



Grammar Induction: An Invitation for Formal Language Theorists

HENNING FERNAU

WSI für Informatik, Universität Tübingen, Germany
Sand 13, D-72076, Tübingen, Germany
E-mail:fernau@informatik.uni-tuebingen.de

COLIN DE LA HIGUERA

EURISE, Université de Saint-Etienne, France
23, rue du Docteur Paul Michelon 42023, Saint-Etienne, France
E-mail:cdlh@univ-st-etienne.fr

1 Introduction

Why should Formal Language specialists get interested and involved in the area of Grammar Induction (GI), likewise known as Grammatical Inference? This is the question we try to answer in this paper. To keep this paper short, we mostly refrain from giving technical details but rather cite according references. So, this is rather a kind of hopefully inspiring (and by no means complete!) annotated bibliography than a technical paper. However, we hope to present an inviting and inspiring panoramic view on the scene.

We believe that there are indeed a number of good reasons, the most obvious being that Grammar Induction, as well as Formal Language Theory, deals with formalisms describing languages, *i.e.*, grammars, automata, expressions, *etc.* This already implies a lot of interesting issues:

- GI makes use of formal language methodologies for constructing learning algorithms and for reasoning about them.
- GI tries to mathematically describe the classes of languages that can be learned by a particular learning algorithm. Usually, little is known about these language families regarding classical formal language questions.
- In GI, methods known within Formal Language Theory have been successfully used to find and describe learnable language classes which contain non-regular languages. Yet, this is a still largely open area in the sense that perhaps only the best known language theoretic properties have been used so far to achieve learning results.

- GI offers a new motivation for exploring the descriptive complexity of language description formalisms, namely by the fact that the time complexity of a learning algorithm typically depends on some measures of descriptive complexity in relation with the language class in question. Sometimes, also the definition of some class of learnable languages depends on some measure of descriptive complexity. Those measures of descriptive complexity could also be “new” ones which deserve further studies from the viewpoint of Formal Language Theory.
- For various reasons, learnability results in GI often rely on the existence of suitable normal forms for the language classes in question. Formal Language Theorists will find it interesting and stimulating to further study these normal forms, or develop similar or alternative normal forms.
- In GI papers, one can sometimes find explicit formulations of open problems which can be read as formal language problems rather than as problems in learning theory.

We will give examples for each of the above-mentioned relations in what follows. However, let us conclude this introduction by mentioning the main motivation for GI: its wide applicability. Here, it actually touches many other areas which are usually not considered to be closely related to formal languages:

- Machine Learning: in actual fact, GI can be seen as a sub-area of Machine Learning. Machine Learning tries to derive concepts from examples in order to solve tasks such as clustering or classification.
- Inductive Logic Programming (ILP) can be also viewed as a sub-area of Machine Learning, the task being to automatically generate logic programs. GI techniques have been used to generate specific forms of logic programs, and ILP techniques have been used for GI purposes, see [8, 11].
- In a certain sense, ILP is also a special case of a technique called Programming By Example. Here, further applications of GI methodologies can be found [35].
- Within the area of Pattern Recognition, GI has found its place in the so-called *structural and syntactic* approach [14]. This is also testified by the fact that in nearly every main conference in that field (which is SSPR), you can find GI papers.
- Techniques known from GI have also been applied in the area of Data Compression [39, 42, 49], although in that case admittedly one of the basic features of GI, namely generalization, is completely missing out. Yet, the way in which “features” are extracted from a sample text looks very similar in this syntactic data compression approach.
- GI has been used within natural language processing systems in a variety of tasks where the construction of a language model is necessary. In speech recognition the inferred finite state automata (usually probabilistic) can have several

thousand states. In machine translation tasks [3], GI techniques have to infer transducers with over 500,000 states and 1,500,000 transitions.

2 Common objects and techniques

Formal Language Theory and Grammar Induction share a lot of areas of interest. We can present the main intersection of areas in two orthogonal ways:

- There are classes of languages and, related to this, classes of grammars, automata, expressions and so forth which are of interest (best from an application point of view).
- There are properties of language and grammar families, as well as questions on language and grammar families, for which people from Grammar Induction would like to see answers.

Let us discuss these two aspects more systematically in the following.

2.1 Common objects

An important part of the research in Grammar Induction has concentrated on the first stage of the Chomsky hierarchy: the regular (string) languages. And as even this class of languages has proved too strong for learning to be achieved in certain paradigms¹ sub-classes have been described and studied. Since there are different ways of describing regular languages, different algorithmic approaches are possible. Let us now comment upon some of the best known classes of languages/generating devices and consider them under the GI point of view. Our goal here is to convince the language theoretician that whatever his favorite object is, there is room for its study in GI.

- Most learning algorithms deal with deterministic finite automata (DFA). The advantage is that here we have a well-defined concept of a normal form in the notion of the minimal DFA. However, since nondeterministic finite automata (or regular grammars) may give exponentially more succinct definitions of languages, this form of description of regular languages offers advantages. Conversely, regular expressions are the target structure for many applications (*e.g.*, XML DTD inference, see [2, 10, 12, 25]). Also there, appropriate normal forms are lacking, also for special forms of regular languages. Formal Language specialists might wish start reading about regular language inference in [4, 5]. One research direction might be to investigate if there are algebraic or logics reasons for the learnability in these cases, keeping in mind similar results, *e.g.*, on tree languages [43, 44].

¹Without entering details, to be able to say that “learning has been achieved” you can rely on a variety of different definitions, depending on whether you want exact learning or approximate learning, on what you are learning from (examples and/or counter-examples), on if you are allowed some additional help and knowledge, and how you count tractability.

- Only few attempts have been undertaken to generalize regular string language learners to the learning of other classes of structures definable by finite automata, as trees [6, 22, 27, 29, 41], graphs, infinite strings (discussed below), power series, pictures [13, 46], traces [1, 23, 24], *etc.* The mentioned references (besides the tree case) are rather exhaustive, as we fear. The importance of these issues is underlined by the fact that recently a special workshop on learning tree languages has been organized [38].
- Automata with outputs, likewise known as transducers [9], deserve further studies from the viewpoint of GI. As mentioned in the introduction, they are of practical use in Machine Translation tasks, where one goal is to construct from a corpus of translation pairs a transducer. Algorithms exist but many open problems subsist, and a better knowledge of these objects would be of real use to the field.
- Barely touched is the area of learning *context-free* languages, be it based on grammars or automata. Here, as the problem is generally believed to be hard, it could be also interesting to consider reasonable subclasses of context-free languages. Actually, the first attempts in this direction were based on transformations of regular inference techniques to the context-free case based on the notion of control languages [50]. A notable exception is here the works of Kanazawa and of Yokomori [28, 52, 56] whose algorithms are not relying on “simple” translation of regular language learners. Works on counter-automata also deserve to be mentioned here [7, 19, 20], as well as on linear languages [30]. Structural properties are underlined by considering pure context-free languages [31] or trees and other forms of structural information, see [43, 44, 47, 48].
 Another approach is undertaken by Starkie (ongoing work) in his PhD thesis. There has also been a number of practical heuristics that have attempted to solve some learning questions related with context-free grammar learning [32, 45].
- In the same line, although finite state automata have been a favored subject for researchers in Grammar Induction, no work (to our knowledge) has taken place on push-down automata learning, except from the counter-automata papers already mentioned.
- To our knowledge, nearly completely untouched is the area of learning classes containing *non-context-free* languages, Lindenmayer systems and regulated grammars being some sort of exception [21, 36, 51, 55, 54].
- There has even been a small number of studies on learning Büchi automata, in order to deal with reactive systems [26, 37, 46].
- Stochastic (or probabilistic) automata, grammars or transducers are of increasing importance due to their capacity of dealing with noisy or ambiguous data [40]. They intervene in areas where language models are needed. It should be added that they generally involve very large alphabets, each symbol corresponding to one word in the dictionary (thus the size of this item should

not be considered as a constant when dealing with complexity issues). Yet little is known about the language theoretical properties of these objects: Can equivalence be decided? What distances (between strings, between strings and stochastic automata, between distributions/automata) can we compute? A crucial question intervenes once the model is learned and used to put probabilities on sentences: For a number of reasons null probabilities are harmful and need to be avoided; This requires smoothing, for which good techniques, adapted to stochastic automata, have yet to be invented.

- Cellular automata have been used to discretely model natural phenomena, see [15]. Is it possible to “explain” natural phenomena by trying to automatically induce suitable cellular automata models? No research has been undertaken in this direction to our knowledge.

Orthogonally, for the development of learning algorithms, Grammar Induction mainly uses the following techniques in each of the cases sketched above. Actually, the exact list of interesting techniques may vary with the concrete learning scenario (a topic which we are deliberately keeping out of our discussion here).

2.2 Common properties

Learnability of language families \mathcal{L} which are of interest to Grammar Induction (for various reasons) will often depend on some specific properties that are also of independent interest:

- The *equivalence problem* for the descriptive devices characterizing \mathcal{L} should be solvable in polynomial time. It is known that if this is not true, the intended class is not even teachable, *i.e.* the quantity of information that a learning algorithm has to be fed in order to learn can be more than polynomial.
- For each language from \mathcal{L} , a *normal form* should be available. Indeed the learner is both learning a language and a grammar. In certain cases it is essential (for reasons depending on the intended application) that some specific form is learned, but in other cases even a black box is good enough, provided it can correctly classify new strings. To illustrate this point take the case of context-free grammars. It is generally admitted that learning these is a hard problem, even if the data we are learning from is bracketed or structured. But on the other hand [43], each context-free language admits a grammar in a special normal form (called *reversible*) for which learning is possible from positive bracketed information only. Mostly, normal forms are available for “deterministic devices.” Formal Language Theorists might find it interesting to propose alternative language families plus normal forms. The study of residual automata [17, 18] could be an inspiring starting point.
- The use of standard representations for the language classes under study is also a problem: even if automata and grammars are appealing, they have serious drawbacks when it comes to learning as numerous studies have shown. For

example, regular expressions may be better readable for humans than finite automata. Alternative representations of languages, generative or recognizing devices should be studied. Unfortunately, the use of non-standard representations in learning theory is scarcely developed: [53] uses context-free expressions, [17] use special forms of nondeterministic finite automata.

- *Parsing* should be possible in linear time if possible. Indeed, during the learning process, strings from the learning set are parsed many times against the candidate grammars. For this, sub-classes of context-free languages should be proposed.²
- Many learning algorithms can be seen as realizing *operators* on language describing devices. For example, state merging is one favorite technique for implementing generalization, since an automaton after merging states will still parse the same sentences as before, together with possible additional ones. Here, operators which preserve inclusion are of specific interest.
- Implementation issues are strong: the fact that in language modeling or computational biology tasks the size of the data, and thus of the grammars or the automata to be manipulated is huge, means that standard *textbook* representations and corresponding algorithms will not be good enough.

3 Two open problems

In this section, we describe only two *concrete* formal language problems that show up either explicitly or implicitly in recent papers from GI. Many more such examples can be found.

Garofalakis *et al.* propose a specific scheme to describe words given a regular expression as a “theory,” see [25] and also [16]. This is needed within their minimum descriptive principle approach to Grammar Induction, where the number of bits needed to encode words by a given regular expression is of interest. The choice of the encoding obviously influences the measure and thereby the choice of the “best” regular expression. Formal Language Theory might wish to systematically study this influence of the encoding scheme. This kind of question is also of interest to applications of information theory and Kolmogorov complexity as outlined in [34].

Lange and Nessel described in [33] so-called decision lists over regular patterns as an alternative to decision trees. They mention that “it is of interest to relate the size of a given decision tree to the shortest decision list that accepts the same regular language.” Especially, the question whether there exists an inherent exponential blow-up when transforming trees into lists is open.

²This issue is also discussed in Brad Starkie’s PhD (in preparation).

4 Conclusion

We have presented here a number of reasons for which Language Theoreticians can find interesting problems in Grammar Induction: the objects that are been manipulated are the same, the properties on which the tractability of learning will depend are properties they are used to study.

The open problems we have proposed are just a few of those that are known in the community. Most of them (as usually is the case when dealing with strongly applicative fields) need yet to be formalized.

There is a final argument towards using theory skills in this field: the problems are real, the applications are numerous. It is possible to see how useful one's theoretic study is on true, motivating and hard tasks.

So, how can you—as a Formal Language Specialist—become involved in GI? As a starting point, simply take one of the papers in the reference list, start reading and possibly contact the authors. Or simply continue reading in this special issue. We are sure you will have fun, and this is not the worst motivation for doing research, isn't it?

References

- [1] N. Abe, “Characterizing PAC-learnability of semilinear sets”, *Information and Computation*, 116, pp. 81-102, 1995.
- [2] H. Ahonen, H. Mannila, E. Nikunen, “Forming grammars for structured documents: an application of grammatical inference”, R. C. Carrasco and J. Oncina, eds., *Grammatical Inference and Applications ICGI, LNCS/LNAI*, 862, pp. 153-167, Springer, 1994.
- [3] J. C. Amengual, J. M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V. M. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, J. M. Vilar, “The EuTrans-I speech translation system”, *Machine Translation*, 15, pp. 75-103, 2001.
- [4] D. Angluin, “Inference of reversible languages”, *Journal of the ACM*, 29(3), pp. 741-765, 1982.
- [5] D. Angluin, “Learning regular sets from queries and counterexamples”, *Information and Computation*, 75, pp. 87-106, 1987.
- [6] H. Arimura, H. Ishizaka, T. Shinohara, “Learning unions of tree patterns using queries”, *Theoretical Computer Science*, 185, pp. 47-62, 1997.
- [7] P. Berman, R. Roos, “Learning one-counter languages in polynomial time”, *Proc. 28th Symposium on Foundations of Computer Science FOCS*, pp. 61-67, IEEE, 1987.

- [8] M. Bernard, C. de la Higuera, “GIFT: Grammatical Inference For Terms”, *Conférence d’Apprentissage, Palaiseau*, May 1999, English version: Late breaking paper of International Conference on Inductive Logic Programming; French journal version: Apprentissage de programmes logiques par inférence grammaticale. *Revue d’Intelligence Artificielle*, 14, pp. 375-396, 2001.
- [9] J. Berstel, *Transductions and Context-Free Languages*, LAMM, 38, Stuttgart: Teubner, 1979.
- [10] J. Berstel, L. Boasson, “Formal properties of XML grammars and languages”, *Acta Informatica*, 38, pp. 649-671, 2002.
- [11] H. Boström, “Theory-guided induction of logic programs by inference of regular languages”, *13th International Conference on Machine Learning ICML*, pp. 46-53, Morgan Kaufmann, 1996.
- [12] A. Brazma, “Learning of regular expressions by pattern matching”, P. M. B. Vitányi, ed., *Computational Learning Theory, Second European Conference, EuroCOLT, LNCS/LNAI*, 904, pp. 392-403, Springer, 1995.
- [13] A. Brüggemann-Klein, P. Fischer, T. Ottmann, “Learning picture sets from examples”, *Results and trends in theoretical computer science (Graz, 1994)*, LNCS, 812, pp. 34-43, Springer, 1994.
- [14] H. Bunke, A. Sanfeliu (eds.), *Syntactic and Structural Pattern Recognition, Theory and Applications*, 7, Series in Computer Science. World Scientific, 1990.
- [15] B. Chopard, M. Droz, *Cellular Automata Modeling of Physical Systems*, Cambridge University Press, 1998.
- [16] D. Conklin, I.H. Witten, “Complexity-based induction”, *Machine Learning*, 16, pp. 203-225, 1994.
- [17] F. Denis, A. Lemay, A. Terlutte, “Learning regular languages using RFSA”, N. Abe, R. Khardon, T. Zeugmann, eds., *Algorithmic Learning Theory ALT, LNCS/LNAI*, 2225, pp. 348-363, Springer, 2001.
- [18] F. Denis, A. Lemay, A. Terlutte, “Residual finite state automata”, *18th Annual Symposium on Theoretical Aspects of Computer Science STACS, LNCS* 2010, pp. 144-157, Springer, 2001.
- [19] A. F. Fahmy, R. Roos, “Efficient learning of real time one-counter automata”, K. P. Jantke, T. Shinohara, Th. Zeugmann, eds., *Proceedings of the 6th International Workshop on Algorithmic Learning Theory ALT, LNCS/LNAI*, 997, pp. 25-40, Springer, 1995.
- [20] A. F. Fahmy, R. S. Roos, “Efficient learning of real time two-counter automata”, S. Arikawa, A. K. Sharma, eds., *Proceedings of the 7th International Workshop on Algorithmic Learning Theory ALT, LNCS/LNAI*, 1160, pp. 113-126, Springer, 1996.

- [21] H. Fernau, “Even linear simple matrix languages: formal language properties and grammatical inference”, *Theoretical Computer Science*, 289, pp. 425-489, 2002.
- [22] H. Fernau, “Learning tree languages from text”, J. Kivinen, R. H. Sloan, eds., *Computational Learning Theory COLT, LNCS/LNAI*, 2375, pp. 153-168, Springer, 2002.
- [23] C. Ferretti, G. Mauri, “Identifying regular languages over partially-commutative monoids”, *Algorithmic Learning Theory ALT, LNCS/LNAI*, 872, pp. 282-289, Springer, 1994.
- [24] C. Ferretti, G. Mauri, “Identifying unrecognizable regular languages by queries” *European Conference on Machine Learning ECML, LNCS/LNAI*, 874, pp. 355-358, Springer, 1994.
- [25] M. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, K. Shim, “XTRACT: Learning document type descriptors from XML document collections”, *Data Mining and Knowledge Discovery*, 7, pp. 23-56, 2003.
- [26] C. de la Higuera, J.-C. Janodet. “Inference of ω -languages from prefixes”, *Theoretical Computer Science*, 313, pp. 296-312, 2004.
- [27] H. Ishizaka, H. Arimura, T. Shinohara, “Finding tree patterns consistent with positive and negative examples using queries”, S. Arikawa and K. P. Jantke, eds., *Proceedings of the 4th International Workshop on Analogical and Inductive Inference and 5th International Workshop on Algorithmic Learning Theory, LNCS/LNAI*, 872, pp. 317-332, Springer, 1994.
- [28] M. Kanazawa, *Learnable Classes of Categorical Grammars*, Cambridge University Press, New York, NY, 1998.
- [29] T. Knuutila, M. Steinby, “The inference of tree languages from finite samples: an algebraic approach”, *Theoretical Computer Science*, 129, pp. 337-367, 1994.
- [30] T. Koshiba, E. Mäkinen, Y. Takada, “Learning deterministic even linear languages from positive examples”, *Theoretical Computer Science*, 185(1), pp. 63-79, 1997.
- [31] T. Koshiba, E. Mäkinen, Y. Takada, “Inferring pure context-free languages from positive data”, *Acta Cybernetica*, 14, pp. 469-477, 2000.
- [32] P. Langley, S. Stromsten. “Learning context-free grammars with a simplicity bias”, *European Conference on Machine Learning ECML, LNCS/LNAI*, 1810, pp. 220-228, Springer, 2000.
- [33] S. Lange, J. Nessel, “Decision lists over regular patterns”, *Theoretical Computer Science*, 298, pp. 71-87, 2003.

- [34] M. Li, X. Chen, X. Li, B. Ma, P. M. B. Vitányi, “The similarity metric” *Symposium on Discrete Algorithms SODA*, pp. 863-872, 2003.
- [35] H. Lieberman ed., *Your Wish is My Command: Giving Users the Power to Instruct Their Software*. Morgan Kaufmann, 2001.
- [36] H. R. Lu, K. S. Fu, “Inferability of context-free programmed grammars”, *International Journal of Computer and Information Sciences*, 13, pp. 33-58, 1984.
- [37] O. Maler, A. Pnueli, “On the learnability of infinitary regular sets”, *Conference on Learning Theory COLT*, pp. 128-136, Morgan Kaufmann, 1991.
- [38] “Mostrare Workshop on Learning Tree Languages”, <http://www.grappa.univ-lille3.fr/twiki/bin/view/Public/WorkshopDec2003>
- [39] C. G. Nevill-Manning, I. H. Witten, D. R. Olsen, Jr., “Compressing semi-structured text using hierarchical phrase identification”, J. A. Storer and M. Cohn, eds., *IEEE Data Compression Conference DCC*, pp. 63-72, IEEE Computer Society Press, 1996.
- [40] H. Ney, “Stochastic grammars and pattern recognition”, P. Laface and R. D. Mori, eds., *Proceedings of the NATO Advanced Study Institute*, pp. 313-344, Springer, 1992.
- [41] J. R. Rico-Juan, J. Calera-Rubio, R. C. Carrasco, “Probabilistic k -testable tree languages”, A. L. Oliveira, ed., *Grammatical Inference: Algorithms and Applications, 5th International Colloquium ICGI, LNCS/LNAI*, 1891, pp. 221-228, Springer, 2000.
- [42] W. Rytter, “Application of Lempel-Ziv factorization to the approximation of grammar-based compression”, *Theoretical Computer Science*, 302, pp. 211-222, 2003.
- [43] Y. Sakakibara, “Learning context-free grammars from structural data in polynomial time”, *Theoretical Computer Science*, 76, pp. 223-242, 1990.
- [44] Y. Sakakibara, “Efficient learning of context-free grammars from positive structural examples”, *Information and Computation*, 97, pp. 23-60, 1992.
- [45] Y. Sakakibara, M. Kondo, “GA-based learning of context-free grammars using tabular representations”, *16th International Conference on Machine Learning ICML*, pp. 354-360, Morgan Kaufmann, 1999.
- [46] A. Saoudi, T. Yokomori, “Learning local and recognizable ω -languages and monadic logic programs”, *European Conference on Learning Theory EURO-COLT*, pp. 157-169, Oxford University Press, 1994.
- [47] J. M. Sempere, A. Fos, “Learning linear grammars from structural information”, L. Miclet and C. de la Higuera, eds., *Proceedings of the Third International Colloquium on Grammatical Inference ICGI: Learning Syntax from Sentences, LNCS/LNAI*, 1147, pp. 126-133, Springer, 1996.

- [48] J. M. Sempere, G. Nagaraja, “Learning a subclass of linear languages from positive structural information”, V. Honavar and G. Slutski, eds., *Proceedings of the Fourth International Colloquium on Grammatical Inference ICGI, LNCS/LNAI*, 1433, pp. 162-174, Springer, 1998.
- [49] A. Shelat, “Evaluating grammar-based data compression algorithms”, PhD Thesis, 2001.
- [50] Y. Takada, “Grammatical inference of even linear languages based on control sets”, *Information Processing Letters*, 28, pp. 193-199, 1988.
- [51] Y. Takada, “A hierarchy of language families learnable by regular language learning”, *Information and Computation*, 123, pp. 138-145, 1995.
- [52] N. Tanida, T. Yokomori, “Inductive inference of monogenic pure context-free languages”, S. Arikawa and K. P. Jantke, eds., *Algorithmic Learning Theory ALT, LNCS/LNAI*, 872, pp. 560-573, Springer, 1994.
- [53] T. Yokomori, “Inductive inference of context-free languages based on context-free expressions”, *International Journal of Computer Mathematics*, 24, pp. 115-140, 1988.
- [54] T. Yokomori, “Inductive inference of 0L languages”, *Lindenmayer Systems*, pp. 115–132, Springer, 1992.
- [55] T. Yokomori, “On learning systolic languages”, K. P. Jantke, S. Doshita, K. Furukawa, T. Nishida, eds., *Proceedings of the 3rd Workshop on Algorithmic Learning Theory ALT, LNCS/LNAI*, 743, pp. 41-52, Springer, 1992.
- [56] T. Yokomori, “Polynomial-time identification of very simple grammars from positive data”, *Theoretical Computer Science*, 298, pp. 179-206, 2003.