmetic sparsity left us: three products, not six. On a real matrix with 50,000 columns and a few hundred nonzeros per row, this is the difference between feasible and not.

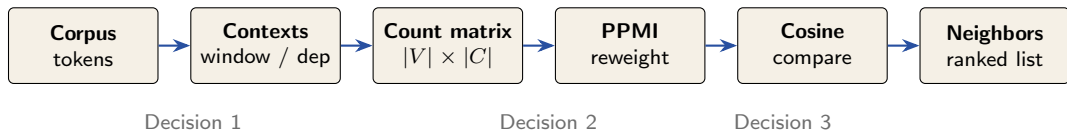
Before and After

Pair	Raw counts	PPMI
coffee / tea	0.999	0.959
laptop / tablet	1.000	0.997
coffee / laptop	0.950	0.000
coffee / tablet	0.946	0.000
tea / laptop	0.947	0.000
tea / tablet	0.943	0.000
Margin	0.05	0.96

Same corpus. Same contexts. Same similarity function. **Only the weighting changed**, and the margin between the correct and incorrect pairs went from 0.05 to 0.96.

If you take one practical lesson from today: in a count-based model, the weighting is not preprocessing. It is the model.

The Method, End to End



Everything in Part II is a variation on this pipeline.

- **Session 8** replaces the wide sparse matrix with a narrow dense one — by factorization (LSA) or by prediction (word2vec), which turn out to be closely related.
- **Session 9** makes the row depend on the occurrence rather than the type, so bank gets a different vector in each sentence.
- **Session 10** asks the question this pipeline cannot answer: given vectors for red and car, what is the vector for *red car*?

Note that nothing in the pipeline has a place to put a composition rule. That absence is the topic of Session 10.

Where Did the Types Go?

Six sessions were organized by **semantic type**. Today every word is the same kind of object: a row.

Expression	Type	6B, exact	Today
<i>Alice</i>	e	order 1 — a vector	order 1
<i>sleeps</i>	$\langle e, t \rangle$	order 1 — a vector	order 1
<i>likes</i>	$\langle e, \langle e, t \rangle \rangle$	order 2 — a matrix	order 1
<i>every</i>	$\langle \langle e, t \rangle, t \rangle$	order 2, then 2 numbers	order 1

The exact encoding preserved the type distinctions as *tensor order*. The distributional encoding flattens every one of them to order 1.

Why this matters more than it looks. In Part I the type did not merely classify a constituent — it **selected the composition rule**, and said which operand was the function and which the argument.

With no types, nothing selects a rule and nothing breaks the symmetry between the operands. The obvious candidates, $\mathbf{v} + \mathbf{w}$ and $\mathbf{v} \odot \mathbf{w}$, are both *commutative* — neither can distinguish a function from its argument.

So Session 10's difficulty is not that we lack an operation, but that we have discarded the discipline that would tell one what to do.

Evaluating a Similarity Model

The standard method: collect human similarity judgments, compute model similarities, report the rank correlation (Spearman's ρ).

Dataset	Size	Asks for	Note
TOEFL synonyms	80	multiple choice	Landauer & Dumais's original LSA test
WordSim-353	353	"relatedness"	conflates similarity and association
MEN	3000	relatedness	larger, crowdsourced
SimLex-999	999	similarity, explicitly	designed to separate the two
SimVerb-3500	3500	similarity, verbs	verbs are much harder

Two warnings about this literature.

- 1 Human inter-annotator agreement on SimLex is around $\rho = 0.67$. A model reporting 0.7 is not "superhuman"; it is at the noise ceiling of the task.
- 2 These are word-pair tasks. Nothing here tests composition, inference, or truth. A model can top every one of these leaderboards and still fail every example from Session 5.

Similarity Is Not Relatedness

The single most important distinction in this literature, and the one most often elided.

Pair	Similar?	Related?
car / automobile	yes	yes
car / bicycle	yes	yes
car / tire	no	yes, strongly
car / highway	no	yes, strongly
doctor / hospital	no	yes
doctor / nurse	yes	yes

Similarity is substitutability — same type, same slot. **Relatedness** is association — same scene, same script.

And now recall Decision 1: a small window gives you similarity, a large window gives you relatedness. The two failure modes and the window parameter are the same fact, seen from two directions.

The Antonym Problem

Context	Fits
<i>the _____ water</i>	hot, cold
<i>too _____ to drink</i>	hot, cold
<i>how _____ is it?</i>	hot, cold
<i>the _____ tap</i>	hot, cold
<i>it got _____ overnight</i>	hot, cold

hot and cold have nearly identical distributions. In a PPMI space they are typically each other's **nearest neighbor**.

But they are not similar in meaning. They are contraries: *the water is hot **entails** the water is not cold*.

The same holds for buy/sell, above/below, always/never, rise/fall — every antonym pair in the language, and antonymy is one of the most robust lexical relations there is.

Why the Antonym Problem Is Principled

It is tempting to treat this as a defect to be patched. It is not. It is the hypothesis working correctly.

- 1 Antonyms are, by definition, words that differ along **exactly one** dimension while agreeing on every selectional, syntactic and topical property.
- 2 Distributional context measures precisely those selectional, syntactic and topical properties.
- 3 Therefore antonyms are *maximally* distributionally similar. The method is not failing to see the difference; the difference is invisible in the data.

The general form

Distribution recovers **substitutability**. Substitutability is a good proxy for similarity of *type* and terrible proxy for similarity of *value*. Any two words that contrast along one dimension and agree on the rest will be confused.

Harris's weak reading, vindicated: the correlation is real, and it is a correlation, not an identity.

The Deeper Version of the Same Point

Return to Session 5 and 5A. What did a determiner's meaning consist of?

	Restrictor	Scope	Licenses any?
<i>every</i>	MON↓	MON↑	in restrictor only
<i>no</i>	MON↓	MON↓	in both
<i>some</i>	MON↑	MON↑	in neither

These three words have **nearly identical distributions**: all are determiners, all precede nouns, all appear in subject and object position, all combine with the same nouns.

A distributional model will place *every*, *no*, and *some* close together — and the property that distinguishes them, monotonicity, is the property that controls inference.

The antonym problem for content words is an inconvenience. The same problem for logical words is a wall, because their entire semantic contribution is the dimension which distributions can't even see.

Back to Session 6B: the Approximate Regime

Session 6B set out two regimes. Today we entered the second; here is exactly what changed.

	Exact (6B)	Distributional (today)
Basis	entities, enumerated	contexts, observed
A word is	a one-hot vector	a sparse profile over contexts
Dimension	n , one per entity	$ C $, one per context
$\mathbf{v} \cdot \mathbf{w}$ means	membership, exactly	similarity, approximately
Two words	orthogonal unless identical	at any angle
Derived from	a model $\mathcal{M}$	a corpus
Guarantee	a theorem	an empirical correlation

What we gained: an angle between any two words — what the exact encoding could not give us, because orthogonality had made every pair equidistant.

What we lost: the inner product no longer computes anything. In 6B, $\mathbf{p} \cdot \mathbf{e}_a$ was the truth value of $P(a)$; here $\mathbf{v} \cdot \mathbf{w}$ is a number that correlates with a judgment.

One caveat: 6B's approximate regime was *dense*, in $\mathbb{R}^d$ with $d \ll n$. Today's vectors are wide and sparse. The compression half of that regime arrives next session.

The Ledger So Far

	Part I: logic	Part II: distribution
Similarity	absent	yes, graded
Learnable from text	no	yes, unsupervised
Handles unseen words	no	degrades gracefully
Covers the whole lexicon	no	yes
Truth conditions	yes	no
Entailment	yes, with proof	no; cosine is symmetric
Quantifier scope	yes	no
Negation	yes	invisible
Composition	yes, by application	open — Session 10

The columns are close to complementary. That is the fact the rest of the course is organized around — and the reason Part IV asks whether a single algebraic setting could support both.

One More Problem: The Matrix Is Enormous

For a realistic vocabulary, the term-term matrix is $|V| \times |V|$.

Vocab	Cells	Dense storage (4-byte floats)
10,000	10^8	400 MB
50,000	2.5×10^9	10 GB
500,000	2.5×10^{11}	1 TB

PPMI makes it sparse (well under 1% nonzero), so it is storable. But three problems remain:

- **Sparsity is also blindness.** If espresso and cappuccino never share a context, their cosine is exactly zero, though they are near-synonyms.
- **No generalization across contexts.** The columns car and automobile are treated as unrelated coordinates.
- **Dimensionality.** Downstream models want a few hundred features, not fifty thousand.

All three are addressed by the same move: find a low-dimensional space in which the observed matrix is approximately recoverable. That is Session 8.

Summary

Point	Detail
Harris 1954	difference in meaning <i>correlates with</i> difference in distribution
Firth 1957	collocation is real information the logic has no slot for
The hypothesis	three independent claims; the third is where trouble lives
Decision 1	context type sets <i>which</i> similarity you measure
Window size	small $\Rightarrow$ paradigmatic; large $\Rightarrow$ topical
Decision 2	raw counts fail: one stopword column carried 71% of mass
PMI	$\log_2$ of observed over expected; the gets ≈ 0
PPMI	clip negatives: sparse, stable, empirically better
$\alpha = 0.75$	fixes PMI's overweighting of rare contexts
Decision 3	cosine, because length is frequency and direction is profile
Result	margin went from 0.05 (raw) to 0.96 (PPMI)
Similarity vs. relatedness	substitutability vs. association; the window is the dial
Antonyms	maximally similar distributionally; a principled failure
Logical words	<i>every/no/some</i> are distributionally near-identical
Next	the matrix is huge and sparse $\Rightarrow$ reduction and prediction

Reading and Exercises

Reading: Lappin & Shanahan §2.2–2.4.

Primary sources, all short and worth the hour:

- Harris (1954), *Distributional Structure*, WORD 10 — read for the hedging
- Church & Hanks (1990), *Word Association Norms, Mutual Information, and Lexicography*
- Levy, Goldberg & Dagan (2015), *Improving Distributional Similarity with Lessons Learned from Word Embeddings*
- Hill, Reichart & Korhonen (2015), *SimLex-999* — the similarity/relatedness separation

Exercises:

- 1 Recompute the PPMI matrix with the the column deleted *before* weighting. Do the cosines change? Explain why, in one sentence.
- 2 Add a fifth row, espresso, with counts (60, 12, 9, 0, 0, 11). Compute its PPMI row and its cosine to coffee and to tea.
- 3 Show that $\text{PMI}(w, c) = \text{PMI}(c, w)$. What does this symmetry rule out as a model of entailment?
- 4 **Harder.** Construct a 3×3 count matrix on which raw-count cosine and PPMI cosine give *opposite* rankings for one pair. Explain what feature of your construction forces this.

Questions?

Next: fifty thousand dimensions down to three hundred —
counting, factorizing, and predicting