Uncertain Temporal Knowledge Graphs

Melisachew Wudage Chekol and Heiner Stuckenschmidt

Data and Web Science Group
University of Mannheim, Germany
{mel|heiner}@informatik.uni-mannheim.de

Abstract. Temporal data can be found in various sources from patient histories, purchase histories, employee histories, to web logs. Recent advances in open information extraction have paved the way for automatic construction of knowledge graphs (KGs) from such sources. Often the extraction tools used to construct KGs produce facts and rules along with their confidence scores, leading to the notion of uncertain temporal KGs. The facts and rules contained in these graphs tend to be noisy and erroneous due to either the accuracy of the extraction tools or uncertainty in the source data. In this work, we use a numerical extension of Markov logic networks to provide formal syntax and semantics for uncertain temporal KGs. Moreover, we propose a set of datalog constraints with inequalities, that extend the underlying schema of the KGs and help in resolving conflicting facts. Finally, we characterize the complexity of two important queries, maximum a-posteriori and conditional probability inference, for uncertain temporal KGs.

1 Introduction

Open Information Extraction (OIE) or 'machine reading' has been announced as a new paradigm for extracting domain independent knowledge from large Web corpora [3, 8]. OIE is of particular interest for the creation of knowledge graphs (KGs) and enriching existing ones. Automated construction of knowledge graphs often results in producing noisy and inaccurate facts and rules. In such graphs, the errors can propagate upon inference or knowledge base expansion as shown in [5]. Hence, it is indispensable to clean KGs at an early stage of their creation in order to avoid maintenance costs and provide clean content. In this work, we propose an approach to tackle this problem. Among others, Google's Knowledge Graph [6], NELL[1], and ReVerb[2] store probabilistic facts – facts along with their confidence scores representing how likely that they are correct. Most of these works have focused on identifying static facts, encoding them as binary relations. However, the vast majority of facts are fluents (dynamic relations whose truth is a function of time), only holding true during an interval of time. Thus, it is very important to extract a fact, for instance *coach(ClaudioRanieri, Chelsea)*, along with its temporal scope *2000–2004*. To overcome this, recently, there is an increased interest in temporal information extraction [12, 18, 19]. As research

[1] http://rtw.ml.cmu.edu/rtw/ [2] http://reverb.cs.washington.edu/

is advancing in building uncertain temporal KGs, it is indispensable to develop efficient techniques and tools to debug and clean such KGs. In this study, we provide a formal characterization of uncertain temporal KGs, and propose a way to debug them.

Related work. A limitation of existing methods for debugging automatically created knowledge [20, 24] is the incapability to deal with probabilistic and temporal information which leads to situations where statements that refer to objects at different points in time are assumed to be inconsistent. In addition, little has been done in debugging uncertain KGs, with the exception of the preliminary results in [11, 5, 7]. In [11], they use Markov Logic Networks (MLNs) and hand-crafted temporal constraints based on Allen's interval calculus, to debug RDF facts that contain date and time values. The shortcoming of this study is that: (i) no formal characterization in terms of syntax and semantics is provided, (ii) only a part of RDF(S) inference rules are considered, and (iii) do not provide constraints for debugging numerical attributes as we do here. Dylla et al. [7] have proposed an approach for resolving conflicts in RDF facts that contain date and time values. The authors use first-order logic Horn formulas with temporal predicates to express temporal and non-temporal constraints. The approach presented in this paper differs in several aspects: (1) they do not make use of MLNs to model the problem. Using MLNs allows to seamlessly integrate our approach with any reasoning tool that supports MAP inference in MLNs. As a consequence, we will benefit from future progress in developing efficient and scalable MAP inference engines. (2) We do not only support uncertain facts, but also uncertain constraints as this will help us model many common sense rules ("a father does not play in the same soccer club as his son") that are often not strict constraints, but express soft constraints. (3) Their approach do not allow debugging facts containing numerical attributes such as age, weight, and so on. Chen and Wang [5] debug erroneous facts by using a set of functional constraints however they do not deal with numerical as well as temporal facts as this is not their objective.

Despite the general complexity of MLNs, it has been shown that it can be used to reason about extracted facts at Web scale using hand-crafted [21] and extracted inference rules [22]. It has also been shown that MLNs can be used to deal with temporal relations in open information extraction [13]. Besides, MLNs are used to check the consistency of knowledge bases [4, 5, 11]. Consequently, in this study, we make use of an extension of MLNs to clean uncertain temporal KGs. Our contributions are the following: (i) we present a formal syntax and semantics, based on a numerical extension of MLN, for uncertain temporal KGs along with a set of temporal inference rules, (ii) we formalize MAP and conditional probability inference problems in uncertain temporal KGs and show that these problems remain NP-hard and #P-hard respectively, and (iii) we propose a set of constraints in order to clean erroneous facts in KGs.

Problem. Given an uncertain temporal KG $\mathcal{G}$, a set of temporal inference rules $\mathcal{F}$, and a set of temporal constraints, what is the most probable and error free temporal KG?

Problem:	**Debug Uncertain (temporal) KGs**
Input:	Uncertain (temporal) KG $\mathcal{G}$, a set of inference rules $\mathcal{F}$, and (optionally) a set of constraints $\mathcal{C}$
Output:	Most probable and conflict-free KG $\mathcal{G}' = clean(\mathcal{G}, \mathcal{F}, \mathcal{C})$.

Consider for instance the following uncertain temporal KG containing the facts that describe the footballer Cristiano Ronaldo:

Fact	**Validy time**	**Weight**
`bdate(CristianoRonaldo,1951)`		0.65
`plays(CristianoRonaldo,Manchester)`	$[2003, 2009]$	0.85

These facts conflict with each other because when Cristiano joined Manchester in 2003, he was already 52 years old while in fact he is only 31 as of 2016. Hence, it is highly unlike that he was playing for Manchester united at the age of 52. Thus, either the valid time[3] of the second fact or his birth date is wrong. In order to identify this kind of conflicts, we can introduce constraints. For instance, 'every player is at most 40 when playing for a club', corresponds to the constraint:

$$\forall p, c, t, t_1, t_2 : bdate(p, t) \wedge plays(p, c, [t_1, t_2]) \wedge \mathsf{NC}(t, [t_1, t_2]) \rightarrow \bot,$$
$$\mathsf{NC}(t, [t_1, t_2]) = t_1 - t < 40$$

If this problem is modeled using MLN and inference is done to obtain the most probable state, then we obtain both facts as a result. However, if the above constraint is included, the result contains only the second fact (the one with the higher weight). Note that NC can be encoded into the numerical extension of MLN [4].

2 Preliminaries

We present a brief introductory of knowledge graphs and Markov logic networks along with their temporal and numerical extensions respectively.

2.1 Knowledge Graphs

RDF is a language used to express structured information on the Web as graphs. We present a compact formalization of RDF [10]. Let $\mathcal{I}$ and $\mathcal{L}$ be two disjoint infinite sets denoting the set of IRIs (identifying a resource) and literals (a character string or some other type of data) respectively. We abbreviate the union of these sets as: $\mathcal{IL} = \mathcal{I} \cup \mathcal{L}$. A triple of the form $(s, p, o) \in \mathcal{I} \times \mathcal{I} \times \mathcal{IL}$ is called an *RDF triple*[4]. s is the *subject*, p is the *predicate*, and o is the *object* of the triple. Each triple can be thought of as an edge between the subject and the object labelled by the predicate, hence a set of RDF triples is often referred to as an *RDF graph*. We use the term *knowledge graph* loosely to refer to an RDF graph.

[3] Valid time is the time period in which a fact is considered valid.

[4] We do not consider blank nodes.

Temporal Knowledge Graphs In [14, 9], it is shown that an RDF graph can be extended with some temporal information by labeling each triple in the graph with some temporal element. The temporal element represents the time period in which the triple is valid, i.e, the *valid time* of the triple. We consider a discrete time domain $\mathcal{T}$ as a linearly ordered finite sequence of *time points*, for instance, days, minutes, or milliseconds. The finite domain assumption ensures that there are finitely many possible worlds in MLNs (see Section 3). A *time interval* is an ordered pair $[t_1, t_2]$ of time points, with $t_1 \leq t_2$ and $t_1, t_2 \in \mathcal{T}$, which denotes the closed interval from t_1 to t_2. We will work with the interval-based temporal domain for defining our data model. Note that time point-based temporal domains can be converted into interval-based by using for every time point t, introduce an interval $[t, t]$.

Definition 1 (Temporal KG). *A temporal* KG *is an* KG *where each fact* (s, p, o) *in the graph has a valid time* $[t_1, t_2]$*, i.e.,* $\texttt{tt} = (s, p, o, [t_1, t_2])$*. We refer to* `tt` *as a temporal fact.*

For a temporal KG G, its *snapshot* at time t is the graph $G(t)$ (the non-temporal KG): $G(t) = \{(s, p, o) \mid (s, p, o, [t, t]) \in G\}$. The KG associated with a temporal KG, denoted $u(G)$, is $\bigcup_t G(t)$, the union of the graphs $G(t)$. We define *temporal entailment* as follows: for temporal KGs G_1, G_2, $G_1 \models_t G_2$ if $G_1(t) \models G_2(t)$ for each t, $\models_t$ denotes temporal entailment [9] and $\models$ is the standard RDF(S) entailment [10].

The *syntax* of temporal KGs is given by reifying temporal facts into non-temporal facts by using the underlying RDF syntax [9]. Another possibility is to extend the RDF syntax and explicitly capture temporal information. In this paper, we do not discuss such implementation details, but instead focus on the conceptual aspects for use in uncertain temporal reasoning (Section 3).

The *semantics* of temporal KGs is given by extending the model theoretic semantics of RDF graphs. The notion of entailment for temporal KGs needs manipulating intervals in order to combine the notion of temporality and deductive properties. A deductive system[5] for temporal KGs, based on a sound and complete set of deduction rules, is presented in [9]. A modified version of these rules is given below:

$$\frac{(a, \mathit{type}, \mathit{class}, T_1)}{(a, \mathit{sc}, a, T_1)} \qquad \frac{(a, \mathit{type}, \mathit{property}, T_1)}{(a, \mathit{sp}, a, T_1)}$$

$$\frac{(a, \mathit{sc}, b, T_1)\ (x, \mathit{type}, a, T_2)\ \mathit{check}(T_1, T_2)}{(x, \mathit{type}, b, T_3)} \qquad \frac{(a, \mathit{sc}, b, T_1)\ (b, \mathit{sc}, c, T_2)\ \mathit{check}(T_1, T_2)}{(a, \mathit{sc}, c, T_3)}$$

$$\frac{(a, \mathit{sp}, b, T_1)\ (b, \mathit{sp}, c, T_2)\ \mathit{check}(T_1, T_2)}{(a, \mathit{sp}, c, T_3)} \qquad \frac{(a, \mathit{sp}, b, T_1)\ (x, a, y, T_2)\ \mathit{check}(T_1, T_2)}{(x, b, y, T_3)}$$

$$\frac{(a, \mathit{dom}, c, T_1)\ (x, a, y, T_2)\ \mathit{check}(T_1, T_2)}{(x, \mathit{type}, c, T_3)} \qquad \frac{(a, \mathit{range}, d, T_1)\ (x, a, y, T_2)\ \mathit{check}(T_1, T_2)}{(y, \mathit{type}, d, T_3)}$$

[5] In the rules, we use the following shorthands, *sp* for `rdfs:subPropertyOf`, *type* for `rdf:type`, *property* for `rdf:Property`, *sc* for `rdfs:subClassOf`, *class* for `rdfs:Class`, *dom* for `rdfs:domain`, and *range* for `rdfs:range`. Equivalently, the non-temporal deduction rules are those without valid time argument and the test $\mathit{check}(T_1, T_2)$.

$T_3 = T_1 \bowtie T_2$, and the definition of $\bowtie$ is given in Figure 1 and T_1, T_2 and T_3 are time intervals define over $\mathcal{T}$. FOL translations of the above rules are used as hard formulas for probabilistic reasoning in uncertain temporal KGs (see Section 3). We use MLNs to extend temporal KGs with uncertainty.

2.2 Markov Logic Networks

Markov Logic Networks (MLNs) can be seen as a first-order template language for log-linear models with binary variables. MLNs combine Markov networks and first-order logic (FOL) by attaching weights to first-order formulas and viewing these as templates for features of Markov networks [17]. Markov Logic networks have been extended with numerical [4] and continuous [25] constraints. In this paper, we will use the numerical extension which is useful for reasoning in uncertain temporal KGs.

Definition 2 (MLN with Numerical Constraints). *A numerical constraint* NC *is composed of numerical constants (such as elements of* $\mathbb{N}$, $\mathbb{I}$, *and so on), variables, elementary operators or functions (such as,* $+$, $*$, $-$, $\div$, $\%$, $\sqrt{\ }$*), standard relations (*$>$, $<$, $=$, $\neq$, $\geq$, $\leq$*), and boolean operators (*$\wedge$, $\vee$, $\neg$*). An MLN* L *with numerical constraints (simply MLN) is a set of pairs* (FC_i, w_i) *where* FC_i *is a formula in FOL that may contain a* NC *and* w_i *is a real number representing the weight of formula* FC_i.

Together with a finite set of constants C, it defines a Markov Network $M_{L,C}$, where $M_{L,C}$ contains one node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true, and 0 otherwise. The probability distribution over possible worlds x specified by the ground Markov network $M_{L,C}$ is given by:

$$P(X = x) = \frac{1}{Z}\exp\big(\sum_{i=1}^{F} w_i n_i(x)\big)$$

where F is the number of formulas in the MLN and $n_i(x)$ is the number of true groundings of FC_i in x. The groundings of a formula are formed simply by replacing its variables with constants in all possible ways.

Example 1. Using MLN it is possible to represent the hard constraint: footballers born before 1850 are not alive:

$\{\forall a, y : \mathit{footballer}(a) \wedge \mathit{bdate}(a, y) \wedge \mathsf{NC}(y) \Rightarrow \mathit{dead}(y),\ \mathsf{NC}(y) = y < \mathit{1850}\}$.

A common inference task over MLNs is finding the most probable state of the world, i.e., finding a complete assignment to all ground atoms which maximizes the probability. This is known as maximum a-posteriori inference (MAP). Finding a most likely world of an MLN is a generalization of the (NP-hard) MaxSAT problem. Another equally important inference problem is, conditional probability inference. This is the task of computing the probability of a set of variables given evidence. The complexity of this problem is known to be #P-hard [17].

3 Reasoning in Uncertain Temporal KGs

Uncertain temporal knowledge graphs (UTKGs) are extensions of temporal KGs with log-linear models that are capable of representing uncertainties and reasoning over temporal knowledge bases. A UTKG is a temporal knowledge graph where each triple has a valid-time and an associated weight or confidence. In other words, each triple has an associated *timestamp* or *valid time*, in addition to a confidence value.

Syntax. A UTKG graph $\mathcal{G} = (\mathcal{G}^{\mathsf{D}}, \mathcal{G}^{\mathsf{U}})$ consists of a deterministic (*hard*) temporal KG $\mathcal{G}^{\mathsf{D}}$ and a UTKG $\mathcal{G}^{\mathsf{U}}$ with $\mathcal{G}^{\mathsf{D}} \cap \mathcal{G}^{\mathsf{U}} = \emptyset$. An uncertain (*soft*) temporal KG is defined as $\mathcal{G}^{\mathsf{U}} = \{\langle \mathtt{tt}_i, w_{\mathtt{tt}_i} \rangle\}$ where $\mathtt{tt}_i$ is a temporal fact and $w_{\mathtt{tt}_i}$ is a real-valued weight assigned to $\mathtt{tt}_i$. The syntax of an uncertain temporal fact is similar to the underlying temporal RDF, besides, each fact has an associated weight, written as $\{(\mathtt{tt}_i, w_{\mathtt{tt}_i})\}$.

Example 2. Consider the following UTKG which represents sport's personality Claudio Raineri's courier:

	Temporal fact	Weight
(1)	`(CRanieri, coach, ChelseaFC, [2000,2004])`	0.9
(2)	`(CRanieri, coach, LeicesterFC, [2015,2016])`	0.7
(3)	`(CRanieri, playsFor, PalermoFC, [1984,1986])`	0.5
(4)	`(CRanieri, bdate,1951)`	1.0
(5)	`(CRanieri, coach, NapoliFC, [2001,2003])`	0.6

Before providing semantics to UTKGs, we need to extend membership ($\in$) to ($\in^*$) and subset ($\subseteq$) to ($\subset^*$) relations as follows. Given a UTKGs $\mathcal{G}$, a temporal fact $(s, p, o, [t_1, t'_1])$, and a UTKG $\mathcal{G}'$, we denote by $(s, p, o, [t_1, t'_1]) \in^* \mathcal{G}$ if $\exists (s, p, o, [t_2, t'_2]) \in \mathcal{G}$ such that $t_2 \leq t_1$ and $t'_1 \leq t'_2$. We denote by $\mathcal{G}' \subseteq^* \mathcal{G}$ if for all $\mathtt{tt} \in^* \mathcal{G}'$, then $\mathtt{tt} \in^* \mathcal{G}$.

Semantics. The semantics of a UTKG is based on a joint probability distribution over the uncertain part of the UTKG. In particular, the weights of the facts in $\mathcal{G}^{\mathsf{U}}$ determine a log-linear probability distribution. As mentioned earlier, we assume that the time domain, in which the validity of triples is expressed is finite as well as discrete, hence the set of possible worlds is finite. Formally, for a given UTKG $\mathcal{G} = (\mathcal{G}^{\mathsf{D}}, \mathcal{G}^{\mathsf{U}})$ and some $\mathcal{G}'$ over the same set of IRIs and literals $\mathcal{IL}$, the probability of $\mathcal{G}'$ is defined as:

$$P(\mathcal{G}') = \begin{cases} \frac{1}{Z} \exp \left(\sum\limits_{\{(\mathtt{tt}_i, w_{\mathtt{tt}_i}) \in^* \mathcal{G}^{\mathsf{U}} : \mathcal{G}' \models_t \mathtt{tt}_i\}} w_{\mathtt{tt}_i} \right) & \text{if } \mathcal{G}' \models_t \mathcal{G}^{\mathsf{D}}, \\ \\ 0 \qquad \text{otherwise} & \end{cases}$$

where $\models_t$ is a temporal entailment relation, and Z is the normalization constant of the log-linear probability distribution P. Note that in MAP inference (i.e., obtaining the most probable temporal KG) Z is not computed. A UTKG can be mapped into a first-order knowledge base as discussed below.

Herbrand Models. The set of formulas, denoted by $\mathcal{F}$, listed in Figure 1 are derived from the deduction rules of Section 2.1. Implicitly, $\mathcal{F}$ also contains the non-temporal equivalents of the inference rules, i.e., those that do not contain time interval arguments and the *check* function (more precisely, FOL translations of RDF/S entailment rules [10]). Let $\mathcal{C}$ be the set of IRIs and Literals that appear in some UTKG $\mathcal{G}$, the Herbrand base of $\mathcal{F}$ can be constructed by instantiating all the variables in $\mathcal{F}$ using the constants in $\mathcal{C}$. The function $\boldsymbol{\theta}$, given a finite set $\mathcal{C}$ and a set of time points $\mathcal{T}$, maps each fact in some UTKG into a subset of the Herbrand base HB of $\mathcal{F}$ with respect to $\mathcal{C}$ and $\mathcal{T}$. Each subset of the Herbrand base is a Herbrand interpretation specifying which ground atoms are true. A Herbrand interpretation H is a Herbrand model of $\mathcal{F}$, denoted as $\models_H \mathcal{F}$, iff it satisfies all groundings of the formulas in $\mathcal{F}$.

Definition 3 (Mapping UTKG into FOL). *Given a UTKG $\mathcal{G}$ over a finite set of IRIs and literals $\mathcal{C}$, a time domain $\mathcal{T}$, and HB the Herbrand base of $\mathcal{F}$ with respect to $\mathcal{C}$ and $\mathcal{T}$, $\boldsymbol{\theta} : \mathcal{P}(\mathcal{G}) \to \mathcal{P}(HB)$ maps $\mathcal{G}$ into subsets of HB as follows:*

$$\boldsymbol{\theta}(\mathcal{G}) = \bigcup_{\mathtt{tt} \in \mathcal{G}} \boldsymbol{\theta}(\mathtt{tt}), \; \textit{where } \boldsymbol{\theta}((s,p,o,T)) = tt(s,p,o,T).$$

The predicate tt is typed, i.e., $s, p \in \mathcal{I}$, $o \in \mathcal{IL}$ and $T \in \mathcal{T}$. At this point we need to show that the function $\boldsymbol{\theta}$ is bijective, i.e., it induces a one-to-one correspondence between Herbrand models of $\mathcal{F}$ and expanded KGs. Applying $\mathcal{F}$ repeatedly on an uncertain KG may generate a set of new facts, this results in an *expanded* KG.

Theorem 1. *Let $\mathcal{C} \subseteq \mathcal{IL}$ be a set of IRIs and literals and let $\mathcal{T}$ be a set of time points. In addition, let $\mathcal{G}$ be a UTKG over $\mathcal{C}$ and let HB be the Herbrand base of $\mathcal{F}$ with respect to $\mathcal{C}$. Then, for any $\mathcal{G}' \subseteq \mathcal{G}$, $\mathcal{G} \models_t \mathcal{G}' \Rightarrow \boldsymbol{\theta}(\mathcal{G}') \models_H \mathcal{F}$ and for any $H \subseteq HB$, $H \models_H \mathcal{F} \Rightarrow \boldsymbol{\theta}^{-1}(H) \models \mathcal{G}''$ and $\mathcal{G} \models_t \mathcal{G}''$.*

3.1 MAP Inference

MAP inference in UTKG corresponds to obtaining the most probable, consistent and non-probabilistic temporal KG. Given a UTKG $\mathcal{G}$, a set of inference rules $\mathcal{F}$, and a translation function $\boldsymbol{\theta}$, we denote the MAP problem by $\mathit{clean}(\boldsymbol{\theta}(\mathcal{G}), \mathcal{F})$. Computing $\mathit{clean}(\boldsymbol{\theta}(\mathcal{G}), \mathcal{F})$ requires to translate $\mathcal{G}$ with the function $\boldsymbol{\theta}$ into an equivalent Markov logic formalization. Then the inference rules $\mathcal{F}$ are added to this translation. The MAP state is computed with the help of a cutting planes algorithm in [4] applied to this input data. To do so, the evidence clauses $\boldsymbol{\theta}(G)$ and the grounding of $\mathcal{F}$ with respect to $\boldsymbol{\theta}(G)$ are given as input to the algorithm. Applying the inverse translation function $\boldsymbol{\theta}^{-1}$ to the MAP state, yields the most probable temporal KG. The MAP problem in MLN can be turned into an integer linear program [15] which allows to integrate extrenal functions (for instance the check function in Figure 1) as already shown in [4].

$$
\begin{array}{lll}
(r_1) & tt(a, type, property, T_1) \rightarrow tt(a, sp, a, T_1) & \\
(r_2) & tt(a, sp, b, T_1) \wedge tt(b, sp, c, T_2) \wedge check(T_1, T_2) \rightarrow tt(a, sp, c, T_3) & T_3 = T_1 \bowtie T_2 \\
(r_3) & tt(a, sp, b, T_1) \wedge tt(x, a, y, T_2) \wedge check(T_1, T_2) \rightarrow tt(x, b, y, T_3) & T_3 = T_1 \bowtie T_2 \\
(r_4) & tt(a, type, class, T_1) \rightarrow tt(a, sc, a, T_1) & \\
(r_5) & tt(a, sc, b, T_1) \wedge tt(b, sc, c, T_2) \wedge check(T_1, T_2) \rightarrow tt(a, sc, c, T_3) & T_3 = T_1 \bowtie T_2 \\
(r_6) & tt(a, sc, b, T_1) \wedge tt(x, type, a, T_2) \wedge check(T_1, T_2) \rightarrow tt(x, type, b, T_3) & T_3 = T_1 \bowtie T_2 \\
(r_7) & tt(a, dom, c, T_1) \wedge tt(x, a, y, T_2) \wedge check(T_1, T_2) \rightarrow tt(x, type, c, T_3) & T_3 = T_1 \bowtie T_2 \\
(r_8) & tt(a, range, d, T_1) \wedge tt(x, a, y, T_2) \wedge check(T_1, T_2) \rightarrow tt(y, type, d, T_3) & T_3 = T_1 \bowtie T_2
\end{array}
$$

$$
[t_1, t'_1] \bowtie [t_2, t'_2] = \begin{cases} [t_1, t'_1] & \text{if } t_1 = t_2 \wedge t'_1 = t'_2 \\ [t'_1, t_2] & \text{if } t'_1 = t_2 \\ [t_2, t'_1] & \text{if } t_1 < t_2 \wedge t_2 < t'_1 \wedge \\ & \quad t'_1 < t'_2 \\ [t_1, t'_1] & \text{if } t_1 < t_2 \wedge t'_1 < t'_2 \\ [t_1, t'_1] & \text{if } t_1 = t_2 \wedge t'_1 < t'_2 \\ [t_2, t'_2] & \text{if } t'_1 < t_1 \wedge t_2 = t'_2 \\ \emptyset & \text{if } t'_1 < t_2 \end{cases} \qquad check(T_1, T_2) = \begin{cases} false & \text{if } T_1 \bowtie T_2 = \emptyset \\ true & \text{otherwise} \end{cases}
$$

Fig. 1. A set of temporal RDF/S inference rules $\mathcal{F}$. If $check(T_1, T_2) = false$, then the head of the rules (r_1) – (r_8) becomes false $(\perp)$. All of the formulas are universally quantified over all the variables.

Theorem 2. *Given the following:*

- *a* UTKG $\mathcal{G} = (\mathcal{G}^{\mathsf{D}}, \mathcal{G}^{\mathsf{U}})$ *over a finite set $\mathcal{IL}$ of IRIs and literals, and a finite set of time points $\mathcal{T}$,*
- *the Herbrand base HB of the formulas $\mathcal{F}$ with respect to $\mathcal{IL}$ and $\mathcal{T}$,*
- *the set of ground formulas $\mathcal{G}_1$ constructed from $\mathcal{G}^{\mathsf{D}}$, and*
- *the set of ground formulas $\mathcal{G}_2$ constructed from $\mathcal{G}^{\mathsf{U}}$.*

The most probable, expanded and consistent *temporal* KG *is obtained with:*

$$
\boldsymbol{\theta}^{-1}(H) = \underset{HB \supseteq H \models \mathcal{G}_1 \cup \mathcal{F}}{\arg\max} \left(\sum_{(tt_j, w_j) \in \mathcal{G}_2 : H \models_{H} tt_j} w_j \right)
$$

From Theorem 1 and the results in [4] taken together, the problem of computing the most probable temporal KG is NP-hard.

Example 3 (MAP state). Given a UTKG which contains the uncertain temporal triples (1)–(5) of Example 2 and the hard temporal constraints (6) and (7), its most probable and consistent temporal KG contains the facts (1)–(4) without their associated weights.

- A person cannot be a coach of two clubs at the same time.
  (6) $\forall x, y, z, T_1, T_2 : tt(x, \texttt{coach}, y, T_1) \wedge tt(x, \texttt{coach}, z, T_2) \rightarrow \texttt{disjoint}(T_1, T_2)$
- A person cannot be a coach before he or she was born.
  (7) $\forall x, y, z, T_1, T_2 : tt(x, \texttt{bdate}, y, T_1) \wedge tt(x, \texttt{coach}, z, T_2) \rightarrow \texttt{before}(T_1, T_2)$

The predicates `disjoint` and `before` are Allen's interval relations [2]. Below, we introduce expressive constraints that allow to identify erroneous facts.

3.2 Debugging numerical attributes in Uncertain KGs

Often uncertain knowledge graphs may contain a large number of numerical data which can be dates, times, latitudes/longitudes, numerical values measured in different units, and so on. For instance, Claudio Ranieri is 1.82 meters tall corresponds to the fact $(\textit{CRanieri}, \textit{height}, \textit{1.82})$. It contains a numeric datum (1.82). Uncertain facts which contain numerical data can be conflicting. One way of resolving this is to compute a MAP state of a given KG which basically throws out facts which have inferior weights or confidences given some inference rules. However, this is not enough. Consider for instance, if there is an uncertain KG that contains two facts: (1) $((\textit{CRanieri}, \textit{height}, \textit{1.80}), \textit{0.3})$ and (2) $((\textit{CRanieri}, \textit{height}, \textit{3.5}), \textit{0.9})$. Assume that these facts are translated into an MLN framework along with the constraint that the property 'height' is functional, i.e., $\forall x, y : tt(x, \textit{height}, y) \land tt(x, \textit{height}, y') \rightarrow y = y'$. In this setting, performing MAP inference results in a KG containing the certain fact $(\textit{CRanieri}, \textit{height}, \textit{3.5})$. However, the correct output should contain only the first triple because normally people cannot be taller than 2.5 meters. In order to *remove* such conflicts, we can add another constraint as discussed below. Constraints are used in description logics and database systems to ensure data validity. In the following, we introduce such constraints in order to ensure validity of temporal facts in uncertain KGs.

Constraints A Datalog [1] constraint is an expression of the form *body* → *head*, where the *head* is an atom (i.e., an expression of the form $p(x_1, \ldots, x_n)$ in which each x_i is either a constant or a variable) and *body* is a set of atoms, such that each variable occurring in the *head* also occurs in some atom in the *body.* Since our choice of MLN with numerical constraints allows us to use external functions whose truth values are computed outside the MLN setting (see Chekol et al. [4]), we can extend datalog constraints (specifically, *inclusion dependencies*, *equality generating dependencies* and *negative constraints* [1]) with numerical constraints. To debug uncertain KGs we can introduce a set of datalog inspired constraints which become *hard* (deterministic) or *soft* (uncertain) formulas in MLNs. For instance, if we want to state that "a person cannot be taller that 2.5 meters", then we can introduce a rule of the form: $\forall x, y : (x, \textit{type}, \textit{person}) \land (x, \textit{height}, y) \rightarrow y < 3.5$. In the following, we introduce three different kinds of constraints.

Inclusion dependencies with inequalities (IDIs). IDIs are first-order formulas of the form $\forall X, Y : \Phi(X, Y) \land \mathsf{NC}(X_i, Y_j) \rightarrow \Psi(Y)$, where $\Phi(X, Y)$ is the body of the formula, it is a conjunction of atoms, $\Psi(Y)$ is the head of the formula, X, Y are sets of variables, and $X_i \subseteq X$ and $Y_j \subseteq Y$. In addition, $\mathsf{NC}(X_i, Y_j)$ denotes a numerical constraint which is an arithmetic expression (see Definition 2).

Example 4. Those who are above the age of 40 are probably retired footballers: $\forall x, y : tt(x, \textit{type}, \textit{Footballer}) \land tt(x, \textit{age}, y) \land \mathsf{NC}(y) \rightarrow tt(x, \textit{type}, \textit{RFootballer})$, $\mathsf{NC}(y) = y > 40$.

(In)equality generating dependencies (IGDs). IGDs are first-order formulas of the form $\forall X : \Phi(X) \rightarrow \mathsf{NC}(X_i)$, where $\Phi(X)$ is the body of the formula which is a conjunction of atoms, X is a set of variables, and $X_i \subseteq X$. In addition, $\mathsf{NC}(X_i)$ denotes a numerical constraint.

Example 5. Temperature Celsius tc can be converted into an equivalent Fahrenheit scale tf using the formula $tf = 1.8tc + 32$: $\forall x, tc, tf : tt(x, tempc, tc) \wedge tt(x, tempf, tf) \rightarrow \mathsf{NC}(tc, tf)$, $\mathsf{NC}(tc, tf) = 1.8tc + 32$. From a practical viewpoint, this rule can be used for checking if two facts extracted from Wikipedia, one containing temperature in Celsius format and the other in Fahrenheit, are conflicting.

Disjointness constraints (DCs). DCs are first-order formulas of the form $\forall X : \Phi(X) \wedge \mathsf{NC}(X_i) \rightarrow \perp$, where $\Phi(X)$ is the body of the formula which is a conjunction of atoms, X is a set of variables, and $X_i \subseteq X$. In addition, $\mathsf{NC}(X_i)$ denotes a numerical constraint.

Example 6. A valid life span of a person is less than 150 years, can be expressed as DCs formula: $\forall bd, dd : tt(x, bdate, bd) \wedge tt(x, ddate, dd) \wedge \mathsf{NC}(bd, dd) \rightarrow \perp$, $\mathsf{NC}(bd, dd) = (dd - bd) > 0 \wedge (dd - bd) < 150$.

These constraints are more expressive than RDF schema because they allow to express disjointness, functionality of properties, inverse properties, among others. Once an uncertain KG is translated into an equivalent Markov logic formalism using the formula $\boldsymbol{\theta}$, and sets of IDIs, IGDs, and DCs constraints over the KG have been constructed, we can apply MAP inference in order to retrieve the most probable and conflict-free KG using $clean(\boldsymbol{\theta}(\mathcal{G}), \mathcal{F}, \mathcal{C})$.

3.3 Conditional Probability Inference

The conditional probability of a temporal fact `tt` given a UTKG $\mathcal{G}$ is the sum of the probabilities of the consistent temporal KGs containing `tt`. In general, a *conditional probability query* is conjunction of a set of temporal facts. Given a query q and a UTKGs $\mathcal{G}$, the conditional probability of q is obtained using:

$$P_q(q \mid \mathcal{G}) = \sum_{\mathcal{G}' : q \subseteq^* \mathcal{G}'} P(\mathcal{G}')$$

$\mathcal{G}'$ is a possible world over the same signature $\mathcal{IL}$ and $\mathcal{T}$ as $\mathcal{G}$. In order to sum over all $\mathcal{G}'$, we need to compare the time intervals in the facts of q with those of $\mathcal{G}'$. To do so, we rewite the query q as follows: for each temporal fact $\mathtt{tt} \in q$ if $\exists \mathtt{tt}' \in \mathcal{G}$ and that $\mathtt{tt} \subseteq^+ \mathtt{tt}'$, then we replace `tt` in q with $\mathtt{tt}'$. The relation $\subseteq^+$ is defined as: for two temporal facts $\mathtt{tt} = (s, p, o, [t_1, t_1'])$ and $\mathtt{tt}' = (s', p', o', [t_2, t_2'])$, $\mathtt{tt} \subseteq \mathtt{tt}'$ if $s = s'$, $p = p'$, $o = o'$, $t_2 \leq t_1$ and $t_1' \leq t_2'$. This allows us to compute conditional probabilities on top of current solvers such as MC-SAT [16]. The rewriting can be done in polynomial time in the size of the UTKG in the worst case. Since no additional computation is required, the complexity of conditional probability inference remains #P-hard for UTKGs. For example the conditional query $tt(\mathit{CRanieri}, \mathit{coach}, \mathit{Chelsea}, [\mathit{2001}, \mathit{2003}]$ given $\mathcal{G}$ (from Example 2), can be rewritten as: $P_q(tt(\mathit{CRanieri}, \mathit{coach}, \mathit{Chelsea}, [\mathit{2000}, \mathit{2004}]) \mid \mathcal{G})$. Since conditional inference is intractable, computing exact probabilities is hard. Thus, it is usual to resort to sampling methods for approximate inference. The state of the art marginal inference algorithm is MC-SAT. A Monte Carlo algorithm must sample consistent or conflict-free temporal KGs according to the distribution P_q. This is very difficult for three reasons: (i) the complexity of reasoning in MLN, (ii) the size of uncertain KGs (such as NELL, ReVerb), and (iii) the presence of deterministic dependencies in the UTKGs. Due to these reasons, we need to use emerging lifted inference techniques for marginal inference [23]. We consider this as a future work.

4 Conclusion and Future Work

In this paper, we have presented an MLN based approach for reasoning over uncertain temporal knowledge graphs. In doing so, we proposed a formal syntax and semantics. Besides, we formalized MAP and conditional probability inference problems in UTKGs and shown that these problems remain NP-hard and #P-hard respectively. Importantly, we extended the datalog constraints in order to clean erroneous facts in UTKGs. Thereby, we are able to apply MAP inference in order to obtain a most probable and conflict-free temporal KG from an uncertain one.

Currently, we are in the process of conducting *experiments* by using an existing implementation, that supports the numerical extension of MLN, called ROCKIT[6]. At present, there is no avaiable uncertain temporal dataset. We are preparing a gold standard from Freebase and ReVerb, by converting those facts that contain dates and times into temporal facts. With that, we can test the efficiency and scalability of the proposed approach. Another direction for future work is, to investigate how lifted inference can be applied. This is important because the complexity of reasoning in MLNs is intractable in general.

References

1. Serge Abiteboul and Victor Vianu. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43(1):62–124, 1991.
2. James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
3. Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.
4. Melisachew Wudage Chekol, Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. Data interlinking through robust linkkey extraction. In *Proc. 22nd european conference on artificial intelligence (ECAI), The Hague (NL)*, 2016. to appear.
5. Yang Chen and Daisy Zhe Wang. Knowledge expansion over probabilistic knowledge bases. In *SIGMOD*, pages 649–660. ACM, 2014.

[6] http://executor.informatik.uni-mannheim.de/systems/n-rockit/

6. Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, pages 601–610. ACM, 2014.
7. Maximilian Dylla, Mauro Sozio, and Martin Theobald. Resolving temporal conflicts in inconsistent rdf knowledge bases. In *BTW*, pages 474–493, 2011.
8. Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.
9. Claudio Gutierrez, Carlos Hurtado, and Alejandro Vaisman. Temporal rdf. In *The Semantic Web: Research and Applications*, pages 93–107. Springer, 2005.
10. Patrick Hayes. RDF semantics. W3C Recommendation, 2004.
11. Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. Applying Markov Logic for Debugging Probabilistic Temporal Knowledge Bases. In *Proceedings of the 4th Workshop on Automated Knowledge Base Construction (AKBC)*, 2014.
12. Xiao Ling and Daniel S. Weld. Temporal information extraction. In *Proceedings of the AAAI 2010 Conference*, pages 1385 – 1390, Atlanta, Georgia, USA, July 11-15 2010. Association for the Advancement of Artificial Intelligence.
13. Xiao Ling and Daniel S Weld. Temporal information extraction. In *AAAI*, volume 10, pages 1385–1390, 2010.
14. Boris Motik. Representing and querying validity time in rdf and owl: A logic-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12:3–21, 2012.
15. Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *AAAI*, 2013.
16. Hoifung Poon and Lucy Vanderwende. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 813–821. Association for Computational Linguistics, 2010.
17. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.
18. Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *SIGKDD*, pages 1104–1112. ACM, 2012.
19. Anisa Rula, Matteo Palmonari, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, Jens Lehmann, and Lorenz Bühmann. Hybrid acquisition of temporal scopes for rdf data. In *European Semantic Web Conference*, pages 488–503. Springer, 2014.
20. Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank Van Harmelen. Debugging incoherent terminologies. *Journal of Automated Reasoning*, 39(3):317–349, 2007.
21. Stefan Schoenmackers, Oren Etzioni, and Daniel S Weld. Scaling textual inference to the web. In *EMNLP*, pages 79–88, 2008.
22. Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. Learning first-order horn clauses from web text. In *EMNLP*, pages 1088–1098, 2010.
23. Parag Singla and Pedro M Domingos. Lifted first-order belief propagation. In *AAAI*, volume 8, pages 1094–1099, 2008.
24. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.
25. Jue Wang and Pedro M Domingos. Hybrid markov logic networks. In *AAAI*, volume 8, pages 1106–1111, 2008.