

CS 135: Computational Semantics

From Logic to Geometry

Brandeis University · Department of Computer Science · Fall 2026

Course Information

Instructor: James Pustejovsky

TA: Bohan Jiang

Class Time: Tuesday, Friday: 11:10 am - 12:30 pm

Course Description

This course studies the computational treatment of meaning in natural language, following a single question across two traditions: what structure does meaning composition require?

The first unit develops the logical answer – the lambda calculus, semantic types, models, and a working interpreted fragment of English – with quantification and scope as the unit's capstone phenomena. The second unit develops the distributional answer: meaning as vectors learned from text. It ends at the composition problem, where vector representations turn out to lose exactly the structural information (who did what to whom) that the logical tradition captures naturally.

The final and largest unit develops a candidate reconciliation. We first recover a forty-year research tradition that standard NLP histories skip: the program of building structured, compositional meaning inside vector spaces (tensor product representations, holographic reduced representations, vector symbolic architectures, categorical models). We then build the mathematics that tradition kept reaching for – geometric, or Clifford, algebra: blades, grades, the geometric product, contraction, and rotors. Finally, we study Functional Geometric Algebra (FGA), a recent framework in which grade encodes arity, the wedge product implements binding, contraction implements function application, and rotors implement type coercion.

You will implement a compositional fragment of English twice – once in the logical tradition and once in the geometric one – and evaluate the comparison yourself. The course closes with genuinely open research problems in this area, and final projects may engage them directly.

Prerequisites

COSI 10A or COSI 12b or equivalent (introductory programming, in any language), and LING 130a (semantics) or COSI 121b (functional programming), or permission of the instructor.

No prior Haskell or functional programming experience is assumed. The first problem set uses a provided Haskell scaffold in which you write semantic definitions, not infrastructure; the language itself is covered in the first three lab sessions. Later programming assignments use Python.

Comfort with basic linear algebra (vectors, inner products, matrices) is strongly recommended for the second half of the course. A short self-assessment goes out in Week 1, along with review materials for anyone who wants them. If you do not meet the stated criteria but are interested in the course, please come talk to us.

Learning Goals

By the end of the course, you will be able to:

- represent and compute meaning in both logical and distributional frameworks, and articulate the strengths and limits of each;
- carry out type-driven semantic derivations, including quantification and scope, and evaluate them against models;
- implement a compositional fragment of English and test it computationally;
- work with the basic constructions of geometric algebra and apply them to semantic representation;
- carry out and evaluate mappings between model-theoretic artifacts and vectorial encodings;
- read, evaluate, and communicate about contemporary research in computational semantics, including work whose central questions are still open;
- integrate these skills in a substantial final project.

Workload. This is a four-credit course. Expect a minimum of nine hours of work per week outside of class, including lab.

Readings

There is no single required textbook to purchase. Chapters and papers are posted on Moodle; several texts are freely available online.

- **EU** — van Eijck & Unger (2010), *Computational Semantics with Functional Programming*. Selected chapters (Moodle).
- **JM** — Jurafsky & Martin, *Speech and Language Processing*, 3rd edition draft. Freely available online.
- **LS** — Lenci & Sahlgren (2023), *Distributional Semantics*. Cambridge University Press (Moodle).
- **D** — Dorst, Fontijne & Mann (2007), *Geometric Algebra for Computer Science* (Moodle).
- **M** — Macdonald (2010), *Linear and Geometric Algebra*. Optional; a gentler entry point if you are reviewing linear algebra.
- **W** — Widdows (2004), *Geometry and Meaning*. CSLI Publications.
- **P** — Papers packet (Moodle): Smolensky (1990); Plate (1995); Coecke, Sadrzadeh & Clark (2010); Pustejovsky (2026) *Toward a Functional Geometric Algebra for Natural Language Semantics*; Pustejovsky & Krishnaswamy (2026) *Frictive Policy Optimization*; Obiso (2024) *Holographic Embeddings for Text and Graphs*; and additional readings announced during the term.
- **SEP** — Stanford Encyclopedia of Philosophy, “Generalized Quantifiers” (optional, Session 5).

Schedule of Topics

The schedule below is our plan; adjustments will be announced in class and on the course site. Session 16 is a deliberate flex session — see the note following the tables.

Part I — Meaning as Logic

Wk	Date	Sess.	Topic	Readings	Assignments
1	Fri 8/28	1	Course introduction; the lambda calculus	—	Linear algebra self-assessment out
2	Tue 9/1	2	Semantic types; type-driven interpretation	EU Ch. 2; JM App. F	
2	Fri 9/4	3	Predicate logic and models	EU 3.6–3.9	PS 1 out
3	Tue 9/8	4	The Fragment of English: lexicon, composition, model checking	EU Ch. 6	
3	Fri 9/11	5	Quantifiers and scope: generalized quantifiers, ambiguity, readings	EU 7.1–7.4; SEP (opt.)	
4	Tue 9/15	6	Beyond the sentence: modality, intensionality, dynamics	Course slides	

Part II — Meaning as Distribution

Wk	Date	Sess.	Topic	Readings	Assignments
4	Fri 9/18	7	The distributional hypothesis; co-occurrence, PPMI	LS 2.2–2.4	
5	Tue 9/22	8	Dimensionality reduction; word2vec; static embeddings	LS 2.4–2.7, Ch. 4	
5	Fri 9/25	9	Contextual embeddings; transformers as semantic models	LS Ch. 5	PS 1 due; Quiz 1 out
6	Tue 9/29	10	The composition problem; Generative Lexicon and word embeddings	LS Ch. 6; slides	Quiz 1 due Mon 9/28; PS 2 out; project guidelines out

Part III — The Algebraic Thread and Geometric Algebra

Wk	Date	Sess.	Topic	Readings	Assignments
6	Fri 10/2	11	The algebraic thread in NLP: tensor products, HRR, VSA, categorical models, Hopf coproducts	P: Smolensky; Plate; Coecke et al.; W Chs. 1–3	
7	Tue 10/6	12	Geometric algebra I: vectors, bivectors, the geometric product	D Chs. 1–2; M as needed	
7	Fri 10/9	13	Geometric algebra II: multivectors, grades, blades	D Chs. 3–5	
8	Tue 10/13	14	Geometric algebra III: contraction, duality, projection	D Chs. 6–7	PS 2 due
8	Fri 10/16	15	Geometric algebra IV: rotors, signatures $Cl(p,q)$, the assembled toolkit	D Ch. 10	Quiz 2 out
9	Tue 10/20	16	Flex session: worked examples across the geometric algebra toolkit	—	Quiz 2 due Mon 10/19

Part IV — Functional Geometric Algebra

Wk	Date	Sess.	Topic	Readings	Assignments
9	Fri 10/23	17	The correspondence: model-theoretic artifacts as vectorial encodings	P: FGA paper §§1–2	PS 3 out
10	Tue 10/27	18	FGA I: grade as arity; wedge as binding	P: FGA §3	
10	Fri 10/30	19	FGA II: contraction as unbinding and function application	P: FGA §4; lambda note Pt. I	
11	Tue 11/3	20	FGA III: predication and quantification	P: lambda note Pt. II	
11	Fri 11/6	21	FGA IV: coercion as rotor action; type-shifting; GL connections	P: FGA §5; GL readings	Project proposals due
12	Tue 11/10	22	FGA over knowledge graphs: hyper-relational representation	P: handout	
12	Fri 11/13	23	Learning in FGA; policy optimization, friction, and alignment	P: FPO paper	
13	Tue 11/17	24	Open problems and the research frontier	Open-problems handout	

Projects

Wk	Date	Sess.	Topic	Readings	Assignments
13	Fri 11/20	25	Project workshop: proposals presented, feedback	—	PS 3 due
14	Tue 11/24	26	AMA: Ask Me Anything Session		
15	Tue 12/1	27	Final project presentations I	—	
15	Fri 12/4	28	Final project presentations II	—	Final writeups due during exam period

A note on the flex session. Sessions 12–15 build a body of mathematics quickly. Session 16 exists so that we can slow down if the material needs more time, and so that we can work examples together if it does not. Please use it: if the geometric algebra is not landing, say so before Session 16 rather than after.

Assignments and Grading

Component	Weight
Problem Set 1 — A Fragment of English	15%
Problem Set 2 — Distributional Representations	15%
Problem Set 3 — The Fragment in Geometric Algebra	15%
Quiz 1 (take-home)	10%
Quiz 2 (take-home)	10%
Final project	30%

Component	Weight
Participation and lab	5%

Problem Set 1 — A Fragment of English (15%)

Out Week 2, due Week 5. You are given a working Haskell file containing a model, type declarations, a test harness, and worked examples. About twenty definitions in it are holes. You fill them in: lexical denotations, composition rules, sentence derivations, and both scope readings of an ambiguous sentence. Everything you write is semantics — you will not build a parser, a model, or a harness, and nearly every answer is a single line.

A short writeup accompanies the code: the scope reasoning, one type error you encountered and what it told you about composition, and one limitation of the fragment. **This fragment is the reference implementation for Problem Set 3.**

Problem Set 2 — Distributional Representations (15%)

Out Week 6, due Week 8. Build distributional representations from a provided corpus (PPMI and learned embeddings) and evaluate them on similarity tasks. A required diagnostic section asks you to compose representations for supplied adjective–noun and noun–noun pairs using addition, multiplication, and concatenation, and to demonstrate precisely where and how each fails. Python.

Problem Set 3 — The Fragment in Geometric Algebra (15%)

Out Week 9, due Week 13. Reimplement the Problem Set 1 fragment in geometric algebra, in Python, using a provided scaffold parallel to the first one: entities and predicates as multivectors, binding via the wedge product, application via contraction, and at least one coercion via a rotor.

The accompanying writeup is the point of the assignment: what does each substrate make easy, what does each make awkward, and what is at least one phenomenon from Problem Set 1 that your geometric implementation does not handle — and why? A well-argued negative result is worth full credit.

Quizzes (10% each)

Both quizzes are take-home, distributed Friday and due the following Monday. Quiz 1 covers Part I: lambda calculus, types, models, quantification and scope. Quiz 2 covers Parts II and III: distributional semantics, the composition problem, and geometric algebra.

Final Project (30%)

Proposals are due Week 11 and workshopped in Week 13; presentations are in Week 15 and the written report (6–8 pages, ACL format) is due during the exam period. Projects may extend any part of the course.

A set of **open-problem tracks** will be distributed with the project guidelines in Week 6 and discussed in detail in Session 24. These are real unsolved questions, and projects that take them on are evaluated on the quality of the investigation — not on whether the approach turns out to work. A rigorous negative result is a successful project.

Course Policies

Collaboration

Discussion of concepts, readings, and general approaches with your classmates is encouraged. Code and written work must be your own. If you discuss an assignment with someone, name them in your submission — this costs you nothing and keeps everyone safe.

Use of AI assistants

You may use AI assistants for conceptual questions, debugging help, and language support. You may not submit generated code or prose as your own. Any substantive use must be disclosed in a short note with the submission describing what you used and how. The assignments in this course are designed so that the understanding, not the artifact, is what is being assessed; outsourcing the former is both detectable and self-defeating.

Late work

If something is going wrong — academically or otherwise — contact us before the deadline rather than after. We are considerably more flexible in advance than in retrospect.

Accessibility

Brandeis seeks to create a learning environment that is welcoming and inclusive of all students, and I want to support your learning. If you have an accommodation letter from Student Accessibility Support, please share it with me early in the term so we can discuss implementation. If you have not yet contacted SAS but believe you may need accommodations, please reach out to them or to me.

Academic integrity

You are expected to be familiar with and to follow the University's policies on academic integrity (see the Office of Academic Integrity website). Instances of alleged dishonesty will be forwarded to the Department of Student Rights and Community Standards. Potential sanctions include failure of the assignment or course, and educational or disciplinary measures.

Getting help

Come to office hours. This is a course that moves through several quite different bodies of material — logic, machine learning, and abstract algebra — and it is entirely normal for one of them to be much harder for you than the others. That is not a signal that you are in the wrong course; it is a signal to come talk to us early.