A Recap of Early Work on Theory and Knowledge Refinement

Raymond J. Mooney[a], Jude W. Shavlik[b]

[a]Dept. of Computer Science, University of Texas at Austin, 2317 Speedway, Stop D9500 Austin, Texas 78712-1757
[b]Dept. of Computer Science, University of Wisconsin – Madison

Abstract
A variety of research on theory and knowledge refinement that integrated knowledge engineering and machine learning was conducted in the 1990's. This work developed a variety of techniques for taking engineered knowledge in the form of propositional or first-order logical rule bases and revising them to fit empirical data using symbolic, probabilistic, and/or neural-network learning methods. We review this work to provide historical context for expanding these techniques to integrate modern knowledge engineering and machine learning methods.

Keywords
Theory Refinement, Knowledge Refinement, Knowledge-Based Neural Networks, Explainable AI

1. Introduction

Combining machine learning (ML) and knowledge engineering (KE) is not a new topic. In the 1990's, there was community of researchers (including the authors) who developed a variety of techniques for taking human-engineered knowledge in the form of propositional or first-order logical rule bases and revising them to fit empirical data using symbolic, probabilistic, and/or neural-network learning methods. Although this work never achieved the substantial lasting impact of some other research of this era, and may not be familiar to many current researchers in machine learning and knowledge engineering, we believe it explored a range of interesting algorithmic and experimental ideas and provides important historical context for any new work on combining ML and KE. It also clearly demonstrated through a range of experimental evaluations in a number of domains, that combining human-engineered and empirically induced knowledge could improve the accuracy of a final intelligent system.

The primary goal of this community was to gain better accuracy than either (a) solely using engineered knowledge for the task at hand in a non-learning manner (recall the 1990's were the tail end of the "expert systems" era) or (b) solely learning a system from labeled training examples, where the only role of domain knowledge was choosing good 'features' with which to represent examples.

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, March 22-24, 2021.
mooney@cs.utexas.edu (R.J. Mooney); shavlik@cs.wisc.edu (J.W. Shavlik)
https://www.cs.utexas.edu/~mooney/ (R.J. Mooney); http://pages.cs.wisc.edu/~shavlik/ (J.W. Shavlik)

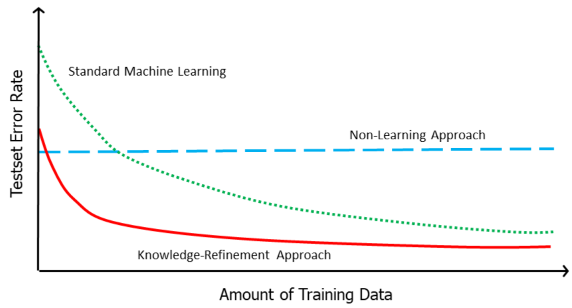

Figure 1: Notional learning curves illustrating the value of knowledge refinement.

Figure 1 illustrates this idea. The X axis is the amount of training data and the Y axis is the system's error rate on novel examples not used during training. The use of domain knowledge provides an error reduction, especially when the number of training examples is small. The cross-over points in the figure show where learning approaches start to exceed non-learning ones, and are indicative of the central role of machine learning in today's AI. In Figure 1, the curve for the non-learning approach is flat since it ignores training examples (though presumably humans did use a few examples to create and represent the domain knowledge). The knowledge-refinement approach starts at a higher error rate to reflect the fact the knowledge-refinement approach may use a more limited knowledge representation than the non-learning approach.

This paper briefly reviews this early work, covering methods that primarily employed logical, probabilistic, and neural-network methods. We believe many of the ideas in this work could be updated and modernized to develop new, effective methods for combining ML and KE. Therefore, we hope that reviewing this prior work serves a valuable resource for current researchers interested in this area.

2. Logical Theory and Knowledge Refinement

A number of systems have integrated KE and ML by using learning methods to revise a human-engineered logical knowledge base (KB) in order to make it fit empirical data. Most of this work employed a rule-based KB, either in propositional logic or in the form of first-order Horn clauses (i.e. Prolog programs). Engineered knowledge was refined by removing conditions from rules to generalize them, adding learned conditions to specialize them, removing rules, and/or learning new rules from constructed subsets of data.

Early work on this thread was by Ginsberg *et al.* [1], which was followed up by a system called RTLS [2]. RTLS flattened a propositional rule base into disjunctive normal form (DNF), revised this DNF to fit labeled training data using learning methods, and then translated the changes back to the multi-level rules. EITHER [3, 4] was a more comprehensive revision system for propositional rule bases that combined deductive, abductive, and inductive reasoning. It used logical abduction to identify "holes" in a theory and used inductive rule learning methods to repair them. NEITHER [5, 6] was a followup to EITHER that focused on revising KBs containing “soft matching” M-of-N rules, which are satisfied as long as at least M of its N antecedents are true. Other systems that refined propositional theories are DUCTOR [7] and the work of Feldman *et al.* [8].

A more challenging problem is revising first-order Horn-clause logical theories that include relations, variables, and quantifiers. Work in this area was tightly connected to early work in Inductive Logic Programming (ILP) [9]. MIS (Model Inference System) [10] was an early system that tried to debug Prolog programs by interactively querying a human oracle. FOCL (First Order Combined Learner) and its derivatives [11, 12] used a first-order theory to bias inductive learning, but required user interaction to determine where to actually make theory revisions. FORTE (First Order Revision of Theories from Examples) [13, 14] was a fully automated system for revising relational KBs and was also used to automatically debug simple Prolog programs developed by students learning logic programming. Other ILP systems that incorporated or revised background knowledge are MLSMART [15], GOLEM [16], GRENDEL [17], and Rx [18].

3. Probabilistic Knowledge Refinement

Logical domain theories in AI have long been criticized for their inability to handle uncertainty in reasoning, which is critical in most real-world applications. Adding *certainty factors* to rules was an early approach to dealing with uncertainty in knowledge-based systems [19]. RAPTURE (Revising Approximate Probabilistic Theories Using Repositories of Examples) [20] was a theory refinement system that was designed to revise certainty-factor rule bases. It adapted backpropagation methods designed for neural-networks [21] to automatically revise the certainty factor parameters through gradient descent. It also uses machine learning methods adapted from decision-tree learning [22] to add features and revise the structure of the rule base. Fu [23] also used backpropagation to revise certainty factors, but his approach was unable to revise the rule-base structure.

Ad hoc methods like certainty factors were criticized for not adhering to the well-founded principles of probability theory and Bayesian reasoning. Consequently, techniques based more firmly in probability theory, such as Bayesian networks [24], came to dominate knowledge-based systems that supported uncertain reasoning. BANNER [25, 26] was a knowledge refinement system designed to revise manually-engineered Bayesian networks to fit empirical data. Like RAPTURE, it uses a variant of backpropagation to adjust the conditional probability parameters of the Bayes-net to fit labeled training data for a classification task. Then, as needed, it alters the structure of the network using learning techniques to add new dependency edges as well as new hidden variables. It focused on networks that used *noisy-or* and *noisy-and* nodes that are probabilistic variants of these logical operators. This allowed it to map an initial purely-logical theory to a Bayes-net and then refine it to fit empirical data. There was also other work on revising Bayes nets [27, 28], but it was unable to add new hidden variables.

4. Knowledge-Based Neural Networks

Starting in the late 1980’s neural networks had a rebirth after their near demise in the 1960’s, due to the ability to train networks with ’hidden units’ [21] lying between the input and output

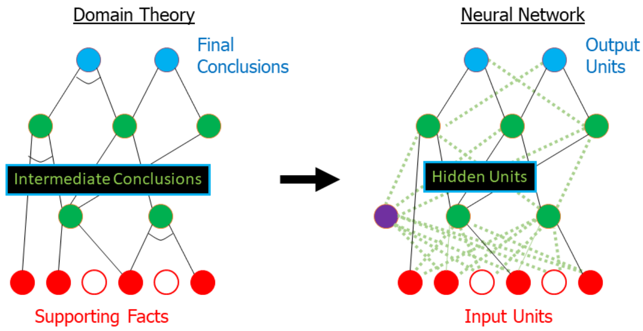

Figure 2: The rule set to neural network mapping of knowledge-based neural networks.

units. Towell and Shavlik [29] recognized the analogy between the dependency graph of a rule set (i.e., a graph where the outputs from some rules serve as the inputs to others) and a neural network. Their KBANN (Knowledge-Based Artificial Neural Networks) algorithm mapped propositional rule sets into neural networks, setting weights so that initially the neural network produced outputs near 1 when the rule set returned true and near 0 when the rule set returned false. Figure 2 illustrates the correspondences. An early test on a gene-finding testbed lead to a halving of the error rate [30].

A disjunctive rule set representing some domain theory is on the left, drawn using the common AND/OR notation. On the right is a corresponding neural network. There are a few aspects of this figure worth noting.

1. Not all the facts about the domain at hand may be referenced by the rule set (these are the open red circles on the bottom), but an important role for them might be discovered during training.
2. Some rule preconditions might be missing, as illustrated by the dashed lines in the neural network; initially these links are given weights near zero, but backpropagation might increase them if doing so helps reduce error. Similarly, some rule antecedents might be pushed toward zero by backpropagation, essentially removing them (backpropagation also converts the Boolean algebra of rule sets into weighted sums that are input to the non-linear sigmoid function).
3. The rule set might be missing some rules, illustrated by the leftmost (purple) hidden unit in the figure, so it can be beneficial to include some initially zero-weighted hidden units [31, 32].
4. A complex rule set can lead to a *deep* neural network, deeper than the traditional one-hidden-layer network of the mid-1980's and early 1990's. A KBANN-followup paper by Towell and Shavlik [33] specifically addressed the use of symbolic knowledge to deal with the challenges of training deep neural networks.

Because neural networks learn in an incremental manner (i.e., one batch of examples at a time), it is possible to consider adding more domain rules in the midst of a long training run [34] (e.g., in the middle of Figure 2's X axis). For example, observing the mistakes made by a robotic reinforcement learner might cause a human teacher to devise some new rules. (This ability to accept rules after learning has begun means one should not think of theory refinement as only using *prior* knowledge.)

Since backpropagation changes the simple logical semantics of propositional rule sets into less intuitive weighted sums, some early researchers [35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47] investigated the task of *rule extraction* where one converts a trained neural network into a more human-readable representations, such as set of rules or a small decision tree. These approaches are generally also applicable to neural networks trained without the use of domain knowledge, and some even can be applied to alternate complex learned representations, such as a forest of decision trees (e.g., [45]). The task of rule extraction closely relates to the current extensive interest in explainable AI, especially in the context of deep neural networks.

Additional early work on refining and/or exploiting symbolic knowledge by neural networks includes Gallant [35], Fu [48], Shavlik and Towell [49], Berenji [50], Frasconi et al. [51], Omlin and Giles [52], Roscheisen et al. [53], Mahoney and Mooney [20], Tresp et al. [54], and Thrun and Mitchell [55] (these citations are sorted by publication year). See Shavlik [56] for a review written in 1992.

5. Application Areas

Theory/knowledge refinement has been applied to a variety of application areas demonstrating that combining human-engineered knowledge and machine learning could develop more accurate intelligent systems than using either approach alone.

Some classic domains in AI and machine learning such as soybean disease diagnosis [57] and human infectious disease diagnosis as performed by the famous MYCIN expert-system [58] were studied. Both EITHER [4] and RAPTURE [20] demonstrated improved performance on soybean diagnosis, and RAPTURE also demonstrated improved performance on MYCIN data.

Another interesting application of logical theory refinement involved improving student modelling for intelligent tutoring systems using a system called ASSERT [59, 60].[1] Using a KB encoding correct knowledge needed to perform a task and examples of a student's behavior for this task, ASSERT modeled student errors by generating refinements to the correct knowledge base sufficient to account for the student's behavior. ASSERT was evaluated using 100 students tested on a classification task covering concepts from an introductory course on C++ programming. Students who received feedback based on student models generated by ASSERT performed significantly better on a post test than students who received just basic instruction.

Applications of knowledge-based neural networks include gene finding [30, 61], protein folding [62], language learning [52, 63], robot training: [34], non-linear control [50, 64], manufacturing [53], computer vision [65], and information extraction [66].

6. Conclusions

This paper has reviewed work from the 1990's on combining knowledge-engineering and machine learning to revise KBs to fit empirical data. This earlier work used a variety of knowledge representation formalisms as well as a range of logical, probabilistic, and neural-network learning methods. It was also evaluated on a range of applications, experimentally demonstrating its ability to achieve improved performance by effectively combining KE and ML. We believe many of the ideas embodied in this early work could be updated to utilize the latest developments in KE and ML, and hope they provide inspiration and guidance in continuing work on combining KE and ML to improve the capabilities and performance of AI systems.

References

[1] A. Ginsberg, S. M. Weiss, P. Politakis, Automatic knowledge based refinement for classification systems, Artificial Intelligence 35 (1988) 197–226.

[2] A. Ginsberg, Theory reduction, theory revision, and retranslation, in: Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), Detroit, MI, 1990, pp. 777–782.

[1] This work was awarded a AAAI Best Paper Award in 1996.

[3] D. Ourston, R. Mooney, Changing the rules: A comprehensive approach to theory refinement, in: Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), Boston, MA, 1990, pp. 815–820.
[4] D. Ourston, R. J. Mooney, Theory refinement combining analytical and empirical methods, Artificial Intelligence 66 (1994) 311–344.
[5] P. T. Baffes, R. J. Mooney, Symbolic revision of theories with M-of-N rules, in: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France, 1993, pp. 1135–1140.
[6] P. T. Baffes, R. J. Mooney, Extending theory refinement to M-of-N rules, Informatica 17 (1993) 387–397.
[7] T. Cain, The DUCTOR: A theory revision system for propositional domains, in: Proceedings of the Eighth International Workshop on Machine Learning, Evanston, IL, 1991, pp. 485–489.
[8] R. Feldman, A. M. Segre, M. Koppel, Incremental refinement of approximate domain theories, in: Proceedings of the Eighth International Workshop on Machine Learning, Evanston, IL, 1991, pp. 500–504.
[9] N. Lavrač, S. Džeroski, Inductive Logic Programming: Techniques and Applications, Ellis Horwood, 1994.
[10] E. Y. Shapiro, Algorithmic Program Debugging, MIT Press, Cambridge, MA, 1983.
[11] M. J. Pazzani, C. Brunk, Detecting and correcting errors in rule-based expert systems: An integration of empirical and explanation-based learning, in: Proceedings of the 5th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, 1990.
[12] M. J. Pazzani, D. F. Kibler, The utility of background knowledge in inductive learning, Machine Learning 9 (1992) 57–94.
[13] B. L. Richards, R. J. Mooney, First-order theory revision, in: Proceedings of the Eighth International Workshop on Machine Learning, Evanston, IL, 1991, pp. 447–451.
[14] B. L. Richards, R. J. Mooney, Automated refinement of first-order Horn-clause domain theories, Machine Learning 19 (1995) 95–131.
[15] F. Bergadano, A. Giordana, A knowledge intensive approach to concept induction, in: Proceedings of the Fifth International Conference on Machine Learning (ICML-88), Ann Arbor, MI, 1988, pp. 305–317.
[16] S. Muggleton, C. Feng, Efficient induction of logic programs, in: Proceedings of the First Conference on Algorithmic Learning Theory, Ohmsha, Tokyo, Japan, 1990.
[17] W. W. Cohen, Compiling prior knowledge into an explicit bias, in: Proceedings of the Ninth International Conference on Machine Learning (ICML-92), Aberdeen, Scotland, 1992, pp. 102–110.
[18] S. Tangkitvanich, M. Shimura, Refining a relational theory with multiple faults in the concept and subconcepts, in: Proceedings of the Ninth International Conference on Machine Learning (ICML-92), Aberdeen, Scotland, 1992, pp. 436–444.
[19] E. H. Shortliffe, B. G. Buchanan, A model of inexact reasoning in medicine, Mathematical Biosciences 23 (1975) 351–379.
[20] J. J. Mahoney, R. J. Mooney, Combining connectionist and symbolic learning to refine certainty-factor rule-bases, Connection Science 5 (1993) 339–364.
[21] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, in: D. E. Rumelhart, J. L. McClelland (Eds.), Parallel Distributed Processing, Vol. I, MIT Press, Cambridge, MA, 1986, pp. 318–362.
[22] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[23] L.-M. Fu, Integration of neural heuristics into knowledge-based inference, Connection Science 1 (1989) 325–339.
[24] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA, 1988.

[25] S. Ramachandran, R. J. Mooney, Revising Bayesian networks parameters using backpropagation, in: International Conference on Neural Networks, Washington D.C., USA, 1996, pp. 82–87.

[26] S. Ramachandran, R. J. Mooney, Theory refinement for Bayesian networks with hidden variables, in: Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), Madison, WI, 1998, pp. 454–462.

[27] W. Buntine, Theory refinement on Bayesian networks, in: Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI-91), 1991.

[28] W. Lam, F. Bacchus, Using causal information and local measure to learn Bayesian networks, in: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93), 1993, pp. 243–250.

[29] G. Towell, J. Shavlik, Knowledge-based artificial neural networks, Artificial Intelligence 70 (1994) 119–165.

[30] G. Towell, J. Shavlik, M. Noordewier, Refinement of approximate domain theories by knowledge-based neural networks, in: Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), Boston, MA, 1990, pp. 861–866.

[31] D. Opitz, J. Shavlik, Heuristically expanding knowledge-based neural networks, in: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France, 1993, pp. 1360–1365.

[32] D. Opitz, J. Shavlik, Dynamically adding symbolically meaningful nodes to knowledge-based neural networks, Knowledge-Based Systems 8 (1995) 301–311.

[33] G. Towell, J. Shavlik, Using symbolic learning to improve knowledge-based neural networks, in: Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), San Jose, CA, 1992, pp. 177–182.

[34] R. Maclin, J. Shavlik, Creating advice-taking reinforcement learners, Machine Learning 22 (1996) 251–281.

[35] S. I. Gallant, Connectionist expert systems, Commun. ACM 31 (1988) 152–169.

[36] L. Fu, Rule learning by searching on adapted nets, in: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, CA, 1991, pp. 590–595.

[37] Y. Hayashi, A neural expert system with automated extraction of fuzzy if-then rules, in: Advances in Neural Information Processing Systems 3, Morgan Kaufmann, San Mateo, CA, 1991, pp. 578–584.

[38] C. McMillan, M. C. Mozer, P. Smolensky, Rule induction through integrated symbolic and subsymbolic processing, in: Advances in Neural Information Processing Systems 4, Morgan Kaufmann, San Mateo, CA, 1992, pp. 969–976.

[39] I. Sethi, J. Yoo, C. Brickman, Extraction of diagnostic rules using neural networks, in: Proceedings of the Sixth Annual 1993 IEEE Symposium Computer-Based Medical Systems, 1993, pp. 217–222.

[40] S. Thrun, Extracting Provably Correct Rules from Artificial Neural Networks, Technical Report, University of Bonn, 1993.

[41] G. Towell, J. Shavlik, The extraction of refined rules from knowledge-based neural networks, Machine Learning 13 (1993) 71–101.

[42] J. Alexander, M. Mozer, Template-based algorithms for connectionist rule extraction, in: Advances in Neural Information Processing Systems 7, 1994.

[43] R. Setiono, H. Liu, Understanding neural networks via rule extraction, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 1995.

[44] C. Omlin, C. Giles, Rule revision with recurrent neural networks, IEEE Transactions on Knowledge and Data Engineering 8 (1996) 183 – 188.

[45] M. Craven, J. Shavlik, Extracting tree-structured representations of trained networks, in: Advances in Neural Information Processing Systems 8, MIT Press, Denver, CO, 1996, pp. 24–30.

[46] R. Andrews, J. Diederich, A. B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-Based Systems 8 (1995) 373–389.

[47] M. Craven, J. Shavlik, Rule Extraction: Where Do We Go from Here?, Technical Report Machine Learning Research Group Working Paper 99-1, Department of Computer Sciences, University of Wisconsin, 1999.
[48] L. Fu, Integration of neural heuristics into knowledge-based inference, Connection Science 1 (1989) 325–340.
[49] J. Shavlik, G. Towell, Combining explanation-based and neural learning: An algorithm and empirical results, Connection Science 1 (1989) 233–255.
[50] H. Berenji, Refinement of approximate reasoning-based controllers by reinforcement learning, in: Proceedings of the Eighth International Workshop on Machine Learning, Morgan Kaufmann, Evanston, IL, 1991, pp. 475–479.
[51] P. Frasconi, M. Gori, M. Maggini, G. Soda, An unified approach for integrating explicit knowledge and learning by example in recurrent networks, in: International Joint Conference on Neural Networks (IJCNN-91), 1991, pp. 811–816.
[52] C. Omlin, C. Giles, Training second-order recurrent neural networks using hints, in: Proceedings of the Ninth International Conference on Machine Learning (ICML-92), Aberdeen, Scotland, 1992, pp. 361–366.
[53] M. Roscheisen, R. Hofmann, V. Tresp, Neural control for rolling mills: Incorporating domain theories to overcome data deficiency, in: Advances in Neural Information Processing Systems 4, volume 4, Morgan Kaufmann, San Mateo, CA, 1992, pp. 659–666.
[54] V. Tresp, J. Hollatz, S. Ahmad, Network structuring and training using rule-based knowledge, in: Advances in Neural Information Processing Systems 5, Morgan Kaufmann, 1992, pp. 871–878.
[55] S. Thrun, T. Mitchell, Integrating inductive neural network learning and explanation-based learning, in: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France, 1993, pp. 930–936.
[56] J. Shavlik, A framework for combining symbolic and neural learning, Machine Learning 14 (1994) 321–331.
[57] R. S. Michalski, R. L. Chilausky, Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis, Journal of Policy Analysis and Information Systems 4 (1980) 126–161.
[58] B. G. Buchanan, E. Shortliffe, Rule-Based Expert Systems:The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Publishing Co., Reading, MA, 1984.
[59] P. T. Baffes, R. J. Mooney, A novel application of theory refinement to student modeling, in: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland, OR, 1996, pp. 403–408.
[60] P. T. Baffes, R. J. Mooney, Refinement-based student modeling and automated bug library construction, Journal of Artificial Intelligence in Education 7 (1996) 75–116.
[61] M. Noordewier, G. Towell, J. Shavlik, Training knowledge-based neural networks to recognize genes in DNA sequences, in: R. Lippmann, J. Moody, D. Touretzky (Eds.), Advances in Neural Information Processing Systems 3, volume 3, Morgan Kaufmann, Denver, CO, 1991, pp. 530–536.
[62] R. Maclin, J. Shavlik, Using knowledge-based neural networks to improve algorithms: Refining the Chou-Fasman algorithm for protein folding, Machine Learning 11 (1993) 195–215.
[63] C. Giles, C. Miller, D. Chen, H. Chen, G. Sun, Y. Lee, Learning and extracting finite state automata with second-order recurrent neural networks, Neural Computation 4 (1992) 393–405.
[64] G. Scott, J. Shavlik, W. Ray, Refining PID controllers using neural networks, in: J. Moody, S. Hanson, R. Lippmann (Eds.), Advances in Neural Information Processing Systems 5, volume 4, Morgan Kaufmann, Denver, CO, 1992, pp. 555–562.
[65] C. Wu, Knowledge-based artificial neural network and the application of it in understanding remotely sensed images, in: X. Shen, J. Liu (Eds.), Neural Network and Distributed Processing,

volume 4555, International Society for Optics and Photonics, SPIE, 2001, pp. 160 – 164.
[66] T. Eliassi-Rad, J. Shavlik, A theory-refinement approach to information extraction, in: Proceedings of 18th International Conference on Machine Learning (ICML-2001), Williamstown, MA, 2001.