Functorial Language Games for Question Answering

Giovanni de Felice†, Elena Di Lavore⋆,
Mario Román⋆, Alexis Toumi†

† Department of Computer Science, University of Oxford.
⋆ Department of Software Science, Tallinn University of Technology.

July 2020

Categories in linguistics

- Syntax with free monoidal categories (e.g. Categorial grammars, CFGs, Lambek, pregroups)
- Semantics with monoidal functors. (Bob Coecke, Mernoosh Sadrzadeh et al. 2008[1]).

[1] Clark, Coecke, and Sadrzadeh, "A Compositional Distributional Model of Meaning", 2008.

Categories in linguistics

- Syntax with free monoidal categories (e.g. Categorial grammars, CFGs, Lambek, pregroups)
- Semantics with monoidal functors. (Bob Coecke, Mernoosh Sadrzadeh et al. 2008[1]).
- Pragmatics: meaning depends on context. The same utterance "Water!" can be a request for help, the answer to a question or the lyrics of a song.

[1] Clark, Coecke, and Sadrzadeh, "A Compositional Distributional Model of Meaning", 2008.

Pragmatics with Wittgenstein's language games

- In the *Philosophical Investigations* (1953), Wittgenstein analysed the pragmatical aspect of language using the concept of **language-game**. He defines it by example: "asking, thanking, cursing, greeting, praying".
- Idea[2]: **Categorical linguistics** $\otimes$ **Compositional game theory**[3]
- In this work we propose two ways of making these theories interact, with applications to game-theoretic pragmatics and question answering.

---

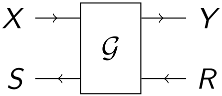
[2] Hedges and Lewis, "Towards Functorial Language-Games", 2018.
[3] Ghani et al., "Compositional game theory", 2018.

Open games

Definition

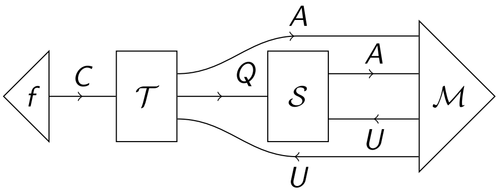
An *open game* $\mathcal{G} : \binom{X}{S} \xrightarrow{\Sigma} \binom{Y}{R}$ that takes *observations* on X, produces *moves* on Y, receives *utilities* on R, and returns *coutilities* on S, is given by:

- $\Sigma_{\mathcal{G}}$ is the set of *strategy profiles*,
- $\pi_{\mathcal{G}} : \Sigma_{\mathcal{G}} \times X \to Y$ is the *play function*,
- $\kappa_{\mathcal{G}} : \Sigma_{\mathcal{G}} \times X \times R \to S$ is the *coplay function*,
- $E_{\mathcal{G}} : X \times (Y \to R) \to \mathcal{P}(\Sigma_{\mathcal{G}})$ is the *equilibrium function*.



Open games

Example (Question Answering)



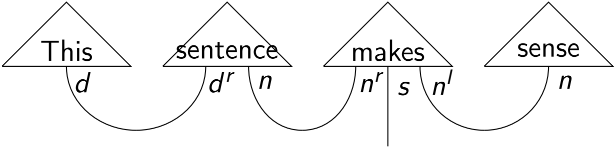
Syntax: pregroup grammar

Definition

A *pregroup grammar* is a tuple $G = (V, B, D, s)$ where:

- V the *vocabulary*
- B the *basic types* with $s \in B$ the *sentence type*
- $D \subseteq V \times P_B$ the *dictionary entries* where P_B the free pregroup generated by B.

$\mathbf{G} = \mathbf{RC}(D)$ is the free rigid category generated by the dictionary.



Semantics: DisCo models

$$\text{Model} : \text{Grammar} \longrightarrow \text{Semantics}$$

Definition (DisCo model)

A DisCo model for a pregroup grammar $G = (V, B, D, s)$ is a rigid monoidal functor $F : \mathbf{G} \to \mathbf{S}$ for $\mathbf{S}$ a rigid monoidal category. The semantics of a list of words $u \in V^*$ with grammatical structure $g : u \to t$ is given by the state $F(g) : 1 \to F(t)$.

Example

Distributional models $\mathbf{G} \to \mathbf{Vect}$.
Relational models $\mathbf{G} \to \mathbf{Rel}$ (Databases)[4]
Quantum models $\mathbf{G} \to \mathbf{FHilb}$ (QNLP).

---

[4] de Felice, Meichanetzidis, and Toumi, "Functorial Question Answering", 2019.

From grammar to games

Can we build functors from **G** to **Game**? **Game** is not a rigid category! To build non-trivial games we use the free completion of open games as a rigid monoidal category.

Proposition

[5] *There is a free completion* $\mathcal{A} : \mathbf{Moncat} \to \mathbf{RigidCat}$, *(freely adding cups and caps) such that the embedding* $\mathbf{C} \to \mathcal{A}(\mathbf{C})$ *is full.*

Solution: build functors $F : \mathbf{G} \to \mathcal{A}(\mathbf{Game})$ with $F(s) \in \mathbf{Game}$, then for any $g : u \to s$, $F(g) \in \mathbf{Game}$.

[5] Delpeuch, "Autonomization of Monoidal Categories", 2014.

Snake removal

Remove snakes from a sentence.

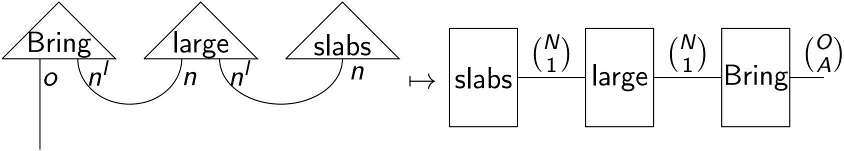
Wittgenstein's example

The language is meant to serve for communication between a builder A and an assistant B. A is building with building-stones: there are blocks, pillars, slabs and beams. B has to pass the stones, in the order in which A needs them. For this purpose they use a language consisting of the words "block", "pillar", "slab", "beam". A calls them out; — B brings the stone which he has learnt to bring at such-and-such a call.
Conceive this as a complete primitive language.

Philosophical Investigations, Ludwig Wittgenstein

Pragmatics with functorial language games

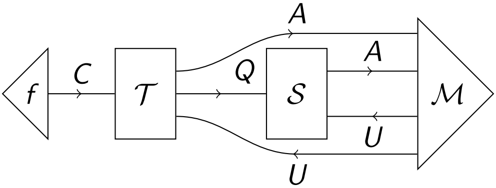
Encode the language game as a functor $F : \mathbf{G} \to \mathcal{A}(\mathbf{Game})$. The imperative "Bring" models the intention of the builder. The apprentice $k : O \to A$ turns orders into actions.



$$E_{F(\text{Bring large slabs})}(k) = \begin{cases} 1 & k(\text{Bring large slabs}) = \texttt{bring}(\text{large slabs}) \\ 0 & \text{otherwise} \end{cases}$$

This gives the pragmatics of orders. Similar treatment for questions.

Instantiating the Q&A game with DisCo models



f is a list of question-answer pairs (q, a) for $q : u \to z$ and $a \in A$.
Student's strategies: $\Sigma_{\mathcal{S}} \subseteq \{\sigma : \mathbf{G} \to \mathbf{Rel} : \ \sigma(z) = A\}$.
Teacher strategies: $\Sigma_{\mathcal{T}} = \{ 0, 1, \ldots n \}$.
Utilities as in an adversarial game.

Nash equilibria in question answering

We can analyse the possible outcomes of this game.

1. There is a pair (q_i, a_i) in C that the student cannot answer correctly. Teacher wins.
2. The corpus is consistent and the student has access to the model σ that answers all the questions correctly. Student wins.
3. For any choice i of the teacher, the student has a model σ_i that answers q_i correctly. And viceversa, for any strategy σ of the student there is a choice j of the teacher such that $\sigma(q_j) \neq a_j$. There is no Nash equilibrium.

Future directions

Language-games as NLP tasks

1. Word sense disambiguation as a collaborative game where the players are words and their strategies are word-senses[6]
2. GANs[7] for Question Answering. (DisCoPy? Catlab?)
3. Quantum supremacy as a language game.

[6] Tripodi and Navigli, "Game Theory Meets Embeddings", 2019.
[7] Goodfellow et al., "Generative Adversarial Nets", 2014.