Compositional Models of Vector-based Semantics: From Theory to Tractable Implementation

Day 1: What's in a vector-based model of compositionality?

Gijs Wijnholds & Michael Moortgat

ESSLLI 2022

Abstract

Vector-based compositional architectures combine a distributional view of word meanings with a modelling of the syntax-semantics interface as a structure-preserving map relating syntactic categories (types) and derivations to their counterparts in a corresponding meaning algebra.

This design is theoretically attractive, but faces challenges when it comes to large-scale practical applications. First there is the curse of dimensionality resulting from the fact that semantic spaces directly reflect the complexity of the types of the syntactic front end. Secondly, modelling of the meaning algebra in terms of finite dimensional vector spaces and linear maps means that vital information encoded in syntactic derivations is lost in translation.

The course compares and evaluates methods that are being proposed to face these challenges. Participants gain a thorough understanding of theoretical and practical issues involved, and acquire hands-on experience with a set of user-friendly tools and resources.

Course Information

Website: https://compositioncalculus.sites.uu.nl/course/

Outline:

Day 1.	What's in a vector-based model of compositionality?	Michael & Gijs
Day 2.	The curse of dimensionality	Gijs
Day 3.	Learning a lexicon for supertagging	Michael
Day 4.	Parsing with Graph Neural Networks	Kokos
Day 5.	Evaluation: Experimenting with semantic tasks	Gijs

Team:

Gijs Wijnholds
Lecturer

Michael Moortgat
Lecturer

Kokos Kogkalidis
Teaching Assistant

Today: what's in a compositional Vector-based model?

Day plan

- What is compositionality?
- What is the syntactic backbone for compositional vector-based models?
- What is vector-based semantics?
- Linking syntax to semantics; using vectors/embeddings
- Ingredients for succes: learnable representations, learnable grammars, learnable parser, model evaluation

Background reading Some useful textbook refs:

- R. Moot and Ch. Retoré, 2012, The Logic of Categorial Grammars, LNCS 6850, Springer. url
- Sheldon Axler, 2015, Linear Algebra Done Right, 3rd edition, Undergraduate Texts in Mathematics, Springer. url

Compositional interpretation: Montague's view

Frege's Principle a central design principle of computational semantics:

> 'the meaning of an expression is a function of the meaning of its parts and of the way they are syntactically combined' (Partee)

Montague gives Frege's Principle a precise mathematical form:

$$\text{Source} \xrightarrow{h} \text{Target}$$

- Source: a multi-sorted algebra for syntax $\langle (A_s)_{s \in S}, (F_\gamma)_{\gamma \in \Gamma} \rangle$
- Target: a multi-sorted algebra for semantics $\langle (B_t)_{t \in T}, (G_\delta)_{\delta \in \Delta} \rangle$
- Interpretation: homomorphism h, i.e. a structure-preserving map

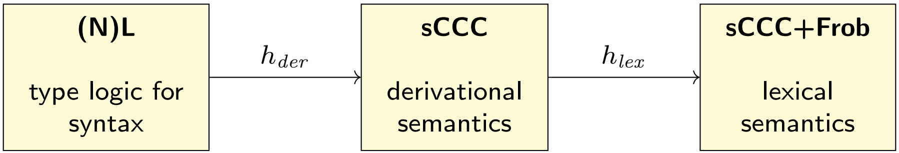
  h respects the sorts: $\{h(a) \mid a \in A_s\} \subseteq B_{\sigma(s)}$

  h respects the operations: $h(F_\gamma(a_1, \ldots, a_n)) = G_{\rho(\gamma)}(h(a_1), \ldots, h(a_n))$

Montague, Universal Grammar, 1970

Sneak Peek: Vector-Based Compositional Interpretation

M's program in a type-logical setting, with two-step interpretation homomorphism:



- Front end: **(N)L**, Lambek-style syntactic calculus

  model: residuated monoid/semigroup/groupoid

- Target for h_{der}: **sCCC**, symmetric compact closed category

  model: **FVect**, finite-dimensional vector spaces and linear maps

- Target for h_{lex}: **FVect** expanded with with Frobenius algebras

  word-internal semantics, allowing for 'duplication'/'merging' of information.

Syntax

Lambek's program for syntax Slogan: 'parsing-as-deduction'

- categories ('noun', 'verb', ...) $\rightsquigarrow$ logical formulas/types:
- well-formedness judgement $\rightsquigarrow$ formal derivation/'proof' in grammatical type logic

THE MATHEMATICS OF SENTENCE STRUCTURE*

JOACHIM LAMBEK, McGill University

The definitions [of the parts of speech] *are very far from having attained the degree of exactitude found in Euclidean geometry.*

—Otto Jespersen, 1924.

1. Introduction. The aim of this paper is to obtain an effective rule (or algorithm) for distinguishing sentences from nonsentences, which works not only for the formal languages of interest to the mathematical logician, but also for natural languages such as English, or at least for fragments of such languages. An attempt to formulate such an algorithm is implicit in the work of Ajdukiewicz.† His method, later elaborated by Bar-Hillel [2], depends on a kind of arithmetization of the so-called *parts of speech*, here called *syntactic types*.‡

Types

Basic types We start from a small set of atomic types for expressions that are considered 'complete'. For example

- s sentences
- n common nouns ('poet', 'politician', ...)
- np noun phrases ('Macron', 'the president', ...)

Types The full set of types is given by the grammar below (p: atomic types)

$$A, B \quad ::= \quad p \mid A \bullet B \mid A/B \mid B\backslash A$$

- $A \bullet B$: concatenation 'A and then B'
- A/B: 'A over B'; right division: combines with B on the right to produce A
- $B\backslash A$: 'B under A'; left division: combines with B on the left to produce A

Syntactic calculus (N)L

Lambek's original presentation of **(N)L** considers statements $A \longrightarrow B$, i.e. derivability is modelled as a relation holding between types.

Pre-order laws The derivability relation is reflexive and transitive:

$$A \longrightarrow A \qquad \frac{A \longrightarrow B \quad B \longrightarrow C}{A \longrightarrow C}$$

Residuation laws

$$B \longrightarrow A\backslash C \quad \textit{iff} \quad A \bullet B \longrightarrow C \quad \textit{iff} \quad A \longrightarrow C/B \qquad \text{[L61]}$$

Structural laws

$$A \bullet (B \bullet C) \longleftrightarrow (A \bullet B) \bullet C \qquad \text{[L58]}$$

$$I \bullet A \longleftrightarrow A \longleftrightarrow A \bullet I \qquad \text{[L88]}$$

L61: **NL**, [Lambek, 1961]; L58: **L**, [Lambek, 1958]; L88: [Lambek, 1988]

Model: residuated monoids/groupoids

The intended models for the type logic are the multiplicative systems freely generated by the words of the language under concatenation.

Types as sets of expressions, i.e. subsets of a groupoid/semigroup/monoid $\langle M, \cdot \rangle$ with

$$\begin{array}{ccl} A \bullet B & = & \{a \cdot b \in M \mid a \in A \wedge b \in B\} \\ C/B & = & \{a \in M \mid \forall_{b \in B}\ a \cdot b \in C\} \\ A \backslash C & = & \{b \in M \mid \forall_{a \in A}\ a \cdot b \in C\} \\ I & = & \{1\} \end{array}$$

- groupoid [L61], types assigned to phrases, bracketed strings
- semigroup [L58], types assigned to strings, associative multiplication
- monoid [L88], multiplicative unit, empty string

Background: category theory

Category theory: abstract study of mathematical structures. Ingredients:

- objects: the elements of a category
- morphisms ('arrows'): transformations, notation $f : A \longrightarrow B$
  - for each object we have the identity morphism, $1_A : A \longrightarrow A$
  - composition: given $f : A \longrightarrow B$ and $g : B \longrightarrow C$, we have $g \circ f : A \longrightarrow C$

$\circ$ associative: $f \circ (g \circ h) = (f \circ g) \circ h$; given $f : A \to B$ we have $f \circ 1_A = f$, $1_B \circ f = f$.

Deductive systems as categories Objects $\rightsquigarrow$ formulas; morphisms $\rightsquigarrow$ derivations; operations on morphisms $\rightsquigarrow$ rules of inference. For **NL** add:

$$\frac{f : A \bullet B \longrightarrow C}{\triangleright f : A \longrightarrow C/B} \qquad \frac{g : A \longrightarrow C/B}{\triangleright^{-1} g : A \bullet B \longrightarrow C} \qquad \frac{f : A \bullet B \longrightarrow C}{\triangleleft f : B \longrightarrow A \backslash C} \qquad \frac{g : B \longrightarrow A \backslash C}{\triangleleft^{-1} g : A \bullet B \longrightarrow C}$$

with appropriate conditions guaranteeing this is a well-behaved category ...

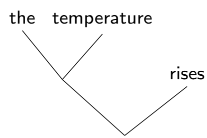
[Wijnholds, 2015]

Example: ‘the temperature rises’

$$\dfrac{\dfrac{\dfrac{\dfrac{}{np/n \longrightarrow np/n}\;1_{np/n}}{(np/n)\bullet n \longrightarrow np}\;\triangleright^{-1} \qquad \dfrac{\dfrac{\dfrac{}{np\backslash s \longrightarrow np\backslash s}\;1_{np\backslash s}}{np\bullet(np\backslash s) \longrightarrow s}\;\triangleleft^{-1}}{np \longrightarrow s/(np\backslash s)}\;\triangleright}{(np/n)\bullet n \longrightarrow s/(np\backslash s)}\;\circ}{(\underbrace{(np/n)}_{\text{the}} \bullet \underbrace{n}_{\text{temperature}}) \bullet \underbrace{(np\backslash s)}_{\text{rises}} \longrightarrow s}\;\triangleright^{-1}$$

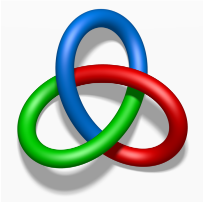
Combinator for this derivation: $\triangleright^{-1}((\triangleright \triangleleft^{-1} 1_{np\backslash s}) \circ (\triangleright^{-1} 1_{np/n}))$

Remark bracketing of the $\bullet$ formula $\approx$ tree structure.



Curry-Howard-Lambek correspondence

The categorical presentation is rather heavy-handed. In what follows we freely move between different perspectives on type logics that are essentially equivalent, thanks to the Curry-Howard-Lambek correspondence



CHL originally:

- math: category theory — Cartesian closed categories
- logic: deductive systems — Intuitionistic logic
- computation: proofs as programs — λ calculus

[Lambek and Scott, 1986]

A user-friendly format: Natural Deduction

Structures, sequents Derivability as a relation between structures and types. A sequent is a statement $\Gamma \vdash A$ with A a type, Γ a structure. Grammar for structures:

$$\Gamma, \Delta \quad ::= \quad A \mid \Gamma \cdot \Delta$$

i.e. atomic structures: types; complex structures are formed with the 2-place $\cdot$, structural counterpart of $\bullet$.

Axiom, logical rules For the base logic, we have the axiom $A \vdash A$ and as logical inference rules, for each connective an elimination rule and an introduction rule.

$$\frac{\Gamma \vdash A \quad \Delta \vdash A\backslash B}{\Gamma \cdot \Delta \vdash B}\ \backslash E \qquad \frac{\Gamma \vdash B/A \quad \Delta \vdash A}{\Gamma \cdot \Delta \vdash B}\ /E$$

$$\frac{A \cdot \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ \backslash I \qquad \frac{\Gamma \cdot A \vdash B}{\Gamma \vdash B/A}\ /I$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma \cdot \Delta \vdash A \bullet B}\ \bullet I \qquad \frac{\Delta \vdash A \bullet B \quad \Gamma[A \cdot B] \vdash C}{\Gamma[\Delta] \vdash C}\ \bullet E$$

Notation: $\Gamma[\Delta]$ for a structure Γ containing a substructure Δ

Examples

Hypothetical reasoning Type lifting.

$$\dfrac{\dfrac{}{np \vdash np}\,Ax \quad \dfrac{}{np\backslash s \vdash np\backslash s}\,Ax}{\dfrac{np \cdot np\backslash s \vdash s}{np \vdash s/(np\backslash s)}\,/I}\,\backslash E$$

Official format vs steno Use words instead of their types left of $\vdash$. Compare

$$\dfrac{\dfrac{\dfrac{}{np/n \vdash np/n}\,Ax \quad \dfrac{}{n \vdash n}\,Ax}{(np/n) \cdot n \vdash np}\,/E \quad \dfrac{}{np\backslash s \vdash np\backslash s}\,Ax}{(\underbrace{(np/n)}_{\text{the}} \cdot \underbrace{n}_{\text{temperature}}) \cdot \underbrace{(np\backslash s)}_{\text{rises}} \vdash s}\,\backslash E$$

$$\dfrac{\dfrac{\dfrac{}{\textsf{the} \vdash np/n}\,Ax \quad \dfrac{}{\textsf{temperature} \vdash n}\,Ax}{\textsf{the} \cdot \textsf{temperature} \vdash np}\,/E \quad \dfrac{}{\textsf{rises} \vdash np\backslash s}\,Ax}{(\textsf{the} \cdot \textsf{temperature}) \cdot \textsf{rises} \vdash s}\,\backslash E$$

N.D.: structural rules

Postulate extensions of the base logic take the form of structural rules. Formula variables $\rightsquigarrow$ structure variables (in context).

Associativity Compare

$$(A \bullet B) \bullet C \longrightarrow A \bullet (B \bullet C) \qquad \rightsquigarrow \qquad \frac{\Gamma[\Delta \cdot (\Delta' \cdot \Delta'')] \vdash D}{\Gamma[(\Delta \cdot \Delta') \cdot \Delta''] \vdash D}$$

$$A \bullet (B \bullet C) \longrightarrow (A \bullet B) \bullet C \qquad \rightsquigarrow \qquad \frac{\Gamma[(\Delta \cdot \Delta') \cdot \Delta''] \vdash D}{\Gamma[\Delta \cdot (\Delta' \cdot \Delta'')] \vdash D}$$

Implicit structural rules A sugared presentation of **L** leaves restructuring under associativity implicit, using sequents $\Gamma \vdash B$ with Γ a list of formulas $A_1, \ldots, A_n$.

Your turn

Monotonicity Use the categorical presentation of **NL** to show that the monotonicity laws below are derived rules of inference.

$$\frac{A \to B \quad C \to D}{A \bullet C \to B \bullet D} \bullet \qquad \frac{A \to B \quad C \to D}{A/D \to B/C} / \qquad \frac{C \to D \quad A \to B}{D\backslash A \to C\backslash B} \backslash$$

Sequents and arrows establish the following:

for every N.D. proof $X \vdash B$ there is an arrow $f : \overline{X} \longrightarrow B$

where $\overline{X}$ is the formula version of X: $\overline{A} = A$, $\overline{X \cdot Y} = \overline{X} \bullet \overline{Y}$.

Ambiguity: Lexical

Distinct derivations resulting from lexical ambiguity.

Noun modification

$$\dfrac{\dfrac{\text{Fallada}}{np} \quad \dfrac{\dfrac{\text{wrote}}{(np\backslash s)/np} \quad \dfrac{\dfrac{\text{a}}{np/n} \quad \dfrac{\dfrac{\text{book}}{n} \quad \dfrac{\dfrac{\text{on}}{(n\backslash n)/np} \quad \dfrac{\text{drugs}}{np}}{(\text{on} \cdot \text{drugs}) \vdash n\backslash n}\,/E}{(\text{book} \cdot (\text{on} \cdot \text{drugs})) \vdash n}\,\backslash E}{(\text{a} \cdot (\text{book} \cdot (\text{on} \cdot \text{drugs}))) \vdash np}\,/E}{\text{wrote} \cdot (\text{a} \cdot (\text{book} \cdot (\text{on} \cdot \text{drugs})))) \vdash np\backslash s}\,/E}{(\text{Fallada} \cdot (\text{wrote} \cdot (\text{a} \cdot (\text{book} \cdot (\text{on} \cdot \text{drugs}))))) \vdash s}\,\backslash E$$

Verb phrase modification

$$\dfrac{\dfrac{\text{Fallada}}{np} \quad \dfrac{\dfrac{\dfrac{\text{wrote}}{(np\backslash s)/np} \quad \dfrac{\dfrac{\text{a}}{np/n} \quad \dfrac{\text{book}}{n}}{(\text{a} \cdot \text{book}) \vdash np}\,/E}{\text{wrote} \cdot (\text{a} \cdot \text{book}) \vdash np\backslash s}\,/E \quad \dfrac{\dfrac{\text{on}}{((np\backslash s)\backslash(np\backslash s))/np} \quad \dfrac{\text{drugs}}{np}}{(\text{on} \cdot \text{drugs}) \vdash (np\backslash s)\backslash(np\backslash s)}\,/E}{(\text{wrote} \cdot (\text{a} \cdot \text{book})) \cdot (\text{on} \cdot \text{drugs}) \vdash np\backslash s}}{(\text{Fallada} \cdot ((\text{wrote} \cdot (\text{a} \cdot \text{book})) \cdot (\text{on} \cdot \text{drugs}))) \vdash s}\,\backslash E$$

Ambiguity: Derivational

Consider the **L** derivation below:

$$\dfrac{\dfrac{\dfrac{\text{French}}{n/n} \quad \dfrac{\text{wine}}{n}}{\text{French wine} \vdash n}\,/E \qquad \dfrac{\text{drinker}}{n\backslash n}}{\text{French wine drinker} \vdash n}\,\backslash E \qquad\qquad \dfrac{\dfrac{\text{French}}{n/n} \qquad \dfrac{\dfrac{\text{wine}}{n} \quad \dfrac{\text{drinker}}{n\backslash n}}{\text{wine drinker} \vdash n}\,\backslash E}{\text{French wine drinker} \vdash n}\,/E$$

Compositional interpretation Meaning must be assigned to derivational history, not to strings. We can keep track of the derivational history by associating a proof with a term recording the inference steps.

Proofs and terms

N.D. derivations are now seen as typing judgements, checking whether a program (=term) is well-typed. Sequents take the form

$$x_1 : A_1, \ldots, x_n : A_n \vdash M : B \qquad\qquad x_i \text{ fresh}$$

meaning: program M is a well-formed expression of type B given type declarations $x_1 : A_1, \ldots, x_n : A_n$ (a typing environment).

We compare the implication fragment of **(N)L** with the semantic type calculus **LP**, with non-directional types $A \multimap B$.

Proofs and terms: syntactic calculus

Types

$$A, B \quad ::= \quad s \mid np \mid n \mid A\backslash B \mid B/A$$

Terms left vs right application/abstraction

$$M, N \quad ::= \quad x \quad \mid \quad \lambda^r x.M \quad \mid \quad \lambda^l x.M \quad \mid \quad (M \ltimes N) \quad \mid \quad (N \rtimes M)$$

Typing rules Axiom $x : A \vdash x : A$

$$\frac{\Gamma \cdot x : A \vdash M : B}{\Gamma \vdash \lambda^r x.M : B/A}\ I/ \qquad \frac{x : A \cdot \Gamma \vdash M : B}{\Gamma \vdash \lambda^l x.M : A\backslash B}\ I\backslash$$

$$\frac{\Gamma \vdash M : B/A \quad \Delta \vdash N : A}{\Gamma \cdot \Delta \vdash (M \ltimes N) : B}\ E/ \qquad \frac{\Gamma \vdash N : A \quad \Delta \vdash M : A\backslash B}{\Gamma \cdot \Delta \vdash (N \rtimes M) : B}\ E\backslash$$

Semantic type calculus: LP

LP extends **L** with product commutativity. **LP** is a.k.a. MILL, Multiplicative Intuitionistic Linear Logic. In MILL, the slashes $/, \backslash$ collapse to linear implication $\multimap$.

Types, terms $A, B ::= e \mid t \mid A \multimap B$; $\qquad M, N ::= x \mid \lambda x.M \mid M\ N$

Typing rules Axiom $x : A \vdash x : A$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \multimap B}\ (\multimap I) \qquad \frac{\Gamma \vdash M : A \multimap B \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash M\ N : B}\ (\multimap E)$$

LP: Set Theoretic Models

Each type A is associated with a semantic domain D_A, based on a non-empty set E (the universe):

$$D_e = E \qquad D_t = \{\text{true}, \text{false}\} \qquad D_{A \multimap B} = D_B^{D_A} \quad \text{linear functions from } D_A \text{ to } D_B$$

Semantic value $D_A \ni [\![M^A]\!]_g$ semantic value of an expression M of type A w.r.t. assignment g:

- Atoms: $[\![x^A]\!]_g = g(x^A)$
- Application: $[\![(M^{A \multimap B}\ N^A)]\!]_g = [\![M^{A \multimap B}]\!]_g [\![N^A]\!]_g$
- Abstraction:

  $[\![\lambda x^A.M^B]\!]_g = f \in D_B^{D_A}$ such that for each $d \in D_A$, $f(d) = [\![M^B]\!]_{g'}$

  where g' is exactly like g except perhaps that x^A is assigned the value d

Add interpretation function for constants, if present: $D_A \ni [\![c^A]\!]_g^I = I(c^A)$

Compositional Interpretation: Toy Example

An example of compositional interpretation taken from your Intro Formal Semantics.

The homomorphism $\lceil\cdot\rceil$ maps types and derivations of the syntactic source logic to their counterparts in the target logic for semantics.

$$\textbf{(N)L}^{s,np,n}_{/,\backslash} \qquad \xrightarrow{\lceil\cdot\rceil} \qquad \textbf{LP}/\mathrm{MILL}^{e,t}_{\multimap}$$

Types

$$\lceil s\rceil = t \quad \lceil np\rceil = e \quad \lceil n\rceil = e \multimap t \qquad \lceil A\backslash B\rceil = \lceil B/A\rceil = \lceil A\rceil \multimap \lceil B\rceil$$

Terms

$$\begin{array}{rcl} \lceil x\rceil & = & \widetilde{x} \\ \lceil \lambda^l x.M)\rceil = \lceil \lambda^r x.M\rceil & = & \lambda\widetilde{x}.\lceil M\rceil \\ \lceil N \rtimes M\rceil = \lceil M \ltimes N\rceil & = & \lceil M\rceil\ \lceil N\rceil \end{array}$$

Illustration: ‘paper that Bob rejected’

$$
\begin{prooftree}
\AxiomC{paper}
\UnaryInfC{n}
\AxiomC{that}
\UnaryInfC{$(n\backslash n)/(s/np)$}
\AxiomC{Bob}
\UnaryInfC{np}
\AxiomC{rejected}
\UnaryInfC{$(np\backslash s)/np$}
\AxiomC{$[np \vdash np]^1$}
\RightLabel{$[/E]$}
\BinaryInfC{$\text{rejected} \cdot np \vdash np\backslash s$}
\RightLabel{$[\backslash E]$}
\BinaryInfC{$\text{Bob} \cdot (\text{rejected} \cdot np) \vdash s$}
\RightLabel{$[Ar]$}
\UnaryInfC{$(\text{Bob} \cdot \text{rejected}) \cdot np \vdash s$}
\RightLabel{$[/I]^1$}
\UnaryInfC{$\text{Bob} \cdot \text{rejected} \vdash s/np$}
\RightLabel{$[/E]$}
\BinaryInfC{$\text{that} \cdot (\text{Bob} \cdot \text{rejected}) \vdash n\backslash n$}
\RightLabel{$[\backslash E]$}
\BinaryInfC{$\text{paper} \cdot (\text{that} \cdot (\text{Bob} \cdot \text{rejected})) \vdash n$}
\end{prooftree}
$$

Source term $M = \text{paper} \rtimes (\text{that} \ltimes \lambda^r x.(\text{Bob} \rtimes (\text{rejected} \ltimes x))) : n$

Translation: derivational semantics word meanings as black boxes

$$\lceil M \rceil = ((\lceil \text{that} \rceil\ \lambda x.((\lceil \text{rejected} \rceil\ x)\ \lceil \text{Bob} \rceil))\ \lceil \text{paper} \rceil) : e \multimap t$$

Illustration (cont'd)

Lexical semantics Assuming the target signature includes constants $\textsc{paper}^{e\multimap t}$, $\textsc{Bob}^{e}$, $\textsc{rejected}^{e\multimap e\multimap t}$, $\wedge^{t\multimap t\multimap t}$, we can unpack the black box word meanings:

WORD	SYN TYPE	$\lceil\cdot\rceil$	SEM TYPE
paper	n	$\textsc{paper}$	$e \multimap t$
that	$(n\backslash n)/(s/np)$	$\lambda x\lambda y\lambda^{!}z.((y\ z) \wedge (x\ z))$	$(e \multimap t) \multimap (e \multimap t) \multimap (!\,e \multimap t)$
Bob	np	$\textsc{Bob}$	e
rejected	$(np\backslash s)/np$	$\textsc{rejected}$	$e \multimap e \multimap t$

non-linear recipe for 'that': z is duplicated

Substituting these in $\lceil M\rceil$

$$\lceil M\rceil = ((\lceil\text{that}\rceil\ \lambda x.((\lceil\text{rejected}\rceil\ x)\ \lceil\text{Bob}\rceil))\ \lceil\text{paper}\rceil) : e \multimap t$$

(and simplifying) produces the final result

$$\lambda x^{!}.((\textsc{paper}\ x) \wedge ((\textsc{rejected}\ x)\ \textsc{Bob})) : !\,e \multimap t$$

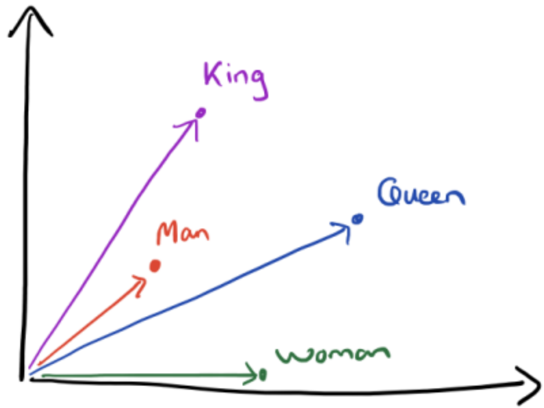
What's next

After the break, you'll get acquainted with a target system that readily lends itself to a vector-based interpretation: sCCC, the (symmetric) compact closed category of **FVect**, finite dimensional vector spaces and linear maps.

Working out a compositional vector-based interpretation will involve the same interplay of derivational and lexical semantics that we just saw.

Source papers [Coecke et al., 2010], [Coecke et al., 2013]

Vectors, Tensors, Linear Maps



Similar words occur in similar contexts

Motivation

Guess the word...

A bottle of	tesguino	is on the table.
Everybody likes	tesguino	.
	Tesguino	makes you drunk.
We make	tesguino	out of corn.

For example Tesguino: a fermented drink like beer but made of corn.

Generally speaking words that often co-occur will have a meaningful relation; words occurring in the same context, will carry similar meaning.

The slogan “You will know a word by the company it keeps”

Count-Based Vectors: Construction

Count! For each **focus** word we count the words that we find within a suitable context window:

sugar, a sliced lemon, a tablespoonful of	**apricot**	preserve or jam, a pinch each of,
their enjoyment. Cautiously she sampled her first	**pineapple**	and another fruit whose taste she likened
well suited to programming on the digital	**computer**.	In finding the optimal R-stage policy from
for the purpose of gathering data and	**information**	necessary for the study authorized in the

Vectors! Focuswords w.r.t context words produce vectors:

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

rows: focus words
columns: context

Normalise! To mitigate the influence of highly frequent words we may normalize the obtained vectors, e.g. using pointwise mutual information:

$$\mathrm{PMI}(w, c) = \log\left(\frac{p(w, c)}{p(w)p(c)}\right)$$

Here, $p(w, c)$ is the frequency of c occurring in the context of w and $p(w)$ the total frequency of w in the corpus.

Count-Based Vectors: Properties

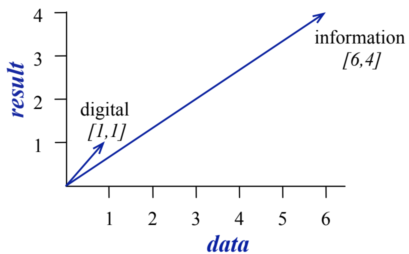
Features The elements of these vectors indicate how strong a word meaning is defined by another word. Aka: each element of a vector is an interpretable feature that gives insight into the meaning of the word.

Features Interpreting operations on features is then relatively straightforward:

$\overrightarrow{w_1} + \overrightarrow{w_2}$ Joining features, we get a less specific 'word'

$\overrightarrow{w_1} \odot \overrightarrow{w_2}$ Intersecting features, we get a more specific 'word'

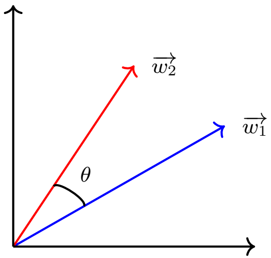
Intersecting context We compare word vectors in a space:



No. of context words = dimension of the vector space

Comparing Vectors

Angles We compute the *similarity* between words using the cosine of the angle between their vectors:



$$\mathrm{sim}(\overrightarrow{w_1}, \overrightarrow{w_2}) = \cos(\theta) = \frac{\overrightarrow{w_1} \cdot \overrightarrow{w_2}}{|\overrightarrow{w_1}||\overrightarrow{w_2}|}$$

Example Say we have $\overrightarrow{\text{digital}} = (2, 1, 0, 1)$ and $\overrightarrow{\text{information}} = (1, 6, 0, 4)$. We calculate:

$$\mathrm{sim}(\overrightarrow{\text{digital}}, \overrightarrow{\text{information}}) = \frac{2 \cdot 1 + 1 \cdot 6 + 0 \cdot 0 + 1 \cdot 4}{\sqrt{2^2 + 1^2 + 1^2}\sqrt{1^2 + 6^2 + 4^2}} = \frac{12}{\sqrt{6}\sqrt{53}} = \frac{12}{17.83} = 0.67$$

So: ‘digital’ and ‘information’ are similar!

The values of sim range from -1 to 1

The future of distributional semantics

Context is everything Because vector representations are suitable neural network input, countless new models have popped up to learn word embeddings.

Skipgram Embeddings that predict the context rather than count it, by maximizing:

$$\sum_{c \in C} \log \sigma(\mathbf{t} \cdot \mathbf{c}) + \sum_{\overline{c} \in \overline{C}} \log \sigma(-\mathbf{t} \cdot \overline{\mathbf{c}})$$

BERT Uses *masked language modelling* on massive text corpora to create a function from a sentential context to an individual word representation:

```
The [MASK] is rising and all life on [MASK] will be [MASK] soon
```

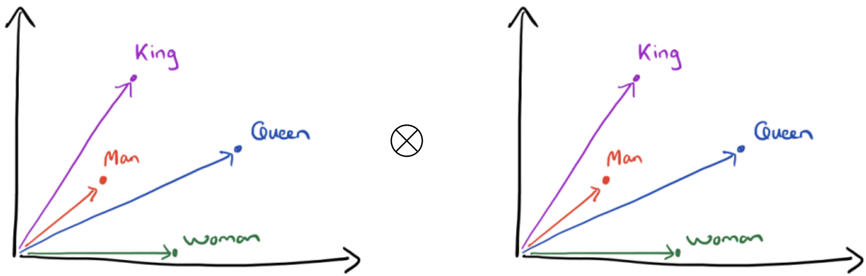
We maximize

$$\frac{1}{n} \sum_{k=1}^{n} \log p(m_i | w_1, ..., w_m)$$

This gives us *dynamic* representations; a change of a single context word changes the embedding for the target word.

In this course We discuss skipgram in more detail tomorrow, and deal with BERT on Friday.

Compositional Vector-based Semantics with Tensors and Linear Maps



Vectors

Given a vector space V spanned by a basis $\{e_i\}_i$, we write a vector $\overrightarrow{v}$ as a linear combination of basis vectors:

$$\overrightarrow{v} = a_1 \overrightarrow{e_1} + ... + a_n \overrightarrow{e_n}$$

We may use shorthand: $v = \sum_i a_i \overrightarrow{e_i}$. When the basis is obvious/irrelevant, we just write down coefficients: $\overrightarrow{v} = (a_1, ..., a_n)$.

Dimension Every vector space V has a dimension corresponding to the number of basis vectors of V.

Ex: vector $\overrightarrow{v} = (1, 2, 3, 4)$ is in the a space V with $dim(V) = 4$.

Dot product The standard inner product between two vector of the same dimension is the sum of their element wise multiplication:

$$\overrightarrow{v} = a_1 \overrightarrow{v_1} + ... + a_n \overrightarrow{v_n}$$
$$\overrightarrow{w} = b_1 \overrightarrow{w_1} + ... + b_n \overrightarrow{w_n}$$

$$\langle \overrightarrow{v} \mid \overrightarrow{w} \rangle = a_1 b_1 + ... + a_n b_n = \mathbf{a}_i \mathbf{b}_i$$

Note: (Orthonormal) basis vectors $\hat{e}_i, \hat{e}_j$ have dot product 1 when $i = j$, and 0 otherwise.

Tensors

Tensor Product We can join two vector spaces V and W with the tensor product.

We write $V \otimes W$ and this space has dimension $dim(V \otimes W) = dim(V)dim(W)$.

The concrete tensor product between two vectors $\overrightarrow{u} = \sum_i a_i \overrightarrow{u_i}$ and $\overrightarrow{v} = \sum_j b_j \overrightarrow{v_j}$ is given by all possible multiplications:

$$\overrightarrow{u} \otimes \overrightarrow{v} = \sum_{ij} a_i b_j (\overrightarrow{u_i} \otimes \overrightarrow{v_j}) = \mathbf{u}_i \mathbf{v}_j$$

Example

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{pmatrix}$$

Bilinearity The tensor product satisfies

$$(a \cdot \overrightarrow{v}) \otimes \overrightarrow{w} = a \cdot (\overrightarrow{v} \otimes \overrightarrow{w}) = \overrightarrow{v} \otimes (a \cdot \overrightarrow{w})$$

$$\begin{pmatrix} 2 \\ 4 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 6 & 8 & 10 \\ 12 & 16 & 20 \end{pmatrix}$$

Brain Teaser 1

Commutativity is the tensor product of vectors commutative? I.e. do we have $\overrightarrow{u} \otimes \overrightarrow{v} = \overrightarrow{v} \otimes \overrightarrow{u}$?

Symmetry is the tensor product between vector spaces symmetric? I.e. do we have $V \otimes W \cong W \otimes V$?

Commutativity

$$\lightning \qquad \begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{pmatrix} \qquad \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 4 & 8 \\ 5 & 10 \end{pmatrix}$$

Symmetry

✓Yes, because $\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \right)^{\perp} = \begin{pmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{pmatrix}^{\perp} = \begin{pmatrix} 3 & 6 \\ 4 & 8 \\ 5 & 10 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

Linear maps

Linear map $f : V \to W$ is a linear map if

$$f(\overrightarrow{u} + \overrightarrow{v}) = f(\overrightarrow{u}) + f(\overrightarrow{v})$$

$$f(a \cdot \overrightarrow{v}) = a \cdot f(\overrightarrow{v})$$

We just write $V \to W$ for all linear maps from V to W (which is a vector space!)

Representation Any linear map can be represented by a matrix. The matrix M of a map f, when multiplied with a vector v, gives $f(\overrightarrow{v}) = M \times \overrightarrow{v} = \mathbf{M}_{ij}\mathbf{v}_j$ (dot product along columns)

Example

$$\begin{pmatrix} 3 & 6 \\ 4 & 8 \\ 5 & 10 \end{pmatrix} \times \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \cdot 2 + 6 \cdot 1 \\ 4 \cdot 2 + 8 \cdot 1 \\ 5 \cdot 2 + 10 \cdot 1 \end{pmatrix} = \begin{pmatrix} 12 \\ 16 \\ 20 \end{pmatrix}$$

Dual spaces We write $V^* = V \to \mathbb{R}$ for the dual space of V. In concrete calculation we often simplify with $V^* \cong V$ (because we have an orthonormal basis) $\rightsquigarrow$ more on this on Thursday!

Standard maps

Identity

$$1_V\left(\sum_i c_i\overrightarrow{v_i}\right) = \sum_i c_i\overrightarrow{v_i} \qquad (=\delta_{ij})$$

Composition $f : V \to W, g : W \to U$ gives $g \circ f : V \to U$

$$(g \circ f)(\overrightarrow{v}) = g(f(\overrightarrow{v}))$$

Composition is matrix multiplication! $L_{ik} = M_{ij}N_{jk}$

Tensor product $f : V \to W, g : U \to Z$ gives $f \otimes g : V \otimes U \to W \otimes Z$:

$$(f \otimes g)\left(\sum_{ij} c_{ij}\overrightarrow{v_i} \otimes \overrightarrow{u_j}\right) = \sum_{ij} c_{ij} f(\overrightarrow{v_i}) \otimes g(\overrightarrow{u_j})$$

Symmetry for any V, W we have $\sigma_{V,W} : V \otimes W \to W \otimes V$

$$\sigma_{V,W}\left(\sum_{ij} c_{ij}\overrightarrow{v_i} \otimes \overrightarrow{w_j}\right) = \sum_{ij} c_{ij}\overrightarrow{w_j} \otimes \overrightarrow{v_i}$$

Symmetry is transposition! $M_{ij} \mapsto M_{ji}$

Contraction For any V we have $\varepsilon_V : V^* \otimes V \to \mathbb{R}$:

$$\varepsilon_V\Big(\sum_{ij} c_{ij}(\overrightarrow{v_i} \otimes \overrightarrow{v_j})\Big) = \sum_{ij} c_{ij}\langle \overrightarrow{v_i} \mid \overrightarrow{v_j}\rangle = \sum_i c_{ii} = \mathbf{M}_{ii}$$

This gives the sum of elements on the diagonal of a tensor (trace).

$$(M_{ij} =) \quad \begin{pmatrix} 3 & 6 & -7 \\ 4 & 8 & 3 \\ 5 & 10 & 2 \end{pmatrix} \mapsto 3 + 8 + 2 = 13 \quad (= M_{ii})$$

Expansion For any V we have $\eta_V : \mathbb{R} \to V \otimes V^*$

$$\eta_V(\lambda) = \sum_i \lambda(\overrightarrow{v_i} \otimes \overrightarrow{v_i}) = \lambda\delta_{ij}$$

Embedding a number on the diagonal of a matrix (0 elsewhere).

$$\lambda \mapsto \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix}$$

Brain teaser 2

Properties We can show that the following properties hold:

$$(1_A \otimes \varepsilon_A) \circ (\eta_A \otimes 1_A) = 1_A \qquad (\varepsilon_A \otimes 1_{A^*}) \circ (1_{A^*} \otimes \eta_A) = 1_{A^*}$$

$$(\varepsilon_{A^*} \otimes 1_A) \circ (1_A \otimes \eta_{A^*}) = 1_A \qquad (1_{A^*} \otimes \varepsilon_{A^*}) \circ (\eta_{A^*} \otimes 1_{A^*}) = 1_{A^*}$$

Brain teaser 2

Properties We can show that the following properties hold:

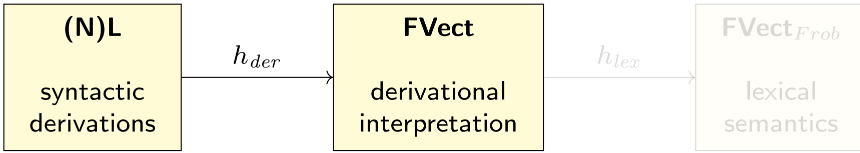
$$(1_A \otimes \varepsilon_A) \circ (\eta_A \otimes 1_A) = 1_A \qquad (\varepsilon_A \otimes 1_{A^*}) \circ (1_{A^*} \otimes \eta_A) = 1_{A^*}$$

$$(\varepsilon_{A^*} \otimes 1_A) \circ (1_A \otimes \eta_{A^*}) = 1_A \qquad (1_{A^*} \otimes \varepsilon_{A^*}) \circ (\eta_{A^*} \otimes 1_{A^*}) = 1_{A^*}$$

Commuting Diagrams

$$\begin{tikzcd}
(A \otimes A^*) \otimes A \arrow[r, "\alpha_{A,A^*,A}"] & A \otimes (A^* \otimes A) \arrow[d, "1_A \otimes \epsilon"] \\
I \otimes A \arrow[u, "\eta \otimes 1_A"] & A \otimes I \arrow[d, "\rho_A"] \\
A \arrow[u, "\lambda_A^{-1}"] \arrow[r, "1_A"'] & A
\end{tikzcd}
\qquad
\begin{tikzcd}
A^* \otimes (A \otimes A^*) \arrow[r, "\alpha^{-1}_{A^*,A,A^*}"] & (A^* \otimes A) \otimes A^* \arrow[d, "\epsilon \otimes 1_{A^*}"] \\
A^* \otimes I \arrow[u, "1_{A^*} \otimes \eta"] & I \otimes A^* \arrow[d, "\lambda_A"] \\
A^* \arrow[u, "\rho^{-1}_{A^*}"] \arrow[r, "1_{A^*}"'] & A^*
\end{tikzcd}$$

Compositional interpretation



- Source: type logic for syntax **(N)L**
- Target: sCCC of finite-dimensional vector spaces and linear maps **FVect**

Linearity Derivational semantics reflects the resource-sensitivity ('linearity') of the source logic. Next step: word-internal semantics allowing for non-linear operations: duplication, merger of information.

Interpretation: Types

Types as vector spaces

$$\lceil np \rceil = \lceil n \rceil = \mathsf{N}$$

$$\lceil s \rceil = \mathsf{S}$$

$$\lceil A \bullet B \rceil = \lceil A \rceil \otimes \lceil B \rceil$$

$$\lceil A \backslash B \rceil = \lceil A \rceil^* \otimes \lceil B \rceil$$

$$\lceil A / B \rceil = \lceil A \rceil \otimes \lceil B \rceil^*$$

Reaching for the stars... As said above, we don't really use dual spaces for our calculations, but we do need them for a correct translation:

$$V^{**} \cong V$$

$$(V \otimes W)^* \cong W^* \otimes V^*$$

Example

$$\lceil s/(np\backslash s) \rceil = \lceil s \rceil \otimes \lceil np\backslash s \rceil^* = \lceil s \rceil \otimes (\lceil np \rceil^* \otimes \lceil s \rceil)^*$$

$$= \mathsf{S} \otimes (\mathsf{N}^* \otimes \mathsf{S})^* = \mathsf{S} \otimes \mathsf{S}^* \otimes \mathsf{N}$$

Interpretation: Proofs

Plan The proof of a sequent $\Gamma \vdash A$ in the source logic **(N)L** is sent to a linear map $f : \lceil\Gamma\rceil \to \lceil A\rceil$ in the target sCCC **FVect**.

Basic rules

$$\left\lceil \; \frac{}{A \vdash A}\,Ax \; \right\rceil \quad = \quad \frac{}{1_{\lceil A\rceil} : \lceil A\rceil \to \lceil A\rceil}\,Ax$$

$$\left\lceil \; \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma \cdot \Delta \vdash A \bullet B}\,\bullet I \; \right\rceil \quad = \quad \frac{f : \lceil\Gamma\rceil \to \lceil A\rceil \quad g : \lceil\Delta\rceil \to \lceil B\rceil}{f \otimes g : \lceil\Gamma\rceil \otimes \lceil\Delta\rceil \to \lceil A\rceil \otimes \lceil B\rceil}$$

$$\left\lceil \; \frac{\Delta \vdash A \bullet B \quad \Gamma[A \cdot B] \vdash C}{\Gamma[\Delta] \vdash C}\,\bullet E \; \right\rceil = \frac{f : \lceil\Delta\rceil \to \lceil A\rceil \otimes \lceil B\rceil \quad g : \Gamma[\lceil A\rceil \otimes \lceil B\rceil] \to \lceil C\rceil}{g[f] : \Gamma[\Delta] \to \lceil C\rceil}$$

Interpretation: Elimination

We're looking to interpret the rules

$$\left\lceil \frac{\Gamma \vdash A \quad \Delta \vdash A\backslash B}{\Gamma \cdot \Delta \vdash B} \backslash E \right\rceil \qquad \left\lceil \frac{\Gamma \vdash B/A \quad \Delta \vdash A}{\Gamma \cdot \Delta \vdash B} /E \right\rceil$$

$\backslash$ E Given maps $f : \lceil\Gamma\rceil \to \lceil A\rceil$, $g : \lceil\Delta\rceil \to \lceil A\rceil^* \otimes \lceil B\rceil$, we create

$$\lceil\Gamma\rceil \otimes \lceil\Delta\rceil \xrightarrow{f \otimes g} \lceil A\rceil \otimes \lceil A\rceil^* \otimes \lceil B\rceil \xrightarrow{\varepsilon_{\lceil A\rceil} \otimes 1_{\lceil B\rceil}} \lceil B\rceil$$

/ E Given maps $f : \lceil\Gamma\rceil \to \lceil B\rceil \otimes \lceil A\rceil^*$, $g : \lceil\Delta\rceil \to \lceil A\rceil$, we create

$$\lceil\Gamma\rceil \otimes \lceil\Delta\rceil \xrightarrow{f \otimes g} \lceil B\rceil \otimes \lceil A\rceil^* \otimes \lceil A\rceil \xrightarrow{1_{\lceil B\rceil} \otimes \varepsilon_{\lceil A\rceil}} \lceil B\rceil$$

Interpretation: Introduction

Now for the introduction rules

$$\left\lceil \frac{A \cdot \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ \backslash I \right\rceil \qquad \left\lceil \frac{\Gamma \cdot A \vdash B}{\Gamma \vdash B/A}\ /I \right\rceil$$

$\backslash$ I Given a map $f : \lceil A \rceil \otimes \lceil \Gamma \rceil \to \lceil B \rceil$ we construct

$$\lceil \Gamma \rceil \xrightarrow{\eta_{\lceil A \rceil} \otimes 1_{\lceil \Gamma \rceil}} \lceil A \rceil^* \otimes \lceil A \rceil \otimes \lceil \Gamma \rceil \xrightarrow{1_{\lceil A \rceil^*} \otimes f} \lceil A \rceil^* \otimes \lceil B \rceil$$

/ I Given maps $f : \lceil \Gamma \rceil \otimes \lceil A \rceil \to \lceil B \rceil$ we construct

$$\lceil \Gamma \rceil \xrightarrow{1_{\lceil \Gamma \rceil} \otimes \eta_{\lceil A \rceil}} \lceil \Gamma \rceil \otimes \lceil A \rceil \otimes \lceil A \rceil^* \xrightarrow{f \otimes 1_{\lceil A \rceil^*}} \lceil B \rceil \otimes \lceil A \rceil^*$$

Derivational Semantics: Example

Let's follow the rules one by one. Taking the derivation for a simple sentence:

$$
\dfrac{\dfrac{}{np \vdash np}\,Ax \qquad \dfrac{\dfrac{}{(np\backslash s)/np \vdash (np\backslash s)/np}\,Ax \qquad \dfrac{}{np \vdash np}\,Ax}{(np\backslash s)/np \cdot np \vdash np\backslash s}\,/E}{\underbrace{np}_{\text{Bob}} \cdot (\underbrace{(np\backslash s)/np}_{\text{rejects}} \cdot \underbrace{np}_{\text{papers}}) \vdash s}\,\backslash E
$$

we end up with

$$(\varepsilon_N \otimes 1_S) \circ (1_N \otimes ((1_{N\otimes S} \otimes \varepsilon_N) \circ (1_{N\otimes S\otimes N} \otimes 1_N)))$$

which simplifies

$$
\begin{aligned}
&= (\varepsilon_N \otimes 1_S) \circ (1_N \otimes (1_{N\otimes S} \otimes \varepsilon_N)) \\
&= (\varepsilon_N \otimes 1_S) \circ (1_{N\otimes N\otimes S} \otimes \varepsilon_N) \\
&= (\varepsilon_N \otimes 1_S \otimes \varepsilon_N)
\end{aligned}
$$

?!?!

Simplification

We can use the 'categorical' rules from before to simplify calculations,
e.g. $1_N \otimes 1_{N\otimes S} = 1_{N\otimes N\otimes S}$, and $f \circ 1_N = f$ and so on.

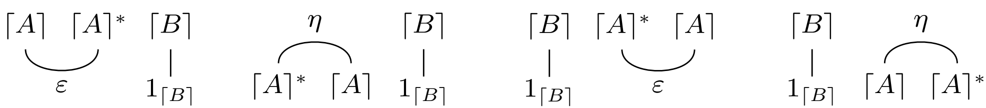
Example Take the left application law:

$$\dfrac{\dfrac{}{1_{\mathsf{N}} : np \vdash np}\,Ax \qquad \dfrac{}{1_{\mathsf{N}^*\otimes\mathsf{S}} : np\backslash s \vdash np\backslash s}\,Ax}{(\varepsilon_{\mathsf{N}^*} \otimes 1_{\mathsf{S}}) \circ (1_{\mathsf{N}} \otimes 1_{\mathsf{N}^*\otimes\mathsf{S}}) : np \cdot np\backslash s \vdash s}\,\backslash E$$

But $(\varepsilon_{\mathsf{N}^*} \otimes 1_{\mathsf{S}}) \circ (1_{\mathsf{N}} \otimes 1_{\mathsf{N}^*\otimes\mathsf{S}}) = (\varepsilon_{\mathsf{N}^*} \otimes 1_{\mathsf{S}}) \circ 1_{\mathsf{N}\otimes\mathsf{N}^*\otimes\mathsf{S}} = \varepsilon_{\mathsf{N}^*} \otimes 1_{\mathsf{S}}$

Visual simplification

$A \otimes A\backslash B \vdash B$ $\qquad$ $B \vdash A\backslash(A \otimes B)$ $\qquad$ $B/A \otimes A \vdash B$ $\qquad$ $B \vdash (B \otimes A)/A$



$\mathbf{a}_i\mathbf{M}_{jk} \mapsto \mathbf{a}_i\mathbf{M}_{ij}$ $\qquad$ $\mathbf{b}_i \mapsto \delta_{jk}\mathbf{b}_i$ $\qquad$ $\mathbf{M}_{ij}\mathbf{a}_k \mapsto \mathbf{M}_{ij}\mathbf{a}_j$ $\qquad$ $\mathbf{b}_i \mapsto \mathbf{b}_i\delta_{jk}$

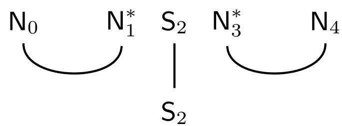
Strategy

Our vector semantics so far can be computed by keeping track of axiom linkings:

- Step 1: derive

$$\dfrac{\dfrac{}{np_0 \vdash np_0}\,Ax \qquad \dfrac{\dfrac{}{(np_1\backslash s_2)/np_3 \vdash (np_1\backslash s_2)/np_3}\,Ax \qquad \dfrac{}{np_4 \vdash np_4}\,Ax}{(np_1\backslash s_2)/np_3 \cdot np_4 \vdash \{(3,4)\} : np_1\backslash s_2}\,/E}{\{(0,1),(3,4)\} : \underbrace{np_0}_{\text{Bob}} \cdot (\underbrace{(np_1\backslash s_2)/np_3}_{\text{rejects}} \cdot \underbrace{np_4}_{\text{papers}}) \vdash s_2}\,\backslash E$$

- Stap 2: translate and connect



- Stap 3: identify!

$$\mathbf{s}_j = \mathbf{bob}_i\mathbf{reject}_{ijk}\mathbf{paper}_k \qquad \text{or} \qquad \overrightarrow{bob}^{\perp} \times (\overline{\overline{reject}} \times \overrightarrow{paper})$$

Summary Linear Algebra: index notation

- A vector is a sequence of numbers $(a_1, a_2, ..., a_n)$, abbreviated by $\mathbf{a}_i$
- A vector is a 1st-order tensor, so it has 1 index. A 2nd-order tensor is a matrix, so: $\mathbf{M}_{ij}$. For a cube (3rd order) we have $\mathbf{C}_{ijk}$, etc.
- Juxtaposing tensors indicates taking a tensor product: $\mathbf{a}_i\mathbf{b}_j$. Why? Because the long-hand is

  $$\mathbf{a}_i\mathbf{b}_j = \sum_i \sum_j a_i b_j (\overrightarrow{a}_i \otimes \overrightarrow{b}_j)$$

  This gives a matrix as we have two free indices.
- Repeating indices means means multiply-and-sum. Examples: dot product of two vectors, trace of a matrix, applying a cube to a matrix:

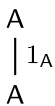
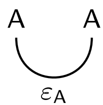
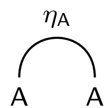
  $$\mathbf{a}_i\mathbf{b}_i = \sum_i a_i b_i \qquad \mathbf{M}_{ii} = \sum_i M_{ii} \qquad \mathbf{C}_{ijk}\mathbf{M}_{jk} = \sum_{ijk} C_{ijk} M_{jk} \overrightarrow{v}_i$$

- The Kronecker delta returns 1 for corresponding indices, and so it's represented by the identity matrix!

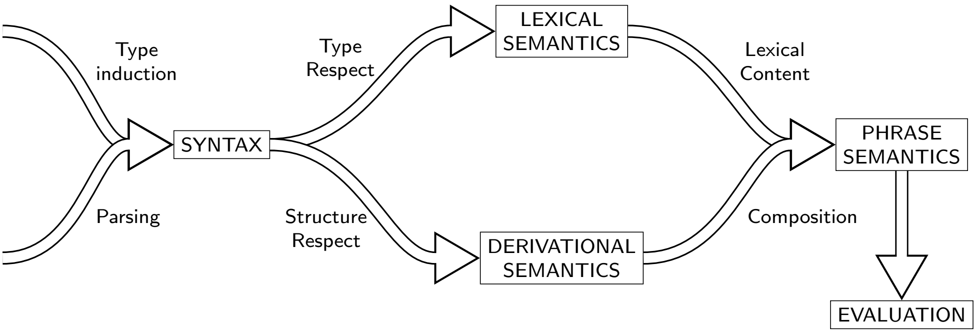
  $$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Summary Vector Interpretation: from proof to indices

We saw that three standard maps are essential for our vector calculations:

Name	**Identity**	**Contraction**	**Expansion**
Derivation	$\ldots A \ldots \vdash \ldots A \ldots$	$\ldots A \ldots A \ldots \vdash \ldots$	$\ldots \vdash \ldots A \ldots A \ldots$
Visual			
Relabelling	$\mathbf{a}_i \mapsto \mathbf{a}_i$	$\mathbf{M}_{ij}/\mathbf{a}_i\mathbf{b}_j \mapsto \mathbf{M}_{ii}/\mathbf{a}_i\mathbf{b}_i$	$1 \mapsto \delta_{ij}$

Summary: The Compositional Process



- Our core methodology provides syntax and the interfacing with semantics,
- Lexical content is learnable, though not always in a tractable way (Tue)
- Syntax doesn't come for free: type induction & parsing as learnable processes (Wed/Thu)
- In the end, the phrase semantics can be applied to NLP tasks (Fri)

References

Sheldon Jay Axler. Linear Algebra Done Right. Undergraduate Texts in Mathematics. Springer, 3 edition, 2015.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. CoRR, abs/1003.4394, 2010. URL http://arxiv.org/abs/1003.4394.

Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. Lambek vs. Lambek: Functorial vector space semantics and string diagrams for lambek calculus. Ann. Pure Appl. Log., 164(11):1079–1100, 2013. doi: 10.1016/j.apal.2013.05.009. URL https://doi.org/10.1016/j.apal.2013.05.009.

Joachim Lambek. The mathematics of sentence structure. The American Mathematical Monthly, 65(3):154–170, 1958.

Joachim Lambek. On the calculus of syntactic types. Structure of language and its mathematical aspects, 12:166–178, 1961.

Joachim Lambek. Categorial and categorical grammars. In Categorial grammars and natural language structures, pages 297–317. Springer, 1988.

Joachim Lambek and Philip J. Scott. Introduction to Higher Order Categorical Logic. Cambridge University Press, 1986.

Richard Montague. Universal grammar. In Richmond H. Thomason, editor, Formal Philosophy: Selected Papers of Richard Montague, number 222–247. Yale University Press, New Haven, London, 1974.

Richard Moot and Christian Retoré. The Logic of Categorial Grammars - A Deductive Account of Natural Language Syntax and Semantics, volume 6850 of Lecture Notes in Computer Science. Springer, 2012. ISBN 978-3-642-31554-1. doi: 10.1007/978-3-642-31555-8. URL https://doi.org/10.1007/978-3-642-31555-8.

Gijs Wijnholds. Categorical foundations for extended compositional distributional models of meaning. MSc Thesis. Institute for Logic, Language and Information, University of Amsterdam., 2015. URL https://eprints.illc.uva.nl/id/eprint/940/.