

Linear Temporal Public Announcement Logic: a new perspective for reasoning the knowledge of multi-classifiers

Amirhoshang Hoseinpour Dehkordi ¹, Majid Alizadeh ², and Ali Movaghar ²

¹Institute for Research in Fundamental Sciences

²Affiliation not available

October 30, 2023

Abstract

Current applied intelligent systems have crucial shortcomings either in reasoning the gathered knowledge, or representation of comprehensive integrated information. To address these limitations, we develop a formal transition system which is applied to the common artificial intelligence (AI) systems, to reason about the findings. The developed model was created by combining the Public Announcement Logic (PAL) and the Linear Temporal Logic (LTL), which will be done to analyze both single-framed data and the following time-series data. To do this, first, the achieved knowledge by an AI-based system (i.e., classifiers) for an individual time-framed data, will be taken, and then, it would be modeled by a PAL. This leads to developing a unified representation of knowledge, and the smoothness in the integration of the gathered and external experiences. Therefore, the model could receive the classifier's predefined -or any external- knowledge, to assemble them in a unified manner. Alongside the PAL, all the timed knowledge changes will be modeled, using a temporal logic transition system. Later, following by the translation of natural language questions into the temporal formulas, the satisfaction leads the model to answer that question. This interpretation integrates the information of the recognized input data, rules, and knowledge. Finally, we suggest a mechanism to reduce the investigated paths for the performance improvements, which results in a partial correction for an object-detection system.

LINEAR TEMPORAL PUBLIC ANNOUNCEMENT LOGIC: A NEW PERSPECTIVE FOR REASONING THE KNOWLEDGE OF MULTI-CLASSIFIERS

Amirhoshang Hoseinpour Dehkordi

School of Computer Science
Institute for Research in Fundamental Sciences,
Tehran, 19538 - 33511, Iran
amir.hoseinpour@ipm.ir

Majid Alizadeh*

School of Mathematics, Statistics and Computer Science
College of Science, University of Tehran,
Tehran, 14155-6455, Iran
majidalizadeh@ut.ac.ir

Ali Movaghar

Department of Computer Engineering,
Sharif University of Technology,
Tehran, 11155-9517, Iran
movaghar@sharif.edu

September 9, 2020

ABSTRACT

Current applied intelligent systems have crucial shortcomings either in reasoning the gathered knowledge, or representation of comprehensive integrated information. To address these limitations, we develop a formal transition system which is applied to the common artificial intelligence (AI) systems, to reason about the findings. The developed model was created by combining the Public Announcement Logic (PAL) and the Linear Temporal Logic (LTL), which will be done to analyze both single-framed data and the following time-series data. To do this, first, the achieved knowledge by an AI-based system (i.e., *classifiers*) for an individual time-framed data, will be taken, and then, it would be modeled by a PAL. This leads to developing a unified representation of knowledge, and the smoothness in the integration of the gathered and external experiences. Therefore, the model could receive the classifier's predefined -or any external- knowledge, to assemble them in a unified manner. Alongside the PAL, all the timed knowledge changes will be modeled, using a temporal logic transition system. Later, following by the translation of natural language questions into the temporal formulas, the *satisfaction* leads the model to answer that question. This interpretation integrates the information of the recognized input data, rules, and knowledge. Finally, we suggest a mechanism to reduce the investigated paths for the performance improvements, which results in a partial correction for an object-detection system.

1 Introduction

Nowadays, artificial intelligence (AI) is inseparable from real-world applications. Besides, the classification, which is considered as the classic application of AI-based algorithms, the cognitive understanding is also gained significant attention more recently. The current works on visual understandings of both image (see [1], [2], and [3]), and video (see [4], [5], and [6]) show the importance of data understandings. Although to date developed approaches get more accuracy via proposing new approaches, there is no certain approach formed to solve such problems, entirely. Moreover, due to the probabilistic nature of most of these approaches, it seems that it is impossible to attain complete accuracy in the near future.

*Corresponding author

To be more precise, followed by the accessing of current works, the process of data understanding could be divided into the following steps: the first and the most popular one is the object recognition from the input data, for which there are many solutions, with fairly great performances [7]. The next step is the knowledge extraction, which is a way to understand the knowledge acquired in the initial step. Herein, objects will be transformed into symbols, in which they could be shared and aggregated in a modal logic approach [8]. Following that, for a reasoning system, it is mandatory to interpret guaranteed rules which could be fed into the system, using predefined protocols (i.e., “cat is an animal”). The final step is to collect a set of knowledge that would lead a model to answer ordinary questions. The last three steps will be discussed further in this study.

1.1 Related Work

Although comprehensive definition of AI has always been a challenge between scientists, the reasoning is one of the most mentioned and agreed dimensions of AI (see [9], [10], [11], [12]). This dimension of AI is trended in most science fields, due to the availability of big and high-quality datasets [13], fast progress of natural language, image, and video processing systems (which could be assumed video as an ordered set of images [14], [15]). Furthermore, it motivates the most recent works in this area to target Visual data understanding and question answering [16], and natural language queries [17], [18]. For instance, a question-answering (QA) reasoning system is developed by [19] which was a combination of methods developed in previous works in QA reasoning systems. Further, they introduced an algorithm to select the best paths of commonsense knowledge, in order to get the whole inference required for QA. Although, the method works well for the specific applications, yet, it is not a general solution for the QA problem. Following this method, and together with systematic analysis of popular knowledge resources and the knowledge-integration methods, [20], stated a modeling solution “non-extractive commonsense QA”, which is more accurate but still does not cover all such problems. More recently, the “CoLlision Events for Video REpresentation and Reasoning” (CLEVRER) method was developed by [16], which is a reasoning system in video streams based on human intelligence. In this research, unlike most previous studies, “causal structures” are taken into account. Also, this model could answer four main varieties of questions, which are *descriptive* (e.g., “what color”), *explanatory* (“what’s responsible for”), *predictive* (“what will happen next”), and *counterfactual* (“what if”). Hereabouts, the method for the extraction of features from the video frames is ResNet-50 [21], which is a well-known object detection method. This method (CLEVRER) is one of the most accepted methods in such a field. Due to the characteristics of the reasoning systems together with the power of the temporal logic in formalizing natural language into formulas, this kind of model gets more interested in reasoning systems [22]. In a new perspective, the Metric Temporal Logic (MTL), which is an extension of the Linear Temporal Logic (LTL), was studied in [23] to handle stochastic state information. Considering the modal logic in such problems would lead to a new way of problem-solving because this approach seems suitable for such problems.

1.2 Motivation

This study aims to provide a flexible reasoning system, which could render the knowledge achieved by AI-based systems. Hither, we are going to develop a model to be applied to the existed classifiers, to translate the information attained from the results. Besides, our model could handle multi-knowledge flow in a multi-classifier scenario. Moreover, in the developed model, the obtained knowledge by classifiers will be aggregated in a defined and unified format. Accordingly, the analysis would be performed considering each scenario of the collaboration for such classifiers (i.e., the investigation of knowledge of each agent, a group of agents, knowledge which is distributed between agents, etc.). Herein, the verification of the investigated formulas will be proposed for such a statement using the definition of “verified formula” which is provided by [8]. Furthermore, a human language to the defined temporal formula translation approach is addressed, to get a more expressive model, and also to confirm the power of the adaption of such a model in real-world applications. Finally, a strategy for determining the probability of each state-transition is also suggested. These probabilities assist us to apply the method in a more optimized fashion. This final adjustment directs us to a “correction for time-series data” strategy for time-series object detection algorithms.

2 Problem Definition

In this section, we are going to define the problem that we will be solved in this study. There are many definitions for intelligent agents. One of the most common out of all is to develop human-like reasoning from input data. There are many approaches in which objects, referred to as knowledge, could be extracted from input data. But still, there is a lack of a generalized system to model to infer from gained knowledge. To reach that kind of intelligence from a classifier, we developed a model in the following steps:

1. Assume the classifier as an artificial neural network (ANN), which is the most common way for classifying approaches, this kind of classifier could not guarantee full accuracy. These classifiers are based on statistical

methods, and the accuracy is based on the architecture and training set. Even by the assumption of good architecture, if a new input data fed into an ANN, the network could not guarantee the correct output. To overcome this problem, [24] presented an approach to verify predefined properties in such models. This method leads these models to verify properties for each input. In the case that the property could not be verified, [8] developed a multi-agent epistemic logic model to find all possible outcomes concerning that property, by that multiple classifiers. By applying this method, the outcome of the classifier would be a set of possible outputs(i.e., possible worlds).

2. Generally, the set of knowledge is directly driven by classifiers. This set of knowledge contains the classes of found objects, yet from a real-world perspective, each detected object would contain more information (i.e., ontology rules). To reflect this, an intelligent agent should understand the category of the object together with sub-categories which are fed in as input rules. In this model, for each possible class of object, predefined inferences would be extracted in a unified and formal manner.
3. The previously developed model was exclusively designed for single-framed data, therefor the time-series data could not be considered. To bring them into consideration, we developed a combination of temporal logic transition systems [25] with the developed epistemic logic model. This leads us to extract all possible sequences (named execution path) of knowledge represented in time-series data.
4. Here, input-output would be in question-answering format. Accordingly, to get an outcome from the developed model, the model should provide an answer to respecting questions that are asked. To do this, first, all possible answers would be extracted due to the asked question. Then, using preexisted approaches, natural language statements will be translated into temporal formulas. Finally, each formula of every possible answer would be investigated in the developed temporal model and the satisfying formula reflects the answers to the question.
5. The verification in this transition system is also defined by modifying the verification definition in [24] and [8]. Accordingly, the situation of the answer would be determined (whether it is a verified answer, possible answer, verified for a single classifier or it needs some information to get a possible/verified answer).
6. The space state explosion, and following that, numerous execution paths could make the proposed method very time and space consuming in large models. To overcome that difficulty, an approach is developed, to find the most probable path and investigate the satisfaction on that path. This method for finding the most probable path could be lead us to a data-stream correction (if there are misclassifications in small fractions of such data), in semi-continuous data-streams.

Here, we are going to do the mentioned steps as an example of a video stream and classifiers. In this case, classifiers could determine objects appearing during the video, following by extracting all possible outcomes for each image known as data-frame. Then, by collecting all input rules of objects (i.e., an input rule could be “elephants” are “animals” so, in each possible situation an “elephant” detected, we infer that an “animal” is also detected) each possible outcome will be enhanced for each image in the video. Next, by placing the model of all possible situations for images in a hierarchical sequencing the transition system would be made. Consequently, we could investigate the satisfaction of formulas that are created by the translation of questions. After that, for the sake of the performance, we will calculate the probability of each transition relation, following that, finding the most probable execution path. In fig 1, a model for two-framed data is illustrated, in which the most probable path is $w_{0,0}w_{1,0}w_{2,0}w_{3,0}$. This path could lead us to correct the misclassifications of classifiers.

3 Public Announcement Logic

In this section, we are going to introduce a logical model for interpreting single-framed data. Following each step of the development of the model, an application of such a model was represented. The epistemic model is based on an extension of PAL that is mentioned in [8]. It will be applied to extract knowledge from a single-framed data (i.e., an image).

Let us, first introduce the syntax and semantics of the logic. The syntax of the language PAL is as follows, in BNF:

$$\phi ::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid K_i\phi \mid D_A\phi \mid [\phi]\phi.$$

Where p as a propositional variable (atomic formula) is a pair of the form (x, c) , in which x denotes the input data and c denotes the target class. We also notice that $K_i\phi$ read as “ i -th classifier knows ϕ ”. In other words, the i -th classifier was assured about the truth value of ϕ , so ϕ could be named a “robust” knowledge for the classifier i . Followed that, $D_A\phi$ read as “ ϕ is a distributed knowledge in a group A of classifiers”. The formula $D_A\phi$ holds exactly when, aggregation of knowledge of agents in the group A , satisfies ϕ . Here, the formula ϕ will be named a “robust” knowledge within group A . The robustness of formula ϕ ensures that the formula is verified by those classifiers [8]. Finally, the formula

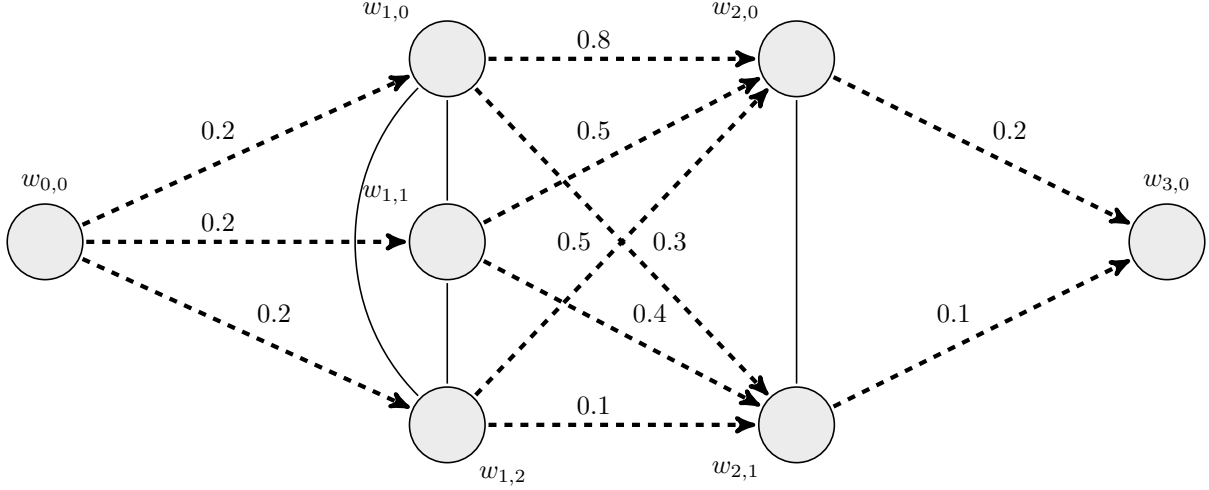


Figure 1: A combination of transition system and epistemic model for two-framed data. The probability of each transition is also determined. The most probable execution path out of all 36 possible ones is $w_{0,0}w_{1,0}w_{2,0}w_{3,0}$

$M_1 = (W_1, R_1, V_1)$, $W_1 = \{w_{1,0}, w_{1,1}, w_{1,2}\}$, $R_1 = \{(w_{1,0}, w_{1,1}), (w_{1,0}, w_{1,2}), (w_{1,1}, w_{1,2})\}$;

$M_2 = (W_2, R_2, V_2)$, $W_2 = \{w_{2,0}, w_{2,1}\}$, $R_2 = \{(w_{2,0}, w_{2,1})\}$;

$\mathcal{TS} = (S, R, s^0, s^{-1}, \rightarrow, L)$,

$S = w_{0,0} \cup W_1 \cup W_2 \cup w_{3,0}$, $s^0 = w_{0,0}$, $s^{-1} = w_{3,0}$,

$R = R_1 \cup R_2$,

$\rightarrow = \{(w_{0,0}, w_{1,0}), (w_{0,0}, w_{1,1}), (w_{0,0}, w_{1,2}), (w_{1,0}, w_{2,0}), (w_{1,0}, w_{2,1}), (w_{1,1}, w_{2,0}), (w_{1,1}, w_{2,1}), (w_{1,2}, w_{2,0}), (w_{1,2}, w_{2,1}), (w_{2,0}, w_{3,0}), (w_{2,1}, w_{3,0})\}$;

$[\psi]\phi$ read as “after a correct announcement of ψ , ϕ will hold”. By employing this operator, it could investigate that, which knowledge is significantly missing. In other words, by adding which information, the system could satisfy the inspecting formulas.

A PAL Kripke model is a tuple $M = (W, R_1, \dots, R_n, V)$, where $W = \{w_1, \dots, w_k\}$ is a set of worlds (hereabouts the set W is a representative for all possible output results of the input data), $R_i \subseteq W \times W$ is an equivalent relation between worlds for each classifier i in $\{1, \dots, n\}$. The intended meaning of $sR_i s'$ relation is, the worlds s and s' cannot be distinguished by the i -th classifier. And finally, $V : W \rightarrow 2^{Prop}$ is the evaluation function specified that the knowledge represented in any world, where $Prop$ is the set of all atoms. We extend the evaluation function V to all formulas as follows:

- $\mathcal{M}, w \models p$ iff $p \in V(w)$,
- $\mathcal{M}, w \models \neg\phi$ iff $\mathcal{M}, w \not\models \phi$,
- $\mathcal{M}, w \models \phi \wedge \psi$ iff $\mathcal{M}, w \models \phi$ and $\mathcal{M}, w \models \psi$,
- $\mathcal{M}, w \models K_i\phi$ iff $\forall v \in R_i(w), \mathcal{M}, v \models \phi$,
- $\mathcal{M}, w \models D_A\phi$ iff $\forall v \in R_A(w), \mathcal{M}, v \models \phi$, where $R_A := \bigcap_{i \in A} R_i$.
- $\mathcal{M}, w \models [\psi]\phi$ iff $\mathcal{M}, w \models \psi$ implies $\mathcal{M}^\psi, w \models \phi$.

The intended meaning of satisfaction of an atomic formula (x, c) in a world w is “class c has appeared as respective classifier’s output class for the input data x ”. We also mention that for every formula ϕ and ψ , $\phi \models \psi$ means that for every model M , and every point $w \in M$, $M, w \models \phi$ implies that $M, w \models \psi$.

In the following example, we applied the model on an image as an input.

Example. (Representation of an Image) Assume that, we have an image as input, with some objects on the picture. The classifiers are trained by the same sets of objects as an output class. Moreover, it is assumed that the classifiers are multi-object detectors, which means they could detect more than one object class in the output if they exist. The next assumption in this section is, the rules of relations between objects and their features are given (i.e. dog is an animal and has two eyes, four legs, etc.). There are many approaches for this purpose, for example, the rules or categories could be derived from object tags in a natural language processing (NLP) model (see [26], [27]).

Remark: 1. (Sub-features of a result class) Hereby, for each image, we have a Kripke model, in which, each word contains possible outcomes from the image, which are detected by classifiers. So, it can be written $M, w \models (x, c_1) \wedge \dots \wedge (x, c_k), P(w) = \{(x, c_1) \wedge \dots \wedge (x, c_k)\}$. Moreover, each output class has some other features that could be driven from it. For instance, a human knows that “cats” are animals, they have two eyes, four legs, etc. To be more formal, for each output class (x, c_i) , it could be written $(x, c_i) \models (x, c_{i_1}) \wedge \dots \wedge (x, c_{i_m})$, in which (x, c_{i_j}) is a concluded as a feature of (x, c_i) . Note that, it could be even more nested, in other words, $(x, c_{i_j}) \models (x, c_{i_{j_1}}) \wedge \dots \wedge (x, c_{i_{j_l}})$ and so on.

2. (Verification of a sub-features) Here, if there is only one possible world for the input image, the outcome is “robust” and this means that the output is verified concerning the predefined property [8]. In the case that there is more than one possible world represented for the image, it is also possible to find some other robust features. To introduce this kind of robustness, assume that two possible worlds w, w' are possible ones for the input image. The world $M, w \models (x, \text{“Chair”}) \wedge (x, \text{“Cat”})$ and $M, w' \models (x, \text{“Chair”}) \wedge (x, \text{“Dog”})$, we could say that $M \models (x, \text{“Chair”}) \wedge (x, \text{“Animal”})$ it represents that the image robust for “Chair” and “Animal”. It is derived from the fact $(x, \text{“Dog”}) \models (x, \text{“Animal”})$, $(x, \text{“Cat”}) \models (x, \text{“Animal”})$. This kind of verification will be more useful in the process of reasoning about questions. For illustration, assume that the question asks information about existing “animals” in the image and doesn’t care which animal showed in the image. So we could provide the verified answer about existing animals. As an illustration, algorithm 1 developed to aggregate all available knowledge, including the classifier’s knowledge and sub-features. The output of this algorithm would be a Kripke model. Moreover, in algorithm 2 satisfaction of PAL formulas will be determined.

Algorithm 1 The Single-Framed Knowledge Extraction (SFKE) function shall collect the knowledge, produced by classifiers, together with the aggregation of external subset rules, we also use MASKS function developed in [8].

Let \mathcal{C} as the set of classifiers, ζ as the set of rules, x_0 as the single framed input data, and η as the defined confident neighborhood

```

1: function SFKE( $\mathcal{C}, \zeta, x_0, \eta$ )
2:                                      $\triangleright V$  could be driven from the model  $\mathcal{M}$ 
3:    $\mathcal{R}, \mathcal{M} := \text{MASKS}(\mathcal{C}, x_0, \eta)$ 
4:   for all possible world  $w \in \mathcal{M}$  do
5:     for all atomic formula  $p \in w$  do
6:       Add all sub rules of  $p$  defined in  $\zeta$  to  $V(w)$ .
7:   return  $\mathcal{M}$ 

```

Algorithm 2 The PAL Satisfaction function (PALS) shall investigate the satisfaction of PAL formulas.

Let ϕ as the PAL formula, \mathcal{M} as the Kripke model, and w as a world

```

1: function PALS( $\phi, \mathcal{M}, w$ )
2:                                      $\triangleright R_i$  and  $V$  could be driven from the model  $\mathcal{M}$ 
3:   if  $\phi$  is an atomic formula then
4:     return  $\phi \in V(w)$ 
5:                                      $\triangleright$  This will be true if  $\phi \in V(w)$  and false otherwise
6:   if  $\phi$  is in form of  $\neg\psi$  then
7:     return  $\neg \text{PALS}(\psi, \mathcal{M}, w)$ 
8:   if  $\phi$  is in form of  $\psi_1 \wedge \psi_2$  then
9:     return  $\text{PALS}(\psi_1, \mathcal{M}, w) \wedge \text{PALS}(\psi_2, \mathcal{M}, w)$ 
10:  if  $\phi$  is in form of  $K_i\psi$  then
11:    return  $\bigwedge_{w'} \text{PALS}(\psi, \mathcal{M}, w'); \forall w' \in R_i(w)$ 
12:  if  $\phi$  is in form of  $D_A\psi$  then
13:     $R_A := \bigcap_{i \in A} R_i$ 
14:    return  $\bigwedge_{w'} \text{PALS}(\psi, \mathcal{M}, w'); \forall w' \in R_A(w)$ 
15:  return false

```

4 Linear Temporal Public Announcement Logic (LTPAL)

In this section to extract all possible sequences, named execution paths, of knowledge represented in time-series data, we introduce an LTL extension of PAL by the following grammar, in BNF:

$$\begin{aligned}\phi &::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid K_j\phi \mid D_A\phi \mid [\phi]\phi \\ \Phi &::= \phi \mid (\neg\Phi) \mid (\Phi \wedge \Phi) \mid X\Phi \mid [\Phi U \Phi]\end{aligned}$$

Here, the temporal operators are $X\Phi$ (in the neXt data-frame Φ must be true) and $[\Phi U \Psi]$ (Ψ must remain true U ntil Φ becomes true). To define the Kripke semantic of this logic, assume that $M_0 = (\{w_{0_0}\}, R_{0_0}, \dots, R_{0_k}, V_0)$,

$$M_1 = (W_1, R_{1_0}, \dots, R_{1_k}, V_1), \dots$$

$$, M_{n-1} = (W_{n-1}, R_{(n-1)_0}, \dots, R_{(n-1)_k}, V_{n-1})$$

$$, M_n = (\{w_{n_0}\}, R_{n_0}, \dots, R_{n_k}, V_n) \text{ are PAL models, where } W_i \text{'s are mutually disjoint sets, and } V_0(w_{0_0}) = V_n(w_{0_0}) = \emptyset.$$

We build a new model $\mathcal{TS} = (S, R, s^0, s^{-1}, \rightarrow, L)$, known as a transition system, in which $S = \bigcup_{i=0}^n W_i$ is the set of states, $R = \{R_{i_j} \mid 0 \leq i \leq n, 0 \leq j \leq k\}$, $s^0 = w_{0_0}$ and $s^{-1} = w_{n_0}$ are initial and final states, $\rightarrow = W_i \times W_{i+1}$, $0 \leq i < n$ are transition relations, and the labeling function $L : S \rightarrow 2^{Prop}$ which assigning propositional letters to states is defined by $L(w_{i_j}) = V_i(w_{i_j})$, for each $w_{i_j} \in W_i$.

An executive path $\pi_{\mathcal{TS}}^{i \dots j}$ in the model \mathcal{TS} is a sequence of worlds $w_i w_{i+1} \dots w_j$ where $w_i \rightarrow w_{i+1} \rightarrow \dots \rightarrow w_j$ for $w_k \in W_k$, $i \leq k \leq j$. An executive path $\pi_{\mathcal{TS}}^{i \dots j}$ is called total and denoted by $\pi_{\mathcal{TS}}$ if $w_i = s^0$ and $w_j = s^{-1}$. The set of all total executive paths will be denoted by $\Pi_{\mathcal{TS}}$. Next, we introduce the semantics for *LTPAL* as follows:

- $\mathcal{TS}, \pi_{\mathcal{TS}}^{i \dots j} \models \phi$ iff $\mathcal{TS}, w_i \models \phi$,
- $\mathcal{TS}, \pi_{\mathcal{TS}}^{i \dots j} \models X\Phi$ iff $\mathcal{TS}, \pi_{\mathcal{TS}}^{i+1 \dots j} \models \Phi$,
- $\mathcal{TS}, \pi_{\mathcal{TS}}^{i \dots j} \models [\Phi_1 U \Phi_2]$ iff there exist m , $i \leq m \leq j$, $\mathcal{TS}, \pi_{\mathcal{TS}}^{m \dots j} \models \Phi_2$, and for all k , $i \leq k < j$, we have $\mathcal{TS}, \pi_{\mathcal{TS}}^{i \dots k} \models \Phi_1$.

We notice that in the first clause above we look at the transition system \mathcal{TS} over PAL formulas simply as a Kripke model. Moreover, algorithm 3 is developed to create a transition system from PAL models.

By the function TEMS developed in algorithm 4, temporal formulas could be investigated. Other temporal operators could be driven from the two operators *next* ($X\Phi$), and *until* ($U\Phi$) in the following way:

- $F\Phi \equiv [\top U \Phi]$
- $[\Phi R \Psi] \equiv \neg[\neg\Phi U \Psi]$
- $[\Phi W \Psi] \equiv [\Psi R(\Phi \vee \Psi)]$
- $G\Phi \equiv [\perp R \Phi]$

The intended meaning of *future* ($F\Phi$) is “eventually Φ becomes true”, *global* ($G\Phi$) is “ Φ must remain true forever”, *release* ($[\Phi R \Psi]$) is “ Ψ remains true until and including when Φ becomes true, if Φ never becomes true, Ψ always remain true”, and *weak until* ($[\Phi W \Psi]$) is “ Φ has to remain true at least until Ψ ; if Ψ never holds, Φ must always remains true” [28], [29].

4.1 Formalizing Questions and answers in LTPAL

Easy interpretation of human language in modal logics (especially in LTL) is one of the most important strengths of such logics (see [30],[31], and [32]). This kind of interpretation could help robots to react to human orders [33]. In order to convert the question into a formula, first of all, we need to extract all possible answers to the question. Then after converting each possible answer to LTL formula using the existing -and abovementioned- approaches we will investigate the satisfactory of each answer. The developed model would lead us a PAL modifications on such LTL formulas. To explain that let $\Phi(p_1, \dots, p_\sigma)$ be an LTL translated formula and ϕ_i , $1 \leq i \leq \sigma$ is a PAL formula. Then $\Phi(\phi_1, \dots, \phi_\sigma)$ is obtained from ϕ by substituting each of p_i with ϕ_i , for all $1 \leq i \leq \sigma$, respectively. For instance, for $\Phi(p_1, p_2) = G[p_1 U X p_2]$, $\phi_1 = \neg D_A \neg p_1$, and $\phi_2 = K_i p_2$, we have $\Phi(\phi_1, \phi_2) = G[(\neg D_A \neg p_1) U (X(K_i p_2))]$, which, for the transition system \mathcal{TS} and the investigated execution path $\pi_{\mathcal{TS}}$, means that “ p_1 should always be a possible answer until in the world which right after, p_2 is a robust knowledge for the i -th agent”. By defining a way for

Algorithm 3 The Time-Series Transition System (TSTS) function shall create the transition system by the given input information.

Let \mathcal{C} as the set of classifiers, ζ as the set of rules, X as the time-series input data of size k , and η as the predefined confident neighborhood

```

1: function TSTS( $\mathcal{C}, \zeta, X, \eta$ )
2:
3:    $S := \{w_0\}$ 
4:    $W' := \{w_0\}$ 
5:    $R_1, \dots, R_n := \{(w_0, w_0)\}$ 
6:    $s^0 := w_0$ 
7:    $s^{-1} := w_0$ 
8:    $\rightarrow := \emptyset$ 
9:    $L(w_0) := \emptyset$ 
10:   $\mathcal{TS} := (S, R_1, \dots, R_n, s^0, s^{-1}, \rightarrow, L)$ 
11:  for all  $x_0$  in  $X$  do
12:     $\mathcal{M} := \text{SFKE}(\mathcal{C}, \zeta, x_0, \eta)$ 
13:
14:     $S := S \cup W$ 
15:     $R_1, \dots, R_n := R_1, \dots, R_n \cup R'_1, \dots, R'_n$ 
16:    for all  $w \in W$  and  $w' \in W'$  do
17:       $\rightarrow := \rightarrow \cup \{(w', w)\}$ 
18:     $W' := W$ 
19:     $W := \{w_{k+1}\}$ 
20:     $S := S \cup W$ 
21:     $R_1, \dots, R_n := R_1, \dots, R_n \cup \{(w_{k+1}, w_{k+1})\}$ 
22:     $s^{-1} := w_{k+1}$ 
23:    for all  $w \in W$  and  $w' \in W'$  do
24:       $\rightarrow := \rightarrow \cup \{(w', w)\}$ 
25:  return  $\mathcal{TS}$ 

```

$\triangleright \mathcal{M} = (W, R_1, \dots, R_n, V)$

\triangleright Kripke model $\mathcal{M} = (W, R'_1, \dots, R'_n, V)$

Algorithm 4 The TEMPoral Satisfaction function (TEMS) function shall investigate the satisfaction of LTL formulas.

Let Φ as the LTL formula, \mathcal{TS} as the transition system, and w as a world

```

1: function TEMS( $\Phi, \mathcal{TS}, \pi_{TS}^{i \dots j}$ )
2:
3:   if  $i > j$  then
4:     return false
5:   if  $\Phi$  is a PAL formula then
6:     return PALS( $\Phi, \mathcal{M}, w_i$ )
7:
8:   if  $\Phi$  is in form of  $\neg\Psi$  then
9:     return  $\neg$  TEMS( $\Psi, \mathcal{TS}, \pi_{TS}^{i \dots j}$ )
10:  if  $\Phi$  is in form of  $\Psi_1 \wedge \Psi_2$  then
11:    return TEMS( $\Psi_1, \mathcal{TS}, \pi_{TS}^{i \dots j}$ )  $\wedge$  TEMS( $\Psi_2, \mathcal{TS}, \pi_{TS}^{i \dots j}$ )
12:  if  $\Phi$  is in form of  $X\Psi$  then
13:    return TEMS( $\Psi, \mathcal{TS}, \pi_{TS}^{i+1 \dots j}$ )
14:  if  $\Phi$  is in form of  $[\Psi_1 U \Psi_2]$  then
15:    if TEMS( $\Psi_2, \mathcal{TS}, \pi_{TS}^{i \dots j}$ ) then
16:      return true
17:    while TEMS( $\Psi_1, \mathcal{TS}, \pi_{TS}^{i \dots j}$ ) do
18:       $i := i + 1$ 
19:      if TEMS( $\Psi_2, \mathcal{TS}, \pi_{TS}^{i \dots j}$ ) then
20:        return true
21:  return false

```

$\triangleright \rightarrow$ could be driven from the model \mathcal{TS}

\triangleright let w_i as first world in path $\pi_{TS}^{i \dots j}$

introducing robustness and possibility for a classifier, together with the definition of verified and possibility concerning a group of classifiers, answers could be labeled to assure the questioner, “how reliable is this answer”. This would lead us to design a more proper system for critical applications. Moreover, by capturing missing knowledge, it could be caught that which knowledge is mandatory for the system which is not discovered by classifiers or external, and which part of the system could be manipulated to obtain the missing information. In the following, we would like to introduce when a formula is verified, possible, robust or it is missing information in our model:

1. The formula $\Phi(p_1, \dots, p_\sigma)$ is *verified* for the group A of classifiers exactly when, for all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$ we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(D_{Ap_1}, \dots, D_{Ap_\sigma})$.
2. The formula $\Phi(p_1, \dots, p_\sigma)$ is a *possible* scenario for the group A of classifiers exactly when, there exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, in which we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(\neg D_A \neg p_1, \dots, \neg D_A \neg p_\sigma)$.
3. The formula $\Phi(p_1, \dots, p_\sigma)$ is *robust* with i -th agent’s perspective exactly when, for all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(K_i p_1, \dots, K_i p_\sigma)$.
4. The formula $\Phi(p_1, \dots, p_\sigma)$ is *possible* with i -th agent’s perspective exactly when, there exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, in which we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(\neg K_i \neg p_1, \dots, \neg K_i \neg p_\sigma)$.
5. A PAL formula ψ is called *verified-missing* information for a formula $\Phi(p_1, \dots, p_\sigma)$ in group A of classifiers exactly when, there exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, in which we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(D_{Ap_1}, \dots, D_{Ap_\sigma})$ and for all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi([\psi](D_{Ap_1}), \dots, [\psi](D_{Ap_\sigma}))$.
6. A PAL formula ψ is called *possible-missing* information for a formula $\Phi(p_1, \dots, p_\sigma)$ in group A of classifiers exactly when, for all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(\neg D_A \neg p_1, \dots, \neg D_A \neg p_\sigma)$ and there exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, in which we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi([\psi](\neg D_A \neg p_1), \dots, [\psi](\neg D_A \neg p_\sigma))$.
7. A PAL formula ψ is called *verified-missing* information for a formula $\Phi(p_1, \dots, p_\sigma)$ of i -th classifiers exactly when, there exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, in which we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(K_i p_1, \dots, K_i p_\sigma)$ and for all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi([\psi](K_i p_1), \dots, [\psi](K_i p_\sigma))$.
8. A PAL formula ψ is called *possible-missing* information for a formula $\Phi(p_1, \dots, p_\sigma)$ of i -th classifiers exactly when, for all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi(\neg K_i \neg p_1, \dots, \neg K_i \neg p_\sigma)$ and there exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, in which we have $\mathcal{T}S, \pi_{\mathcal{T}S} \models \Phi([\psi](\neg K_i \neg p_1), \dots, [\psi](\neg K_i \neg p_\sigma))$.

Example. (Representation of a Video) In the previous chapter, a PAL logic for representing image data was illustrated. In this example, assume that the given finite video stream contains $n - 1$ images, and models of epistemic information extraction for these images are assumed to be $M_1 = (W_1, R_{1_0}, \dots, R_{1_k}, V_1), \dots, M_{n-1} = (W_{n-1}, R_{(n-1)_0}, \dots, R_{(n-1)_k}, V_{n-1})$. By adding two dummy models M_0 and M_n the transition model $\mathcal{T}S = (S, R, s^0, s^{-1}, \rightarrow, L)$ could be created.

For clarification, we will describe a real-world situation: assume that, there is a medical *clean-room* in an Operating theater, which is monitored by a camera. The recorded video will be fed into a set of classifiers A , which are fairly accurate. The room will be cleaned with an air conditioner together with an ultra-violet (UV) led. As it is known that the UV led would damage humans, thus, it must be turned off while a person is in the room. On the opposite, the air conditioner should work during the appearance of a person in that room, it also should be stopped in an empty room for saving electricity power. So, the first protocol is to “shout down UV while any person monitored”, and the second one is “shout down the air conditioner when no one is in the room”. The classifiers should answer the vital question of “How much the protocols are robust?”.

Let p as “at least a human exists” and q as “the UV is on”, and r as “air conditioner is on”. The LTL formulas for the first question is:

- $G(p \Rightarrow X \neg q)$: which is the translation of “by observation of human, the UV should be shouted down”.
- For all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, $\mathcal{T}S, \pi_{\mathcal{T}S} \models G(D_{Ap} \Rightarrow X \langle D_A \neg q \rangle)$: by the satisfaction of this formula the satisfaction of the property would be verified within group A .
- There exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, $\mathcal{T}S, \pi_{\mathcal{T}S} \models G(\neg D_A \neg p \Rightarrow X \langle \neg D q \rangle)$: by the satisfaction of this formula, there is a scenario in which the formula holds. It means that this property is possible within group A .
- For all $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, $\mathcal{T}S, \pi_{\mathcal{T}S} \models G(K_i p \Rightarrow X \langle K_i \neg q \rangle)$: by the satisfaction of this formula, this property would be robust for the i -th classifier.
- There exists $\pi_{\mathcal{T}S} \in \Pi_{\mathcal{T}S}$, $\mathcal{T}S, \pi_{\mathcal{T}S} \models G(\neg K_i \neg p \Rightarrow X \langle \neg K_i q \rangle)$: by the satisfaction of this formula, the i -th classifier, consider this property as a possible one.

More handily, for all $\pi_{\mathcal{TS}} \in \Pi_{\mathcal{TS}}$, $TS, \pi_{\mathcal{TS}} \models G(\neg D_A \neg p \Rightarrow X(D_A \neg q))$ will ensure the system in any possible situation, the possible existence of anybody should enforce UV-led to turned off. Accordingly, this system leads us to investigate whether protocols are followed or not. Besides, it can be observed that any error by the system is caused by which classifier.

5 Finding most probable paths in a semi-continuous stream

Traversing all paths in the developed transition system could be very time-consuming, concerning the size of PAL models. Fortunately, the computation could be reduced in special conditions to reach an applicable approach. In particular, and for instance, image frames of video streams are strongly correlated with the next and previous image frames [34]. In other words, it could be assumed, the video stream as semi-continues data and the overall changes between frames should be minimal. It also could be assumed as a Markov-chain memoryless data flow, which means every data frame is just related to its previous data frame. Using this property of video streams together with recent technology of NLP the most probable path could be determined. Here, we will calculate the changes by the label of objects represented in frames and finding the most probable path among all possible paths. As it mentioned before, the set of paths $\Pi_{\mathcal{TS}}$ in the transition system \mathcal{TS} are created by worlds of $n + 1$ models M_0, \dots, M_n , in which relations are between two successive worlds of models M_i and $M_{(i+1)}$. To find out the similarities and differences between two worlds, first, we extract word-tags of true atomic formulas in each world, and then we will find the probability of each relation by NLP approaches, by finding the tags' similarities in both worlds (see [35], [36], [37]). After finding the similarity score of each transition relation, the most possible path could be found in time complexity $O(|\rightarrow|)$ by following a greedy algorithm. To introduce the approach, we assume the transition system was made of $n + 1$ PAL models $\mathcal{M}_i = (W_i, R_{i1}, \dots, R_{in}, V)$, $0 \leq i \leq n$. Let $\min \pi_{\mathcal{TS}}^{0 \dots (i-1)} \rightarrow w_{i_j}$ be the best-found path of length $i + 1$, starting at the initial state s^0 to the world w_{i_j} , and also let $\mathcal{P}_{\rightarrow_{j,k}^i}$ be the represented probability score between worlds w_{i_j} and $w_{(i+1)_k}$ (which is calculated similarity between tags in worlds w_{i_j} and $w_{(i+1)_k}$ by the NLP algorithm, i.e. Siamese [37]). The possibility score of the most probable path from s^0 to $w_{(i+1)_k}$ is $\mathcal{P}_{\min \pi_{\mathcal{TS}}^{0 \dots i} \rightarrow w_{(i+1)_k}} = \min_j (\mathcal{P}_{\min \pi_{\mathcal{TS}}^{0 \dots (i-1)} \rightarrow w_{i_j}} \times \mathcal{P}_{\rightarrow_{j,k}^i})$ and the path would be $\min \pi_{\mathcal{TS}}^{0 \dots i} \rightarrow w_{(i+1)_k} \equiv \min \pi_{\mathcal{TS}}^{0 \dots (i-1)} \rightarrow w_{i_j} \rightarrow w_{(i+1)_k}$. By continuing this approach to the state s^{-1} the best path would be converged into one and the most possible path would be found. This path could be applied in many real-world applications, i.e., a frequent mistake in the video stream object detection systems is misclassifications in some special frames, because of the exceptional situation of the object. Employing this algorithm would lead to getting rid of the partial misclassifications. Moreover, the above mentioned epistemic formula could be applied in the most probable path for reducing the time of computations. The algorithm 5 is also developed to find the most probable path among the transition system.

Algorithm 5 The Most Probable Path Extraction (MPPE) function shall collect the most possible path of a data-stream. The similarity score of tags in the 3rd line could be found with any arbitrary NLP algorithms.

Let \mathcal{TS} as a transition system

```

1: function MPPE( $\mathcal{TS}$ )
2:   for all  $(w_i, w_{i+1}) \in \rightarrow$  do
3:      $\mathcal{P}_{(w_i, w_{i+1})} :=$  similarity score of tags in  $w_i$  and  $w_{i+1}$ 
4:   for  $index \in \{1, \dots, \text{length}(\pi_{\mathcal{TS}})\}$  do
5:     for all  $w_i \in W_i$  do
6:        $\min P_{w_i} := 0$ 
7:        $\min \pi_{w_i} := \emptyset$ 
8:       for all  $\pi_{\mathcal{TS}}^{0 \dots i} = \min \pi_{\mathcal{TS}}^{0 \dots i-1} \rightarrow w_i$  do
9:         if  $\mathcal{P}_{\pi_{\mathcal{TS}}^{0 \dots i}} > \min P_{w_i}$  then
10:            $\min P_{w_i} := \mathcal{P}_{\pi_{\mathcal{TS}}^{0 \dots i}}$ 
11:            $\min \pi_{w_i} := \pi_{\mathcal{TS}}^{0 \dots i}$ 
12:   return  $\min \pi_{s^{-1}}$ 

```

6 Conclusions

The statistical nature of most common classification algorithms, together with reaching a high accuracy in most applied applications (i.e. classification, NLP, etc.) made them so popular, but unfortunately, the reasoning about answers seems so complicated. Accordingly, the logical point of view in classification problems to extract information and

reasoning should be get noticed in critical cases. In the introduced approach, first, a straight way is represented to aggregate knowledge acquired by any classification or by any input rules, in a unified manner. Next, by introducing an LTL-based transition system, the flow of knowledge in time-series data is modeled. Later, to answer any question, all possible answers will be converted to LTL formulas. Such formulas could be investigated in the developed model, so, satisfied formulas could be reported as answers. The reliability of answers could also be investigated using defined robustness and verification concepts. Moreover, it is possible to catch missing information that could lead the system to the answer, for any level of reliability. The captured missing information could lead us to find out classifiers’ shortages of information and the cause of the wrong decision makings. In the final part, an approach was developed to find the most probable path in order to reduce the state-space in larger models.

References

- [1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.
- [2] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016.
- [3] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.
- [4] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2758–2766, 2017.
- [5] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.
- [6] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8807–8817, 2019.
- [7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [8] Amirhoshang Hoseinpour Dehkordi, Majid Alizadeh, Ebrahim Ardeshtir-Larijani, and Ali Movaghar. MASKS: A Multi-Artificial Neural Networks System’s verification approach. 7 2020.
- [9] Patrick H Winston. Artificial intelligence 3rd edition. *Addison-Wesley, Reading, MA*, 34:167–339, 1992.
- [10] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.
- [11] Nils J Nilsson. *The quest for artificial intelligence*. Cambridge University Press, 2009.
- [12] Robert P Goldman and Eugene Charniak. Probabilistic text understanding. *Statistics and Computing*, 2(2):105–114, 1992.
- [13] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.
- [14] Chen Sun and Ram Nevatia. Active: Activity concept transitions in video event classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 913–920, 2013.
- [15] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1250–1257. IEEE, 2012.
- [16] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.
- [17] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.
- [18] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

- [19] Lisa Bauer, Yicheng Wang, and Mohit Bansal. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.
- [20] Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv preprint arXiv:1910.14087*, 2019.
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [22] Brendan Fong, Alberto Speranzon, and David I Spivak. Temporal landscapes: A graphical temporal logic for reasoning. *arXiv preprint arXiv:1904.01081*, 2019.
- [23] Daniel de Leng and Fredrik Heintz. Approximate stream reasoning with metric temporal logic under uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2760–2767, 2019.
- [24] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In Rupak Majumdar and Viktor Kuncak, editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24–28, 2017, Proceedings, Part I*, volume 10426 of *Lecture Notes in Computer Science*, pages 3–29. Springer, 2017.
- [25] Rob Gerth, Doron Peled, Moshe Y Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *International Conference on Protocol Specification, Testing and Verification*, pages 3–18. Springer, 1995.
- [26] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*, 2019.
- [27] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.
- [28] Akash Hossain and François Laroussinie. From quantified ctl to qbf. *arXiv preprint arXiv:1906.10005*, 2019.
- [29] Marta Kwiatkowska, Alessio Lomuscio, and Hongyang Qu. Parallel model checking for temporal epistemic logic. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 543–548, 2010.
- [30] Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. Translating structured english to robot controllers. *Advanced Robotics*, 22(12):1343–1359, 2008.
- [31] Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *2009 IEEE International Conference on Robotics and Automation*, pages 4163–4168. IEEE, 2009.
- [32] Rani Nelken and Nissim Francez. Automatic translation of natural language system specifications into temporal logic. In *International Conference on Computer Aided Verification*, pages 360–371. Springer, 1996.
- [33] Cameron Finucane, Gangyuan Jing, and Hadas Kress-Gazit. Ltlop: Experimenting with language, temporal logic and robot control. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1988–1993. IEEE, 2010.
- [34] Jawadul H Bappy, Sujoy Paul, Ertem Tuncel, and Amit K Roy-Chowdhury. Exploiting typicality for selecting informative and anomalous samples in videos. *IEEE Transactions on Image Processing*, 28(10):5214–5226, 2019.
- [35] Wei Yang, Wei Lu, and Vincent W Zheng. A simple regularization-based algorithm for learning cross-domain word embeddings. *arXiv preprint arXiv:1902.00184*, 2019.
- [36] Atish Pawar and Vijay Mago. Calculating the similarity between words and sentences using a lexical database and corpus statistics. *arXiv preprint arXiv:1802.05667*, 2018.
- [37] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *thirtieth AAAI conference on artificial intelligence*, 2016.