Toward a Neural-Symbolic Framework for Automated Workflow Analysis in Surgery

Hirenkumar Nakawala[1(✉)], Elena De Momi[2], Roberto Bianchi[3], Michele Catellani[3], Ottavio De Cobelli[3], Pierre Jannin[4], Giancarlo Ferrigno[2], and Paolo Fiorini[1]

[1] Department of Computer Science, University of Verona, Via Str. le Grazie 15, 37134 Verona, Italy
hirenkumarchandrakant.nakawala@univr.it
[2] Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy
[3] Department of Urology, European Institute of Oncology, Via Guiseppe Ripamonti 435, 20141 Milan, Italy
[4] LTSI, INSERM U1099, Universitè de Rennes 1, Rennes 35000, France

Abstract. Learning production rules from continuous data streams, e.g. surgical videos, is a challenging problem. To learn production rules, we present a novel framework consisting of deep learning models and inductive logic programming (ILP) system for learning surgical workflow entities that are needed in subsequent surgical tasks, e.g. "what kind of instruments will be needed in the next step?" As a prototypical scenario, we analyzed the Robot-Assisted Partial Nephrectomy (RAPN) workflow. To verify our framework, first consistent and complete rules were learnt from the video annotations which can classify RAPN surgical workflow and temporal sequence at high-granularity e.g. steps. After we found that RAPN workflow is hierarchical, we used combination of learned predicates, presenting workflow hierarchy, to predict the information on the next step followed by a classification of step sequences with deep learning models. The predicted rules on the RAPN workflow was verified by an expert urologist and conforms with the standard workflow of RAPN.

Keywords: Production rules · Robot-Assisted Partial Nephrectomy · Surgical workflow · Deep learning · Inductive logic programming

1 Introduction

Learning production rules from continuous data streams, e.g. surgical videos, is a challenging problem. In our previous work [1, 2], we used production rules, i.e. "if-then" rules, inside an ontology to define the surgical workflow and to discretize surgical activities through symbol grounding by assigning semantic meaning to the objects in the environment. However, pre-defining a fixed set of production rules limits the scalability of expert systems in dynamic situations because it limits understanding of the streaming data and the real-time reasoning on it. The production rules are generally very discriminative and easily comprehensible by humans. The latter could

© Springer Nature Switzerland AG 2020
J. Henriques et al. (Eds.): MEDICON 2019, IFMBE Proceedings 76, pp. 1–8, 2020.
https://doi.org/10.1007/978-3-030-31635-8_192

be helpful to understand the surgical workflow, for example displaying information on the consequent surgical activities which are automatically extracted from the rules. Rules could also be helpful as constraints in automation of the surgery, e.g. verification of step sequences for the safe task execution. A full surgical activity is expected to consist of surgical entities, e.g., *phases*, *steps*, *instruments*, *actions* and *anatomy*. Recognition of a surgical activity needs a large set of annotated data on these entities as well as robust neural network architectures. On the contrary, scene recognition is still outside the capabilities of symbolic systems which are good at recognizing relations between entities in surgical activities. Integrating neural and symbol systems [3] for analyzing surgical workflows may provide complementary benefits interpreting a surgical activity.

Inductive Logic Programming (ILP) is formed at the intersection of machine learning and logic programming. ILP systems learn predicate descriptions, first-order causal theories, in the form of rules from the examples, e.g. real-world instances, and background knowledge. Examples, e.g. surgical step sequences, background knowledge, which are predicates describing relational information on surgical entities, and final descriptions i.e. a hypothesis or a rule all are described as logic programs. Recently, ILP has been extensively used in classification, data mining and information extraction (IE) tasks [4]. In the medical field, ILP has been applied to diagnostics e.g. to detect the breast cancer [5] and to do knowledge discovery from the mammography reports [6]. As far as our knowledge allows, ILP is yet to be applied for the analysis of surgical workflows which can be used to predict information about workflow steps. In [7], ILP system is used with an instance detection system, using deep learning models, for interpretation of the scene. In this manuscript, we present a preliminary analysis of a use of ILP to analyze surgical workflow for learning complete and consistent rules for classification of RAPN workflow as well as its use in a pipeline with an instance detection system which was used to recognize surgical steps and sequences.

As a prototypical scenario, in this work, we automatically analyzed the surgical workflow of Robot-Assisted Partial Nephrectomy (RAPN). RAPN is performed to remove the kidney tumor. RAPN is associated with high complicate rates from 12.3 to 33% with different surgical approaches [8]. Learning production rules to classify relations between the surgical entities, following a recognition of an instance e.g. a sequence of surgical step, would be of great importance to surgical assistance, especially for novice surgeons, by providing information on a workflow which is expected to improve surgical outcomes.

2 Methods

In our previous work [1], we extracted features from 9 videos (996,373 frames) representing 10 RAPN steps and classified it using the deep learning models. As shown in Fig. 1, the deep learning models were consisted of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) cells to classify “Current step” and “Next step” respectively. In this work, we combined predictions of these two networks as predicates representing the current step and corresponding subsequent step, for example “hasNextStep”(mobilization, dissection), and used correctly predicted step

sequences, as positive examples, in an ILP system. We then applied ILP to classify a dataset on RAPN workflow entities to find relational information between instances of surgical entities, for example, instances of instruments needed in the next step.

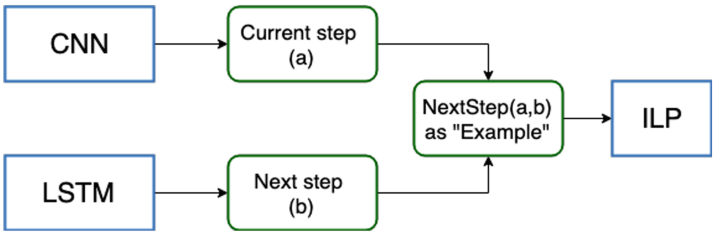

Fig. 1. A prototypical framework consisting of two components, i.e. (1) deep learning models – CNN and LSTM, and (2) ILP. In the framework, outputs of the deep learning models, instances of current and next step, are forwarded as relational predicates "Example" to an ILP system.

Terminology

The first-order logic alphabet is composed of *predicate* symbol (e.g., "step"), *function* symbols (e.g., "hasAction"), *constant* (e.g., "dissection") and *variables* (e.g., *x*). A *term* is any constant, variable, or function applied to a term (e.g., *dissection, x, has Action (x)).* An *atomic formula* is a predicate symbol together with its arguments, each argument being a "term". A *ground atom (or fact)* is an atomic formula with no variables (e.g., *hasNextStep (mobilization, dissection)).* The dataset created from video annotations are *ground atoms* and they also constitute the *background knowledge.* A *literal* is an atomic formula and a clause is a disjunction of literals whose values are assumed to be universally quantified. "Horn clause" is a clause with exactly one positive literal.

Aleph

ILP classifiers attempt to learn a set of rules (definite clauses), given a dataset of positive and negative instances, that will correctly discriminate between the sets. These rules cover most or all of the positive instances, and little or none of the negative instances. The aim of ILP is to find a theory that is complete and consistent. More formally, given a positive example $pos(x_i)$, e.g. a predicate representing surgical step consequence, let $\perp_i$ be the bottom clause, consisting of action, phase and instruments for example *i*. $\perp_i$ is the most specific hypothesis that together with the background knowledge *B*, a procedural knowledge base, entail $x_i : (B \wedge \perp_i \wedge x_i) \vdash pos(x_i)$. In this work, we used "Aleph" (A learning engine for proposing hypotheses) [9], a type of ILP classifier, to learn first-order rules in the form of Horn clauses. Aleph also uses bottom clauses, i.e. the most specific hypothesis, to guide the search. Moreover, it runs also without any negative examples. The predicate in the rule *head* is not in the background knowledge. We followed the four basic steps in the Aleph greedy learning approach:

(1) Select a positive example. Each instance of the relation can be seen as pair of step sequences, e.g. *hasNextStep*(mobilization, dissection).
(2) Build bottom clause. The bottom clause is conjunction of all the relations as pair of action and step, step and instrument, phase and step and between the steps.

For example, if (mobilization *hasNextStep* A) then (*actionperformedinstep* in A are dissect and cut) and (A *stephasinstrument* monopolarCurvedScissors and fenestratedBipolar and sucker and hemolokClipApplicator) and (hilumDissection *phasehasstep* A) and (A *haspreviousstep* mobilization).

(3) Search. This step uses greedy best-first search to find a clause consistent with the data.

(4) Removed covered positive examples. Coverage of each hypothesis is counted and the positive examples from the training set are removed.

As a refinement operator, we only considered examples where we cannot find a consistent generalization as exceptions. In those cases, we added the bottom clause as the consistent rule.

Dataset From Video Annotation and Background Knowledge Construction

To extract the background knowledge, 9 videos at 24 Hz with an approximate length of 82.49 ± 37.54 min on RAPN were acquired using the da Vinci Xi surgical system and the da Vinci Xi endoscope (Intuitive Surgical Inc., CA, USA) at European Institute of Oncology (Milan, Italy). As mentioned in [1], we used Anvil annotation tool [10] to annotate the videos, consisting the frame-by-frame annotations of "Phase", "Step", "Instrument", "Assistant-Instrument1", "Assistant-Instrument2", "Anatomy", and "Actions". Each track specifies different surgical workflow entities in synchrony without specifying any relations between them. Definitions of classes of annotations and more details can be found in [1]. Video annotations were saved in a comma separated values as a database, where the relational predicates were extracted considering the time-series in each video. To form a background knowledge, we extracted tuples specifying relations between *phase*, *step*, *action*, *instrument* and *anatomy* and temporal relations between *step*. Background knowledge is represented in Prolog language. Each relational predicate is represented in the form: predicate (ModeType, Modetype, ..), where ModeType is one of: (1) $+$ ModeType specifies the input, (2) $-$ ModeType specifies the output, and (3) $\#$ Modetype specifies a constant. Examples of Prolog facts derived from video annotations are shown in Table 1.

Experimental Protocols

The goal of the experimental protocols is to assess the complete and consistent rules to solve the problem of classification of surgical workflow and prediction of information in the next step following a classification of steps and sequences from the videos.

1. As shown in Table 1, "Rule" i.e. hypotheses 1, 2 and 3 are evaluated if they are consistent and complete over a set of 5 annotations. In "EP1", all the datasets are combined, and redundant examples are removed. Then, the dataset is divided into 65%, a total of 17 examples, as a training set and 35%, a total of 9 examples, as a testing set to get the baseline results (a single split). Furthermore, in "EP2", we did fivefold LOOCV (Leave-one-out cross-validation) to learn the rules and to check whether they are useful for classifying the workflow.
2. In "EP3", predicted instances of steps predicates and sequences from deep learning models are forwarded to Aleph, with background facts and hypothesis as shown in "Rule-4" in Table 1, to predict workflow entities of the next step. The predictions are qualitatively assessed by an expert urologist "MC".

Table 1. Predicates, "ground atoms", used in background knowledge

Rules	Type of information	Predicates	Description	An exemplary predicate showing examples
1	Step transitions induced by temporal relations of actions in steps	"hasNextStep" (+step, −step) "stepHasAction" (+step, −action) "actionPerformedInPreviousStep" (+action, −step)	Relations between actions and steps for predicting next step	*hasNextStep*(mobilization, dissection) as a positive example *hasNextStep*(dissection, mobilization) as a negative example
2	Relations between step and phase induced by relations between phase, actions and instruments	"phaseHasStep" (+phase, −step) "phaseHasInstrument" (+phase, −instrument) "actionPerformedbyInstrument" (+action, −instrument) "actionPerformedInStep" (+action, −step)	Relations between actions, instrument and phases to classify steps in phases	*phaseHasStep*(mobilization, mobilize) as a positive example *phaseHasStep*(mobilization, suturing) as a negative example
3	Temporal relations	"hasPreviousStepOfPreviousStep" (+step, −step) "hasPreviousStep" (+step, −step) "hasPreviousStep" (+step,−step)	Classifying previous steps	*hasPreviousStepOfPreviousStep* (removal, drainage) as a positive example *hasPreviousStepOfPreviousStep* (dissection, drainage) as a negative example
4	Step transitions induced by the instances of workflow entities.	"hasNextStep" (#step, +step) "actionPerformedInStep" (#action, +step) "stepHasInstrument" (+step, #instrument) "phaseHasStep" (#phase, +step) "hasPreviousStep" (+step, #step)	Classify instances of workflow entities, e.g. actions, instruments, phases and step of a next step	*hasNextStep*(mobilization, dissection) as a positive example

3 Results and Discussion

Aleph's learning is done by restricting the clause length from minimum of 2 literals to maximum of 8 literals. The results are summarized as below:

Table 2. Induced rules, i.e. for hypothesis representing rule-4, from the instances recognized from deep learning models. The rules are represented in horn clauses and show instances of surgical entities that are required in the next step ("A").

Predicted step sequence	Generated rules
hasnextstep (mobilization, dissection)	hasnextstep(mobilization, A):- actionperformedinstep(dissect, A), actionperformedinstep(cut, A), stephasinstrument(A, monopolarCurvedScissors), stephasinstrument(A, fenestratedBipolar), stephasinstrument(A, sucker), stephasinstrument(A, hemolokClipApplicator), phasehasstep (hilumDissection, A), haspreviousstep(A, mobilization)
hasnextstep(dissection, ultrasound)	hasnextstep(dissection, A):- actionperformedinstep(detect, A), stephasinstrument(A, fenestratedGraspingForcep), stephasinstrument(A, monopolarCurvedScissors), stephasinstrument(A, fenestratedBipolar), stephasinstrument(A, laparoscopicUltrasoundProbe), phasehasstep(tumorExposure, A), haspreviousstep(A, identificaion)
hasnextstep(marking, clamping)	hasnextstep(marking, A):- actionperformedinstep(clamp, A), actionperformedinstep(dissect, A), stephasinstrument(A, monopolarCurvedScissors), stephasinstrument(A, fenestratedBipolar), stephasinstrument(A, laparoscopicBulldog), phasehasstep(tumorResection, A), haspreviousstep(A, marking)
hasnextstep(clamping, resection)	hasnextstep(clamping, A):- actionperformedinstep(resect, A), stephasinstrument(A, monopolarCurvedScissors), stephasinstrument(A, fenestratedBipolar), stephasinstrument(A, fenestratedGraspingForcep), stephasinstrument(A, sucker), stephasinstrument(A, metalClipApplicator), phasehasstep (tumorResection, A), haspreviousstep(A, clamping)
hasnextstep(resection, suturing).	hasnextstep(resection, A):- actionperformedinstep(suture, A), stephasinstrument(A, roboticLargeNeedleDriver), stephasinstrument(A, monopolarCurvedScissors), stephasinstrument(A, hemolokClipApplicator), phasehasstep (renorraphy, A), haspreviousstep(A, resection)
hasnextstep(suturing, unclamping).	hasnextstep(suturing, A):- actionperformedinstep(unclamp, A), stephasinstrument(A, monopolarCurvedScissors), stephasinstrument(A, roboticLargeNeedleDriver), stephasinstrument(A, laparoscopicBulldog), phasehasstep (renorraphy, A), haspreviousstep(A, suturing)
hasnextstep (reconstruction, drainage)	hasnextstep(reconstruction, A):- actionperformedinstep(put, A), stephasinstrument(A, fenestratedGraspingForcep), stephasinstrument(A, roboticLargeNeedleDriver), stephasinstrument(A, monopolarCurvedScissors), phasehasstep (closure, A), haspreviousstep(A, reconstruction)

Quantitative Evaluation: Rule 1 and 2 represent hypothesized relations between the workflow entities, while rule-3 verifies the temporal relations at higher-granularity level, i.e. steps. With both experiments, we obtained 100% accuracy, precision and recall predicting hypotheses with rule-2 and rule-3. In "EP-1", with rule-1, we obtained 71% accuracy, 33% sensitivity, and 100% specificity. However, in "EP-2", for the Rule 1, we obtained 100% accuracy, precision, and recall. While doing these experiments, we excluded other 4 video annotations considering some missing data. The results present that the relations between surgical entities can be used to classify the hierarchical RAPN surgical workflow. The latter is also supported by the Discrete-Time Markov Chain (DTMC) constructed for RAPN step transitions from the similar set of video annotations in our previous work [1].

Qualitative Evaluation: In "EP3", since we extracted constants in "Rule-4", i.e. relations between the instances of workflow entities, we inputted prediction of deep learning models as examples to check the hypothesis whether Aleph can predict consistent and complete information of the subsequent step. As shown in Table 2, the generated rules represent surgical resources, e.g. instruments, needed in the next step and information on actions and phases. Generated rules on the workflow were verified by an expert urologist ("MC") and found it to be true for the sequence of steps in RAPN workflow.

4 Conclusions

In this feasibility study, we have shown that implementation of ILP with an instance detection system can be useful to learn explanatory rules on the RAPN surgical workflow. We have tested four different types of hypotheses: (1) relations between surgical workflow entities; (2) temporal relations; (3) relations between instances of workflow entities and (4) relations for verifying the step sequences. The system was be able to learn rules for RAPN workflow representing all these hypotheses. The learned predicates represent workflow decomposition from higher-granularity e.g. phases to lower-granularity e.g. actions. Instruments can be considered as task resources to execute actions in steps. The relational information between surgical entities can be useful to predict information on the next step.

In future, we will find the exact temporal sequence especially at lower granularity, e.g. prediction of sequences of actions in the next predicted step. We will consider rules as constraints while training the deep learning networks [3] for recognizing surgical entities and ontology to learn the strict-order. Moreover, we will also compare it with the performance of other rule learners, such as associative rule learning [11].

Acknowledgement. This research is funded by the European Research Council (ERC) under the European Union's H2020 research and innovation programme (grant agreement No. 742671 "ARS"). This work was supported by European Union's Horizon 2020 research and innovation programme (grant agreement No. 732515 "SMARTsurg").

Conflict of Interest. The authors confirm that there are no known conflicts of interest associated with this publication.

References

1. Nakawala, H., Bianchi, R., Pescatori, L.E., De Cobelli, O., Ferrigno, G., De Momi, E.: "Deep-Onto" network for surgical worflow and context recognition. Int. J. Comput. Assist. Radiol. Surg. **14**(4), 685–696 (2019)
2. Nakawala, H., Ferrigno, G., De Momi, E.: Developement of an intelligent training system for Thoracentesis. Artif. Intell. Med. **84**, 50–63 (2018)
3. Garcez, A., Besold, T.R., de Raedt, L., Földiak, P., Hitzler, P., Icard, T., Kühnberger, K.-U., Lamb, L.C., Miikkulainen, R., Silver, D.L.: Neural-symbolic learning and reasoning: contributions and challenges. In: 2015 AAAI Spring Symposium, Palo Alto, California (2015)
4. Bock, H.-H.: Data Mining Tasks and Methods: Classification: the Goal of Classification. Handbook of Data Mining and Knowledge Discovery. Oxford University Press, New York (2002)
5. Dutra, I., Nassif, H., Page, D., Shavik, J., Strigel, R.M., Wu, Y., Elezaby, M.E., Burnside, E.: Integrating machine learning and physician knowledge to improve the accuracy of breast biopsy. In: AMIA Annual Symposium Proceedings, pp. 349–355 (2011)
6. Burnside, E.S., Davis, J., Costa, V.S., de Castro Dutra, I., Kahn Jr., C.E., Fine, J., Page, D.: Knowledge discovery from structured mammography reports using inductive logic programming. In: AMIA Annual Symposium Proceedings, pp. 96–100 (2005)
7. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. J. Artif. Intell. Res. **61**(1), 1–64 (2018)
8. Hu, J.C., Treat, E., Filson, C.P., McLaren, I., Xiong, S., Stepanian, S., Hafez, K.S., Weizer, A.Z., Porter, J.: Technique and outcomes of robot-assisted retroperitoneoscopic partial nephrectomy: a multicenter study. Eur. J. Urol. **66**, 542–549 (2014)
9. Srinivasan, A.: The Aleph Manual (1999). https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html
10. Kipp, M.: Anvil–a generic annotation tool for multimodal dialogue. In: Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech), pp. 1367–1370 (2001)
11. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data - SIGMOD 1993, p. 207 (1993)