CSIT Conference 2021, Yerevan, Armenia, September 27 - October 1

Automating Acquisition and Explanation of Strategy Knowledge

Sedrak Grigoryan
IIAP NAS RA
Yerevan, Armenia
e-mail: addressforsd@gmail.com

***Abstract*—Knowledge acquisition in different cognitive architectures and knowledge-based systems is mainly performed by manually inserted knowledge, as well as by learning from new situations, revealed from current knowledge base. In RGT Solvers of combinatorial games, knowledge is acquired hand to hand with experts and by revelation. Earlier, certain procedures of acquiring and explaining strategy RGT knowledge were outlined. This work aimed to implement these procedures, thus, to advance in automating human-computer communications.**

***Keywords*—Combinatorial problems, strategy knowledge, acquisition, explaining, communication, chess.**

I. INTRODUCTION

Problems of communications in knowledge-based systems.

In knowledge-based systems, one of the difficulties is to inject the knowledge into the problem-solving solutions. When it comes to manually define the knowledge, it is time-consuming and error-prone, even if an expert of the problem is defining the knowledge, however, this is still one of the most common way in cognitive architectures, such as SOAR [2], object-oriented models, also usually require manual definition of each OOP class and object, while there are some simplifications, e.g., UML [5]. In RGT Solvers, knowledge acquisition is performed by certain developed means [3], as well as knowledge revelation algorithms were developed [6]. On the other hand, a problem comes to the natural presentation and explanation of the knowledge already acquired by the system, which can be used in explanations of concepts, tutoring to those problems and relevant to them knowledge, where in [7] an approach for knowledge explanation in scope of personalized tutoring was suggested [7].

In [4], learning of mental systems is discussed and a constructive approach is proposed as follows: 1. Mentals can be represented as systems of do classifiers of the types of one and two place relations. 2. Learning mental systems, as a rule, are reduced to assembling the already learned mentals and relations into mental systems equal to those of communities and this assembling is supervised, explicitly or not, by instructions provided by communicatives of these mental systems. 3. Storages of thesauri Th of mentals induced by the libraries of OO classes can, in principle, be organized as storages for representing these mental systems of do classifiers of the types of one- and two- place relations.

Since the above assumption does not refer to a particular representation of 1-/2-place relations they can, particularly, be realized by a certain module of artificial neuron nets (ANN), or a unit ANN (unn), thus, allowing to state that Mentals, in principle, can be adequately modeled by ANN that, at first, represent 1-/2- place relations of mentals equal to unit ANN, then, compose these units into ANN equal to the basic constituents of mentals and, finally, unite these constituents into ANN equal to the target mentals.

To provide a certain assessment, progress and statements, we concentrate on a class of combinatorial problems, which is regularized and where space of solutions is reproducible game trees (RGT) [4, 9, 10].

RGT is a class of problems that satisfies the following conditions: 1. There are (a) interacting actors (players, competitors, etc.,) performing (b) identified types of actions in the (c) specified moments of time and (d) specified types of situations. 2. There are identified benefits for each of the actors. 3. There are descriptions of the situations the actors act in and transformed after actions. Chess and chess-like games, network intrusion protection, management in oligopoly competition are considered as RGT class problems. As it is shown [9], that the kernel of these problems is unique, which lets having a unified framework for achievements and experiment solutions and achievements for a certain kernel problem (say, chess), and then spread the solution to the whole class.

Thus, in the following work we concentrate on RGT Solver.

One of the two problems we aim to solve is the extraction of knowledge from widely used spaces, texts. Knowledge extraction is the creation of knowledge from structured (relational databases, XML) and unstructured (text, documents, images) sources. The resulting knowledge needs to be in a machine-readable and machine-interpretable format and must represent knowledge in a manner that facilitates inference. It requires either the reuse of existing formal knowledge (reusing identifiers or ontologies) or the generation of a schema based on the source data. E.g., [8] provides enterprise solutions to reveal knowledge from raw data. In [11], learning of procedural knowledge is discussed, that requires the presence of goal-directed procedures defined in the text. The approach uses cognitive and natural language processing algorithms to solve the problems, however, HBD models of knowledge presentation used in RGT Solvers vary from procedural paradigms, remaining close to human presentation of mental systems.

Since we can't assume the existance of structured forms of knowledge available in general for RGT problems, we aim to construct the pieces of knowledge by extraction from texts, where there are a lot of Natural Language Processing-based (NLP) approaches, our approach attempts to provide a combined solution, based on both NLP advancements and algorithms that can be used to extract knowledge from texts with certain expected structures.

The second problem, as mentioned, is the knowledge explanation problem, where the problem is reduced to the problem of translation of the knowledge to texts that can be properly understood by the learners, i.e., the users that aim to learn certain concepts in the space of a given problem. In [7], an approach to represent RGT knowledge for tutoring purposes is discussed. Here we aim to enhance the suggested approach.

We follow the above statements provided in [4] and attempt to concentrate on learning by extracting one- and two-place relations from the given descriptions, as well as enhance them by NN classifiers to provide examples and descriptions of those classifiers and mental systems in adequate to human means.

Problems statement for RGT Solvers:

Currently RGT Solvers require experts to inject the knowledge manually by specific interface, which causes difficulties and consumes much time. Extraction of knowledge from text, e.g from chess teaching books enhances the solver to learn knowledge more efficiently. As discussed, this is a knowledge extraction problem and we concentrate on development of algorithms for extraction of the knowledge from strictly formatted text and for solutions that can enhance revelation of such texts from free form texts using NLP-based solutions.

Tutoring and explanation of knowledge learned by RGT Solver can improve usage efficiency and understanding of RGT problems and relevant knowledge.

Experiments will be provided for chess to evaluate the adequacy of provided solutions in learning and explaining RGT problems and knowledge by RGT Solver.

II. Learning By Solvers

Algorithms for providing strict format text to RGT Solver solution.

In [3, 10], it is shown, that Solver is able to adequately acquire expert knowledge for RGT problems with a developed interface for user interactions. The RGT knowledge is represented in a network, where each node and its relations can be considered as a piece of knowledge. It is achieved by representation of the knowledge in HBD (have-be-do) model of presentation, which is based on Have, Be and Do relations close to natural languages and OOP languages. The adequacy of this presentation is demonstrated in [3, 10]. This lets us use main relations in texts to acquire the knowledge from it. Particularly, at first stage, we consider having strictly formatted texts, which describe each piece of knowledge with the following main restrictions:

The description starts from the name of the concepts, then its relations and their names come. E.g., a chess knowledge description text like "Knight is figure, has color, shape, coordinate X, coordinate Y". This simple description requires additional restriction definitions, relations and dependencies in between the attributes of the knowledge piece to be more descriptive and correct, thus a correct form would require specification of some attributes and relations, for which we use "where" keyword here "Knight is figure, has color, shape, where shape equals to knight's shape (to differentiate from other figures), coordinate X and coordinate Y". In general, when considering chess concepts, they are meaningful when discussing over the board, not out of chess context, thus, coordinates are expected attributes for chess figures, fields and other appearing concepts referring to them. To extract knowledge from the given text, the algorithm creates a node named knight, finds figure as a parent, if the parent is defined but not available, the node cannot be created. Next, it finds all appropriate attributes of the parent (figure) node, in this case there are 4 of them: x coordinate, y coordinate, shape, color, where only shape has updates, thus, others are copied in the node presentation and shape is set equal to the value appropriate for knight's shape (in RGT Solver we define their numerical representations [4] and knight's shape numerical representation is usually set to 2). The knight chess concept is now available in Solver extracted from the described text. Similarly, other types of chess knowledges can be acquired based on similar strict format texts following the mentioned format restrictions. In case when any of HBD relations are missing, they are not searched or created.

To handle free form texts and generate our expected formatted texts, NLP solutions are investigated. Currently we rely on START (http://start.csail.mit.edu/), syntactic parser that covers a wide range of English constructions and for given texts generates a set of nested triples of the form [subject relation object].

To enhance learning adequacy of RGT knowledge Solvers for lower levels in the Network of Classifiers, we also generate NN classifiers for various nodes. In [12], various chess classifiers, particularly, figures' classifiers integration is described. To enhance the adequacy of learning from examples, as well as explanation of classifiers (with showing their appearances), we enhance the NN classifiers list and concentrate on inclusion of newer ones. Particularly, already various types of "field under attack" concept specifications are integrated, "doubled pawns" concept is integrated, etc. Integration of more NN classifiers is one of the ongoing tasks.

III. Explanation By Solvers

In [7], approaches and solutions provided Solvers in explanation problems, particularly for personalized interactive tutoring to chess. The explanation of chess concepts by RGT Solvers is discussed, where the explanation is mainly reduced to providing the description of each chess concept relevant knowledge, and the regularities appearing in it, while providing ways for addressing concepts, to which the explained one is referred to.

Here we concentrate on the task of explaining RGT knowledge by constructing text descriptors for them from the nodes in the network of classifiers.

a. Algorithms for providing strict format text (human-understandable) for queried knowledge by RGT Solver.

At the first step, we aim to provide human-readable strictly formatted texts for the descriptions of concepts. As discussed in Section 2, the structure of network of classifiers is based on HBD English language main dimensions and OOP, which lets us developing algorithms by concentrating on proper descriptions of these main relations' extraction.

Logic is based on the following main ideas: first find the parent concept of the desired concept and describe with "is" relation, e.g., "Knight is figure", similarly search for have relations, e.g., "Knight has coordX, coordY, color and shape". If concepts attribute name and attribute's parent name are different, it shows both, e.g., "field under attack has attacker figure", where the first one is the name and the sconed one is the parent's name, for the "do" relations it can be described "Knight does knightMove", after these descriptions, each of the above concepts referred by Knight can be queried. When reaching the nuclear nodes [4], the mapping of each node and the real live representation of their values are kept or appropriate NN classifier exists with named classes, e.g., shape = 2 value is being replaced by "knight" and its icon when searching for that value on chess board interface, color = 1 is replaced with "white".

When dealing with virtual classifiers [4] and actions, the algorithm behaves differently. Let's first discuss the actions.

The above example of knightMove can be described as the following "knightMove is move by knight", however, it also has regularities describing relation of knight and the field where it moves, and it has at most 8 different specifications, thus, we describe it "knightMove1 is move by knight, where knight.coordX = field.coordX + 1 and knight.coordY = field.coordY + 2, move is perfomed by setting knight.coordX to field.coordX and knight.coordY = field.Y" (overall the algorithm can apply "knight on field's position" text when it finds application of X and Y coordinates from one concept to another in the action, so it would be "…by setting knight on field's position"), similarly, other 7 specifications of knightMove can be provided. As a result, when describing actions, algorithm searches for their *specifications* and describes them (similar to learning by examples).

It is also useful to provide samples on board and query NN classifiers to demonstrate those classes on images, which can be achieved for classes that already have their NN classifiers.

Virtual classifiers, other than *specifications* have *usages* [4]. In case of explanation of virtual classifiers themselves, similar to actions algorithms are applied. E.g., "field under attack has attacker figure, target field", "field under attack of queen has attacker queen, target field". Similarly, if the specifications themselves are also virtual and have their own specifications, when their descriptions are queried, they are also described as discussed.

In case of usages a concept contains an attribute of virtual classifier. In this case the attribute is named as the virtual parent, not itself, e.g. "check has attribute field under attack, king, where field under attack target is king" is the result of description of check where check.fua.target=king. In this case, direct attribute of check.field_under_attack itself cannot be queried with its own values, but only with its parent, the description of which is the same as above.

IV. Experiments And Examples For Chess Knowledge

As in the above examples of discussed algorithms, the experiments were provided for chess in RGT Solvers. The algorithms were developed to provide a text for the queried concept by its name, and the creation of a new node can be triggered with the text formats discussed in II.

Various chess concepts of various levels, including cr1s [10], composite ones, actions and virtual concepts were properly described in text forms from Network of Classifiers, particularly figures, fields, virtual classifiers, such as "field under attack" and examples on the board images were properly classified.

From strictly formatted texts chess main cr1s, types of figures, composite classifiers were acquired by the solver. Some freely formed knowledge describing texts were successfully transferred to the mentioned format and then learned by Solvers as well.

V. Conclusion

In the following work, we discussed means of automation of acquisition and explanation in communication between users and RGT Solvers, particularly the acquisition from texts.

1. Algorithms for extracting RGT knowledge from texts were developed and discussed, which included classifiers learning from strictly formatted texts, particularly virtual classifiers, as well as approaches for overcoming strict format restrictions were discussed and some solutions shown.

Meantime enhancement of acquisition of classifiers by examples, using neural networks is performed, which allows adequately demonstrate and learn from images, a way of natural presentation of classifiers, as well as allows using these classifiers to properly demonstrate examples in when explaining them.

2. Algorithms for explaining acquired by RGT Solvers knowledge by texts were provided, which, based on HBD model of knowledge presentation allows providing texts for RGT classifiers using have, be, do dimensions of English.

3. Experiments to validate the algorithms were conducted for chess classifiers.

References

[1] A. Samsonovich, "*Toward a Unified Catalog of Implemented Cognitive Architectures*", *Biologically Inspired Cognitive Architectures*, Washington, DC, USA, 50 pp. 2010.

[2] J. Laird, *The Soar Cognitive Architecture*, England, MIT Press, 2012.

[3] K. Khachatryan and S. Grigoryan, "*Java Programs for Presentation and Acquisition of Meanings in SSRGT Games*", *Proceedings of SEUA Annual conference*, Yerevan, Armenia, pp. 135-141, 2013.

[4] E. Pogossian, *Constructing Models of Being by Cognizing*, Academy of Sciences of Armenia, Yerevan, Armenia, 496 pp., 2020.

[5] [Online]. *UML*® Web Resource Page Available: http://www.uml.org/.

[6] K. Khachatryan, "*A Combinative Approach to Generalization of Meanings*", Mathematical Problems of Computer Sciences, vol. vol. 39, pp. 54-65, 2013.

[7] S. Grigoryan and L. Berberyan, "*Developing Interactive Personalized Tutors in Chess*", Mathematical Problems of Computer Science, vol. 44, pp. 116-132, 2015.

[8] [Online] https://www.rocketsoftware.com/products/rocket-corvu

[9] E. Pogossian, "*Effectiveness enhancing knowledge based strategies for SSRGT class of defense problems*", *NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems*, Salamanca, Spain, 16 pp., 2011.

[10] S. Grigoryan, *Research and Development of Algorithms and Programs of Knowledge Acquisition and Their Effective Application to Resistance Problems*, Yerevan, Armenia, 2016.

[11] P. Langley, H. Shrobe and Boris Katz, "*Cognitive Task Analysis of Rapid Procedure Acquisition from Written Instructions*", *Annual Conference on Advances in Cognitive Systems*, held virtually on slack and zoom, 19 pp., 2020.

[12] N. Hakobyan, "*A System for Transforming Images to Symbolic Presentation for Combinatorial Defense and Competition Problems*", *ISSN 0002-306X. Proc. of the RA NAS and NPUA Ser. Of tech. sc.* vol. LXXII, N2, pp. 199-209, 2019.