Connectedness: Papers by and for Sarah VanWagenen
UCLA Working Papers in Linguistics 18: 90–101
Carson T. Schütze & Linnaea Stockall (eds.), 2014

Interpreted Learning

A framework for investigating the contribution of various information sources to the learning problem

John Case · Jeffrey Heinz · Gregory M. Kobele

Natural language utterances presents learners with more information about the underlying grammar than is typically encoded in the orthographic string. While multiple information sources such as prosody, and semantics can be encoded as a single object, allowing the results of typical learning frameworks to apply, this coding obscures the question of exactly how the learner can draw inferences about a single object from information made available by these multiple different perspectives. Here we tease apart the contribution of different information sources to the learning problem by generalizing Gold's learning paradigm. The main result is a proof that multiple sources of information can interact synergistically to facilitate learning of the target underlying grammar.

Keywords Inductive inference, Strong generative capacity

Introduction

For a long time, work on formal learning theory studied (sub)classes of languages strictly *weaker* than those considered 'language-like' by theoretical (computational) linguists. Recently, work by Clark (2010) and others (Becerra-Bonache et al. 2010; Yoshinaka 2011) has discovered classes of string languages which are (feasably) learnable under various criteria, and which fall within the class of mildly context sensitive languages, a language family which is thought to be sufficiently rich to describe all natural language patterns (Joshi 1985). Although some of these learning theoretic results are known in linguistic circles, they are dismissed as being linguistically irrelevant, for the reason that the learned grammars do not assign the right structures to the strings they generate (emphasis in the original):

> it does not address the original POS ['Poverty of the Stimulus'] question, which [. . .] depends on which *structures* a language generates (i.e., a language's strong generative capacity). (Berwick et al. 2011)

The emphasis on the structures generated by grammars (rather than on the strings) is fully explicit in Chomsky (1990), where the set of strings generated by the grammar is dismissed as linguistically uninteresting. At first blush this is puzzling, as strings are observables (more or less), while structures are not. But, of course, arguments *for* certain structures often involve semantic considerations (e.g. the syntactic level of LF), and one might suspect

© 2014 John Case, Jeffrey Heinz, Gregory M. Kobele
This is an open-access article distributed under the terms of a Creative Commons Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/).

that what linguists are actually objecting to is the fact that these learning algorithms do not seem to support compositional semantic interpretation (which, as noted by Berwick et al. (2011), is recognized already by the authors). Indeed, we suggest that the empirical content of ‘rightness of structure’ is exhausted by the relevant observable facts about the string, such as its semantic interpretation, its prosodic structure, etc. If this is in fact the substance of the objection, one might wonder how to address it. As hinted at above, we believe that the linguistic deficiencies discussed in Berwick et al. (2011) are due primarily to the fact that the linguistic arguments for assigning certain structures to strings (and hence for favoring one from among a number of weakly equivalent grammars) are at least in part based on non-distributional (i.e. non-syntactic) properties of strings. Accordingly, a natural way to proceed is to investigate how information about these non-syntactic properties of strings can be incorporated into the learning setup, and what affects this may have.

In this paper we do just this. We explore the idea that “having the right structure” is synonymous with “allowing for the derivation of the relevant facts,” propose a variant of the Gold learning paradigm (‘interpreted learning’) which makes sense of these notions, and prove some first theorems about learnability in this setup. Note that this is not the only reasonable way to proceed. It is, in fact, something of a formal retreat. Clark (2013) has proposed that, instead of assuming that the structures we want to derive require more than just information about word order to reconstruct, we investigate learnable classes for which there is a notion of a canonical grammar for each language. We believe, however, that this sort of proposal does not address the objections of Berwick et al. (2011), who are not interested in learning *a* grammar for a language (canonical or not), but rather the right grammar, which supports semantic interpretation among other things.

1 Mathematical Preliminaries

$\mathbb{N}$ is the set of non-negative integers. For $n \in \mathbb{N}$, we write $[n]$ for the set $\{i \in \mathbb{N} : i < n\}$. $[0] = \emptyset$. A sequence of length $n \in \mathbb{N}$ over some set A is a function from $[n]$ to A. An infinite sequence over A is a function from $\mathbb{N}$ to A. We write $\mathtt{length}(s)$ for the length of a sequence s ($\mathtt{length}(s) = \omega$ if s is infinite). Given a sequence s, we write s_i for the i^{th} element of s (starting at 0. For s a sequence, and $i \in \mathbb{N}$ (such that i is less than the length of s, if s is finite), $s[i]$ denotes the initial segment of s of length i; i.e. the sequence of length i such that for all $k < i$, $s[i]_k = s_k$. For a sequence s over A, we write $\mathtt{content}(s)$ for the range of s ($\mathtt{content}(s) = \{s(n) : n < \mathtt{length}(s)\}$). We write (short, finite) sequences in the usual way; *aba* is the sequence of length 3 whose first element is a, second b, and third a. We write ε for the empty string. Note that $\mathtt{length}(aba) = 3$, $\mathtt{content}(abc) = \{a,b\}$, $\mathtt{length}(\varepsilon) = 0$, and $\mathtt{content}(\varepsilon) = \emptyset$. We write A^n for the set of all sequences of length n over A, and $A^* = \bigcup_{n \in \mathbb{N}} A^n$ for the set of all finite sequences over A. A subset $B \subseteq A^2$ is a (binary) relation over A, and we write aBc to indicate that the sequence $ac \in B$. For any A, we write Δ_A for the identity relation over A; the least reflexive relation over A.

Given a set A, we write $B \subseteq_{fin} A$ if B is a finite subset of A (i.e. if $B = \mathtt{content}(s)$ for some finite sequence s over A). We write 2^A for the powerset of A.

We write $f : A \to B$ for functions from A to B; injections (where $f(a) = f(b)$ implies $a = b$) are sometimes written $f : A \rightarrowtail B$. Given a function $f : A \to B$, we extend it pointwise to sequences ($f^* : A^* \to B^*$) and sets ($2^f : 2^A \to 2^B$) as usual. We systematically abuse notation and abbreviate f^* and 2^f as f.

We recall some notions of the identification in the limit learning paradigm (Gold 1967), adopting the presentation of Kanazawa (1998). A *grammar system* is a triple $\langle \Omega, \mathscr{P}, L\rangle$, where Ω is a denumerable set of grammars, $\mathscr{P}$ is the set of possible sentences, and $L : \Omega \to 2^{\mathscr{P}}$ is the map assigning to each grammar $G \in \Omega$ the language it generates. A *text* is an infinite sequence $s : \mathbb{N} \to \mathscr{P} \cup \{\#\}$ where #, a special symbol not in $\mathscr{P}$, is a 'pause' indicating the absence of information. The sequence s is a text for some $L \subseteq \mathscr{P}$ iff $\texttt{content}(s)/\{\#\} = L$. A *learner* $A : (\mathscr{P} \cup \{\#\})^* \to \Omega$ is a function mapping finite sequences to grammars. Given a text s, A *converges* to G on s iff there is some $n \in \mathbb{N}$ such that for all $m \geq n$, $A(s[m]) = A(s[n]) = G$; i.e. A converges to G on s iff it maps all but finitely many initial segments of s to G. If s is a text for L, A *identifies* s iff A converges on s to a grammar for L. A identifies L iff A identifies every text for L. Finally, A identifies a class of grammars $\mathscr{G} \subseteq \Omega$ iff A identifies each $L(G)$, for every $G \in \mathscr{G}$. We may say that A identifies a class $\mathscr{L}$ of languages iff A identifies a class $\mathscr{G}$ of grammars such that $\mathscr{L} = \{L(G) : G \in \mathscr{G}\}$. Gold (1967) proved that any finite class is identifiable in the limit. He also proved that no learner identifies any strict superset of the class of all finite languages.

2 Interpreted Grammars

Here we present a variant of the identification-in-the-limit paradigm, which allows for the incorporation of information about the structure which should be assigned to a datum. This generalization is based on the linguistic distinction between *tecto* and *pheno* structure (Curry 1961; de Groote 2001; Muskens 2001). According to this view, a grammar generates tectostructures (derivation or 'deep' structures), which are then transduced into phenostructures (derived or 'surface' structures). 'Syntactic universals' are viewed as a description of regularities either in tectostructures or in their translation into phenostructures. In a number of modern grammar formalisms (for example, tree adjoining grammars, minimalist grammars), the language-specific (i.e. non-universal) aspects of individual grammars are reduced to a specification of the tecto- and pheno- properties of a finite number of (tecto-)leaves (the lexical items), with the tecto- and pheno- properties of all other tecto-formatives being completely determined in terms of these. A grammar, then, is completely determined by its choice of lexical items. From this perspective, grammar induction is the problem of identifying a finite set of lexical items through the obscuring effect of the *a priori* tecto primitives and their known pheno-properties. In other words, the *observed* structures are the result of combining in *known* ways a finite number of *unknown* objects.

2.1 Definitions

We formalize this perspective as follows. We let $\mathscr{G}$ be a countably infinite set (of grammars), and $\mathscr{T}$ and $\mathscr{P}$ countably infinite sets (of tectostructures and phenostructures, respectively). Given an interpretation scheme $f : \mathscr{T} \to \mathscr{P}$ which interprets tectostructures as phenostructures, we write $\mathscr{A}(G) \subseteq \mathscr{T}$ for the set of tectostructures associated with a grammar G, and $\mathscr{O}_f(G) \subseteq \mathscr{P}$ for the set of phenostructures generated by G; crucially, we require that $\mathscr{O}_f(G) = \{f(t) : t \in \mathscr{A}(G)\}$. For a given grammar G, $\mathscr{A}(G)$ is its *abstract* or tecto-language, and $\mathscr{O}_f(G)$ is its *observable* or pheno-language (with respect to f). We call two grammars G_1, G_2 f-equivalent (written $G_1 \equiv_f G_2$) iff their observable languages are identical. We will sometimes suppress the interpretation scheme f when it is clear from context.

Consider a simple example, the case of finite state machines. We fix countable sets Q and Σ to be the logically possible states and alphabet symbols respectively, and take a grammar $G \in \mathscr{G}$ to be a selection of some $q_s^G \in Q$, a finite $Q_f^G \subset Q$, and a finite $\delta^G \subset Q \times \Sigma \times Q$.[1] A tectostructure is a possible run, i.e. a sequence of the form $t_0, \ldots, t_n$, where each $t_i = \langle q_i, \sigma, q_{i+1} \rangle$; the tectostructures associated with a grammar G are those sequences $t_0, \ldots, t_n$ whose first component is of the form $t_0 = \langle q_s^G, \sigma_0, q_1 \rangle$, whose last is of the form $t_n = \langle q_n, \sigma_n, q_f \rangle$ for some $q_f \in Q_f^G$, and where each $t_i \in \delta^G$. The standard interpretation scheme f_{str} acts homomorphically on tectostructures: $f_{str}(\langle q, \sigma, q' \rangle ^\frown t) = \sigma ^\frown f_{str}(t)$. The function f_{str} is simply the function mapping each run of the machine to the string it recognizes. This is not the only possible interpretation scheme of course, another is given by $f_{set}(\langle q, \sigma, q' \rangle ^\frown t) = \{\sigma\} \cup f_{set}(t)$, which treats strings as the sets of letters they contain. Clearly, f_{str} gives more information about a tectostructure t than does f_{set}. (This is clear, in this case, because $f_{set} = g \circ f_{str}$, for the string homomorphism g which maps letters to their unit sets, and concatenation to set union.)

A (positive) f-interpreted text for grammar $G \in \mathscr{G}$ is an infinite sequence $m \in \mathbb{N} \to \mathscr{O}_f(G) \cup \{\#\}$ such that $\texttt{content}(m)/\{\#\} = \mathscr{O}_f(G)$. A learner is a function $A : (\mathscr{O}_f(G) \cup \{\#\})^* \to \mathbb{N}$. We say that a learner converges to H on an interpreted text m iff there is some $i \in \mathbb{N}$ such that for all $j > i$, $A(m[j]) = A(m[i]) = H$. A learner f-identifies G if it converges to some f-equivalent H on all texts for G, and it f-identifies a class of interpreted grammars $\mathscr{G}$ if it f-identifies each $G \in \mathscr{G}$.

Returning to our example of finite state machines, note that there is a subtle interplay regarding the amount of information about tectostructures which is preserved by an interpretation scheme f and the requirement that identification be only up to f-equivalence: it is clear that the class of finite state machines is f_{set}-identifiable (the set $\mathscr{O}_{f_{set}}(G)$ for any G is a finite set), while it is equally clear that this class is not f_{str}-identifiable (as shown already by Gold).[2] Even though f_{str} gives more information about tectostructures than does f_{set}, f_{str}-equivalence is much harder to achieve than is f_{set}-equivalence.

2.2 *Intuition*

The relation between the perspective on natural language grammars described above in §2 and the formalization in §2.1 is as follows. A grammar G is a finite subset of the (countable) set of all possible lexical items, and the set $\mathscr{T}$ of all possible tectostructures is given with respect to this set of all possible lexical items. The interpretation scheme f is the universal mapping associating tecto- with pheno-structures. As such, f offers a *perspective* or *view* of a tectostructure through a phenostructure. Different choices for f can provide more or less information about the possible tectostructures underlying a given phenostructure. In the cases we find linguistically interesting, $f = g_1 \times \cdots \times g_k$, where each g_i represents a different map from tectostructures into some observable domain (such as semantics, prosody, etc.). Note that the relation between structures and (pronounced) strings is often taken to be

[1] We can think of $\mathscr{G}$ as a 'family of grammars', on par with the more familiar notion of a family of languages.

[2] These claims are justified in theorem 1.

a function of structure (see e.g. Morawietz (2003)). Many linguists assume in addition there to be significant portions of the prosodic properties of an utterance which are determined by its syntactic structure (Wagner 2005), and there is evidence to suggest that prosodic information is even emphasized in mother-child interaction (Nelson et al. 1989). While we do not find the thought that a complete semantic representation is accessible to a human language learner particularly convincing, it does seem reasonable to grant that some basic facets of semantics are recoverable from the context of use, such as might be represented in simple semantic dependency graphs (where are represented argument structural notions such as agent-of, patient-of, instrument-of, and so on), which are easily represented as compositions of 'forgetful' functions with the standard semantic interpretation maps (Shieber 2006; Kobele 2012).[3] Indeed, recent work (Kallmeyer and Kuhlmann 2012) on relating TAG derivation trees to dependency structures (which might be thought of as encoding just this kind of reduced semantic information) can be seen as approximating the effects of a lambda homomorphism (de Groote 2001) which *lowers* the types of quantified noun phrases (from $(et)t$ to e) on the lambda term obtained by semantically interpreting the derivation term.

Section 2.4 below addresses the learnability of the tectostructures given such a multidimensional f.

2.3 Relations to Gold Learning

The standard Gold paradigm is the special case of this one where the tectostructures are languages and the function f is the identity function; our f-identification is then simply identification of an actually equivalent grammar. Given a (language) learning problem in the standard Gold paradigm (i.e. a class $\mathscr{G}$ of grammars, and a mapping L associating grammars with their languages, we recast it into the terms here by setting $\mathscr{G} = \mathscr{G}$, $\mathscr{T} = \bigcup_{G \in \mathscr{G}} L(G)$, $\mathscr{A}(G) = L(G)$, and f the identity function. Furthermore, this is the special case of the standard Gold paradigm where the mapping from grammars to languages is factored in a particular way (into the composition of f and $\mathscr{A}$). We make this precise by, for a grammar G and an interpretation scheme f, defining G_f to be some grammar such that $\mathscr{A}(G_f) := \{f(w) : w \in \mathscr{A}(G)\}$ (i.e. $\mathscr{A}(G_f) = \mathscr{O}_f(G)$). Note that G and G' are f-equivalent iff G_f and G'_f are equivalent. For a set of grammars $\mathscr{G}$, we write $\mathscr{G}_f$ for the pointwise extension of $\cdot_f$ to $\mathscr{G}$.

Theorem 1. *$\mathscr{G}$ is f-identifiable if and only if $\mathscr{G}_f$ is identifiable in the Gold sense.*

Proof. Let $\mathscr{G}$ be f-identifiable by A, and define $A_f(s) := A(s)_f$. Let $G_f \in \mathscr{G}_f$ and H a text for G_f be arbitrary. Then H is also an (f-)text for G, and A converges on it to an f-equivalent G' to G, and so A_f converges to G'_f, which is equivalent to G_f.

For the right-to-left direction, let $\mathscr{G}_f$ be identifiable by A_f. To define a learner A for $\mathscr{G}$, we need to invert $\cdot_f$; we accordingly set, for $G_f \in \mathscr{G}_f$, $(G_f)^f$ to be a fixed element of $\{G' : G'_f = G_f\}$. Then $A(s) = (A_f(s))^f$. Let $G \in \mathscr{G}$ and H an f-text for G be arbitrary. The H is a text for G_f and therefore A_f converges on it to an equivalent G'_f. But then A converges to $(G'_f)^f$, which is f-equivalent to G. □

[3] Siskind (1996) explores procedures for associating words with semantic properties in the face of noise, ambiguity, etc. Kanazawa (2001) recasts this work in a framework more congenial to the setting investigated here.

Despite the formal equivalence, making this factorization explicit allows for a different landscape of relatedness of particular learning problems. As shown in theorems 5 and 6, this landscape is non-trivial. We think that this is of sufficient importance to justify this perspective shift.

2.4 The Landscape of Learning

The distinction between the tectostructures generated by the grammar and the phenostructures associated with them gives us an intuitive notion of observable equivalence, which, in the definition of identification, requires the learner to converge to a grammar which assigns the right structure to its strings. From the following simple theorem we infer a number of desirable corollaries.

Theorem 2. *Fix $\mathscr{G}$ and tectostructures $\mathscr{T}$. Let $f : \mathscr{T} \to \mathscr{P}$, and $h : \mathscr{P} \rightarrowtail \mathscr{Q}$ an injection. Then $\mathscr{G}$ is f-identifiable iff $\mathscr{G}$ is $h \circ f$-identifiable.*

Proof. Let f and h be as in the theorem. Note that for any $t, t' \in \mathscr{T}$, $f(t) = f(t')$ iff $(h \circ f)(t) = (h \circ f)(t')$, and thus f-equivalence and $(h \circ f)$-equivalence are equivalent, and that for any grammar $G \in \mathscr{G}$, an $h \circ f$-text for G is an $h^{-1} \circ h \circ f = f$-text for G. For the left to right direction, let A be a learner which f-identifies $\mathscr{G}$. We define a learner B which will $h \circ f$-identify $\mathscr{G}$ by setting $B = A \circ h^{-1}$ (i.e. $(\langle a_1, \ldots, a_k \rangle) = A(\langle h^{-1}(a_1), \ldots, h^{-1}(a_k) \rangle)$). Let T be a $h \circ f$-text for G. Then as A converges to some G' on $h^{-1}(T)$, $B = A \circ h^{-1}$ converges to G' on T. As G' is f-equivalent to G, they are also $(h \circ f)$-equivalent. As T and G were arbitrary, B $(h \circ f)$-identifies $\mathscr{G}$. For the other direction, we proceed symmetrically; let B be a learner which $(h \circ f)$-identifies G, we define $A = B \circ h$. Then for T an f-text for G, B converges on $h(T)$ to G', and thus A converges on T to G'. As B identifies $\mathscr{G}$, $G' \equiv_{h \circ f} G$, and so $G' \equiv_f G$, whence, as T and G were arbitrary, A f-identifies $\mathscr{G}$. □

Theorem 2 entails that certain structural ways of combining data sources are equivalent.

Corollary 3. *$\mathscr{G}$ is*

1. *f-identifiable iff it is $\langle f, f \rangle$-identifiable*
2. *$\langle f, g \rangle$-identifiable iff it is $\langle g, f \rangle$-identifiable.*
3. *$\langle f, \langle g, h \rangle \rangle$-identifiable iff it is $\langle \langle f, g \rangle, h \rangle$-identifiable.*

Proof. By theorem 2, as witnessed by the injections $h_1(x) = (x, x)$, $h_2(x, y) = (y, x)$, and $h_3(x, (y, z)) = ((x, y), z)$. □

Furthermore, adding extra 'predictable' information about a source does not change the learnability of a class.

Corollary 4. *Let $\mathscr{G}$ be f-identifiable, and let g be arbitrary. Then $\mathscr{G}$ is $\langle f, g \circ f \rangle$-identifiable.*

Proof. By theorem 2, as witnessed by $h(x) = (x, g(x))$. □

We now turn to the main results of this paper. In particular corollary 3 suggests that the manner in which combination of information sources affects learning is purely structural and independent of the manner of combination. That this is not so is demonstrated by theorems 5 and 6. Theorem 5 states that the learnability of classes of tectostructures is not necessarily preserved under increased information about the identity of tectostructures. Theorem 6 states, intuitively, that combining information from multiple sources can have a synergetic effect on learning.

We recall that a class of languages $\mathscr{L}$ has an *accumulation point* $L \in \mathscr{L}$ iff there is an infinite sequence of finite sets $S_1, S_2, \ldots$ such that

1. $S_i \subseteq S_{i+1}$, for all i,
2. $\bigcup_{i \in \mathbb{N}} S_i = L$
3. for all i, there is some $L_i \in \mathscr{L}$ such that $S_i \subseteq L_i$ and $L_i \subset L$

Kapur (1992) proves that a class of languages $\mathscr{L}$ is identifiable in the limit iff $\mathscr{L}$ does not have an accumulation point.

Theorem 5. *It is not the case that, for every f and g, f- and g-identifiability imply $\langle f, g\rangle$-identifiability.*

Proof. Let $L \subseteq \Sigma^*$ be arbitrary but infinite, and let $x_0, x_1, \ldots$ be an enumeration of the words of L. We define $S_0 = \{\langle x_0, x_0\rangle\}$ and $S_{i+1} = S_i \cup \{\langle x_{i+1}, x_{i+1}\rangle, \langle x_0, x_{i+1}\rangle\}$, and define $L_i = \Delta_L \cup S_i$. Let $L_\infty = \Delta_L \cup (\{x_0\} \times L)$. We let the set of possible tectostructures $\mathscr{T} = \{L_i : i \in \mathbb{N}\} \cup \{L_\infty\}$. Let G_∞ be a grammar such that $\mathscr{A}(\mathscr{G}_\infty) = L_\infty$, let $\mathscr{G} = \{G_i : i \in \mathbb{N}\} \cup \{G_\infty\}$ and let $\mathscr{A}(G_i) = L_i$.

We set $f = \pi_1$ and $g = \pi_2$ be the left and right projection functions respectively. Since every tectostructure contains Δ_L, it follows that $\mathscr{A}(\mathscr{G}_f) = \{L\} = \mathscr{A}(\mathscr{G}_g)$, and thus $\mathscr{G}$ is both f- and g-identifiable (by Theorem 1 since $\{L\}$ is a class of finite cardinality).

We show that L_∞ is an accumulation point for $\mathscr{G}_{\langle f,g\rangle}$. By construction, it is the case that $S_i \subseteq S_{i+1}$ and $S_i \subset_{fin} L_i$ for all i. Each S_i is contained in L_∞ as S_i contains pairs of the form $\langle x, x\rangle$, which is in Δ_L, or of the form $\langle x_0, y\rangle$, which is in $\{x_0\} \times L$. As each L_i is the union of subsets of L_∞, $L_i \subseteq L_\infty$ for each i. The properness of the inclusion is witnessed by the pair $\langle x_0, x_{i+1}\rangle$ for each L_i. It remains to show that $\bigcup_{i \in \mathbb{N}} S_i = L_\infty$. The left to right direction follows from the fact that the left-hand term is the union of subsets of the right-hand term. For the right to left direction let $\langle x_i, x_j\rangle \in L_\infty$. Then either $i = j$ or $i = 0$. In both cases, $\langle x_i, x_j\rangle \in S_j \subseteq \bigcup_{k \in \mathbb{N}} S_k$.

As $\mathscr{G}_{\langle f,g\rangle}$ contains an accumulation point L_∞, it is not identifiable in the limit, and thus by Theorem 1 $\mathscr{G}$ is not $\langle f, g\rangle$-identifiable. □

We recall that no superfinite class of languages (one which contains all finite languages and at least one infinite one) is identifiable in the limit, and that the class of all finite languages *is* identifiable in the limit.

Theorem 6. *It is not the case that, for every f and g, $\langle f, g\rangle$-identifiability implies f- or g-identifiability.*

Proof. Fix a non-empty alphabet Σ. Set $\mathscr{G}$ to be the set of all pairs $\langle L_1, L_2\rangle$ such that one of the following (mutually exclusive) conditions holds:

1. $L_1 = \{\varepsilon\}$ and $L_2 = \Sigma^*$
2. $L_1 = \Sigma^*$ and $L_2 = \{\varepsilon\}$
3. L_1 and L_2 are finite subsets of Σ^* distinct from $\{\varepsilon\}$

We define $f = \pi_1$ and $g = \pi_2$. Then $\mathscr{G}_f = \mathscr{G}_g = \{L : L \subset_{fin} \Sigma^*\} \cup \{\Sigma^*\}$, and thus by theorem 1 $\mathscr{G}$ is neither f- nor g-identifiable.

The following learner will $\langle f, g\rangle$-identify $\mathscr{G}$. On $\langle u_1, v_1\rangle, \ldots, \langle u_n, v_n\rangle$, the learner conjectures $\langle\{\varepsilon\}, \Sigma^*\rangle$ if $u_1 = \ldots = u_n = \varepsilon$, the learner conjectures $\langle\Sigma^*, \{\varepsilon\}\rangle$ if $v_1 = \ldots = v_n = \varepsilon$, and the learner conjectures $\langle\{u_1, \ldots, u_n\}, \{v_1, \ldots, v_n\}\rangle$ otherwise. □

The implications of these theorems are discussed in the conclusion.

3 Learning from Szilard Languages

In this section we consider how to recast the result of Mäkinen (1992) into the present framework. Mäkinen, in work very much anticipating this one, investigates learning context-free grammars from structural information (as an alternative to Sakakibara (1992), who investigated learning with structural information in the form of unlabeled derivation trees). Recasting this work into the present framework provides a useful perspective not only on how the present framework works, but also on the result itself. We will see that the present framework forces one to be very explicit about the distinction between the properties which are shared by the entire class of grammars (linguistic universals) and those which must be learned (language particulars).

Mäkinen defines texts for a grammar G to consist of pairs $\langle \mathtt{yield}(t), \mathtt{sz}(t)\rangle$, where t is a derivation tree generated by G, and where `yield` is the standard yield mapping and `sz` is the *Szilard* mapping which takes a derivation tree to the sequence of production names encountered in a preorder traversal. He observes that the Szilard languages of CFGs are *very simple*, and are thus efficiently identifiable (Yokomori 2003).[4] As the language of a CFG in GNF is the image of a symbol-to-symbol mapping applied to its Szilard language,[5] the efficient identifiability of GNF grammars in Mäkinen's setting is straightforward.

It is instructive to consider how to present this learning scenario in our setting. We help ourselves to an countably infinite set $\mathscr{N}$ (of possible non-terminals), and a countably infinite set Σ (of possible terminals), in terms of which we define the countable alphabet $\mathscr{R} = \mathscr{N} \times (\mathscr{N} \cup \Sigma)^*$ (of possible rules). In order to describe Mäkinen's learning problem, we need to provide a definition of the Szilard mapping, the kernel of which associates a rule with its (unique) name. One possibility would be to take the Szilard kernel to be antecedently given (i.e., for some infinite set $\mathscr{S}$ of szilard names, $\mathtt{sz} : \mathscr{R} \rightarrowtail \mathscr{S}$). This however would give the learner complete information about the identity of each production of the target grammar; the learner has access to the functions in terms of which observable languages are defined. Once the learner knows that the rule named 'o' was used, he may simply look up that the rule $r \in \mathscr{R}$ is that rule, which makes the learning problem trivial.

[4] A CFG in GNF is very simple iff for every terminal symbol a, there is exactly one rule with that terminal symbol (recall that a CFG is in GNF iff all of its rules are of the form $A \rightarrow aB_1 \ldots B_n$).

[5] A slightly more complex mapping mediates between the Szilard language of a grammar in CNF and its yield.

Instead, we must somehow parameterize the Szilard mapping on the grammar. We do this by defining a grammar to be a triple $G = \langle S, R, \mathit{ker} \rangle$ where $S \in \mathscr{N}$, $R \subset_{fin} \mathscr{R}$ and $\mathit{ker} : R \rightarrowtail \mathscr{S}$. This means that the learner must do more than simply identify a context-free grammar (up to equivalence); he must also 'decode' the names used to refer to the rules, which is represented as *ker*.

The Szilard mapping itself, as a mapping from tectostructures to strings over $\mathscr{S}$, does not make any reference to the grammar, and so information about names must already be present in the tectostructures. The possible tectostructures should then be the subset $\mathscr{T} \subset (\mathscr{R} \times \mathscr{S})^*$ of local trees containing a subterm $\langle (N \to w_0 N_1 w_1 \ldots N_k w_k), o \rangle (t_1, \ldots, t_n)$ iff $n = k$ and the root of each $t_i = \langle (N_i \to u_i), o_i \rangle$, and the set of tectostructures of a grammar G are the possible tectostructures containing only nodes of the form $\langle r, \mathit{ker}(r) \rangle$, and whose roots are of the form $\langle (S \to w), o \rangle$.[6]

Now we can define the given mappings $\texttt{yield} : \mathscr{T} \to \Sigma^*$ and $\texttt{sz} : \mathscr{T} \to \mathscr{S}^*$ in the obvious (and standard) way.[7] It follows, as Mäkinen points out, from the fact that Szilard languages are very simple that the class of context-free grammars is sz-learnable. It is equally clear that the class of context-free grammars is not yield-learnable (they are a strict superset of the class of finite languages). Mäkinen shows that two subclasses of context-free grammars (those in GNF and those in CNF) are in fact $\langle \texttt{yield}, \texttt{sz} \rangle$-learnable. Note that this does not (quite) follow from corollary 4, because even though yield is in fact definable in terms of the Szilard mapping in the following way: $\texttt{yield} = h \circ \mathit{ker}^{-1} \circ \texttt{sz}$, for $h(N \to aw) = a$, *ker* is not accessible to the learner, being dependent on the particular choice of grammar. It is, however (as Mäkinen observes), very easy to determine what *ker* must be, given the *a priori* restrictions on grammar form, and the input $\langle \texttt{yield}(t), \texttt{sz}(t) \rangle$. Indeed, for a grammar in GNF, for each tectostructure t, $\texttt{length}(\texttt{yield}(t)) = \texttt{length}(\texttt{sz}(t))$, and for r_i the rule with name $\mathit{ker}(\texttt{sz}(t)_i)$, the unique nonterminal on its right-hand side is $\texttt{yield}(t)_i$. The remainder of each rule, its shape and the identity of the left- and right-hand nonterminals, is exactly the same as that of the very simple grammar for its Szilard language, which is obtained by Yokomori's algorithm.[8]

We see then that Mäkinen's result just barely avoids falling under the purview of corollary 4; the yield perspective on tectostructures is almost completely predictable given the sz perspective.

Conclusion

We have made some first steps toward a better understanding of how combining multiple sources of information about the same objects can affect learning, and thus to the goal of meeting the challenges by linguists to the relevance of the learning results mentioned in the introduction to linguistics. Theorems 5 and 6 demonstrate that whether adding or removing information sources about the underlying objects affects learning depends on the

[6] For the sake of readability, we write throughout $N \to w$ for the rule $\langle N, w \rangle$.

[7] The mapping sz is simply the preorder traversal of the second projection function extended over trees: $\texttt{sz}(\langle r, o \rangle (t_1, \ldots, t_k)) = o ^\frown \texttt{sz}(t_1) ^\frown \ldots ^\frown \texttt{sz}(t_k)$. The mapping yield is the composiiton of the standard one with the first projection function (as the nodes of our trees are pairs), which satisfies $\texttt{yield}(\langle (N \to w_0 N_1 w_1 \ldots N_k w_k), o \rangle (t_1, \ldots, t_k)) = w_0 ^\frown \texttt{yield}(t_1) ^\frown w_1 ^\frown \ldots ^\frown \texttt{yield}(t_k) ^\frown w_k$

[8] If the grammar is in CNF this procedure is more complicated (but still efficient); see Mäkinen (1992) for more details.

way in which the objects in effect coordinate these sources.[9] In the proof of Theorem 5, the complexity of the tectostructures was only visible when viewed simultaneously through two information sources. On the other hand, in Theorem 6, the underlying simplicity of the tectostructures was obscured when viewed from only a single perspective. Theorem 2 and its corollaries demonstrate that the addition of 'completely predictable' information does not influence the outcome of learning, as is to be expected.

In other words, there is a trade-off between the amount of information we have about some objects, and the complexity of that set of objects. The proof of theorem 5 works because we have a complex set of objects, but the information the individual projections give about that set is minimal, and that of theorem 6 works because, although we have a simple set of objects, the individual projections make it look more complicated than it is.

From a linguistic perspective, the proof of theorem 5 is interesting precisely because the tectostructures exhibit a degree of arbirtrariness that we do not expect to find in natural language. Indeed, linguists often assume that the computation of the prosodic form of an utterance and the computation of its logical form proceed in a similarly compositional manner and that these relations are constrained in nontrivial ways. While in the interpreted learning setting explicated here, these computations are expressed by the function f, it is not unreasonable to expect that complex tectostructures will be similarly constrained.

We plan to investigate desirable kinds of coordination between linguistically-motivated information sources in future work. We anticipate that the advances in distributional learning and the grammatical inference of mildly context-sensitive languages more generally, when augmented with additional sources of information plausibly available to children, will be able to assign the 'right structures' to the observed strings. We also hope that the present framework will help us understand at a deeper level recent advances in supervised learning of sound-meaning mappings (Zettlemoyer and Collins 2005; Kwiatkowksi et al. 2010).

References

Becerra-Bonache, Leonor, John Case, Sanjay Jain, and Frank Stephan. 2010. Iterative learning of simple external contextual languages. *Theoretical Computer Science* 411:2741–2756. Special Issue for *ALT'08*.

Berwick, Robert C., Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science* 35:1207–1242.

Chomsky, Noam. 1990. On formalization and formal linguistics. *Natural Language and Linguistic Theory* 8:143–147.

Clark, Alexander. 2010. Efficient, correct, unsupervised learning of context-sensitive languages. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 28–37. Uppsala, Sweden: Association for Computational Linguistics. URL papers/conll2010.pdf.

[9] This seems as though it should have something to do with the notion of mutual information, though note that even in the proof of theorem 5, f and g are not completely independent: for any grammar $G \in \mathscr{G}$, and any tectostructure $p \in G$, $f(p) = x_{i+1}$ guarantees that $g(p) = x_{i+1}$.

Clark, Alexander. 2013. Learning trees from strings: A strong learning algorithm for some context-free grammars. *Journal of Machine Learning Research* 14:3537–3559.

Curry, Haskell B. 1961. Some logical aspects of grammatical structure. In *Structure of language and its mathematical aspects*, ed. Roman O. Jakobson, volume 12 of *Symposia on Applied Mathematics*, 56–68. Providence: American Mathematical Society.

Gold, E. Mark. 1967. Language identification in the limit. *Information and Control* 10:447–474.

de Groote, Philippe. 2001. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, 148–155.

Joshi, Aravind K. 1985. How much context-sensitivity is necessary for characterizing structural descriptions. In *Natural language processing: Theoretical, computational and psychological perspectives*, ed. David Dowty, Lauri Karttunen, and Arnold Zwicky, 206–250. NY: Cambridge University Press.

Kallmeyer, Laura, and Marco Kuhlmann. 2012. A formal model for plausible dependencies in lexicalized tree adjoining grammar. In *Proceedings of the Eleventh International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+11)*, 108–116. Paris.

Kanazawa, Makoto. 1998. *Learnable classes of categorial grammars*. Stanford University.: CSLI Publications.

Kanazawa, Makoto. 2001. Learning word-to-meaning mappings in logical semantics. In *Proceedings of the Thirteenth Amsterdam Colloquium*, ed. R. van Rooy and M. Stokhof, 126–131. University of Amsterdam.

Kapur, Shyam. 1992. Computational learning of languages. Doctoral Dissertation, Cornell.

Kobele, Gregory M. 2012. Importing montagovian dynamics into minimalism. In *Logical Aspects of Computational Linguistics*, ed. Denis Béchet and Alexandre Dikovsky, volume 7351 of *Lecture Notes in Computer Science*, 103–118. Berlin: Springer.

Kwiatkowksi, Tom, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1223–1233. Cambridge, MA: Association for Computational Linguistics.

Mäkinen, Erkki. 1992. Remarks on the structural grammatical inference problem for context-free grammars. *Information Processing Letters* 44:125–127.

Morawietz, Frank. 2003. *Two-step approaches to natural language formalisms*, volume 64 of *Studies in Generative Grammar*. Mouton de Gruyter.

Muskens, Reinhard. 2001. Lambda grammars and the syntax-semantics interface. In *Proceedings of the Thirteenth Amsterdam Colloquium*, ed. Robert van Rooij and Martin Stokhof, 150–155. Amsterdam.

Nelson, Deborah G. Kemler, Kathy Hirsh-Pasek, Peter W. Jusczyk, and Kimberly Wright Cassidy. 1989. How the prosodic cues in motherese might assist language learning. *Journal of Child Language* 16:55–68.

Sakakibara, Yasubumi. 1992. Efficient learning of context-free grammars from positive structural examples. *Information and Computation* 97:23–60.

Shieber, Stuart M. 2006. Unifying synchronous tree-adjoining grammars and tree transducers via bimorphisms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 377–384. Trento.

Siskind, Jeffrey Mark. 1996. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition* 61:39–91.

Wagner, Michael. 2005. Prosody and recursion. Doctoral Dissertation, Massachusetts Institute of Technology.

Yokomori, Takashi. 2003. Polynomial-time identification of very simple grammars from positive data. *Theoretical Computer Science* 298:179–206.

Yoshinaka, Ryo. 2011. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science* 412:1821–1831.

Zettlemoyer, Luke S., and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, 658–666. AUAI Press.

Affiliations

John Case
Department of Computer
and Information Sciences
University of Delaware
case@cis.udel.edu

Jeffrey Heinz
Department of Linguistics
University of Delaware
heinz@udel.edu

Gregory M. Kobele
Computation Institute and
Department of Linguistics
University of Chicago
kobele@uchicago.edu