A Probabilistic Abduction Engine for Media Interpretation based on Ontologies

Oliver Gries, Ralf Möller, Anahita Nafissi, Maurice Rosenfeld, Kamil Sokolski, Michael Wessel

Hamburg University of Technology, Germany

Abstract. We propose an abduction-based formalism that uses description logics for the ontology and Horn rules for defining the space of hypotheses for explanations, and we use Markov logic to define the motivation for the agent to *generate* explanations on the one hand, and for *ranking* different explanations on the other. The formalism is applied to media interpretation problems in a agent-oriented scenario.[1]

1 Introduction

For multimedia interpretation in the context of an agent-based scenario, and for the combined interpretation of information coming from different modalities in particular, a semantically well-founded formalization is required. Low-level percepts, which are represented symbolically, define the observations of an agent w.r.t. some content, and interpretations of the content are defined as explanations for the observations. In [1] we have proposed an abduction-based formalism that uses description logics for the ontology and Horn rules for defining the space of hypotheses for explanations (i.e., the space of possible interpretations of media content). An evaluation of the abduction approach based on description logics and rules is presented in [3]. A discussion of related work can be found in [4].

In this paper, we propose the use of a probabilistic logic to define the motivation for the agent to *generate* explanations on the one hand, and for *ranking* different explanations on the other. Furthermore, we discuss how the interpretation process is performed, possibly with uncertainty and inconsistency in the input data. We also introduce a new approach for ranking interpretation Aboxes. The explanationranking process is performed based on a probabilistic scoring function (as opposed to the proof-theoretic scoring function used in [3]). A termination condition is also defined which determines how long the interpretation process should be performed. The approach is evaluation using a detailed example.

Due to space restrictions, not all preliminaries could be specified in this paper. For an introduction to description logics, grounded conjunctive queries, and rules we refer to [3]. For specifying the ontology used to describe low-level analysis results as well as high-level interpretation results, a less expressive description logic is applied to facilitate fast computations. We decided to represent the domain knowledge with the DL $\mathcal{ALH}_f{}^-$ (restricted attributive concept language with role hierarchies, functional roles and concrete domains). The motivation to only allow a restricted use of existential restrictions is to support a well-founded integration of the description logic part of the knowledge base with the probabilistic part, based on Markov logics.

The Markov logic formalism [2] provides a means to combine the expressivity of first-order logic augmented with the formalism of Markov networks [6]. The Markov logic formalism uses first-order logic to define "templates" for constructing Markov networks. The basic notion for this is called a Markov logic network [2].

A Markov logic network $MLN = (\mathcal{F}_{MLN}, \mathcal{W}_{MLN})$ consists of a sequence of first-order formulas $\mathcal{F}_{MLN} = \langle F_1, ..., F_m \rangle$ and a sequence of real number weights $\mathcal{W}_{MLN} = \langle w_1, ..., w_m \rangle$. The association of a formula to its weight is by position in the sequence. For a formula $F \in \mathcal{F}_{MLN}$ with associated weight $w \in \mathcal{W}_{MLN}$ we also write $w\,F$ (weighted formula). Weights are used to specify probability

[1] This work has been funded by the European Community with the project CASAM (Contract FP7-217061 CASAM) and by the German Science Foundation with the project PRESINT (DFG MO 801/1-1).

distributions. For a more detailed introduction to description logics and their combination with Markov logic networks, we refer to [5].

The central idea is to use abduction to compute possible explanations for observations of an agent, which are seen as high-level interpretations. The space of abducibles is defined in terms of Horn rules in combination with ontologies (see [3] for details). Compared to other approaches (e.g., [7]) also the ontology is used for checking whether something must be abduced. In addition, a motivation for computing explanations (or interpretations) using abduction is given by assuming that the agent would like to increase increase the probability that the observations are true. If there is no significant increase (due to a threshold ϵ), possible interpretations are considered as irrelevant for the agent.[2] Another important idea is that, given a "current" interpretation, the agent should be able to compute what must be added due to new percepts and what must be retracted (for this purpose, an Abox difference operator is used).

The abduction and interpretation procedures are discussed in detail in Section 2. In Section 3, a complete example is given showing the main approach using intermediate steps. Section 4 summarizes this paper.

2 Probabilistic Interpretation Engine

At the beginning of this section, the most important preliminaries to the abduction process are specified. Afterwards, functions are introduced for the abduction procedure, interpretation procedure, and the media interpretation agent.

2.1 Computing Explanations

In general, abduction is formalized as $\Sigma \cup \Delta \models_{\mathcal{R}} \Gamma$ where background knowledge (Σ), rules ($\mathcal{R}$), and observations (Γ) are given, and explanations (Δ) are to be computed. In terms of DLs, Δ and Γ are Aboxes and Σ is a pair of Tbox and Abox. Abox abduction is implemented as a non-standard retrieval inference service in DLs. In contrast to standard retrieval inference services where answers are found by exploiting the ontology, Abox abduction has the task of acquiring what should be added to the knowledge base in order to answer a query. Therefore, the result of Abox abduction is a set of hypothesized Abox assertions. To achieve this, the space of abducibles has to be defined. We do this in terms of rules. We assume that a set of rules $\mathcal{R}$ as defined in [3] are specified, and use a function *explanation_step*, see [3] or [5] for details.

2.2 The Abduction Procedure

In the following, we devise an abstract computational engine for "explaining" Abox assertions in terms of a given set of rules. Explanation of Abox assertions w.r.t. a set of rules is meant in the sense that using the rules some high-level explanations are constructed such that the Abox assertions are entailed. The explanation of an Abox is again an Abox. For instance, the output Abox represents results of the content interpretation process. Let the agenda $\mathfrak{A}$ be a set of Aboxes Γ and let Γ be an Abox of observations whose assertions are to be explained. The goal of the explanation process is to use a set of rules $\mathcal{R}$ to derive "explanations" for elements in Γ. The explanation algorithm implemented in the Conceptual Abduction Engine (CAE) works on a set of Aboxes $\mathfrak{I}$.

The complete explanation process is implemented by the CAE function:

[2] Obviously, there is a horizon problem, which we neglect for the time being.

Function CAE(Ω, Ξ, Σ, $\mathcal{R}$, S, $\mathfrak{A}$):
Input: a strategy function Ω, a termination function Ξ, a knowledge base Σ, a set of rules $\mathcal{R}$, a scoring function S, and an agenda $\mathfrak{A}$
Output: a set of interpretation Aboxes $\mathfrak{I}'$
$\mathfrak{I}' := \{assign\_level(l, \mathfrak{A})\}$;
repeat
  $\mathfrak{I} := \mathfrak{I}'$;
  $(\mathcal{A}, \alpha) := \Omega(\mathfrak{I})$;
  $l = l + 1$;
  $\mathfrak{I}' := (\mathfrak{A} \setminus \{\mathcal{A}\}) \cup assign\_level(l, explanation\_step(\Sigma, \mathcal{R}, S, \mathcal{A}, \alpha))$;
until $\Xi(\mathfrak{I})$ *or no* $\mathcal{A}$ *and* α *can be selected such that* $\mathfrak{I}' \neq \mathfrak{I}$;
return $\mathfrak{I}'$

where $assign\_level(l, \mathfrak{A})$ is defined as follows:

$$assign\_level(l, \mathfrak{A}) = map(\lambda(\mathcal{A}) \bullet assign\_level(l, \mathcal{A}), \mathfrak{A}) \tag{1}$$

$assign\_level(l, \mathfrak{A})$ takes as input a superscript l and an agenda $\mathfrak{A}$. In the following, $assign\_level(l, \mathcal{A})$ is defined which superscripts each assertion α of the Abox $\mathcal{A}$ with l if the assertion α does not already have a superscript:

$$assign\_level(l, \mathcal{A}) = \left\{\alpha^l \mid \alpha \in \mathcal{A}, \alpha \neq \beta^i, i \in \mathbb{N}\right\} \tag{2}$$

The motivation for adding levels to assertions is to support different strategies Ω. Note that l is a global variable, its starting value is zero, and it is incremented in the CAE function. The map[3] function is defined as follows:

$$map(f, X) = \bigcup_{x \in X} \{f(x)\} \tag{3}$$

It takes as parameters a function f and a set X and returns a set consisting of the values of f applied to every element x of X. The CAE function applies the strategy function Ω in order to decide which assertions to explain, uses a termination function Ξ in order to check whether to terminate due to resource constraints and a scoring function S to valuate an explanation. The function Ω for the explanation strategy and Ξ for the termination condition are used as an oracle and must be defined in an application-specific way.

In the next Section we explain how probabilistic knowledge is used to (i) formalize the effect of the "explanation", and (ii) formalize the scoring function S used in the CAE algorithm explained above. In addition, it is shown how the termination condition (represented with the parameter Ξ in the above procedure) can be defined based on the probabilistic conditions.

2.3 The Interpretation Procedure

The interpretation procedure is completely discussed in this section by explaining the interpretation problem and presenting a solution to this problem. The solution is presented by a probabilistic interpretation algorithm which calls the CAE function described in the previous section. In the given algorithm, a termination function, and a scoring function are defined. The termination function determines if the interpretation process can be stopped since at some point during the interpretation process it makes no sense to continue the process. The reason for stopping the interpretation process is that no significant changes can be seen in the results. The defined scoring function in this section assigns probabilistic scores to the interpretation Aboxes.

[3] Please note that in this report, the expression *map* is used in two different contexts. The first one MAP denotes the Maximum A Posteriori approach which is a sampling method whereas the second one *map* is a function used in the $assign\_level(l, \mathfrak{A})$ function.

Problem The objective of the interpretation component is the generation of interpretations for the observations. An interpretation is an Abox which contains high level concept assertions. Since in this paper we adopt the view that agents are used for solving the problems while acquiring information, in the following the same problem is formalized in the perspective of an agent: Consider an agent given some percepts in an environment where the percepts are the analysis results of the multimedia documents.[4] The objective of this agent is finding explanations for the existence of percepts. The question is how the interpretation Aboxes are determined and how long the interpretation process must be performed by the agent. The functionality of this Media Interpretation Agent is presented in the *MI_Agent* algorithm in Section 2.4.

Solution In the following, an application for a probabilistic interpretation algorithm is presented which gives a solution to the mentioned problem. This solution illustrates a new perspective to the interpretation process and the reason why it is performed. In this approach, we define a probabilistic scoring function which assigns probabilities to the interpretation Aboxes. Additionally, we define a termination function which determines whether the interpretation process can be terminated. The central idea is to check whether interpretation results computed by a call to CAE substantially increase the probability the the observations are true. If there is no significant increase (due to a threshold ϵ, possible interpretations are considered as irrelevant for the agent.[5] Another important idea is that, given a "current" interpretation, the agent should be able to compute what must be added due to new percepts and what must be retracted (for this purpose, an Abox difference operator is used).

We are now ready to define the algorithm. Assume that the media interpretation component receives a weighted Abox $\mathcal{A}$ which contains observations. In the following, the applied operation $P(\mathcal{A}, \mathcal{A}', \mathcal{R}, \mathcal{WR}, \mathcal{T})$ in the algorithm is explained:

The $P(\mathcal{A}, \mathcal{A}', \mathcal{R}, \mathcal{WR}, \mathcal{T})$ function determines the probability of the Abox $\mathcal{A}$ with respect to the Abox $\mathcal{A}'$, a set of rules $\mathcal{R}$, a set of weighted rules $\mathcal{WR}$, and the Tbox $\mathcal{T}$ where $\mathcal{A} \subseteq \mathcal{A}'$. Note that $\mathcal{R}$ is a set of forward and backward chaining rules. The probability determination is performed based on the Markov logic formalism as follows:

$$P(\mathcal{A}, \mathcal{A}', \mathcal{R}, \mathcal{WR}, \mathcal{T}) = P_{MLN(\mathcal{A},\mathcal{A}',\mathcal{R},\mathcal{WR},\mathcal{T})}(\vec{Q}(\mathcal{A}) \mid \vec{e}(\mathcal{A}')) \tag{4}$$

$\vec{Q}(\mathcal{A})$ denotes an event composed of the conjunction of all assertions which appear in the Abox $\mathcal{A}$. Assume that the Abox $\mathcal{A}$ contains n assertions $\alpha_1, \ldots, \alpha_n$. Consequently, the event for the Abox $\mathcal{A}$ is defined as follows:

$$\vec{Q}(\mathcal{A}) = \langle \alpha_1 = true \wedge \ldots \wedge \alpha_n = true \rangle \tag{5}$$

Consider Abox $\mathcal{A}$ contains m assertions $\alpha_1, \ldots, \alpha_m$. Then, the evidence vector $\vec{e}(\mathcal{A})$ is defined by:

$$\vec{e}(\mathcal{A}) = \langle \alpha_1 = true, \ldots, \alpha_m = true \rangle \tag{6}$$

Note that $\alpha_1, \ldots, \alpha_n$ denote the boolean random variables of the MLN. In order to answer the query $P_{MLN(\mathcal{A},\mathcal{A}',\mathcal{R},\mathcal{WR},\mathcal{T})}(\vec{Q}(\mathcal{A}) \mid \vec{e}(\mathcal{A}'))$ the function $MLN(\mathcal{A}, \mathcal{A}', \mathcal{R}, \mathcal{WR}, \mathcal{T})$ is called. This function returns a Markov logic network $MLN = (\mathcal{F}_{MLN}, \mathcal{W}_{MLN})$ where $\mathcal{F}_{MLN}$ and $\mathcal{W}_{MLN}$ are ordered sets initialized as follows: $\mathcal{F}_{MLN} = \emptyset$ and $\mathcal{W}_{MLN} = \emptyset$. In the following, it is described how the MLN is built based on the Aboxes $\mathcal{A}$ and $\mathcal{A}'$, the rules $\mathcal{R}$ and $\mathcal{WR}$ and the Tbox $\mathcal{T}$:[6]

$$MLN(\mathcal{A}, \mathcal{A}', \mathcal{R}, \mathcal{WR}, \mathcal{T}) = \begin{cases} \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{\alpha\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{w\} & \text{if } w\,\alpha \in \mathcal{A} \\ \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{\alpha\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{\infty\} & \text{if } \alpha \in \mathcal{A} \\ \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{\alpha\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{w\} & \text{if } w\,\alpha \in \mathcal{A}' \\ \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{\alpha\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{\infty\} & \text{if } \alpha \in \mathcal{A}' \\ \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{\alpha\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{\infty\} & \text{if } \alpha \in \mathcal{R} \\ \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{\alpha\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{w\} & \text{if } w\,\alpha \in \mathcal{WR} \\ \mathcal{F}_{MLN} = \mathcal{F}_{MLN} \cup \{FOL(\alpha)\}; \quad \mathcal{W}_{MLN} = \mathcal{W}_{MLN} \cup \{\infty\} & \text{if } \alpha \in \mathcal{T} \end{cases}$$

[4] The analysis might also be carried out by the agent.

[5] Obviously, there is a horizon problem, which we neglect for the time being.

[6] $FOL(\phi)$ represents the GCI ϕ in first-order notation.

where w and α denote a weight and an assertion, respectively. In the following, the interpretation algorithm *Interpret* is presented:

Function Interpret($\mathfrak{A}$, $CurrentI$, Γ, $\mathcal{T}$, $\mathcal{FR}$, $\mathcal{BR}$, $\mathcal{WR}$, ϵ)
Input: an agenda $\mathfrak{A}$, a current interpretation Abox $CurrentI$, an Abox of observations Γ, a Tbox $\mathcal{T}$, a set of forward chaining rules $\mathcal{FR}$, a set of backward chaining rules $\mathcal{BR}$, a set of weighted rules $\mathcal{WR}$, and the desired explanation significance threshold ϵ
Output: an agenda $\mathfrak{A}'$, a new interpretation Abox $NewI$, and Abox differences for additions Δ_1 and omissions Δ_2
$i := 0$;
$\mathcal{R} := \mathcal{FR} \cup \mathcal{BR}$;
$p_0 := P(\Gamma, \Gamma, \mathcal{R}, \mathcal{WR}, \mathcal{T})$;
$\Xi := \lambda(\mathfrak{A}) \bullet \left\{ i := i + 1; p_i := \max_{\mathcal{A} \in \mathfrak{A}} P(\Gamma, \mathcal{A} \cup \mathcal{A}_0, \mathcal{R}, \mathcal{WR}, \mathcal{T}); \textbf{return} \mid p_i - p_{i-1} \mid < \frac{\epsilon}{i} \right\}$;
$\Sigma := (\mathcal{T}, \emptyset)$;
$S := \lambda((\mathcal{T}, \mathcal{A}_0)), \mathcal{R}, \mathcal{A}, \Delta) \bullet P(\Gamma, \mathcal{A} \cup \mathcal{A}_0 \cup \Delta, \mathcal{R}, \mathcal{WR}, \mathcal{T})$;
$\mathfrak{A}' := CAE(\Omega, \Xi, \Sigma, \mathcal{R}, S, \mathfrak{A})$;
$NewI = argmax_{\mathcal{A} \in \mathfrak{A}'}(P(\Gamma, \mathcal{A}, \mathcal{R}, \mathcal{WR}, \mathcal{T}))$;
$\Delta^+ = AboxDiff(NewI, CurrentI)$; `// additions`
$\Delta^- = AboxDiff(CurrentI, NewI)$; `// omissions`
return $(\mathfrak{A}', NewI, \Delta^+, \Delta^-)$;

In the above algorithm, the termination function Ξ and the scoring function S are defined by lambda calculus terms. The termination condition Ξ of the algorithm is that no significant changes can be seen in the successive probabilities p_i and p_{i-1} (scores) of the two successive generated interpretation Aboxes in two successive levels $i-1$ and i. In this case, the current interpretation Abox $CurrentI$ is preferred to the new interpretation Abox $NewI$. The CAE function is called which returns the agenda $\mathfrak{A}'$. Afterwards, the interpretation Abox $NewI$ with the maximum score among the Aboxes $\mathcal{A}$ of $\mathfrak{A}'$ is selected. Additionally, the Abox differences Δ^+ and Δ^-, respectively, for additions and omissions among the interpretation Aboxes $CurrentI$ and $NewI$ are computed. In this paper, we formalize AboxDiff as set difference, knowing that a semantic approach would be desirable (see [5] for a semantics-based definition of *AboxDiff*).

In the following, the strategy condition Ω is defined which is one of the parameters of the CAE function:

Function $\Omega(\mathfrak{I})$
Input: a set of interpretation Aboxes $\mathfrak{I}$
Output: an Abox $\mathcal{A}$ and a fiat assertion α
$\mathfrak{A} := \left\{ \mathcal{A} \in \mathfrak{I} \mid \neg \exists \mathcal{A}' \in \mathfrak{I}, \mathcal{A}' \neq \mathcal{A} : \exists {\alpha'}^{l'} \in \mathcal{A}' : \forall \alpha^{l} \in \mathcal{A} : l' < l \right\}$;
$\mathcal{A} := random\_select(\mathfrak{A})$;
$min\_\alpha s = \left\{ \alpha^l \in \mathcal{A} \mid \neg \exists {\alpha'}^{l'} \in \mathcal{A}', {\alpha'}^{l'} \neq \alpha^l, l' < l \right\}$;
return $(\mathcal{A}, random\_select(min\_\alpha s))$;

In the above strategy function Ω, the agenda $\mathfrak{A}$ is a set of Aboxes $\mathcal{A}$ such that the assigned superscripts to their assertions are minimum. In the next step, an Abox $\mathcal{A}$ from $\mathfrak{A}$ is randomly selected. Afterwards, the min_α_s set is determined which contains the assertions α from $\mathcal{A}$ whose superscripts are minimal. These are the assertions which require explanations. The strategy function returns as output an Abox $\mathcal{A}$ and an assertion α which requires explanation.

2.4 The Media Interpretation Agent

In the following, the MI_Agent function is presented which calls the *Interpret* function:

Function MI_Agent(Q, $Partners$, Die, $(\mathcal{T}, \mathcal{A}_0), \mathcal{FR}, \mathcal{BR}, \mathcal{WR}, \epsilon$)
Input: a queue of percept results Q, a set of partners $Partners$, a function Die, a background knowledge base $(\mathcal{T}, \mathcal{A}_0)$, a set of forward chaining rules $\mathcal{FR}$, a set of backward chaining rules $\mathcal{BR}$, a set of weighted rules $\mathcal{WR}$, and the desired precision of the results ϵ
Output: –
$CurrentI = \emptyset$;
$\mathfrak{A}'' = \{\emptyset\}$;
repeat
  $\Gamma := extractObservations(Q)$;
  $W := MAP(\Gamma, \mathcal{WR}, \mathcal{T})$;
  $\Gamma' := select(W, \Gamma)$;
  $\mathfrak{A}' := filter(\lambda(\mathcal{A}) \bullet consistent_{\Sigma}(\mathcal{A}),$
    $map(\lambda(\mathcal{A}) \bullet \Gamma' \cup \mathcal{A} \cup \mathcal{A}_0 \cup forward\_chain(\Sigma, \mathcal{FR}, \Gamma' \cup \mathcal{A} \cup \mathcal{A}_0),$
      $\{select(MAP(\Gamma' \cup \mathcal{A} \cup \mathcal{A}_0, \mathcal{WR}, \mathcal{T}), \Gamma' \cup \mathcal{A} \cup \mathcal{A}_0) \mid \mathcal{A} \in \mathfrak{A}''\}))$;
  $(\mathfrak{A}'', NewI, \Delta^+, \Delta^-) := Interpret(\mathfrak{A}', CurrentI, \Gamma', \mathcal{T}, \mathcal{FR}, \mathcal{BR}, \mathcal{WR} \cup \Gamma, \epsilon)$;
  $CurrentI := NewI$;
  $Communicate(\Delta^+, \Delta^-, Partners)$;
  $\mathfrak{A}'' := manage\_agenda(\mathfrak{A}'')$;
until $Die()$;

The body of MI_Agent uses a set of auxiliary functions, which are defined as follows.

$$filter(f, X) = \bigcup_{x \in X} \begin{cases} \{x\} & \text{if } f(x) = true \\ \emptyset & \text{else} \end{cases} \tag{7}$$

The function *filter* takes as parameters a function f and a set X and returns a set consisting of the values of f applied to every element x of X. In the MI_Agent function, the current interpretation $CurrentI$ is initialized to the empty set and the agenda $\mathfrak{A}''$ to a set containing the empty set. Since the agent performs an incremental process, it is defined by a repeat-loop. The percept results Γ are sent to the queue Q. In order to take the observations Γ from the queue Q, the MI_Agent calls the $extractObservations$ function.

In the following we assume that Γ represents an ordered set. $MAP(\Gamma, \mathcal{WR}, \mathcal{T})$ determines the most probable world of observations Γ with respect to a set of weighted rules $\mathcal{WR}$ and the Tbox $\mathcal{T}$. This function performs the MAP process defined in Section **??**. It returns a vector W which consists of zeros and ones assigned to indicate whether the ground atoms of the considered world are true (positive) and false (negative), respectively. The function $select(W, \Gamma)$ then selects the positive assertions in the input Abox Γ using the bit vector W. The selected positive assertions are the assertions which require explanations. The *select* operation returns as output an Abox Γ' which has the following characteristic: $\Gamma' \subseteq \Gamma$. The determination of the most probable world by the MAP function and the selection of the positive assertions is also carried out on $\Gamma' \cup \mathcal{A} \cup \mathcal{A}_0$.

In the next step, a set of forward chaining rules $\mathcal{FR}$ is applied to $\Gamma' \cup \mathcal{A} \cup \mathcal{A}_0$. The generated assertions in this process are added to the to $\Gamma' \cup \mathcal{A} \cup \mathcal{A}_0$. In the next step, only the consistent Aboxes are selected and the inconsistent Aboxes are removed. Afterwards, the *Interpret* function is called to determine the new agenda $\mathfrak{A}''$, the new interpretation Abox $NewI$ and the Abox differences Δ_1 and Δ_2 for additions and omissions among $CurrentI$ and $NewI$. Afterwards, the $CurrentI$ is set to the $NewI$ and the MI_Agent function communicates the Abox differences Δ_1 and Δ_2 to the $Partners$. The *manage_agenda* function is also called. This function is explained in Section **??**. The termination condition of the MI_Agent function is that the $Die()$ function is true. Note that the MI_Agent waits in the function call $extractObservations(Q)$ if $Q = \emptyset$.

The $manage\_agenda(\mathfrak{A})$ function is called in the MI_Agent function to improve its performance. The agent can eliminate, shrink, or combine Aboxes.

After presenting the above algorithms, the mentioned unanswered questions can be discussed. A reason for performing the interpretation process and explaining the fiat assertions is that the probability

of $P(\mathcal{A}, \mathcal{A}', \mathcal{R}, \mathcal{WR}, \mathcal{T})$ will increase through the interpretation process. In other words, by explaining the observations the agent's belief to the percepts will increase. This shows a new perspective for performing the interpretation process. The answer to the question whether there is any measure for stopping the interpretation process, is indeed positive. This is expressed by $\mid p_i - p_{i-1} \mid < \frac{\epsilon}{i}$ which is the termination condition Ξ of the algorithm. The reason for selecting $\frac{\epsilon}{i}$ and not ϵ as the upper limit for the termination condition is to terminate the oscillation behaviour of the results. In other words, the precision interval is tightened step by step during the interpretation process. In Section 3, we discuss an example for interpreting a single video shot.

3 Preference-Based Video Shot Interpretation

One of the main innovations introduced in the previous section, namely the introduction of a probabilistic preference measure to control the space of possible interpretations, is demonstrated here using examples from an environmental domain.

We have to mention that this example is not constructed to show the possible branchings through the interpretation process. The purpose of this example is to show how the probabilities of the most probable world of observations $P(\mathcal{A}_0, \mathcal{A}, \mathcal{R}, \mathcal{WR}, \mathcal{T})$ behave during the interpretation process.

At the beginning of this example, the **signature** of the knowledge base is presented. The set of all concept names **CN** is divided into two disjoint sets **Events** and **PhysicalThings** such that

$\mathbf{CN} = \mathbf{Events} \cup \mathbf{PhysicalThings}$ where these two sets are defined as follows:

$\mathbf{Events} = \{CarEntry, EnvConference, EnvProt, HealthProt\}$

$\mathbf{PhysicalThings} = \{Car, DoorSlam, Building, Environment, Agency\}$

EnvConference, *EnvProt* and $HealthProt$ denote respectively environmental conference, environmental protection and health protection. The set of role names **RN** is defined as follows:

$\mathbf{RN} = \{Causes, OccursAt, HasAgency, HasTopic, HasSubject, HasObject, HasEffect, HasSubEvent, HasLocation\}$

The set of individual names **IN** is defined as follows:

$\mathbf{IN} = \{C_1, DS_1, ES_1, Ind_{42}, Ind_{43}, Ind_{44}, Ind_{45}, Ind_{46}, Ind_{47}, Ind_{48}\}$

In the following, the set of the forward chaining rules $\mathcal{FR}$ is defined:

$\mathcal{FR} = \{\forall x \;\; CarEntry(x) \rightarrow \exists y \;\; Building(y), OccursAt(x, y),$
$\forall x \;\; EnvConference(x) \rightarrow \exists y \;\; Environment(y), HasTopic(x, y),$
$\forall x \;\; EnvProt(x) \rightarrow \exists y \;\; Agency(y), HasAgency(x, y)\}$

Similarly, the set of backward chaining rules $\mathcal{BR}$ is given as follows:

$\mathcal{BR} = \{Causes(x, y) \leftarrow CarEntry(z), HasObject(z, x), HasEffect(z, y), Car(x), DoorSlam(y),$
$OccursAt(x, y) \leftarrow EnvConference(z), HasSubEvent(z, x), HasLocation(z, y), CarEntry(x), Building(y),$
$HasTopic(x, y) \leftarrow EnvProt(z), HasSubEvent(z, x), HasObject(z, y), EnvConference(x), Environment(y),$
$HasAgency(x, y) \leftarrow HealthProt(z), HasObject(z, x), HasSubject(z, y), EnvProt(x), Agency(y)\}$

In the following, a set of weighted rules $\mathcal{WR}$ is defined where the weight of each rule is 5:

$\mathcal{WR} = \{5 \; \forall x, y, z \; CarEntry(z) \wedge HasObject(z, x) \wedge HasEffect(z, y) \rightarrow Car(x) \wedge DoorSlam(y) \wedge Causes(x, y),$
$5 \; \forall x, y, z \; EnvConference(z) \wedge HasSubEvent(z, x) \wedge HasLocation(z, y) \rightarrow$
$CarEntry(x) \wedge Building(y) \wedge OccursAt(x, y),$
$5 \; \forall x, y, z \; EnvProt(z) \wedge HasSubEvent(z, x) \wedge HasObject(z, y) \rightarrow$
$EnvConference(x) \wedge Environment(y) \wedge HasTopic(x, y),$
$5 \; \forall x, y, z \; HealthProt(z) \wedge HasObject(z, x) \wedge HasSubject(z, y) \rightarrow$
$EnvProt(x) \wedge Agency(y) \wedge HasAgency(x, y)\}$

The selected value for ϵ in this example is 0.05. In the following, Δ_1 and Δ_2 denote respectively the set of assertions hypothesized by a forward chaining rule and the set of assertions generated by a backward chaining rule at each interpretation level. Let us assume that the media interpretation agent receives the following weighted Abox $\mathcal{A}$:

$\mathcal{A} = \{1.3 \; Car(C_1), 1.2 \; DoorSlam(DS_1), -0.3 \; EngineSound(ES_1), Causes(C_1, DS_1)\}$

The first applied operation to $\mathcal{A}$ is the MAP function which returns the bit vector $W = \langle 1, 1, 0, 1 \rangle$. By applying the *select* function to W and the input Abox $\mathcal{A}$, the assertions from the input Abox $\mathcal{A}$ are selected that correspond to the positive events in W. Additionally, the assigned weights to the positive assertions are also taken from the input Abox $\mathcal{A}$. In the following, Abox A_0 is depicted which contains the positive assertions:

$\mathcal{A}_0 = \{1.3\ Car(C_1), 1.2\ DoorSlam(DS_1), Causes(C_1, DS_1)\}$

At this step, $p_0 = P(\mathcal{A}_0, \mathcal{A}_0, \mathcal{R}, \mathcal{WR}, \mathcal{T}) = 0.755$. Since no appropriate forward chaining rule from $\mathcal{FR}$ is applicable to Abox $\mathcal{A}_0$, $\Delta_1 = \emptyset$ and as a result $\mathcal{A}_0 := \mathcal{A}_0 \cup \emptyset$. The next step is the execution of the *backward_chain* function where the next backward chaining rule from $\mathcal{BR}$ can be applied to Abox $\mathcal{A}_0$:

$Causes(x, y) \leftarrow CarEntry(z), HasObject(z, x), HasEffect(z, y), Car(x), DoorSlam(y)$

Consequently, by applying the above rule the next set of assertions is hypothesized:

$\Delta_2 = \{CarEntry(Ind_{42}), HasObject(Ind_{42}, C_1), HasEffect(Ind_{42}, DS_1)\}$

which are considered as strict assertions. Consequently, $\mathcal{A}_1$ is defined as follows: $\mathcal{A}_1 := \mathcal{A}_0 \cup \Delta_2$. In the above Abox, $p_1 = P(\mathcal{A}_0, \mathcal{A}_1, \mathcal{R}, \mathcal{WR}, \mathcal{T}) = 0.993$. As it can be seen, $p_1 > p_0$ i.e. $P(\mathcal{A}_0, \mathcal{A}_i, \mathcal{R}, \mathcal{WR}, \mathcal{T})$ used in Ξ increases by adding the new hypothesized assertions. This shows that the new assertions are considered as additional support. The termination condition of the algorithm is not fulfilled therefore the algorithm continues processing. At this level, it is still not known whether Abox $\mathcal{A}_1$ can be considered as the final interpretation Abox. Thus, this process is continued with another level. Consider the next forward chaining rule:

$\forall x \;\; CarEntry(x) \rightarrow \exists y \;\; Building(y), OccursAt(x, y)$

By applying the above rule, the next set of assertions is generated, namely:

$\Delta_1 = \{Building(Ind_{43}), OccursAt(Ind_{42}, Ind_{43})\}$

The generated assertions are also considered as strict assertions. In the following, the expanded Abox $\mathcal{A}_1$ is defined as follows: $\mathcal{A}_1 := \mathcal{A}_1 \cup \Delta_1$.

Let us assume the next backward chaining rule from $\mathcal{BR}$:

$OccursAt(x, y) \leftarrow EnvConference(z), HasSubEvent(z, x), HasLocation(z, y), CarEntry(x), Building(y)$

Consequently, by applying the above abduction rule the next set of assertions is hypothesized:

$\Delta_2 = \{EnvConference(Ind_{44}), HasSubEvent(Ind_{44}, Ind_{42}), HasLocation(Ind_{44}, Ind_{43})\}$

which are considered as strict assertions. Consequently, $\mathcal{A}_2 := \mathcal{A}_1 \cup \Delta_2$.

In the above Abox, $p_2 = P(\mathcal{A}_0, \mathcal{A}_2, \mathcal{R}, \mathcal{WR}, \mathcal{T}) = 0.988$. As it can be seen, $p_2 < p_1$ i.e. $P(\mathcal{A}_0, \mathcal{A}_i, \mathcal{R}, \mathcal{WR}, \mathcal{T})$ decreases slightly by adding the new hypothesized assertions. Since the termination condition of the algorithm is fulfilled, Abox $\mathcal{A}_1$ can be considered as the final interpretation Abox. To realize how the further behaviour of the probabilities is, this process is continued for the sake of illustration. Consider the next forward chaining rule from $\mathcal{FR}$:

$\forall x \;\; EnvConference(x) \rightarrow \exists y \;\; Environment(y), HasTopic(x, y)$

By applying the above rule, new assertions are generated.

$\Delta_1 = \{Environment(Ind_{45}), HasTopic(Ind_{44}, Ind_{45})\}$

In the following, the expanded Abox $\mathcal{A}_2$ is defined: $\mathcal{A}_2 := \mathcal{A}_2 \cup \Delta_1$.

Consider the next backward chaining rule from $\mathcal{BR}$:

$HasTopic(x, y) \leftarrow EnvProt(z), HasSubEvent(z, x), HasObject(z, y), EnvConference(x), Environment(y)$

By applying the above abduction rule, the following set of assertions is hypothesized:

$\Delta_2 = \{EnvProt(Ind_{46}), HasSubEvent(Ind_{46}, Ind_{44}), HasObject(Ind_{46}, Ind_{45})\}$

which are considered as strict assertions. In the following, $\mathcal{A}_3$ is defined as follows $\mathcal{A}_3 := \mathcal{A}_2 \cup \Delta_2$. In the above Abox $\mathcal{A}_3$, $p_3 = P(\mathcal{A}_0, \mathcal{A}_3, \mathcal{R}, \mathcal{WR}, \mathcal{T}) = 0.99$. As it can be seen, $p_3 > p_2$, i.e. $P(\mathcal{A}_0, \mathcal{A}_i, \mathcal{R}, \mathcal{WR}, \mathcal{T})$ increases slightly by adding the new hypothesized assertions.

Consider the next forward chaining rule:

$\forall x \;\; EnvProt(x) \rightarrow \exists y \;\; Agency(y), HasAgency(x, y)$

By applying the above rule, the next assertions are generated:

$\Delta_1 = \{Agency(Ind_{47}), HasAgency(Ind_{46}, Ind_{47})\}$

As a result, the expanded Abox $\mathcal{A}_3$ is presented as follows: $\mathcal{A}_3 := \mathcal{A}_3 \cup \Delta_1$.

Let us consider the next backward chaining rule from $\mathcal{BR}$:

$HasAgency(x, y) \leftarrow HealthProt(z), HasObject(z, x), HasSubject(z, y), EnvProt(x), Agency(y)$

Consequently, new assertions are hypothesized by applying the above abduction rule, namely:
$\Delta_2 = \{HealthProt(Ind_{48}), HasObject(Ind_{48}, Ind_{46}), HasSubject(Ind_{48}, Ind_{47})\}$
which are considered as strict assertions. Consequently, $\mathcal{A}_4$ is defined as follows: $\mathcal{A}_4 := \mathcal{A}_3 \cup \Delta_2$.
In the above Abox, $p_4 = P(\mathcal{A}_0, \mathcal{A}_4, \mathcal{R}, \mathcal{WR}, \mathcal{T}) = 0.985$. As it can be seen, $p_4 < p_3$, i.e. $P(\mathcal{A}_0, \mathcal{A}_i, \mathcal{R}, \mathcal{WR}, \mathcal{T})$ decreases slightly by adding the new hypothesized assertions.

Discussion of the Results:

The determined probability values $P(\mathcal{A}_0, \mathcal{A}_i, \mathcal{R}, \mathcal{WR}, \mathcal{T})$ of this example are summarized in the next table which shows clearly the behaviour of the probabilities stepwise after performing the interpretation process. In this table, variable i denotes the successive levels of the interpretation process.

i	*Abox* $\mathcal{A}_i$	$p_i = P(\mathcal{A}_0, \mathcal{A}_i, \mathcal{R}, \mathcal{WR}, \mathcal{T})$
0	$\mathcal{A}_0$	$p_0 = 0.755$
1	$\mathcal{A}_1$	$p_1 = 0.993$
2	$\mathcal{A}_2$	$p_2 = 0.988$
3	$\mathcal{A}_3$	$p_3 = 0.99$
4	$\mathcal{A}_4$	$p_4 = 0.985$

In this example, the interpretation process is consecutively performed four times. As it can be seen, through the first interpretation level the probability p_1 increases strongly in comparison to p_0. By performing the second, third and the forth interpretation levels, the probability values decrease slightly in comparison to p_1. This means no significant changes can be seen in the results. In other words, the determination of $\mathcal{A}_3$ and $\mathcal{A}_4$ were not required at all. But the determination of $\mathcal{A}_2$ was required to realize the slight difference $|p_2 - p_1| < \frac{\epsilon}{2}$. Consequently, Abox $\mathcal{A}_1$ is considered as the final interpretation Abox.

4 Summary

For multimedia interpretation, a semantically well-founded formalization is required. In accordance with previous work, a well-founded abduction-based approach is pursued. Extending previous work, abduction is controlled by probabilistic knowledge, and it is done in terms of first-order logic. Rather than merely using abduction for computing explanations with which observations are entailed, the approach presented in this paper, uses a probabilistic logic to motivate the explanation endeavor by increasing the belief in the observations. Hence, there exists a certain utility for an agent for the computational resources it spends for generating explanations. Thus, we have presented a first attempt to more appropriately model a media interpretation agent and evaluated it using a detailed example.

References

1. S. Castano, S. Espinosa, A. Ferrara, V. Karkaletsis, A. Kaya, R. Möller, S. Montanelli, G. Petasis, and M. Wessel. Multimedia interpretation for dynamic ontology evolution. In *Journal of Logic and Computation*. Oxford University Press, 2008.
2. Pedro Domingos and Matthew Richardson. Markov logic: A unifying framework for statistical relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, pages 339–371. Cambridge, MA: MIT Press, 2007.
3. Sofia Espinosa-Peraldi, Atila Kaya, and Ralf Möller. Formalizing multimedia interpretation based on abduction over description logic aboxes. In Bernardo Cuenca-Grau, Ian Horrocks, and Boris Motik, editors, *Proc. of the 22nd International Workshop on Description Logics (DL2009)*, 2010.
4. Sofia Espinosa-Peraldi, Atila Kaya, and Ralf Möller. Logical formalization of multimedia interpretation. In George Tsatsaronis Georgios Paliouras, Constantine D. Spyropoulos, editor, *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, volume 6050. Springer, 2010.
5. Oliver Gries, Ralf Möller, Anahita Nafissi, Maurice Rosenfeld, Kamil Sokolski, and Michael Wessel. A probabilistic abduction engine for media interpretation. Technical report, Hamburg University of Technology, 2010. http://www.sts.tu-harburg.de/tech-reports/2010/GMNR10.pdf.
6. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA., 1988.
7. David Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.