Synthesizing Theories of Human Language with Bayesian Program Induction

John Smith,[1*] Jane Doe,[1] Joe Scientist[2]

[1]Department of Chemistry, University of Wherever,
An Unknown Address, Wherever, ST 00000, USA
[2]Another Unknown Address, Palookaville, ST 99999, USA

[*]To whom correspondence should be addressed; E-mail: jsmith@wherever.edu.

Introduction

An age-old aspiration within artificial intelligence research is to build a machine that helps automate the scientific process by synthesizing theories or models (*1–3*). This aspiration remains largely unrealized: despite a few small-scale successes on toy problems (*4*), practicing scientists do not use machines to generate theories. In contrast, AI has made great strides on problems like machine vision and natural language processing. How can the artificial intelligence community get theory induction off the ground? This is an especially difficult question, because the current mainstream in machine intelligence focuses on qualitatively different classes of problems (e.g., prediction tasks like classification and regression), whereas theory induction requires synthesizing human-understandable causal models of real-world phenomena (*5*), so that human scientists can understand and learn from the AI's outputs.

We propose theory induction research start with theories of human language, and introduce a model of theory induction for a key module of natural language: *morphophonology*, the relationship between word pronunciation and meaning. Acquiring the morphophonology of a language involve solving a basic problem confronting both linguists and children: given a collection of utterances, together with aspects of their meaning, what is the causal relationship between form and meaning?

Our contribution is a model for synthesizing theories of natural language morphophonology. Like linguists, the model starts with a collection of utterances paired with meanings, and then constructs a causal, interpretable model explaining how those meanings gave rise to the utterances, roughly mirroring the process by which theorists construct theories starting from experimental data. In addition to building theories, scientists distill those theories into higher-level kinds of knowledge spanning multiple theories (e.g., energy conservation and Lorentz invariance in physics; XXX in chemistry; universal grammar in linguistics). We argue that this higher-level abstract knowledge is a crucial part of theory induction, imparting prior knowledge and constraints on what would otherwise be an ill-posed inductive reasoning problem. Our model also acquires this higher-level knowledge by jointly inferring theories for multiple languages, and then shifting its inductive bias over theories to more closely match the attested distribution of languages. We evaluate our algorithm on # data sets spanning # languages, automatically finding theories that can model a wide swath of a core component of human language.

Discovering Theories by Synthesizing Programs

We frame our approach as Bayesian Program Learning (BPL: see (*6*)), where the model explains a set of utterance/meaning pairs $\{(u_n, m_n)\}_{n=1}^{N}$ by inferring a theory T, which we model as a program. Formalizing grammars (theories) as generative programs has a long history in linguistics (*7*), for two intuitive reasons: being procedural, programs can capture the causal nature of both grammars and theories; and being highly structured, programs can be interpreted

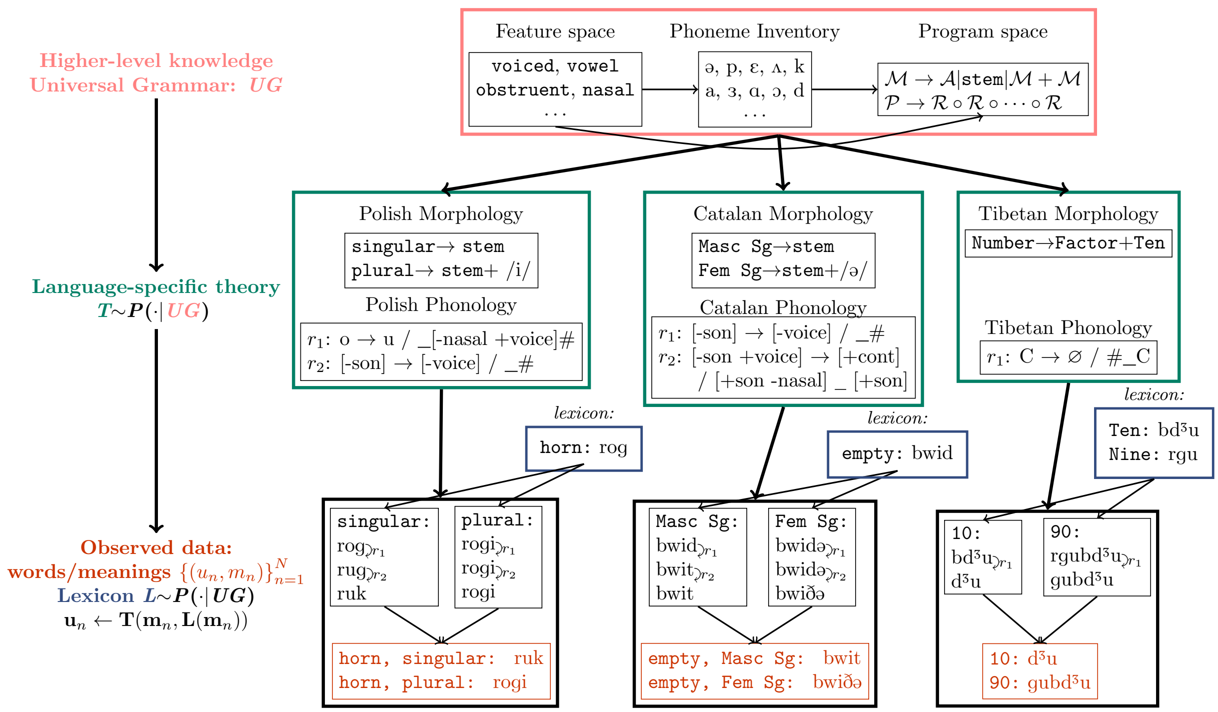

Figure 1: Agent induces theories (teal) for a range of languages, given observed form/meaning pairs (orange). Grammars are expressed as programs drawn from a universal grammar, or programming language (magenta).

by human scientists. Written as a probabilistic inference problem, our model seeks the theory T maximizing $P(\{u_n, m_n\}_{n=1}^{N}|T)P(T|UG)$, where UG is a "universal grammar" encapsulating higher-level abstract knowledge across different languages. In this BPL setting we model UG as a prior distribution over theories (programs), and represent programs as Context-Sensitive Rewrites, a Turing-complete program representation, but restrict the rewrites in such a way as to make them equivalent to finite state transducers — this approach comes from the computational linguistics literature (*8*).

A language's morphophonology can generate infinitely many utterances, depending on what words are in the language, just as theories in other sciences can generate an infinite set of possible observations — in Newtonian mechanics, the theory prescribes how bodies will interact, but does not prescribe the number of bodies or their masses. Thus, for a theory to explain a set of observations, it must introduce additional latent (unobserved) variables on a dataset-by-dataset basis. For the theories of language considered here, this latent variable is the *lexicon*, which is a mapping between the meaning of a stem and its pronunciation. Taking into account the latent lexicon, we refine the theory-induction objective into finding the theory T and lexicon L maximizing $\left[\prod_{n=1}^{N} \mathbb{1}[T \text{ and } L \text{ predict } u_n \text{ for } m_n]\right] P(L)P(T|UG)$. Fig. 1 illustrates this set up for three different languages.

Although this framing captures the problem a BPL theory inductor needs to solve, it offers no guidance on how to solve that problem: the space of all programs (theories) is infinitely large and sharply discontinuous, lacking the local smoothness that enables local optimization algorithms (e.g., gradient descent; MCMC) to succeed. We adopt a strategy based on constraint–based program synthesis, where the optimization problem is translated into a combinatorial constraint satisfaction problem and solved using a SMT solver (*9*). To scale these solvers to large and complex theories we wrap the solver in an outer loop that incrementally introduces new (utterance, meaning) pairs, incrementally modifying the theory to explain new data points

(Supplementary materials).

We apply our model to textbook morphophonology problems taken from (*10*). Each textbook problem requires synthesizing a causal theory of a subset of a language. Fig. 3 compares model outputs against ground-truth textbook solutions. These problem span a range of difficulties and cover a diverse set of natural language phenomena: systems for assigning tone (e.g., in Kerewe, 'to count' is kubala, but 'to count it' is kukíbála, where accents mark high tones), for "harmonizing" vowels (found across many languages, e.g. Kikuria has siika meaning 'close' but seekera meaning 'close for'; Latin has [adeps] meaning 'fat' (nominative)' but [adipis] for the genitive 'fat'), and many other linguistic phenomena like assimilation, epenthesis, and degemination (Fig. 2 and Supplementary materials).

Our theory induction model covers a wider range of languages and phenomena than prior grammar induction algorithms from the computational linguistics literature: prior approaches either recover interpretable causal models (e.g., (*11–13*)) but do not scale to a wide range of challenging and realistic data sets, or abandon theory induction and instead learn opaque probabilistic models (*14*) that may nonetheless predict the data well but which do not help human scientists generate theories. Our improvement here stems from two modeling choices: First, we use a generic computational substrate — context-sensitive rewrites — giving the expressive power needed to explain diverse linguist phenomena. Second, we leverage solver-based program synthesis techniques, borrowing decades of research in the programming languages community that have honed these tools to the point that we can scale them to realistic data sets using rich program representations.

The Role of Higher-Level Knowledge

No theory is built from scratch: instead, researchers borrow concepts and constraints from other successful theories. Linguistics has long acknowledged the importance of constraints and

	Tibetan Count System		Catalan Nouns	
Subset of Data			əkelj∼əkeljə	mal∼malə
			siβil∼siβilə	əskerp∼əskerpə
			ʃop∼ʃopə	sɛk∼sɛkə
			əspɛs∼əspɛsə	gros∼grosə
			baʃ∼baʃə	koʃ∼koʃə
			tot∼totə	brut∼brutə
			pɔk∼pɔkə	prəsis∼prəsizə
			frənses∼frənsezə	gris∼grizə
	1	j̽ig	kəzat∼kəzaðə	bwit∼bwiðə
	4	š̽i	rɔt^{ʃ}∼rɔʒə	bot^{ʃ}∼boʒə
	5	ŋa	orp∼orβə	ljark∼ljarɣə
	9	gu	sek∼seɣə	fəʃuk∼fəʃuɣə
	10	j̽u	grok∼groɣə	puruk∼puruɣə
	11 $(= 10 + 1)$	j̽ugj̽ig	kandit∼kandiðə	frɛt∼frɛðə
	14 $(= 10 + 4)$	j̽ubš̽i	səɣu∼səɣurə	du∼durə
	15 $(= 10 + 5)$	j̽uŋa	səɣəðo∼səɣəðorə	kla∼klarə
	19 $(= 10 + 9)$	j̽urgu	nu∼nuə	kru∼kruə
	40 $(= 4 + 10)$	š̽ibj̽u	flɔñd^{ʒ}u∼flɔñd^{ʒ}ə	dropu∼dropə
	50 $(= 5 + 10)$	ŋabj̽u	əgzaktə∼əgzaktə	əlβi∼əlβinə
	90 $(= 9 + 10)$	gubj̽u	sa∼sanə	pla∼planə
			bo∼bonə	sərɛ∼sərɛnə
			suβlim∼suβlimə	al∼altə
			fɔr∼fɔrtə	kur∼kurtə
			sor∼sorðə	bɛr∼bɛrðə
			san∼santə	kəlɛn∼kəlɛntə
			prufun∼prufundə	fəkun∼fəkundə
			dəsen∼dəsentə	dulen∼dulentə
			əstuðian∼əstuðiantə	blaŋ∼blaŋkə
Theory	C → ∅ / #_C (*Upon encountering a consonant (C), delete it (→ ∅) at the beginning of a word (#_) if followed by another consonant (_C)*)		Morphology: stem∼stem+ə [+coronal +sonorant -lateral] → ∅ / _# [-sonorant] → [-voice] / _# C → k / ŋ_ [+voice -nasal] → [+continuant] / [-nasal]_V V → ∅ / [-sonorant]_V t → ∅ / C_#	

Figure 2: Example morphophonologies

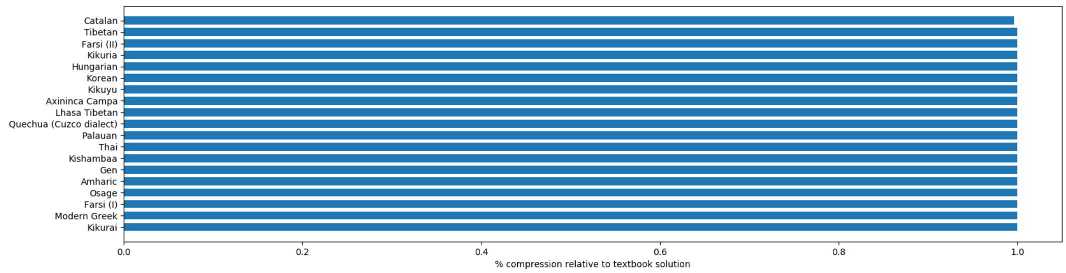

Figure 3: Basically a placeholder – we still need to decide exactly what to put in here and how to measure success.

inductive biases in language learning, and these principles collectively go by the name Universal Grammar: innate for child language learners and empirically probed by linguists. In practice, both children and linguists may have very little data to go off, and thus success often hinges upon having the right kind of high-level knowledge.

Our model represents and acquires this cross-theory knowledge by jointly inferring UG along with the grammars for each data set. Assuming we have D datasets (e.g., from different languages), notated $\left\{\left\{(u_n^d, m_n^d)\right\}_{n=1}^{N_d}\right\}_{d=1}^{D}$, we propose that a theory inductor construct D theories, $\{T_d\}_{d=1}^{D}$, along with a universal grammar UG, maximizing

$$P(UG)\prod_{d=1}^{D}P(T_d|UG)P(\left\{(u_n^d, m_n^d)\right\}_{n=1}^{N_d}|T_d)$$

where $P(UG)$ is a prior distribution over universal grammars. As a first approximation to this goal, we modeled the space of universal grammars using a formalism known as Fragment Grammars (*15*), which work by saving and reusing pieces ('fragments') of the symbolic structure of tree-shaped representations. Here the trees are programs, so by inferring a Fragment Grammar across the theories for the different data sets, the model learns pieces of the higher-level structure found across languages. This automatically learned higher-level knowledge serves two functions: First, it is human interpretable: manually inspecting the contents of the fragment grammar reveals cross-language motifs previously discovered by linguists (e.g., word-final devoicing of obstruents, which occurs in XX% of the world's languages). Second, it aids further theory induction: learning UG allows the model to find better solutions to the textbook problems than if it solves each problem in isolation using a fixed, uninformative UG (Fig. 3).

Discussion

Theory induction is a grand challenge for AI, and our work here captures only small slices of the theory building process. Like our model, human theorists do craft models by examining experimental data, but also propose new theories by unifying existing theoretical frameworks, performing 'thought experiments', and inventing new formalisms. Humans also deploy their theories more richly than our model: proposing new experiments to test theoretical predictions, engineering new tools based on the conclusions of a theory, and distilling higher-level knowledge that goes far beyond what our Fragment-Grammar approximation can represent. Continuing to push theory induction along these many dimensions remains a prime target for future research.

References

1. W. Paul, R. J. Solomonoff (1990).
2. P. Langley, *Scientific discovery: Computational explorations of the creative processes* (MIT Press, 1987).
3. M. Schmidt, H. Lipson, *science* **324**, 81 (2009).
4. P. Langley, G. L. Bradshaw, H. A. Simon, *IJCAI* (1981).
5. J. Pearl, *Causality* (Cambridge university press, 2009).
6. B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, *Science* **350**, 1332 (2015).

7. N. Chomsky, M. Halle, *The sound pattern of English*, Studies in language (Harper & Row, 1968).

8. K. R. Beesley, L. Karttunen .

9. A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, V. Saraswat, *ACM Sigplan Notices* (ACM, 2006), vol. 41, pp. 404–415.

10. D. Odden, *Introducing Phonology* (Cambridge University Press, 2005). Cambridge Books Online.

11. A. Albright, B. Hayes, *Cognition* **90**, 119 (2003).

12. D. Gildea, D. Jurafsky, *Computational Linguistics* **22**, 497 (1996).

13. E. Rasin, I. Berger, R. Katzir (2015).

14. R. Cotterell, N. Peng, J. Eisner, *Transactions of the Association for Computational Linguistics* **3**, 433 (2015).

15. T. J. O'Donnell, *Productivity and Reuse in Language: A Theory of Linguistic Computation and Storage* (The MIT Press, 2015).