Learnability

Greg Kobele

Summer Semester, 2024

1 Categorial grammar

- Expressions are pairs of strings and their categories
- categories have a regular structure that indicates their distribution, and their combinatory options
- Categories CAT are an *infinite set*, constructed out of a finite set of *basic categories*, BASCAT in the following way: for α, β arbitrary categories, so too are the following:
  - α/β
  - $\alpha\backslash\beta$
- Rules are left and right function application:

$$(u, \alpha/\beta) + (v, \beta) \mapsto (uv, \alpha) \qquad \text{(FA)}$$
$$(u, \beta) + (v, \beta\backslash\alpha) \mapsto (uv, \alpha) \qquad \text{(BA)}$$

- Each type can be thought as denoting a set (here of strings).
  - associate each basic category with a finite set of strings
  - then we can compositionally define the denotation of any category:

$$[\![\alpha/\beta]\!] = \{v \mid \forall u \in [\![\beta]\!], uv \in [\![\alpha]\!]\}$$
$$[\![\beta\backslash\alpha]\!] = \{u \mid \forall v \in [\![\beta]\!], uv \in [\![\alpha]\!]\}$$

- We can extend this to non-empty *sequences* of categories using concatenation:

$$[\![\alpha \cdot \beta]\!] = \{uv \mid u \in \alpha \wedge v \in \beta\}$$

- Now we can see that the rules of categorial grammar are *sound*; e.g. if a string w is in the denotation of $\beta \cdot \beta\backslash\alpha$ then it is also in the denotation of α

  Theorem 1. *For all $\alpha, \beta \in$ CAT, $[\![\alpha/\beta \cdot \beta]\!] \subseteq [\![alpha]\!]$.*

  Proof. let $w \in [\![\alpha/\beta \cdot \beta]\!]$. Then there are strings $u \in [\![\alpha/\beta]\!]$ and $v \in [\![\beta]\!]$ such that $w = uv$. By the definition of $[\![\cdot/\cdot]\!]$, as $v \in [\![\beta]\!]$, $w \in [\![\alpha]\!]$. □

  Theorem 2. *For all $\alpha, \beta \in$ CAT, $[\![\beta \cdot \beta\backslash\alpha]\!] \subseteq [\![alpha]\!]$.*

  Proof. The proof is similar. □

- A categorial lexicon is a finite assignment of types to strings. For example:

string	type
the	d/n
boy	n
girl	n
happy	n/n
laughs	$d\backslash s$
loudly	$(d\backslash s)\backslash(d\backslash s)$
praises	$fc(d\backslash s)d$

  - We can understand this as assigning 'extra' strings to the denotation of certain categories. This is why the previous theorems only talk about *containment* (not identity).

- A grammar is completely determined by a lexicon and the basic category for sentences (usually s), as the grammatical functions are assumed to be constant.
  - The language of a categorial grammar is the set of all derivable sentences of its basic category

- The name of categories is only important in that we restrict who can combine with whom.
  - if we replaced every n with a q in the lexicon, we would still derive the same sentences in the same way.
  - We could even rename every category, without changing things, as long as we didn't accidentally give the same name to two (previously) different categories.

2 Substitutions and unification

- A (first-order) term is built out of function symbols each with a particular arity $f_1, \ldots, f_k$ and constant symbols $a_1, \ldots, a_j$, together with variables for constants (X,Y,Z)
  - $g(g(a))$, $f(a, X)$, $f(g(X), f(Y, g(g(X))))$
  - a term without variables is called *ground*
- each term can be viewed as a tree
  - The *head* of a term is its left-most symbol, i.e. its root
  - the subterm beginning with a particular symbol is the tree rooted at that symbol
- Two terms unify with one another iff there is a way to replace variables with terms that makes them identical; this is called a *substitution*
  - to unify $\texttt{man}(X)$ with $\texttt{man}(\texttt{socrates})$, we substitute the variable X with the term $\texttt{socrates}$; this can be written $\{X/\texttt{socrates}\}$, or $[X \mapsto \texttt{socrates}]$
- A substitution can replace multiple (different) variables with terms *simultaneously*
  - Let $\theta = \{X/\texttt{s}(Y), Y/\texttt{z}\}$, then $\texttt{sum}(X, Y, X)\theta = \texttt{sum}(\texttt{s}(Y), \texttt{z}, \texttt{s}(Y))$
- substitutions can be *composed*: $A(\theta\rho) = (A\theta)\rho$
  - the composed substitution must
    1. on $X \in \mathsf{dom}(\theta)$, replace this with $(X\theta)\rho$
    2. on $Y \in \mathsf{dom}(\rho) - \mathsf{dom}(\theta)$, replace this with $X\rho$
  - $\{X_1/\texttt{s}(Y_1, X_2), X_2/t\}\{Y_1/u, X_2/v\} := \{X_1/\texttt{s}(u, v), X_2/t, Y_1/u\}$
- *unification* is the problem of finding a substitution that makes two expressions identical
- There is a simple algorithm to find a most general unifier $\mathsf{mgu}(E, F)$ for two expressions E and F (if one exists):
  1. set $k = 0$ and $\theta_0 = \{\}$
  2. if $E\theta_k = F\theta_k$ stop, and output θ_k.

- otherwise, find the leftmost subtree E' and F' where $E\theta_k$ and $F\theta_k$ differ
- if one of E' and F' is a variable X and the other any term t, then if X does **not** occur in t, then set $\theta_{k+1} = \theta_k\{X/t\}$, increment k to $k+1$, and return to step 2.
  * otherwise, they are not unifiable

3 Substitutions and unifications in CG

- We formalize the intuition about renaming categories via substitutions
- Our terms are categories.
  - the start category s and the slashes are treated as symbols
  - the other basic categories are treated as variables
  - What are the mgus of
    * $\{c_1/c_2, c_3\}$
    * $\{c_1/c_2, c_1\backslash c_2\}$
    * $\{c_1/(c_2\backslash c_3), (\mathsf{s}/c_5)/c_6\}$
- Let $G\theta = \{(w, c\theta) \mid (w, c) \in G\}$ be the result of applying θ to each lexical category

Theorem 3. *If $G_1\theta \subseteq G_2$ for any grammars with the same start category s, then $[\![s]\!]_{G_1} \subseteq [\![s]\!]_{G_2}$.*

Proof. Consider any derivation tree of G_1 with root s. Applying any substitution to the categories in this tree results in a derivation tree of G_2 with root s and the same yield. □

4 Elasticity

- Consider the following sets:
  1. $\mathbb{N} = \{0, 1, 2, \ldots\}$, the set of natural numbers
  2. $\mathbb{E}$, the set of finite subsets of even numbers
  3. $\mathbb{O}$, the set of finite subsets of odd numbers
- It is easy to think of a learner for the class $\mathcal{L}_1 = \{\mathbb{N}\} \cup \mathbb{E} \cup \mathbb{O}$

    - if you see e and o, then guess $\mathbb{N}$, else guess exactly what you have seen

- The set $\mathcal{L}_2 = \{L_1 \cup L_2 \mid L_1, L_2 \in \mathcal{L}\}$ is not identifiable
    - it contains all finite subsets of $\mathbb{N}$, together with $\mathbb{N}$
    - seeing e and o together does *not* indicate that you should generalize

Definition 4.1. A class $\mathcal{L}$ has a **limit point** L iff there are languages $L_0, L_1, \ldots \in \mathcal{L}$ such that

$$L_0 \subset L_1 \subset L_2 \cdots$$

and $L \in \mathcal{L}$ is such that

$$L = \bigcup_{n \in \mathbb{N}} L_n$$

- $\mathcal{L}_2$ has a limit point $\mathbb{N}$, witness

$$\{0\} \subset \{0, 1\} \subset \{0, 1, 2\} \subset \cdots$$

- but $\mathcal{L}_1$ has no limit point

Theorem 4 (Kanazawa, 1998)**.** *If $\mathcal{L}$ has a limit point, it is not identifiable.*

Proof. Towards a contradiction, assume that ϕ identifies $\mathcal{L}$, which has limit point L. Then there must be a locking sequence t for ϕ and L. Since t is finite and $L = \bigcup_{n \in \mathbb{N}} L_n$, there must be some L_i which contains all of t. Then on any text for L_i which begins with t, ϕ converges instead on L, thus failing to identify L_i. □

- Why do we care about unions of languages?
    - $\mathsf{German} \cup \mathsf{English}$ is a rough approximation of my children's mind
    - Note: we are usually comfortable saying $\mathsf{High\text{-}class\ English} \cup \mathsf{Colloquial\ English} = \mathsf{English}$; what's the difference?

- *Code-switching* suggests that multilingualism is more than just language *union*
    - but it seems a reasonable first approximation

Definition 4.2. A class $\mathcal{L}$ has **infinite elasticity** iff there is a sequence of sentences $s_0, s_1, \ldots$ and a sequence of languages $L_0, L_1, \ldots$ in $\mathcal{L}$ such that for all $n \geq 0$:

- $s_n \notin L_n$, and
- $\{s_0, \ldots, s_n\} \subseteq L_{n+1}$

A class has **finite elasticity** iff it does not have infinite elasticity.

- $\mathcal{L}_1$ has infinite elasticity (as does $\mathcal{L}_2$):

$$\begin{array}{ccccc} 0 & 2 & 4 & 6 & \ldots \\ \emptyset & \{0\} & \{0,2\} & \{0,2,4\} & \ldots \end{array}$$

Theorem 5 (Wright, 1989)**.** *If $\mathcal{L}$ has finite elasticity, it is identifiable.*

Proof. Let $\mathcal{L}$ have finite elasticity. Assume some sort of ordering on the sentences of each language (e.g. alphabetical). We show that each $L \in \mathcal{L}$ has a distinguished set in Angluin's sense. Let $L \in \mathcal{L}$ be arbitrary; here is its distinguished set:

$$D_L = \{w \mid \exists L_i \in \mathcal{L}.w \text{ is least in } L - L_i\}$$

Note that if $L_i \supset L$, then $L - L_i = \emptyset$, and so supersets don't contribute to D_L. Now we show that D_L has the desired properties.

if $D_L \subseteq L'$ then $L' \not\subseteq L$: Let $L' \in \mathcal{L}$ st $D_L \subseteq L'$. Then the least element in L but not L' cannot exist, and so $L' \supset L$.

D_L is finite: Now suppose D_L is infinite (for a contradiction). Then for any $n > 0$, some element of D_L is longer than n. Consider some $s_0 \in D_L$, and some language $L_0 \in \mathcal{L}$ such that s_0 is least in $L - L_0$. There must also be some other language L_1 such that $L - L_1 \neq \emptyset$ and $s_0 \in L_1$; otherwise s_0 would be the (assumed non-existent) longest string in D_L. Thus for every $s_i \in D_L$, there is a language $L_{i+1} \in \mathcal{L}$ such that $L - L_i \neq \emptyset$ and $\{s_0, \ldots, s_i\} \subseteq L_{i+1}$. But then there is an infinite sequence of sentences $s_0, s_1, \ldots$ and languages $L_0, L_1, \ldots$ witnessing the infinite elasticity of $\mathcal{L}$.

□

Theorem 6 (Wright, 1989; Motoki, Shinohara and Wright, 1989; Kanazawa, 1998)**.** *If $\mathcal{L}_1$ and $\mathcal{L}_2$ have finite elasticity, then so does $\{L_1 \cup L_2 \mid L_1 \in \mathcal{L}_1$ and $L_1 \in \mathcal{L}_1\}$.*

5 Rigid CGs have finite elasticity

Kanazawa [1998] shows that a particular subclass of CGs has finite elasticity.

Definition 5.1. (Rigidity) A grammar is **rigid** iff there are no two distinct lexical items with the same string component.

- Derivation trees are written with leaves labeled with strings, and with internal nodes labeled with categories
  - every leaf must be immediately dominated by a unary branching node, labeled with its category.
  - every other internal node must be binary branching, and be licensed by either the FA or the BA rule.
- *f-structures* are binary branching trees with strings at the leaves and either a right arrow or a left arrow at the internal nodes.
- To go from a derivation tree to a f-structure,
  1. erase the unary branching nodes, and
  2. for each binary branching node, replace its label with an arrow pointing to the slash-category
- $\mathcal{F}(G)$ is the set of f-structures of derivations of s in G
- $\mathcal{G}_r$ is the set of rigid grammars over some particular vocabulary V. We assume they share a common start category, s.
  - $\mathcal{F}_r$ is the set $\{\mathcal{F}(G) \mid G \in \mathcal{G}_r\}$
  - $\mathcal{L}_r$ is the set $\{[\![s]\!]_G \mid G \in \mathcal{G}_r\}$

Theorem 7 (Kanazawa, 1998)**.** *$\mathcal{F}_r$ has finite elasticity.*

A learner for $\mathcal{F}_r$ on input $t_i = \{F_1, \ldots, F_i\}$

1.
   (a) assign s to the root of each
   (b) assign unique basic categories to each argument node
   (c) determine the categories of the functor nodes
2. Collect the lexical categories to obtain a grammar $GF(t_i)$
3. Unify all the different categories assigned to each individual word to obtain a rigid grammar $RG(t_i)$