Learning about Actions and Events in Shared NeMuS

Milena Rodrigues Tenório[1], Edjard de Souza Mota[1]
Jacob M. Howe[2], Artur S. d'Avila Garcez[2]

[1] Universidade Federal do Amazonas,
Instituto de Computação, Campus Setor Norte
Coroado - Manaus - AM - Brasil CEP: 69080-900
{mrt,edjard}@icomp.ufam.edu.br,
[2] City, University of London, London, EC1V 0HB, UK
{J.M.Howe,a.garcez}@city.ac.uk

1 Introduction

The categorization process of information from pure data or learned in unsupervised artificial neural networks is still manual, especially in the labeling phase. Such a process is fundamental to knowledge representation [6], especially for symbol-based systems like logic, natural language processing and textual information retrieval. Unfortunately, applying categorization theory in large volumes of data does not lead to good results mainly because there is no generic and systematic way of categorizing such data processed by artificial neural networks and joining investigated conceptual structures.

Connectionist approaches are capable of extracting information from artificial neural networks, but categorizing them as symbolic knowledge have been little explored. The obstacle lies on the difficulty to find logical justification from response patterns of these networks [2]. This gets worse when considering inductive learning from dynamic data which is very important to Cognitive Sciences that considers categorization as a mental operation of classifying objects, actions and events [1].

We shall address the discoveries of our on-going investigation on the problem of inductively learning (IL) from dynamic data by applying a novel framework for neural-symbolic representation and reasoning called share Neural Multi-Space (NeMuS) used in the Amao system[4]. Instead of woking like traditional approaches for ILP, e.g. [5], Amao uses a shared NeMuS of a give background knowledge (BK) and uses inverse unification as the generalization mechanism of a set of logically connected expressions from the Herbrand Base (HB) of BK that defines positive examples.

2 Discussion on Inductive Learning about Dynamic Data

Our scenario on the realm of Driving offences considering a small portion of it in a usual traffic light. Consider four cars `c1`, `c2`, `c3` and `c4`, and a traffic light `tl1` on a street `s1`, as depicted in Figure 1.

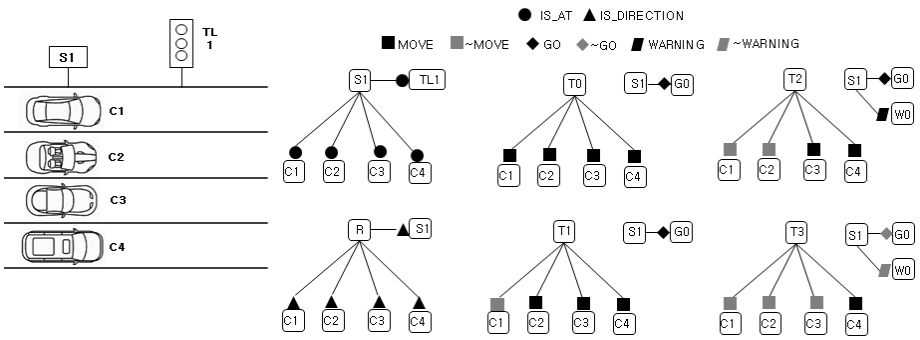

Fig. 1. BK representation of example.

The logical entities considered are based on dynamic nature of the data[3], like a historical database about a traffic light. No external action is assumed to influence this scenario and the passage of time is based on actions on it.

```
car(c1).                is_at(c1,s1).          is_direction(c1,r).
car(c2).                is_at(c2,s1).          is_direction(c2,r).
car(c3).                is_at(c3,s1).          is_direction(c3,r).
car(c4).                is_at(c4,s1).          is_direction(c4,r).
street(s1).             is_at(tl1,s1).         is_direction(s1,r).
traffic_light(tl1).
```

There are four moments of interest.**M1**: All cars are moving; **M2**: `c1` stopped on street; **M3**: `tl1` signals warning, so `c2` has stopped; and **M4**: `tl1` stops blinking yellow and goes to stop sign (red), then `c3` has stopped.

M1	**M2**	**M3**	**M4**
			`~warning(tl1,w1).`
`go(tl1,g0).`			`~go(tl1,g1).`
`move(c1,t0).`	`~move(c1,t1).`	`~move(c1,t2).`	`~move(c1,t3).`
`move(c2,t0).`	`move(c2,t1).`	`~move(c2,t2).`	`~move(c2,t3).`
`move(c3,t0).`	`move(c3,t1).`	`move(c3,t2).`	`~move(c3,t3).`
`move(c4,t0).`	`move(c4,t1).`	`move(c4,t2).`	`move(c4,t3)..`

```
during(t0,t2,g0).          during(t2,t2,w0).
during(t3,t3,g1).          during(t3,t3,w1).
```

Change through time of an entity (car or traffic light) is represented by a change from positive to negative literal of `move`, `warning` and `go`. The goal is to identify which cars have committed a traffic light or driving offense. Traffic regulations would point that `c1` and `c4` violated such laws. `c1` is blocking (it stopped) traffic on `s1` while `tl1` indicates to go, and `c4` is moving when `tl1` is indicating to stop. For lack of space, we consider just the target `block_offence(C)` with positive example `block_offence(c1)` and negative `~block_offence(c2)`, and the target `driving_offence(C)` with positive example `driving_offence(c4)` and negative `~driving_offence(c3)`. The following (unsual long) hypothesis should be generated.

```
block_offence(C); car(C); traffic_light(TL); is_at(C,S); is_at(TL,S);
    move(C,TA); go(TL,GA); ~move(C,TB); warning(TL,W1);
    during(TA,TB,GA); during(TX,TY,W1).
driving_offence(C); car(C); traffic_light(TL); is_at(C,S); is_at(TL,S);
   move(C,TA); go(TL,GA); move(C,TB); ~go(TL,GB);
   during(TA,TX,GA); during(TY,TB,GB).
```

Shared NeMuS codes allow us to know that (1) the car and traffic light are on the same street, and (2) the car has the same direction of the street. Predicate code 2 can be ignored inasmuch as it is unnecessary information, because it is a rule that involves carriage movement and obedience to traffic light. Amao gets the bindings relations (occurrences) the given objects as positive and negative examples. As `c1`, `c2`, `c3`, and `c4` are cars they belong to the same region of predicate codes. `is_at()` relating to `s1`, both have `is_direction()` with `r` (right), and have a single object different that also relates through `is_at ()` with `s1`, the `tl1`. The difference between the positive and negative examples lies in their *action predicates*, in which `c1` performs an action before any change occurs in `t1`. We, not in Amao a automatic mechanism, also noticed that the predicate `is_direction ()` and `r` do not relate in any way to another object or to *action predicates*. Thus, we can ignore this information in the construction of a hypothesis. By inverse unification, Amao finds a linkage pattern between *is_at(C,S)* and *is_at(TL,S)*, and thus connecting the car and the traffic light.

3 Concluding Remarks

The example explored here is small, and yet we have a long rule. An example with more information, such as velocity, position of the car and people on the street,their relations we will have many ways to find hypotheses. We aim to overcome this challenge by using shared NeMuS weight to to group the predicates that form an intermediate concept, an abstraction, so that we can add only the predicates needed for the rules.

References

1. Cohen, H., Lefebvre, C.: Handbook of categorization in cognitive science. Elsevier (2005)
2. d'Avila Garcez, A.S., Broda, K., Gabbay, D.: Neural-Symbolic Learning Systems: Foundations and Applications, Perspectives in Neural Computing. Springer-Verlag (2002)
3. Gardenfors, P.: Concept learning and non-monotonic reasoning. In: Handbook of Categorization (2005)
4. Mota, E.d.S., Howe, J., Garcez, A.: In: Besold, T.R., d'Avila Garcez, A., Noble, I. (eds.) To appear NeSy 2017 Neural-Symbolic Learning and Reasoning (July)
5. Muggleton, S.H.: Inductive Logic Programming. New Generation Computing 8(4), 295–318 (1991)
6. Sowa, J.F., et al.: Knowledge representation: logical, philosophical, and computational foundations, vol. 13. MIT Press (2000)