TeCoRe: Temporal Conflict Resolution in Knowledge Graphs*

Melisachew W. Chekol
DWS Group
University of Mannheim
mel@informatik.uni-mannheim.de

Giuseppe Pirrò
ICAR-CNR
Rende(CS), Italy
pirro@icar.cnr.it

Joerg Schoenfisch
DWS Group
University of Mannheim
joerg@dwslab.de

Heiner Stuckenschmidt
DWS Group
University of Mannheim
heiner@informatik.uni-mannheim.de

ABSTRACT

The management of uncertainty is crucial when harvesting structured content from unstructured and noisy sources. Knowledge Graphs (KGs), maintaining both numerical and non-numerical facts supported by an underlying schema, are a prominent example. Knowledge Graph management is challenging because: (i) most of existing KGs focus on static data, thus impeding the availability of timewise knowledge; (ii) facts in KGs are usually accompanied by a confidence score, which witnesses how likely it is for them to hold.

We demonstrate TeCoRe, a system for temporal inference and conflict resolution in uncertain temporal knowledge graphs (UTKGs). At the heart of TeCoRe are two state-of-the-art probabilistic reasoners that are able to deal with temporal constraints efficiently. While one is scalable, the other can cope with more expressive constraints. The demonstration will focus on enabling users and applications to find inconsistencies in UTKGs. TeCoRe provides an interface allowing to select UTKGs and editing constraints; shows the maximal consistent subset of the UTKG, and displays statistics (e.g., number of noisy facts removed) about the debugging process.

1. INTRODUCTION

The automated construction of knowledge graphs is an active area of research [2]. Open Information Extraction (OIE) has been used for creating and enriching knowledge graphs (KGs) such as YAGO, Google Knowledge Vault, Freebase, DBpedia, Wikidata, ProbBase, ProbKB, FootballDB, and ReVerb. Some of these KGs like YAGO, FootballDB, and Wikidata also contain temporal facts – facts with validity time. Besides, most of them store probabilistic facts, i.e., facts along with confidence scores witnessing how likely these facts are to hold. The automated construction of KGs produces noisy and inaccurate facts and rules with errors that can propagate upon inference or knowledge expansion.

Harvesting KGs poses some key challenges. The first concerns the need to *clean KGs from noisy facts* to avoid maintenance costs and provide reliable content. A limitation of existing methods (e.g., [9]) is the lack of capabilities to deal with both probabilistic and temporal facts. This leads to situations where statements that refer to objects at different points in time are assumed to be inconsistent. The second challenge is *providing temporal information.* Most existing approaches focus on identifying static facts encoded as binary relations. However, the vast majority of facts are fluents (dynamic relations whose truth is a function of time), only holding during an interval of time. Facts like (ClaudioRanieri, coach, Chelsea) loose relevance without a temporal scope (2000–2004 in this case). In addition, temporal inference rules and consistency checking constraints are useful to both deriving implicit (or new) facts from existing ones and identifying conflicting facts.

We demonstrate the TeCoRe (Temporal Conflict Resolution) system. TeCoRe provides a Web interface allowing to choose inference rules, build constraints (via an autocompletion), and compute conflicting temporal facts. The tool is built on top of the temporal extensions of nRockIt and PSL solver. In a nutshell, TeCoRe translates UTKGs, inference rules and constraints into weighted first-order logic that can be represented by MLNs (Markov Logic Networks) and PSL (Probabilistic Soft Logic). TeCoRe can be applied in several contexts among which debugging UTKGs and making temporal inference scalable due to the magnitude of many existing KGs. TeCoRe has been successfully tested in a highly noisy setting where there are as many erroneous temporal facts as the correct ones. Besides, TeCoRe allows to set a threshold value and remove derived facts below that.

The intended audience of the demo is broad; TeCoRe is particularly useful for people with some knowledge of constraints and temporal databases. TeCoRe's UI provides guidance on how to construct temporal constraints and rules as well as showing which temporal facts are inconsistent.

Related work. Temporal databases [5, 7] extend databases with a temporal dimension. In this respect, one of the key research challenges is the study of temporal query evaluation under constraints. In the context of KGs another relevant

*Authors listed in alphabetical order.

Proceedings of the VLDB Endowment, Vol. 10, No. 12

problem is the management of uncertainty. Indeed, building and extending KGs using open domain information extraction, will often lead to uncertainty about the correctness of schema information. Moreover, a large variety of temporal inference rules and constraints, some of which will be domain specific, can also be the subject of uncertainty. One key peculiarity of TeCoRe with respect to related research is the focus on maximum a posteriori inference instead of marginal inference. Moreover, existing approaches are limited to a small set of temporal patterns and only deal with uncertainty in facts. Knowledge base expansion and query-driven inference based on Markov Logic Networks (MLNs) have been studied in [10]. Contrary to TeCoRe, the knowledge bases considered are not temporal. Despite the general complexity of MLNs, it has been shown that this tool can be used to reason about facts extracted at Web scale using a combination of hand-crafted and extracted inference rules. MLNs can be used to deal with temporal relations in open information extraction [6] or check the consistency of knowledge bases [4].

2. SYSTEM OVERVIEW

TeCoRe is a tool that allows to detect temporal conflicts thanks to the combination of: (i) temporal KGs with uncertain information expressed by assigning to each temporal fact a confidence value; (ii) temporal inference rules that include arithmetic predicates; (iii) temporal constraints based on Allen's relations.

Data Model. In the demonstration, we focus on data encoded in the W3C standard RDF. Specifically, uncertain Temporal Knowledge Graphs (UTKGs) are represented as set of triples. Each triple of the form (s, p, o) can be thought of as an edge between the subject s and the object o labeled by the predicate p; hence a set of RDF triples is referred to as an *RDF graph*. We use the term *knowledge graph* loosely to refer to an RDF graph. An RDF graph can be extended with temporal information by labeling each triple in the graph with a temporal element. The time period in the temporal element represents 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. Besides, each temporal fact is assigned a confidence value representing how likely is for it to hold.

Hence, UTKGs are extensions of temporal KGs with probabilistic graphical models that are capable of representing uncertainties and reasoning over temporal knowledge bases. The semantics of a UTKG is based on a joint probability distribution over Herbrand interpretations of the uncertain part of the UTKG. In particular, the weights of the facts determine a log-linear probability distribution (in the case of MLN). As mentioned earlier, we assume that the time domain, in which the validity of facts is expressed, is finite as well as discrete; hence, the set of possible worlds is finite. An example of a UTKG which represents sport's personality Claudio Raineri's (CR) career, is shown in Figure 1.

Temporal Inference Rules. Inference rules are useful to derive new knowledge from existing knowledge. One way to use rules is to learn them from data. Alternatively, rules can also be handcrafted by domain experts. In this demonstration, applications and users provide their own temporal inference rules for driving implicit facts or relationships within data. A temporal inference rule has the following

```
(1) (CR, coach, Chelsea, [2000,2004])        0.9
(2) (CR, coach, Leicester, [2015,2017])      0.7
(3) (CR, playsFor, Palermo, [1984,1986])     0.5
(4) (CR, birthDate, 1951, [1951,2017])       1.0
(5) (CR, coach, Napoli, [2001,2003])         0.6
```

Figure 1: A UTKG G about coach Claudio Raineri (CR).

form: Body $\wedge$ [Condition] $\rightarrow$ Head. If a *Body* together with a *Condition* holds in a formula, we can infer the *Head*. The condition is an optional parameter which is used to embed Allen's interval relations and other relevant arithmetic predicates (e.g. $age > 40$).

Temporal Constraints. In relational databases, integrity constraints are used to detect inconsistencies in data. Similarly, in order to detect conflicts in UTKGs, we use constraints. We provide users a language –based on Datalog– to design constraints. To debug uncertain KGs we introduce a set of constraints that become *hard* (deterministic) or *soft* (uncertain) formulas in MLNs and PSL. We introduce three different kinds of constraints: (i) inclusion dependencies with inequalities, (ii) (in)equality generating dependencies, and (iii) disjointness constraints [4].

2.1 Architecture

The system architecture of TeCoRe is shown in Figure 2. In what follows we provide a brief description of its main components.

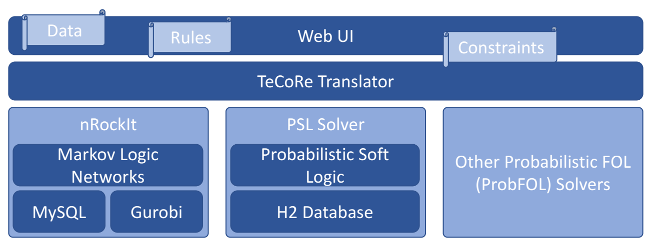

Figure 2: TeCoRe system overview.

Web UI. Users interact with TeCoRe through a Web interface. Users can select temporal KGs, inference rules, and constraints in order to compute the most probable conflict free temporal KG. Within TeCoRe it is possible to specify temporal constraints for relations using Allen's interval algebra. Once a temporal KG has been selected, users will be redirect to an interface where relations between predicates can be set by using Allen's relations. For instance, if a user selects the relations birthDate and worksFor, and specifies the Allen relation before, because a person must be born before she works for a company.

TeCoRe Translator. The translator parses data, inference rules, and temporal constraints, and transforms those into the specific syntax of the chosen solver (e.g. nRockIt, PSL). Special care is taken to verify that the input adheres to the expressivity of the solver.

Markov Logic Networks. In TeCoRe, *Markov Logic Networks* (MLNs) are used to reason over UTKGs. In particular, MLNs are templates for constructing Boolean Markov Random Fields (MRFs). They combine MRFs and first-order logic (FOL) by attaching weights to first-order formulas [8].

In this demonstration, we use an extension of MLNs with numerical constraints [3], which is useful for reasoning in uncertain temporal KGs.

An MLN L with numerical constraints is a set of pairs (F_i, w_i), where F_i is a FOL formula that may contain a numerical constraint and w_i is a real number representing the weight of formula F_i. Together with a finite set of constants C, a MLN with numerical constraints 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:

$$P(X = x) = Z^{-1} \exp \Big[\sum_{i=1}^{N} w_i n_i(x) \Big]$$

where N is the number of formulas in the MLN and $n_i(x)$ is the number of true groundings of F_i in x. The groundings of a formula are formed simply by replacing its variables with constants in all possible ways. In this demo we use the state-of-the-art MLN solver *nRockIt* [3].

Probabilistic Soft Logic. PSL [1] allows scalable inference over large KGs by restricting the expressivity of the rules and constraints. In particular, PSL uses first-order logic to specify templates for Hinge-Loss Markov Random Fields that are defined over continuous variables. That is, PSL is defined over random variables with soft truth values in the interval $[0, 1]$ whereas MLNs are defined over Boolean variables. In addition, PSL formulas are restricted to rules with conjunctive bodies.

A common inference task with MLNs and PSL 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* (MAP) inference. Our experimental findings indicate that MLN solvers do not scale well. This comes as no surprise due to the complexity of inference in MLN [8]. Thus we also offer the possibility to use PSL, which trades expressiveness for scalability. We *implemented* a numerical extension on top of PSL for temporal reasoning called *nPSL*.

ProbFOL solvers. In this demonstration we run TeCoRe on nRockIt (for MLNs) and PSL solver. An additional benefit of the architecture is that any off-the-shelf probabilistic first-order logic (ProbFOL) system (e.g., Tuffy, ProbLog, DeepDive) with numerical support can be seamlessly integrated into the TeCoRe system by extending the translator.

Temporal Inference. A UTKG can be mapped into a first-order knowledge base by transforming every temporal fact into a *quad* atom. Given a UTKG $\mathcal{G}$, a set of temporal inference rules $\mathcal{F}$, a set of temporal constraints $\mathcal{C}$, and a translation function $\boldsymbol{\theta}$, we denote the MAP problem by $map(\boldsymbol{\theta}(G), \mathcal{F} \cup \mathcal{C})$. Computing this function requires to translate $\mathcal{G}$ with the function $\boldsymbol{\theta}$ into an equivalent Markov logic formalization. Then, the inference rules $\mathcal{F}$ and constraints $\mathcal{C}$ are added to this translation. The MAP state is computed with the help of nRockIt (resp. PSL solver) applied to this input data. To do so, the evidence clauses $\boldsymbol{\theta}(G)$ and the grounding of $\mathcal{F}$ with respect to $\boldsymbol{\theta}(\mathcal{G})$ are given as input. TeCoRe applies MAP inference to compute, $\mathrm{map}(\boldsymbol{\theta}(G), \mathcal{F} \cup \mathcal{R})$, the most probable and expanded $G_{inferred}$. After inference is completed, users and applications have access to the expanded KG, i.e., implicit facts from G have been made explicit using the inference rules upon reasoning.

3. RUNNING EXAMPLE

The goal of the demonstration is to cover a variety of scenarios rather than focusing on a single one. We will specifically stimulate the audience to pick examples from different knowledge domains; nevertheless for exposition we develop a main example in the domain of sport. Given the UTKG in Figure 1 we want to compute the most probable conflict-free temporal knowledge graph. The user selects a UTKG and a set of temporal inference rules and constraints. These pieces of information are dealt with via the interface shown in Figure 3.

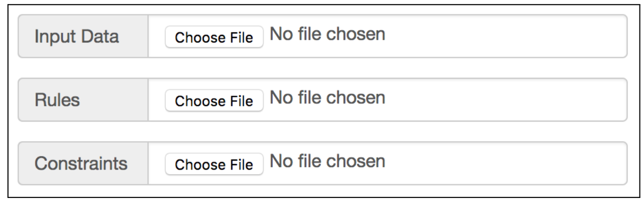

Figure 3: Interface to select the input data, inference rules, and temporal constraints.

For the example above, an exemplary set of temporal inference rules is shown in Figure 4; f_1 expresses the fact that if a footballer plays for a club, she works for that club; f_2 tells us that if a footballer works for a club and that club is located in a city, then she lives in the same city; and f_3 deals with the fact that if a footballer plays for a club and is less than 20 years, then she is a teen footballer.

$$f_1 \quad \mathsf{quad}(x, \mathsf{playsFor}, y, t) \rightarrow \mathsf{quad}(x, \mathsf{worksFor}, y, t) \qquad \mathbf{w} = 2.5$$

$$f_2 \quad \mathsf{quad}(x, \mathsf{worksFor}, y, t) \wedge \mathsf{quad}(y, \mathsf{locatedIn}, z, t') \wedge \mathit{overalps}(t, t') \rightarrow \mathsf{quad}(x, \mathsf{livesIn}, z, t'' = t \cap t') \qquad \mathbf{w} = 1.6$$

$$f_3 \quad \mathsf{quad}(x, \mathsf{playsFor}, y, t) \wedge \mathsf{quad}(x, \mathsf{birthDate}, z, t') \wedge t' - t < 20 \rightarrow \mathsf{quad}(x, \mathsf{type}, \mathsf{TeenPlayer}) \qquad \mathbf{w} = 2.9$$

Figure 4: Temporal inference rules $\mathcal{F}$.

To facilitate the writing of constraints TeCoRe features the Constraints Editor shown in Figure 5. Here, predicates in a UTKG can be selected (also with the help of auto-completion) and constrained via Allen's temporal relations.

For this example, the set of constraints in Figure 6 express that a person must be born before she dies (c_1), a person cannot coach two clubs at the same time (c_2), and a person cannot be born in two different cities (c_3). The predicates *overlaps*, *disjoint* and *before* are Allen's interval relations.

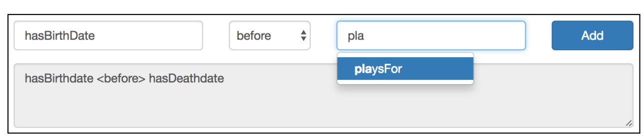

Figure 5: Constraints editor (predicate auto-completion).

Results. The UTKG in Figure 7 is obtained after computing $map(\boldsymbol{\theta}(G), \mathcal{F}, \mathcal{C})$. It represents the most probable, expanded and conflict free temporal KG. The temporal fact (5) is

c_1 $\mathsf{quad}(x, \mathsf{birthDate}, y, t) \wedge \mathsf{quad}(x, \mathsf{deathDate}, z, t')$

$\rightarrow \mathit{before}(t, t')$ $\quad \mathbf{w} = \infty$

c_2 $\mathsf{quad}(x, \mathsf{coach}, y, t) \wedge \mathsf{quad}(x, \mathsf{coach}, z, t')$

$\wedge\, y \neq z \rightarrow \mathit{disjoint}(t, t')$ $\quad \mathbf{w} = \infty$

c_3 $\mathsf{quad}(x, \mathsf{bornIn}, y, t) \wedge \mathsf{quad}(x, \mathsf{bornIn}, z, t')$

$\wedge\, \mathit{overlap}(t, t') \rightarrow y = z$ $\quad \mathbf{w} = \infty$

Figure 6: Temporal constraints $\mathcal{C}$.

removed during inference because of the constraint c_2, i.e., a coach cannot manage two clubs at the same time. Due to this, there is a clash between the temporal facts (1) and (5) (see Figure 3), the later is removed since it has inferior weight. On the other hand, as shown in Figure 8, we used TeCoRe to compute the number of conflicting facts (19,734) from a UTKG containing 243,157 temporal facts.

```
(1) (CR, coach, Chelsea, [2000,2004])
(2) (CR, coach, Leicester, [2015,2017])
(3) (CR, playsFor, Palermo, [1984,1986])
(4) (CR, birthDate, 1951, [1951,2017])
```

Figure 7: Temporal facts $G_{inferred}$ after MAP inference.

Performance of MAP Inference. We used nRockIt and PSL to conduct the experiments. nRockIt, using MLNs, allows to use more expressive constraints than PSL. However, PSL scales well since it computes a soft approximation of the discrete MAP state. The running times on a desktop machine for performing MAP inference, on the uncertain temporal graph FootballDB, for nRockIt and nPSL is 12,181ms and 6,129ms, respectively. Running times are averaged over 10 runs.

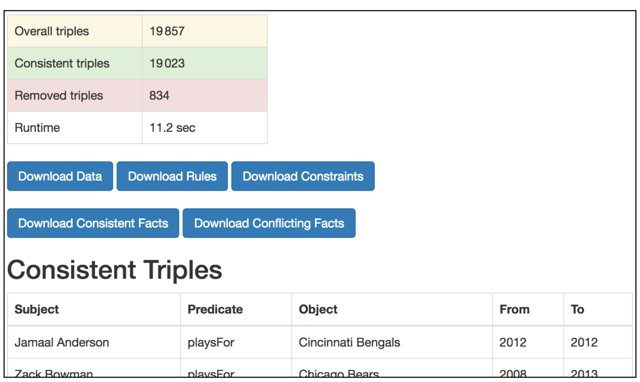

Figure 8: Display of result statistics and result data (browsable consistent and conflicting statements).

4. DEMONSTRATION

The demonstration will start by providing a very brief overview about the notion of UTKG (e.g., showing excerpts of Google Knowledge Vault or YAGO). Then, we will proceed by presenting concrete examples of data, inference rules and constraints, along the same line of the example discussed in Section 3, and discuss the results. This will serve as a basis to underline the following aspects: (i) managing both temporal and uncertain information is a real need; (ii) the set of constraints and inference rules is domain-specific and can be provided by domain experts (also with the help of automatic tools).

During the demo we will focus on the following datasets: (i) *FootballDB*: we extracted temporal facts about American football players from footballdb.com, that contains two important relations: *playsFor* and *birthDate*. In particular, we will focus on a set of >13K temporal facts for the *playsFor* relation and >6K facts for the *birthDate* relation; (ii) *Wikidata*: we extracted over 6.3 million temporal facts. Some of the relations that will be used include *playsFor* (>4 million facts), *educatedAt* (>6K), *memberOf* (>23K), *occupation* (>4.5K) and *spouse* (>20K).

We will let the audience have the opportunity to modify a set of predefined constraints and inference rules that we will provide, or suggest new constraints and inference rules. The Web UI of TeCoRe allows to involve a broader audience since anybody can interact with the tool on their own laptop.

Goals. The broad goal of the demonstration is to underline the importance of managing *both* temporal and uncertain facts in existing KGs. The desideratum is that by experiencing with TeCoRe, the audience will realize its potential and the importance of the problem that it addresses. Another goal that we set is stimulating new research on this topic. The idea is to discuss the pros and cons of our approach with particular emphasis on the following aspects: (i) inference expressiveness and scalability (i.e., nRockIt versus PSL); (ii) automatic derivation or suggestion of constraints and inference rules.

5. REFERENCES

[1] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor. Hinge-loss markov random fields and probabilistic soft logic. *CoRR*, abs/1505.04406, 2015.

[2] M. J. Cafarella, A. Halevy, and J. Madhavan. Structured data on the Web. *Communications of the ACM*, 54(2):72–79, 2011.

[3] M. W. Chekol, J. Huber, C. Meilicke, and H. Stuckenschmidt. Markov logic networks with numerical constraints. In *Proc. of ECAI 2016*, pages 1017–1025, 2016.

[4] M. W. Chekol, G. Pirrò, J. Schoenfisch, and H. Stuckenschmidt. Marrying Uncertainty and Time in Knowledge Graphs. In *Proc. of AAAI 2017*, 2017.

[5] M. Dylla, I. Miliaraki, and M. Theobald. A Temporal-probabilistic Database Model for Information Extraction. *VLDB Endowment*, 6(14):1810–1821, 2013.

[6] X. Ling and D. S. Weld. Temporal Information Extraction. In *AAAI*, pages 1385–1390, 2010.

[7] A. Papaioannou and M. H. Böhlen. TemProRA: Top-k temporal-probabilistic results analysis. In *ICDE*, pages 1382–1385, 2016.

[8] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[9] S. Schlobach, Z. Huang, R. Cornet, and F. Van Harmelen. Debugging Incoherent Terminologies. *J. Automated Reasoning*, 39(3):317–349, 2007.

[10] X. Zhou, Y. Chen, and D. Z. Wang. ArchimedesOne: Query Processing over Probabilistic Knowledge Bases. *VLDB Endowment*, 9(13):1461–1464, 2016.