

Simple semantics for compositional language learning

Michael Goodale

December 12, 2024

This document outlines a potential approach for integrating semantic information into a learning approach using minimalist grammars. The idea is that by pairing **semantic scenarios** with sentences of natural language, it can be possible to learn a fragment of a natural language grammar.

Semantic scenarios consist of a simple, finite model that consists of a small number of individuals in a given scenario, along with various predicates that apply to them as well as various events that involve them. These have a simple ontology which can be used to describe different scenarios in which a sentence is uttered.

1 Semantic scenarios

The basic ontology consists of four kinds of semantic objects:

- Individuals: An individual is a simple atom that belong to the set of individuals, \mathcal{D} .
- Events: An event is a simple atom that belong to the set of events, \mathcal{E} .
- Thematic relations: A thematic relation is a tuple, $\langle x, y, \theta \rangle \in \Theta$ consisting of one event, $x \in \mathcal{E}$ and an individual, $y \in \mathcal{D}$ and a theta role, $\theta \in \{\text{AGENT}, \text{PATIENT}\}$.
- Properties: A property is a tuple of an atom paired with a set of individuals or events. The atoms in the tuple are members of the set of properties, \mathcal{P} .¹

This ontology (along with some other capacities) will likely be sufficient for a large swathe of natural language phenomena. Crucially, they also consist things which are likely innate (or at least acquired incredibly young), namely agent-patient distinction (Papeo et al. 2024), event vs individual objects and properties.

Formally, we say a given scenario is a 4-tuple consisting of subsets of \mathcal{D} , \mathcal{E} , Θ , \mathcal{P} . For example, a scenario could be the following:

¹This is a bit odd, but it is a way to allow a word to refer to the same property across scenarios, even as that property has different members

$$\begin{aligned} &\langle \{j, m, p\}, \\ &\{e_1, e_2\}, \\ &\{\langle e_1, j, \text{AGENT} \rangle, \langle e_1, m, \text{PATIENT} \rangle, \langle e_2, j, \text{AGENT} \rangle, \langle e_1, p, \text{PATIENT} \rangle\} \\ &\{\langle S, \{e_1, e_2\} \rangle\} \end{aligned}$$

Here there are three individuals, and two events occurring (to which the property S applies). The agent of both events is j while m is the patient of the first event and p the patient of the second. Unsurprisingly, the scenario refers to one where “John saw Mary and John saw Phil” is true, although the specific scenario might be paired with different sentences in training (e.g. “John saw Mary”, “Someone saw someone”, “Someone saw at least two people”, etc...).

1.1 Missing elements that may or may not be added in the future

There are a number of essential things which are missing from this ontology that would likely be necessary. Here’s a small list of ones which might be in the scope of the thesis (though it seems extremely unlikely all could be done):

- Relations (e.g. prepositions, more thematic roles, etc)
- Plural individuals (as some kind of lattice).
- Time (could be as simple as a linear order of events).
- Alternatives (e.g. with a Hamblin semantics adapted for simple scenarios)
- Degrees for properties (c.f. Kennedy and McNally 2005).
- Modal scenarios

Here’s a list of ones which, while evidently necessary, probably are not in the scope of the thesis:

- Event trajectories (c.f. Unger 2024).
- Causal reasoning (e.g. DAG of (possibly plural) events).
- Sense
- “Rich” free variables.

2 Semantic primitives/syntax

Given this model theory, we need a simple set of primitives and a syntax which can operate on top of given scenario. (This is obviously underformalised but the interpretations should be easy to deduce).

Here are some core primitives that are almost certainly necessary.

- x is a well-formed term.
- If a is an individual in \mathcal{D} , then a is a well-formed term.
- If e is an event in \mathcal{E} , then e is a well-formed term.
- If S is a property-atom in \mathcal{P} and t is a well-formed terms then $S(t)$ is a well-formed formula.
- If t_1, t_2 are well-formed terms, then $\text{AGENT}(t_1, t_2)$ and $\text{PATIENT}(t_1, t_2)$ are well-formed formula.
- If ϕ and ψ are well-formed formulae then $\neg\phi, \phi \vee \psi, \phi \wedge \psi, \phi \Rightarrow \psi, \exists x\phi$ and $\forall x\psi$ are well-formed formulae.

Here are some possible other primitives that may be useful.

- If ϕ is a wff, then $\iota(\phi)$ is a term.
- If ϕ and ψ are wff with a free variable, then $\phi \subset \psi, \phi \subseteq \psi, \phi = \psi$.
- If a is an atom, then $\text{ALT}(a)$ is the set of alternatives to a and if b is an atom, $a \in \text{ALT}(a)$ is a wff.²
- Exh (lol)
- Some kind of dynamic assignment to variables.

3 Montagovian Minimalist Grammars

Given our logic and semantic models, we need a way to connect it to natural language. We can do this with Minimalist Grammars by attaching simple lambda expressions to each lexical entry (a similar approach could be done for CCG). As for movement, this can be handled in various ways (c.f. Kobele 2012 for one way)

We define each lexical item of Montagovian MG as tuple consisting of:

- The word
- The features of the word (including both selectors and selectees)

²This might be done by “magic”; i.e. a pre-defined set of valid alternatives in each scenario, which is fine as we are interested in *language*-learning here not concept learning.

- A lambda expression

In principle, these could be maximally unconstrained. However, the interest of this thesis is to describe precisely what constraints are necessary to learn language. As such, I will write **meta-grammars** which define possible lexical entries under some theory. Eventually it might be nice to formally describe how these are written, but for now I will write them informally.

For example, the following meta-grammar only allows grammars which have proper names and intransitive verbs. Here, I'll write the changable parts of a meta-grammar in **red**.

- $w::d::e$ A word, w which has a single feature (showing its type d) and whose lambda expression *must* be a atomic individual.
- $w::d = d \quad v::\lambda x.\exists y \textcolor{red}{S}(y) \wedge \text{AGENT}(y, x)$ A word, w which has a two features (it takes a d as a complement and is itself of type v) and whose lambda expression requires there to be an event with property S such that its complement is the agent.

4 An example

The following section outlines a way of putting this all together and outlines possible learning pressures (e.g. potential functions to optimize grammars with).

Let our grammars be restricted to the following meta-grammar:

- $w::d::e$
- $w::d = d \quad v::\lambda yx.\exists y \textcolor{red}{S}(y) \wedge \text{AGENT}(z, x) \wedge \text{PATIENT}(z, y)$

This is a simple meta-grammar that allows sentences consisting of a proper name, a transitive verb and a proper name and only in SVO order. In principle, one could allow meta-grammars that allow exclusively $w::d = d \quad v::\lambda yx.P(x, y)$ or exclusively $w::d = d \quad v::\lambda yx.P(x, y)$, providing a dumb Principles and Parameters approach with a parameter switching between SOV and SVO.

We then also define an alphabet, $\{\text{John, Mary, Phil, Sue, sees, hears}\}$, and our concepts for our meta-grammars. The concepts are the following individuals, $\{j, m, p, s\}$ and a set of properties, $\{S, H\}$.

Given this meta-grammar, alphabet and conceptual repertoire, the possible grammars are defined by the power set of all possible lexical entries. We call the set of possible lexical entries, Λ and the set of possible grammars is thus $\mathcal{P}(\Lambda)$. The elements of Λ for this problem are the following³

³I'm using $S(x)$ as a shorthand for $\lambda yx.\exists y S(y) \wedge \text{AGENT}(z, x) \wedge \text{PATIENT}(z, y)$

- John::d::j
- John::d::m
- John::d::p
- John::d::s
- Mary::d::j
- Mary::d::m
- Mary::d::p
- Mary::d::s
- Phil::d::j
- Phil::d::m
- Phil::d::p
- Phil::d::s
- Sue::d::j
- Sue::d::m
- Sue::d::p
- Sue::d::s
- sees::d::j
- sees::d::m
- sees::d::p
- sees::d::s
- sees::d= =d v:: $\lambda yx.S(x, y)$
- sees::d= =d v:: $\lambda yx.H(x, y)$
- hears::d::j
- hears::d::m
- hears::d::p
- hears::d::s
- hears::d= =d v:: $\lambda yx.S(x, y)$
- hears::d= =d v:: $\lambda yx.H(x, y)$
- Phil::d= =d v:: $\lambda yx.S(x, y)$
- John::d= =d v:: $\lambda yx.S(x, y)$
- Mary::d= =d v:: $\lambda yx.S(x, y)$
- Sue::d= =d v:: $\lambda yx.S(x, y)$
- Phil::d= =d v:: $\lambda yx.H(x, y)$
- John::d= =d v:: $\lambda yx.H(x, y)$
- Mary::d= =d v:: $\lambda yx.H(x, y)$
- Sue::d= =d v:: $\lambda yx.H(x, y)$

4.1 Learning a grammar

Finding solutions

The toy examples I give here are simple enough to be solved with enumeration over all possibilities. They can be tested directly using the `toy.py` script accompanying this document.

In practice, actual solutions would be found using genetic algorithms or Monte-Carlo Markov Chains or some other method.

Learning a lexicon, then, consists of identifying the right subset in $\mathcal{P}(\Lambda)$. The right subset, of course, depends on the data provided and what learning pressures we would like to optimize. For example, table 1, shows the data as pairs of scenarios and sentences. One learning pressure then, should likely be to maximise the number of sentences which can be parsed. Another natural pressure, would be to maximise the number of sentences who have a parse that makes them true in that scenario.

Provided string	Minimal situation	Situation provided to model
John sees Mary	$\langle \{j, m\}, \{ \langle S, \{ \langle j, m \rangle \} \} \rangle$	$\langle \{j, m, p, s\}, \{ \langle S, \{ \langle j, m \rangle, \langle p, s \rangle \} \} \rangle$
Phil sees Sue	$\langle \{p, s\}, \{ \langle S, \{ \langle p, s \rangle \} \} \rangle$	$\langle \{j, m, p, s\}, \{ \langle S, \{ \langle j, m \rangle, \langle p, s \rangle \} \} \rangle$

Table 1: An example set of two sentences paired with different scenarios. I've elided the representation to make it easier to represent: $\langle \{j, m\}, \{ \langle S, \{ \langle j, m \rangle \} \} \rangle$ is properly rendered as $\langle \{j, m\}, \{e_1\}, \{ \langle e_1, j, \text{AGENT} \rangle, \langle e_1, p, \text{PATIENT} \rangle \}, \{ \langle S, \{e_1\} \} \rangle$

Solvable situation sets

Some situations are solvable for a certain meta-grammar, while others are not. We say a situation is solveable such that there exists a lexicon such that:

- i. Every sentence has a parse.
- ii. Every sentence can be made true in its paired scenario.

For example, there would be no way to learn ditransitive verbs with this meta-grammar. Curiously, there is reason to suspect that the input of a child could very often be unsolvable (c.f. creolisation and home-sign) and that linguistic structure would be imposed by finding the best possible solution.

To solve this dataset, one possible lexicon could be to simply take the language defined by Λ , that is, all possible lexical entries at once. While this grammar will produce parses for every sentence, and every sentence will have a parse which is true in a scenario (this will always be true provided that the model is **solvable**). The grammar will however have enormous syntactic ambiguity and will not be an ideal solution.

So, to actually find a good target grammar, we could minimise lexical ambiguity as well. This would mean that we prefer the grammar with the fewest cases of lexical ambiguity. Under these constraints, there would be two possible grammars as shown in Figure 1:

- | | |
|--|--|
| • sees::d= =d v:: $\lambda yx.S(x, y)$ | • sees::d= =d v:: $\lambda yx.S(x, y)$ |
| • John::d::j | • Phil::d::j |
| • Mary::d::m | • Sue::d::m |
| • Phil::d::p | • John::d::p |
| • Sue::d::s | • Mary::d::s |

(a) Grammar 1

(b) Grammar 2

Figure 1: Possible grammars for the data in table 1 .

While this data would not be sufficient to allow us to narrow down the grammar to a single solution, it does eliminate $2^3 - 2$ grammars. If we have one more scenario in which Phil is referred to while John is not present or vice versa, or the same thing with Sue and Mary, then we would be able to deduce the actual target grammar.

4.2 Learning pressures

- **Minimise lexical ambiguity!** Try to minimise the number of lexical items which share the same lemma.
- **Maximise truth!** Try to make the maximal number of sentences true in their scenarios.
- **Maximise parseability!** Try to find a valid parse for every sentence (n.b. probably need a mechanism for choosing the closest possible parse to do creolisation).
- **Maximise presupposition!**⁴ If a sentence can't be made true, its presuppositions should at least be true (in our simple model, a parse that mentions an absent atom has an undefined truth-value (i.e. its presuppositions are unsatisfied)).

References

- [1] Christopher Kennedy and Louise McNally. “Scale Structure, Degree Modification, and the Semantics of Gradable Predicates”. en. In: *Language* 81.2 (2005). Number: 2, pp. 345–381. ISSN: 1535-0665. DOI: 10.1353/lan.2005.0071. URL: <http://muse.jhu.edu/content/crossref/journals/language/v081/81.2kennedy.pdf> (visited on 05/15/2022).
- [2] Gregory M. Kobele. “Importing Montagovian Dynamics into Minimalism”. en. In: *Logical Aspects of Computational Linguistics*. Ed. by Denis Béchet and Alexander Dikovsky. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 103–118. ISBN: 978-3-642-31262-5. DOI: 10.1007/978-3-642-31262-5_7.
- [3] Liuba Papeo et al. “Abstract thematic roles in infants’ representation of social events”. eng. In: *Current biology: CB* 34.18 (Sept. 2024), 4294–4300.e4. ISSN: 1879-0445. DOI: 10.1016/j.cub.2024.07.081.
- [4] Baptiste Loreau Unger. *Toward a topological theory of the aspectual conceptualization of events*. LingBuzz Published In: Master’s thesis, Université Paris Cité. Sept. 2024. URL: <https://ling.auf.net/lingbuzz/008561> (visited on 12/11/2024).

⁴This is tongue in cheek, but the actual maximise presupposition condition might be useful down the road.