

Contextual Embeddings

Why a Word Should Not Have One Vector

James Pustejovsky

Brandeis University
Department of Computer Science

CS 135: Computational Semantics

Fall 2026

Where We Left Off

Session 8 compressed the co-occurrence matrix to a few hundred dimensions by three routes that turned out to be one, and closed with a result: SGNS factorizes shifted PMI.

Along the way we gained generalization, compactness, and differentiability. The last of those matters today.

The commitment nobody examined

Every method so far — PPMI, LSA, word2vec, GloVe — assigns **one vector to each word type**. Not one per occurrence. One per entry in the vocabulary.

That was never argued for. It was inherited from the matrix: rows are indexed by vocabulary items, so a word gets a row, and that is the end of it.

Today we examine the commitment, find it is wrong, and look at what it costs to abandon it.

Today's Target: Claim (2)

Session 7 broke the distributional hypothesis into three independent claims.

Claim	Status
1 Contexts are informative	holds
2 The signal is summarizable in one fixed object	today
3 Geometric closeness tracks semantic closeness	holds, with the Session 7 caveats

Claim (2) was called “an engineering bet” when we first stated it, with a promise that this session would revisit it.

The verdict, stated now so you can watch the argument for it: the claim is false, and the field's response was not to repair it but to **abandon** it — while leaving claims (1) and (3) exactly as they were.

The Word That Breaks It

*She walked along the **bank** of the river.*

*She deposited the check at the **bank**.*

Two occurrences of one string, with unrelated meanings. A type-level model must give them the same vector.

The usual account of what goes wrong: the vector is an *average* of the two senses, and lands near neither.

That account is right for dense embeddings. For the sparse PPMI representation you built in Session 7 something more drastic happens, and it is worth seeing first.

What PPMI Does to a Two-Sense Word

Suppose we could observe the two senses separately. Financial sense at 80% of tokens, river sense at 20%.

PPMI	money	loan	account	river	water	shore	the
bank, financial sense	0.49	0.48	0.48	0.00	0.00	0.00	0.00
bank, river sense	0.00	0.00	0.00	1.81	1.79	1.81	0.00
credit_union	0.46	0.47	0.49	0.00	0.00	0.00	0.01
stream	0.00	0.00	0.00	1.82	1.84	1.82	0.00
bank, as actually observed	0.17	0.16	0.16	0.00	0.00	0.00	0.00

Look at the last row's river columns. They are not diluted. They are **zero**.

Pooling the senses raised bank's marginal frequency, which lowered the PMI of every cell; the minority sense's cells fell below zero, and the clip removed them.

Winner-Take-All, Not Averaging

Cosine with the observed bank vector	
bank, financial sense	1.000
credit_union	0.999
bank, river sense	0.000
stream	0.000

The sparse representation does not compromise between the senses. It awards the vector to the majority sense and deletes the other entirely.

Change the frequency split so the river sense dominates, and the vector flips wholesale to the river sense — cosine 0.996 with stream, 0.000 with the financial sense.

The consequence for a downstream system is severe and invisible: for a word whose senses are unevenly distributed in your corpus, the minority sense is not underrepresented. It is absent, and nothing in the representation records that it was ever there.

The Dense Case: Genuine Averaging

Dense embeddings do not clip, so the standard account applies. If the two senses would have had unit vectors **a** and **b** at right angles, the learned vector sits at

$$\mathbf{v} \propto f \mathbf{a} + (1 - f) \mathbf{b}.$$

Frequency split	$\cos(\mathbf{v}, \mathbf{a})$	$\cos(\mathbf{v}, \mathbf{b})$	Angle from a
50/50	0.707	0.707	45°
70/30	0.919	0.394	23°
80/20	0.970	0.243	14°
95/5	0.999	0.053	3°

At an even split the vector is 45° from both senses — a point that denotes neither, and that no occurrence of the word ever meant.

Two representations, two different failures: the sparse one loses the minority sense, the dense one invents a meaning that is not a sense of the word. Both follow from one vector per type.

It Is Not a Handful of Words

Word	WordNet senses	A few
run (verb)	41	move fast; operate; flow; compete; manage
set	47	put; collection; harden; a group; ready
line	29	mark; queue; rope; verse; business
break	59	shatter; pause; news; opportunity; violate

And the distribution is the familiar one: a long tail of monosemous words, and the **most frequent** words are the most polysemous.

So the words the representation handles worst are exactly the ones that appear most often. Frequency does not rescue you here — it is the cause.

Sense counts are inventory-dependent and WordNet's are famously fine-grained; the qualitative point survives any inventory.

Two Different Phenomena Being Conflated

	Example	What is going on
Homonymy	bank	two unrelated lexemes that happen to share a spelling
Polysemy	book	<i>one</i> lexeme with related senses: physical object and informational content

The book is heavy and *the book is boring* predicate of different aspects, and *she picked up and finished the book* accesses both at once, without contradiction.

This is the distinction Generative Lexicon is built to handle: a structured lexical entry with a qualia structure, where the senses are *generated* from the entry rather than enumerated alongside it.

A type-level vector conflates homonymy with polysemy by treating both as noise in one profile. A token-level vector conflates them differently — a different point per use, with nothing to say that the book points are systematically related and the bank points are not.

We return to this in Session 10, where GL meets the composition problem directly.

What Part I Did About This

Nothing subtle. It stipulated.

$$\llbracket \text{bank}_1 \rrbracket = \lambda x. \text{financial-institution}(x), \quad \llbracket \text{bank}_2 \rrbracket = \lambda x. \text{riverbank}(x)$$

Two lexical entries, disambiguated before composition begins, with the choice made by something outside the semantics.

	Part I	Part II so far
Senses	enumerated by hand	averaged or annihilated
Disambiguation	assumed done	not represented
Cost	a sense inventory	the minority sense

Part I's answer is not available to us. Requiring a hand-built sense inventory is precisely the commitment distributional semantics was trying to escape — and sense inventories do not agree with each other, which is its own literature.

The Senses Are Still In There

Before abandoning type-level vectors, note a result that complicates the picture.

Arora, Li, Liang, Ma & Risteski (2018)

A static embedding of a polysemous word is approximately a **sparse linear combination** of a small number of “atoms of discourse” — and the individual senses can be recovered from it by sparse coding, without a sense inventory.

So the superposition is real, it is approximately linear, and it is partially invertible.

This matters for two reasons. First, it says the type-level vector is not simply corrupted — it is a *superposition*, with structure. Second, it is one of the few results in Part II that describes a vector as a **sum of components with distinct semantic roles**.

Hold that second observation. A representation that is a graded sum of components with different roles is the shape Part IV will propose on entirely different grounds.

The Move

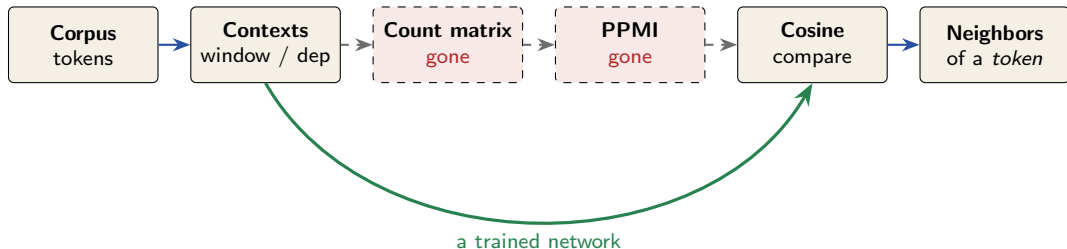
Static	$w \mapsto \mathbf{v}_w$
Contextual	$(w, \text{context}) \mapsto \mathbf{v}_{w, \text{context}}$

The representation becomes a **function of the occurrence**, not a lookup keyed by the type.

Consequence	Detail
No embedding table	there is no finite set of vectors to print
No vocabulary of meanings	every token in every sentence gets its own point
Sense emerges	disambiguation is not a separate step; it is the representation
The model is the lexicon	you cannot inspect an entry, only run the network

Session 8 made embeddings differentiable parameters. Once a representation is a layer rather than a table, nothing requires it to be a lookup — and that is the whole idea.

The Session 7 Pipeline, Broken



The two middle boxes — the explicit matrix and the weighting — disappear. What replaces them is a trained function from a token and its context to a vector.

Notice what did *not* change: the ends. A corpus goes in, and a vector you compare by cosine comes out. Claims (1) and (3) of the distributional hypothesis are untouched — the epistemology is the same, and only the summarization step was replaced.

ELMo: the First Widely Used Version

Peters, Neumann, Iyyer, Gardner, Clark, Lee & Zettlemoyer (2018)

Train a two-layer bidirectional LSTM language model. For a token, take a task-weighted combination of the hidden states at every layer.

Two ideas arrived together and are worth separating:

- **Contextualization** — the representation depends on the sentence. This is the idea that lasted.
- **Layer mixing** — different layers carry different information, and the downstream task should choose the blend. This foreshadowed the probing results.

ELMo's architecture was replaced within a year. Its central claim was not: a word's representation should be computed, not retrieved.

The recurrence is what the transformer removed, and removing it is what made the scale of Session 9's models possible.

Where the Mechanics Live

The transformer's forward pass — tokenization, positional encoding, layer normalization, the projections, the residual stream, the feed-forward block, the stack, and the modern replacements for each — is covered step by step in the companion deck.

Session 9A: *Inside the Transformer*

This session assumes it and asks a different question:

What kind of *semantic* operation is attention?

That question has a precise answer, it is answerable from one equation, and the answer is what Session 10 needs. Everything else about the architecture is engineering we can take as given.

The One Equation

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

For a single query token with vector $\mathbf{q}$, against context tokens with keys $\mathbf{k}_1, \dots, \mathbf{k}_n$ and values $\mathbf{v}_1, \dots, \mathbf{v}_n$:

$$\mathbf{o} = \sum_{i=1}^n \alpha_i \mathbf{v}_i, \quad \alpha_i = \frac{\exp(\mathbf{q} \cdot \mathbf{k}_i / \sqrt{d_k})}{\sum_j \exp(\mathbf{q} \cdot \mathbf{k}_j / \sqrt{d_k})}.$$

$\mathbf{q} \cdot \mathbf{k}_i$	a learned similarity between this token and context token i
softmax	turn those similarities into weights summing to 1
$\sum \alpha_i \mathbf{v}_i$	a weighted average of the context tokens' value vectors

That is the whole operation: a soft retrieval, whose result is a convex combination of what was retrieved.

Worked by Hand

Let $d_k = 2$. One query — the token bank — against two sentences.

$\mathbf{q}_{\text{bank}} = (1, 1)$	$\mathbf{k}_{\text{river}} = (3, 0), \mathbf{v}_{\text{river}} = (1, 0)$	$\mathbf{k}_{\text{bank}} = (1, 1), \mathbf{v}_{\text{bank}} = (0.5, 0.5)$
	$\mathbf{k}_{\text{money}} = (0, 3), \mathbf{v}_{\text{money}} = (0, 1)$	

Scores (identical in both sentences, by the symmetry of the setup):

$$\mathbf{q} \cdot \mathbf{k}_{\text{river}} / \sqrt{2} = 3 / 1.414 = 2.121, \quad \mathbf{q} \cdot \mathbf{k}_{\text{bank}} / \sqrt{2} = 2 / 1.414 = 1.414$$

$$\alpha = \text{softmax}(2.121, 1.414) = (0.670, 0.330)$$

Outputs (different, because the *values* differ):

$$\mathbf{o}_A = 0.670 (1, 0) + 0.330 (0.5, 0.5) = (0.835, 0.165)$$

$$\mathbf{o}_B = 0.670 (0, 1) + 0.330 (0.5, 0.5) = (0.165, 0.835)$$

$\cos(\mathbf{o}_A, \mathbf{o}_B) = 0.380$. One token, one string, two representations — which is exactly what we asked for.

What That Example Shows, and What It Does Not

Shows: the type/token problem is solved, and solved by construction rather than by a disambiguation module. bank has as many representations as it has occurrences, and they separate along the dimension the context distinguishes.

Does not show: that anything was composed.

The output is	a convex combination of vectors that were already present
The weights are	a learned similarity, symmetric in form
Nothing marks	which token is the function and which the argument
Nothing produces	an object of a different type from its inputs

Attention selects and mixes. Function application consumes an argument and returns something of a new type. These are different operations, and no amount of stacking turns the first into the second.

This is the sentence Session 10 is built on. It is worth writing down.

Order Enters Exactly Once

Strip the positional encodings and compute attention over a bag of tokens. Permute the tokens.

The output is the same, up to the same permutation.

Self-attention is **permutation-equivariant**. The mechanism has no access to word order at all; order is injected by adding position vectors to the inputs, before the first layer.

dog bites man

man bites dog indistinguishable without positional encoding

You have met this pair before, or will. Vector addition is commutative, so a summed representation cannot tell these apart; attention has the same defect at its core, repaired by a separate additive signal rather than by the composition operation itself.

RoPE and its relatives change how the signal is injected, not the fact that it must be injected.

Multi-Head, and the Feed-Forward Block

Multi-head attention runs h of these weighted averages in parallel with different learned projections, and concatenates.

Semantically: h different notions of “relevant,” each producing its own mixture. Heads specialize — some track syntactic dependencies, some coreference, some positional offsets — but the specializations are emergent and not stable across training runs.

The feed-forward block applies the same position-wise MLP to every token independently.

Attention	moves information <i>between</i> positions
FFN	transforms information <i>at</i> a position

The division matters for us: only attention combines material from two places, and we have just seen that what it does there is average. The FFN is where a nonlinear map on a single representation could live — and it is applied uniformly, with no notion of which position is functor and which argument.

BERT

Devlin, Chang, Lee & Toutanova (2019)

A deep bidirectional transformer trained by **masked language modeling**: delete 15% of the tokens and predict them from both directions of context.

	BERT-base	BERT-large
Layers	12	24
Hidden size	768	1024
Attention heads	12	16
Parameters	110M	340M

The 768 is worth noting — when Session 10 opens with **alice, sleeps, likes** $\in \mathbb{R}^{768}$, this is where the number comes from.

Masked language modeling is the distributional hypothesis restated as a training objective: a word is predicted from the company it keeps, and the representation supporting the prediction is what we keep.

Probing: the Method

The representations are opaque. The standard way to ask what is in them:

- 1 Freeze the model.
- 2 Extract the vector for each token at some layer.
- 3 Train a **small supervised classifier** — usually linear — to predict some linguistic property from that vector.
- 4 If the probe succeeds, conclude the property is encoded.

Property probed	part of speech, dependency label, coreference, semantic role
Probe	logistic regression, or a single linear map
Claim	the information is <i>linearly accessible</i> in the representation

Step 4 is where the trouble is, and we come back to it. First, the findings.

Tenney: the Classical Pipeline, Rediscovered

Tenney, Das & Pavlick (2019), *BERT Rediscovered the Classical NLP Pipeline*

Probe each layer for each task, and ask at which layer the information becomes available.

Lower layers	part of speech, morphology
Middle layers	constituents, dependency structure
Upper layers	semantic roles, coreference

The order recapitulates the traditional NLP pipeline — tag, parse, then resolve — which nobody built in and which the training objective never mentions.

Two readings, and the paper is careful about the difference. The strong one: the model discovered that this decomposition is the right one. The weak one: the tasks form a difficulty ordering, and depth correlates with difficulty. The evidence underdetermines the choice.

Hewitt and Manning: the Structural Probe

Hewitt & Manning (2019)

Find a single linear transformation B such that, for every pair of tokens i, j in a sentence,

$$\|B(\mathbf{h}_i - \mathbf{h}_j)\|^2 \approx d_{\text{tree}}(i, j)$$

the distance between them in the dependency parse.

It works. A single learned B , applied to frozen BERT representations, recovers parse-tree distances well enough to reconstruct the tree.

The claim is sharper than “syntax is in there somewhere.” It is that the parse tree is embedded as a **metric structure** in a linear subspace — squared distance in the right coordinates *is* tree distance.

For this course that is notable: a discrete combinatorial object recovered as geometry, with the recovery map linear. It is the closest thing in Part II to the correspondence Part IV is after — and it is about syntax, not meaning.

And Then the Control

Hewitt & Liang (2019), *Designing and Interpreting Probes with Control Tasks*

Build a **control task**: the same inputs, but labels assigned at random and fixed per word type. Train the same probe on it.

If the probe fits the random labels nearly as well as the real ones, it was powerful enough to learn the task by itself, and its success said nothing about the representation.

Selectivity	(real-task accuracy) – (control-task accuracy)
Finding	some widely cited probes have <i>low</i> selectivity
Implication	the bigger the probe, the weaker the conclusion

This is the methodological spine of the literature and it is routinely skipped. Read any probing result with one question first: *compared to what?*

The structural probe holds up well on this criterion, which is part of why it is the one worth teaching.

A Geometric Caution: Anisotropy

Ethayarajh (2019)

Contextual representations are highly **anisotropic**: in upper layers, randomly chosen word vectors have average cosine similarity well above zero — in some layers above 0.9.

The vectors occupy a narrow cone rather than spreading over the sphere.

Consequence	a raw cosine of 0.8 may mean nothing at all
Fix	subtract the mean, or compare against the random baseline

Compare Session 7, where one high-frequency column made every pair of rows look similar and the diagnosis was that we were measuring the wrong thing. This is the same failure, in a learned space, and it is corrected the same way — by asking how similar two things are *relative to chance*.

PMI was our answer to that question for counts. There is no equally clean answer here.

What Probing Has Not Found

A decade of probing has located morphology, part of speech, constituency, dependencies, coreference, semantic roles, entity types, and factual associations.

Not located	
Truth conditions	no subspace in which a sentence's vector encodes when it is true
Quantifier scope	the two readings of <i>every student read a book</i>
Monotonicity	the property that licenses the inference
A composition operation	no operation identified that takes two constituents to a third

These absences are not for want of looking — each has been the subject of dedicated studies, and the negative results are as careful as the positive ones.

Note the pattern in the two lists. What is found is **structural and recoverable from form**. What is not found is **relational to the world, or compositional**. That is the same boundary Session 7 drew with the antonym argument, holding at a much larger scale.

Fixed: Senses

	Static	Contextual
bank _{river} vs. bank _{finance}	one vector	two, far apart
Minority sense	clipped or swamped	represented whenever it occurs
Disambiguation	a separate module	not a separate step
Novel senses	absent	handled if the context supports them

This is a real solution to a real problem, and it is the reason contextual representations displaced static ones within about two years.

What it does not give is a *sense inventory*. There are no senses in the model — only a cloud of token points, denser in some regions than others. Whether that cloud has the discrete structure a lexicon requires is an open empirical question, and the answer differs for homonymy and for polysemy.

Which is the GL question again, now posed geometrically.

Fixed: Unknown Words, at a Price

Subword tokenization (WordPiece, BPE) splits rare strings into known pieces, so nothing is out of vocabulary.

<i>unbelievable</i>	un · ##believ · ##able
<i>Pustejovsky</i>	Pu · ##ste · ##jov · ##sky

No OOV, graceful handling of morphology and of names, a fixed vocabulary of about 30,000 pieces.

And the unit of representation is no longer the word. The pieces are chosen by a compression algorithm to minimize corpus encoding length. They are not morphemes, they do not respect morphological boundaries, and they have no lexical status.

Part I's basic object was the lexical entry. Part II's is now a statistically convenient substring. That is a large quiet change, and it is almost never remarked on.

Not Fixed: Antonymy and Negation

Antonymy	hot and cold still occupy near-identical contexts, so their contextual representations remain close
Negation	representations of <i>the water is hot</i> and <i>the water is not hot</i> are highly similar

Contextualization does not touch either, and the reason is the argument from Session 7, unchanged: distribution recovers **substitutability**, and negation preserves almost every distributional property of the sentence it operates on.

Ettinger (2020) made this concrete with a psycholinguistic diagnostic: BERT assigns high probability to completions that are true of the affirmative sentence even under explicit negation — *a robin is not a _____* still favors bird.

The scale did not help. Neither did the contextualization, because the problem was never that the model lacked context.

Not Fixed: Monotonicity

Return to Session 5A. *No student who owns a car drives to campus* licenses an inference to *no student who owns a red car drives to campus*, and reversing the determiner reverses the direction.

What the inference needs	the determiner's monotonicity profile, per argument position
What distribution supplies	<i>every</i> , <i>no</i> , <i>some</i> occupy nearly the same contexts

This is the case we examined at length in the positioning deck, and nothing in this session changes the verdict. The property that controls the inference is the one property distribution cannot see, because the determiners are distributionally near-identical.

Models do get many NLI items right. The question the benchmarks cannot settle is whether they are computing the inference or matching a pattern that usually accompanies it — and the constructed-adversarial results suggest the latter more often than the aggregate scores imply.

Not Fixed: Composition

A contextual model gives a vector per token. Applications need one vector per sentence. The standard answers:

Method	Operation	Property
Mean pooling	average the token vectors	commutative
Max pooling	coordinatewise maximum	commutative
[CLS] token	read one designated position	attention-weighted average

The first two are commutative, so they cannot in principle distinguish *dog bites man* from *man bites dog*. They do not, in practice, entirely fail at it — because the *token* vectors already carry positional and contextual information from the layers below.

Read that carefully: the pooling operation is not doing the compositional work. Whatever compositional behavior a sentence embedding shows was performed by attention, whose semantic character we characterized three sections ago as **weighted averaging**.

Which returns the question to where Session 10 picks it up.

The Ledger

	Part I	S7	S8	S9
Similarity	—	✓	✓	✓
Generalizes to unseen pairs	—	—	✓	✓
Word senses	by stipulation	—	—	✓
Unknown words	—	—	—	✓
Syntax recoverable	✓	—	—	✓
Antonyms distinguished	✓	—	—	—
Negation	✓	—	—	—
Quantifier scope	✓	—	—	—
Monotonicity	✓	—	—	—
Composition	✓	—	—	— S10

Two new rows went green, and one Part I had only by stipulation is now earned. The bottom block has not moved since Session 7 — three sessions, three orders of magnitude of scale, no change.

The Verdict on Claim (2)

Claim		Verdict
1	Contexts are informative	survives
2	The signal is summarizable in one fixed object	false — abandoned
3	Geometric closeness tracks semantic closeness	survives

Claim (2) was not repaired. It was dropped, and the representation became a function of the occurrence.

What is striking is how little else changed. A corpus still goes in, the representation is still a summary of contexts, similarity is still cosine — and the failures of Session 7 are still the failures of Session 9.

The shape of the result

Dropping claim (2) fixed everything that was a consequence of the *type* being the unit — senses, unknown words, contextual modulation. It fixed nothing that was a consequence of **distribution being the evidence**.

The Question, and Why It Is Worth Stating Carefully

By now the models we are discussing produce fluent text, pass professional examinations, and are used daily by hundreds of millions of people. It is reasonable to ask whether the distinctions this course draws still matter.

Do these systems have meaning, or only form?

The debate is live and the participants are serious on both sides. Two slides for the case each side makes, and then the version of the question this course can actually settle.

The Case for Form Without Meaning

Bender & Koller (2020), *Climbing towards NLU*

Meaning is a relation between form and **communicative intent**. A system trained only on form has no access to the second term, so it cannot learn meaning in that sense, however well it models form.

The octopus. Two people on separate islands communicate by telegraph. A hyperintelligent octopus taps the cable, learns the statistical structure of the exchanges, and eventually impersonates one party convincingly. Then one asks how to defend against an attacking bear. The octopus has never seen a bear, a coconut, or a rope.

The argument's force is that the octopus's failure is not a matter of scale. More tapping does not supply what was never on the wire.

Related: the symbol grounding problem (Harnad 1990), and the arguments about reference that Part I's model theory answers by fiat — by *giving* the model a domain.

The Case Against

Line of reply	Content
Conceptual role	meaning may be constituted by inferential role rather than by reference; text encodes a great deal of inferential role
Structure is learned	probing finds syntactic and relational structure nobody supplied
The premise is too strong	human language acquisition is also heavily distributional, and grounding is partial for most vocabulary
Multimodality	the grounding gap is an artifact of the training data, not of the architecture

Pavlick (2023) is the most careful statement of the middle position: the question is empirical, the evidence mixed, and both the strong and the dismissive readings outrun what is known.

Few of the abstract words in this lecture — *monotonicity*, *conservativity*, *entailment* — are grounded in perception for anyone, and we manage.

Where This Course Stands

We do not need to settle it.

The grounding debate concerns whether a system trained on form can *mean* anything. This course asks a narrower question, and one that admits an answer:

Is there a compositional operation in the representation, and if so, what is it?

Answerable by	inspecting the architecture, which we did three sections ago
Independent of	whether the representations refer to anything
The answer so far	attention averages; pooling is commutative; nothing applies

A system could be fully grounded and still lack a composition operation, or have one and be entirely ungrounded. Keeping the two questions apart is what lets us make progress on the second.

Practice

1. In the two-sense PPMI example, the minority sense was clipped to zero. Would the same happen without the PPMI clip — under plain PMI? Under raw counts? Say what each would give.
2. The attention example gave the same weights (0.670, 0.330) in both sentences. Which feature of the construction forced that, and what would you change to make the weights differ?
3. Mean pooling is commutative and therefore cannot distinguish *dog bites man* from *man bites dog*. Explain precisely why a transformer nevertheless does distinguish them, and identify which component is responsible.
4. A probe recovers dependency labels from layer 7 at 91% accuracy. The control task scores 88%. What do you conclude, and what would you need to know to conclude more?
5. **Harder.** *book* is polysemous and *bank* is homonymous. Describe the geometry you would expect for each in a contextual space, and propose a measurement that would distinguish the two cases. What would a negative result mean?

Exercise 5 is a real open question, and a reasonable project topic.

Summary

Point	Detail
The commitment	one vector per <i>type</i> was inherited from the matrix, never argued for
Sparse failure	PPMI is winner-take-all: the minority sense is clipped to zero, not diluted
Dense failure	a frequency-weighted average; at 50/50 the vector is 45° from both senses
Scope of the problem	the most frequent words are the most polysemous
Homonymy vs polysemy	bank is two lexemes; book is one with structure — the GL distinction
Arora et al.	senses are recoverable by sparse coding; the superposition is structured
The move	$w \mapsto \mathbf{v}_w$ becomes $(w, \text{context}) \mapsto \mathbf{v}_{w, \text{context}}$
Consequence	no embedding table; the model <i>is</i> the lexicon
Attention	$\mathbf{o} = \sum \alpha_i \mathbf{v}_i$ — a weighted average under a learned similarity
Attention is not application	it selects and mixes; it does not consume an argument or change type
Order	self-attention is permutation-equivariant; order is an additive repair
Probing	the classical pipeline by layer; parse distance as a linear metric
Control tasks	selectivity, not accuracy — the question is always “compared to what?”
Anisotropy	raw cosine misleads in upper layers; Session 7’s problem, relearned
Not found	truth conditions, scope, monotonicity, a composition operation
Claim (2)	false, and abandoned rather than repaired
What it fixed	everything caused by the <i>type</i> being the unit
What it did not	anything caused by <i>distribution</i> being the evidence

Reading

Reading: Lappin & Shanahan Ch. 5. **Companion deck:** Session 9A, *Inside the Transformer*.

Primary sources:

- Peters et al. (2018), *Deep Contextualized Word Representations* (ELMo)
- Devlin, Chang, Lee & Toutanova (2019), *BERT*
- Tenney, Das & Pavlick (2019), *BERT Rediscovered the Classical NLP Pipeline*
- Hewitt & Manning (2019), *A Structural Probe for Finding Syntax in Word Representations*
- Hewitt & Liang (2019), *Designing and Interpreting Probes with Control Tasks* — read this one alongside the two above, not after
- Ethayarajh (2019), *How Contextual are Contextualized Word Representations?*
- Ettinger (2020), *What BERT Is Not*
- Arora, Li, Liang, Ma & Risteski (2018), *Linear Algebraic Structure of Word Senses*
- Bender & Koller (2020), *Climbing towards NLU*; Pavlick (2023), *Symbols and Grounding in LLMs*

Rogers, Kovaleva & Rumshisky (2020), *A Primer in BERTology*, is the survey if you want the whole probing landscape in one place.

Questions?

Next: we have a vector for every token, in context,
and still cannot say what *red car* is