TCL : Apprentissage de contraintes temporelles à partir de graphes de connaissances temporels

Djawad Bekkoucha[1], Nadjib Lazaar[1], Pierre-Henri Paris[1], Fatiha Saïs[1]

[1] Université Paris-Saclay, CNRS,
Laboratoire Interdisciplinaire des Sciences du Numérique,
91190, Gif-sur-Yvette, France

firstname.lastname@lisn.fr

Résumé

Les graphes de connaissances temporels (Temporal Knowledge Graphs, TKG) encodent de grandes collections de faits horodatés. Ces informations présentent fréquemment des incohérences temporelles, ce qui limite l'abstraction et le raisonnement. Nous modélisons les relations temporelles entre faits via des réseaux de contraintes temporelles (Temporal Constraint Networks, TCN) et proposons TCL, une approche apprenant un ensemble compact de contraintes temporelles à partir des TKG. Afin d'en garantir la cohérence, nous proposons un algorithme de propagation inspiré de la cohérence de chemin. Les TCN résultants fournissent une représentation compacte de la structure temporelle implicite des TKG et peuvent servir d'entrées structurées pour le raisonnement temporel, la détection d'incohérences, la validation et l'inférence prédictive. Expérimentalement, nous montrons la qualité des TCN produits par TCL à travers le taux de réduction de domaines obtenu, la qualité des inférences générées et le support des contraintes apprises, mettant ainsi en évidence l'intérêt de combiner informations observées et connaissances inférées.

Mots-clés

Graphe de connaissances temporel, Cohérence des graphes de connaissances, Réseau de contraintes qualitatives, Acquisition de contraintes

Abstract

Temporal Knowledge Graphs (TKGs) encode large collections of time-stamped facts. However, this information is often temporally inconsistent, which limits the effectiveness of abstraction and temporal reasoning. In this work, we model temporal relations between facts using Temporal Constraint Networks (TCNs). We introduce TCL, an approach that learns a compact set of temporal constraints directly from TKGs. To ensure the coherence of the learned network, we develop a propagation algorithm inspired by path consistency. The resulting TCNs provide a compact and informative representation of the temporal structure implicitly encoded in TKGs. They can serve as structured inputs for downstream tasks such as temporal reasoning, inconsistency detection, knowledge graph validation, and predictive inference. Experimentally, we demonstrate the quality of the TCNs produced by TCL through the obtained domains reduction rate, the quality of the generated inferences, and the support of the learned constraints, thereby highlighting the benefit of combining observed information with inferred knowledge.

Keywords

Temporal Knowledge Graph, Knowledge Graph Consistency, Qualitative Constraint Network, Constraint Acquisition

1 Introduction

Les graphes de connaissances temporels (*Temporal Knowledge Graphs*, TKGs) sont des structures de données fondamentales pour représenter des faits complexes associés à des intervalles de validité temporelle explicites, difficiles à modéliser à l'aide de représentations tabulaires. En enrichissant les faits par des dimensions temporelles, les TKGs permettent l'analyse de phénomènes dynamiques et facilitent l'extraction de connaissances ancrées dans le temps. De grandes bases de connaissances collaboratives telles que Wikidata Vrandecic and Krötzsch [2014] et YAGO Suchanek et al. [2024] intègrent de plus en plus d'informations temporelles, offrant ainsi des ressources précieuses pour le raisonnement temporel. Toutefois, la nature collaborative de ces plateformes les rend vulnérables à des problèmes de qualité des données. Les contributeurs peuvent introduire des annotations temporelles incorrectes, imprécises ou incohérentes avec les faits associés. Les intervalles temporels peuvent contredire d'autres faits temporels ou violer des contraintes d'ordre implicites entre propriétés. Par exemple, un fait peut associer à *Zinédine Zidane* une date de fin de carrière en 2006, tandis qu'un autre fait indique une participation à la Coupe du monde 2010. Cette configuration est incohérente, car elle viole une contrainte temporelle implicite selon laquelle toute participation à une compétition doit être incluse dans l'intervalle de carrière d'un joueur. Ces incohérences se propagent ensuite dans les processus analytiques en aval, introduisant des biais dans l'extraction de connaissances, le raisonnement et les tâches de complétion. Assurer la cohérence temporelle est donc une étape

nécessaire pour une exploitation fiable des TKGs. En pratique, la cohérence temporelle est souvent négligée dans les ontologies existantes.

Le traitement de ce problème requiert des structures de données capables de représenter efficacement la structure temporelle globale d'un TKG, ainsi que des mécanismes de détection des incohérences globales. Une validation à l'échelle des faits est insuffisante, car de nombreuses incohérences sont dues à des violations de régularités structurelles entre propriétés plutôt qu'à des erreurs isolées. Une vue à l'échelle du schéma est donc nécessaire pour capturer ces régularités de manière concise et interprétable.

Dans ce travail, nous présentons *TCL*, une approche qui modélise la structure temporelle des TKGs à l'aide de réseaux de contraintes temporelles (Temporal Constraint Networks, TCNs), une sous-classe des réseaux de contraintes qualitatives. Ces réseaux s'inscrivent dans les cadres de raisonnement temporel qualitatif tels que l'algèbre d'Allen Allen [1983] et ont été largement étudiés dans les travaux sur l'acquisition de contraintes qualitatives Sioutis and Condotta [2017]; Belaid et al. [2024]; Condotta et al. [2016]. Ils offrent un formalisme symbolique et interprétable pour représenter et raisonner sur des relations temporelles. De plus, leurs mécanismes de propagation permettent de réduire l'ensemble des relations temporelles admissibles tout en évaluant la cohérence globale du réseau. Ces mécanismes permettent à la fois l'élimination des configurations temporelles non supportées et la détection d'incohérences structurelles issues de combinaisons de contraintes.

Contrairement aux approches existantes qui considèrent des TCNs avec des domaines de relations binaires (où une relation est simplement présente ou absente), *TCL* associe une valeur de support à chaque relation temporelle. Cette valeur reflète la confiance dérivée des intervalles observés. Ces valeurs de support sont obtenues via un oracle automatisé intégré dans l'algorithme d'apprentissage. Cet oracle fournit des informations de référence sur les relations temporelles observées entre des paires de faits comparables. Dans notre approche, deux faits sont comparables s'ils décrivent une même entité par des propriétés temporelles distinctes. Ces relations sont estimées en comparant uniquement des faits associés à une même entité, car seuls ces faits sont temporellement comparables. Cette formulation, prenant en compte le support permet aux mécanismes de propagation et de réparation de raisonner non seulement sur les relations possibles, mais aussi sur leur niveau de confiance relatif, guidant ainsi la résolution de conflits et l'inférence de relations manquantes. En outre, *TCL* intègre un mécanisme de propagation spécifiquement conçu pour le contexte des TKGs. Ce mécanisme propage les relations observées afin de réduire l'ensemble des relations possibles, en ne conservant que celles qui sont essentielles. Il permet également d'inférer de nouvelles relations pour les paires de propriétés pour lesquelles l'oracle ne fournit pas d'information directe. Un mécanisme de réparation est intégré afin de traiter les domaines temporellement incohérents, garantissant que le TCN résultant est localement cohérent.

Via une étude expérimentale, nous démontrons la qualité du TCN produit par notre approche en répondant à plusieurs questions de recherche. Tout d'abord, nous évaluons les taux de réduction des domaines obtenus par *TCL*, en montrant qu'il ne conserve que les relations structurellement nécessaires. Nous évaluons également la capacité d'inférence de notre propagateur, qui permet d'inférer un grand nombre de nouvelles relations à partir d'un ensemble restreint de relations observées. Enfin, nous analysons la qualité des relations restantes en examinant leurs valeurs de support à l'issue du processus d'apprentissage. Ces supports peuvent être soit observés via l'oracle, soit inférés par le propagateur.

L'article est organisé comme suit. La section 2 situe notre travail par rapport à l'état de l'art. La section 3 présente les notions préliminaires et les notations. La section 4 présente en détail *TCL*. La section 5 présente l'évaluation expérimentale de notre méthode, et enfin, la section 6 conclut et discute des perspectives de recherche.

2 État de l'art

La découverte de contraintes temporelles à partir de TKGs n'a pas reçu une grande attention dans les travaux antérieurs. Les approches existantes en profilage de données pour les bases de données relationnelles Abedjan et al. [2018], telles que Consonni et al. [2019]; Xiao et al. [2022], se concentrent principalement sur l'identification de relations d'ordre à l'aide d'opérateurs de comparaison (par exemple, $\leq$, $<$). Cependant, ces méthodes n'intègrent pas d'opérateurs spécifiques aux intervalles, pourtant essentiels pour un raisonnement temporel expressif et robuste. De manière similaire, les approches qualitatives fondées sur des algèbres d'intervalles, telles que l'algèbre d'Allen Allen [1983] ou le raisonnement basé sur les QCN Belaid et al. [2024]; Mensi et al. [2018]; Paparrizou and Sioutis [2025], traitent le problème de la cohérence temporelle, mais ne sont pas appliquées à la découverte automatique de contraintes à partir de TKGs.

Plus récemment, Soulard et al. [2025] proposent une méthode de découverte de contraintes temporelles utilisées ensuite pour la vérification de faits temporels. Dans leur approche, les contraintes sont extraites via un mécanisme d'exploration du TKG. Une contrainte temporelle candidate est acceptée si elle est satisfaite dans le TKG par un nombre de paires de faits supérieur à un seuil prédéfini. Cependant, la cohérence des contraintes découvertes entre elles n'est pas vérifiée, et le choix d'un seuil approprié est difficile, car une valeur inadéquate peut soit éliminer des contraintes pertinentes, soit conserver des contraintes bruitées.

Les approches fondées sur des motifs, telles que *PaTeCon* Chen et al. [2023], définissent des langages de motifs de graphes structurels afin de capturer des configurations fréquentes de faits temporels. Les contraintes sont évaluées à l'aide de mesures statistiques locales, notamment le support et la confiance au niveau des entités. Bien que cela permette une découverte automatique de contraintes temporelles, l'évaluation est réalisée indépendamment pour chaque motif, ce qui limite la capacité à garantir une co-

hérence globale ou à inférer des informations temporelles manquantes. Cette approche souffre également de la difficulté intrinsèque du choix des seuils, le niveau optimal de support ou de confiance étant souvent inconnu et dépendant du jeu de données.
Notre approche garantit la cohérence entre l'ensemble des contraintes temporelles découvertes et les intègre dans un cadre de raisonnement unifié. Chaque contrainte est associée à un seuil de support interprété comme un indicateur de confiance plutôt que comme un critère strict de filtrage. Les contraintes présentant un faible support empirique sont conservées lorsqu'elles sont structurellement nécessaires au maintien de la cohérence globale du TKG. Cela permet d'effectuer des inférences sur des faits pour lesquels aucune information temporelle explicite n'est disponible, améliorant ainsi la fiabilité des tâches de validation et de raisonnement temporels.

3 Préliminaires et Notations

Cette section introduit le cadre formel sous-jacent à notre approche. Nous définissons les graphes de connaissances temporels ainsi que la représentation temporelle qualitative utilisée pour modéliser les relations entre faits temporels. Ces fondations permettent la formalisation du problème d'apprentissage de contraintes temporelles traité dans ce travail.

Definition 1 (Graphe de connaissances temporel) *Soit E un ensemble fini d'entités, P un ensemble fini de propriétés, et L un ensemble fini de valeurs littérales. Un* graphe de connaissances temporel *est noté $\mathcal{TKG}$ et défini comme un ensemble de faits temporels de la forme (e, p, o, I), où $e \in E$ est l'entité sujet, $p \in P$ est une propriété, $o \in E \cup L$ est l'objet (entité ou littéral), et $I = [d, f]$ est un intervalle temporel fermé représentant la période de validité du fait. Les faits instantanés sont représentés par des intervalles dégénérés tels que $I.d = I.f$.*
Dans ce travail, nous adoptons une sémantique en monde clos pour $\mathcal{TKG}$, ce qui signifie que seuls les faits temporels explicitement observés sont considérés dans le processus de raisonnement ultérieur.

Les définitions suivantes reposent sur un langage temporel qualitatif $\mathcal{L} = \{r_1, \ldots, r_m\}$ dérivé de l'algèbre des intervalles d'Allen Allen [1983]. En particulier, $\mathcal{L}$ est composé des 13 relations temporelles élémentaires entre intervalles temporels introduites par Allen : *before*, *meets*, *overlaps*, *starts*, *during*, *finishes* et *equals*, ainsi que leurs relations converses respectives *after*, *met-by*, *overlapped-by*, *started-by*, *contains* et *finished-by*. Les relations de $\mathcal{L}$ sont conjointement exhaustives et mutuellement exclusives (JEPD). L'algèbre relationnelle induite par $\mathcal{L}$ est finie, fermée par converse et composition faible, et contient la relation identité.
Chaque relation $r \in \mathcal{L}$ est interprétée comme un prédicat binaire sur des intervalles temporels. Étant donnés deux intervalles I_1 et I_2, le prédicat $r(I_1, I_2)$ signifie que la relation r est vérifiée entre I_1 et I_2 selon la sémantique d'Allen. Par exemple, la relation $equal(I_1, I_2)$ est vérifiée si et seulement si

$$I_1.d = I_2.d \ \wedge \ I_1.f = I_2.f. \tag{1}$$

Definition 2 (Valeur de support) *Soit $\mathcal{TKG}$ un graphe de connaissances temporel et $p_i, p_j \in P$ deux propriétés temporelles telles que $p_i \neq p_j$. Soit $r \in \mathcal{L}$ une relation temporelle.*
L'ensemble des paires de faits comparables associées à (p_i, p_j) est défini par

$$\Omega_{ij} = \left\{ \left((e, p_i, o, I), (e, p_j, o', I') \right) \in \mathcal{TKG} \times \mathcal{TKG} \right\}.$$

Le sous-ensemble des paires satisfaisant la relation r est

$$\Omega_{ij}^r = \{ ((e, p_i, o, I), (e, p_j, o', I')) \in \Omega_{ij} \mid r(I, I') \}.$$

La fonction de support

$$s : \mathcal{L} \times (P \times P) \to [0, 1] \cup \{\nu\}$$

associe à chaque relation r et paire ordonnée (p_i, p_j) la valeur

$$s(r, (p_i, p_j)) = \begin{cases} \dfrac{|\Omega_{ij}^r|}{|\Omega_{ij}|} & \textit{si } |\Omega_{ij}| > 0, \\ \nu & \textit{sinon}. \end{cases}$$

La valeur ν désigne un support indéfini résultant de l'absence de paires de faits comparables.

Definition 3 (Contrainte temporelle binaire) *Soient $p_i, p_j \in P$ deux propriétés temporelles telles que $i < j$, et soit $\mathcal{L}$ l'ensemble des relations temporelles élémentaires d'Allen.*
Une contrainte temporelle binaire entre p_i et p_j, notée C_{ij}, est un ensemble fini $C_{ij} \subseteq \mathcal{L} \times ([0, 1] \cup \{\nu\})$, où chaque paire (r, s) associe une relation temporelle $r \in \mathcal{L}$ à sa valeur de support $s = s(r, (p_i, p_j)) \in ([0, 1] \cup \{\nu\})$.
Intuitivement, C_{ij} représente une disjonction de relations temporelles admissibles entre p_i et p_j, chacune pondérée par son support empirique.
La contrainte inverse est définie par

$$C_{ji} = C_{ij}^{-1} = \{ (r^{-1}, s) \mid (r, s) \in C_{i,j} \}.$$

On note $\perp$ la contrainte vide, et $\top$ la contrainte universelle autorisant toutes les relations $r \in \mathcal{L}$ avec des supports non nuls ou indéfinis.

Definition 4 (Réseau de contraintes qualitatives) *Soit $\mathcal{TKG}$ un graphe de connaissances temporel et soit $\mathcal{K}$ une classe d'entités dans $\mathcal{TKG}$. Le réseau de contraintes temporelles (TCN) associé à $\mathcal{K}$ est défini comme un couple (V, E) où :*

- *V est l'ensemble des propriétés temporelles instanciées par les entités de la classe $\mathcal{K}$;*
- *$E : V \times V \to 2^{\mathcal{L} \times ([0,1] \cup \{\nu\})}$ associe à chaque paire de propriétés (p_i, p_j) une contrainte temporelle binaire C_{ij}.*

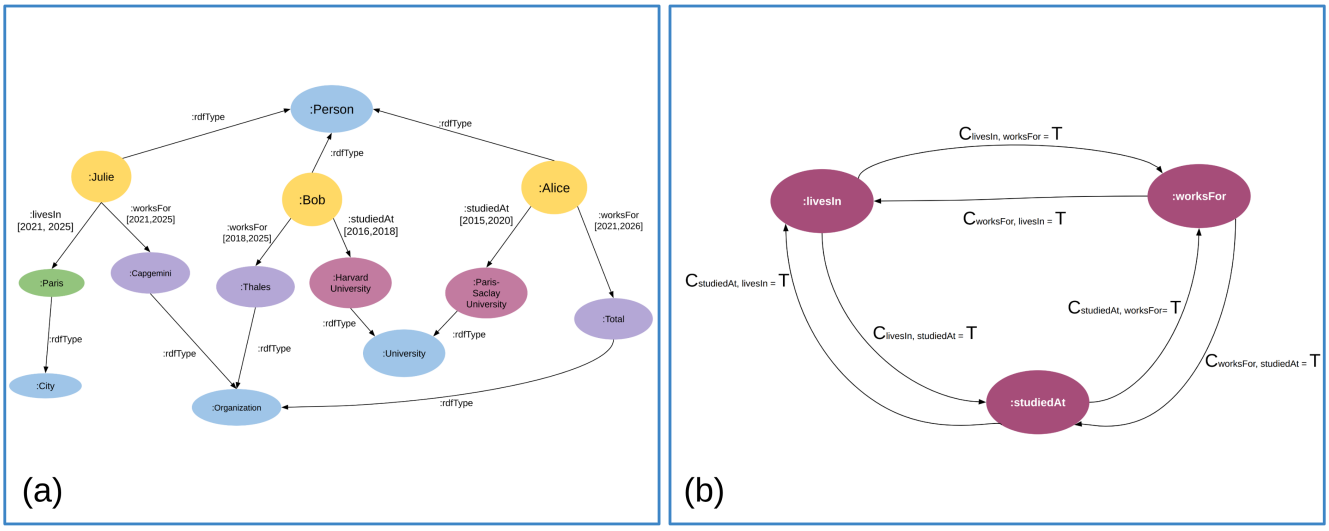


FIGURE 1 – (a) Un graphe de connaissances temporel et (b) le réseau de contraintes temporelles correspondant.

Example 1 *La Figure 1 illustre la construction d'un réseau de contraintes temporelles à partir d'un graphe de connaissances temporel. Dans la Figure 1(a), nous présentons un graphe de connaissances temporel contenant les entités* `:Julie`, `:Bob` *et* `:Alice`, *les propriétés temporelles* `:livesIn`, `:worksFor` *et* `:studiedAt`, *ainsi que les entités* `:Capgemini`, `:Thales`, `:Total`, `:HarvardUniversity` *et* `:ParisSaclayUniversity` *comme objets. Dans la Figure 1(b), nous présentons un réseau de contraintes temporelles construit à partir de ce graphe, où chaque paire de propriétés temporelles est initialement associée à l'ensemble complet des relations d'Allen, i.e., la relation universelle* $(\top)$.

Definition 5 (Cohérence d'un TCN) *Soit* $N = (V, E)$ *un réseau de contraintes temporelles. Le réseau* N *est dit* cohérent *(ou satisfiable) s'il existe une affectation* α *associant à chaque propriété temporelle* $p_i \in V$ *un intervalle temporel* $\alpha(p_i)$ *telle que toutes les contraintes temporelles soient satisfaites simultanément, i.e., pour toute paire* $(p_i, p_j) \in V^2$, *il existe une relation* $r \in C_{ij}$ *telle que* $r(\alpha(p_i), \alpha(p_j))$.

Definition 6 (Apprentissage de contraintes temporelles) *Soit* $\mathcal{TKG}$ *un graphe de connaissances temporel. Le problème d'apprentissage de contraintes temporelles (TCL) consiste à apprendre, à partir des faits temporels contenus dans* $\mathcal{TKG}$, *un réseau de contraintes temporelles* $\mathcal{N} = (V, E)$ *tel que :*

- V *est un ensemble de propriétés temporelles extraites de* $\mathcal{TKG}$ *;*
- E *est un ensemble de contraintes temporelles binaires* $C_{i,j}$ *apprises entre paires ordonnées de propriétés* (p_i, p_j) *à partir des intervalles temporels observés ;*

et tel que le réseau $\mathcal{N}$ *admette au moins une solution, i.e., qu'il soit cohérent.*

Dans la suite de cet article, pour simplifier, nous utilisons le terme *contrainte binaire* ou *contrainte* pour désigner une contrainte temporelle binaire, le terme *relation* pour désigner une relation temporelle d'Allen, et le terme *propriété* pour désigner une arête d'un $\mathcal{TKG}$ tel que défini dans la définition. 1.

4 *TCL* : Un cadre pour l'apprentissage de contraintes temporelles à partir de TKGs

Cette section introduit *TCL*, un cadre pour l'acquisition de réseaux de contraintes temporelles qualitatives cohérents, à partir de graphes de connaissances temporels. L'approche proposée formule l'apprentissage de contraintes temporelles comme un problème d'acquisition de contraintes, reposant sur une interaction entre des données observationnelles et un oracle automatisé Belaid et al. [2022, 2024]. Étant donné un graphe de connaissances temporel, *TCL* apprend progressivement des contraintes temporelles tout en garantissant la cohérence globale du réseau résultant. Le cadre repose sur deux phases étroitement couplées. Dans un premier temps, un oracle automatisé est interrogé afin d'extraire des relations temporelles candidates, soutenues par les faits temporels observés dans le graphe de connaissances. Dans un second temps, ces relations sont intégrées dans un processus de raisonnement par contraintes qui propage l'information temporelle et infère des relations supplémentaires par propagation qualitative. Cette interaction entre acquisition guidée par l'oracle et propagation logique permet de construire un réseau de contraintes temporelles qui reste globalement cohérent. Les deux phases sont détaillées ci-dessous.

4.1 Interrogation de l'oracle

L'interaction avec l'oracle nécessite un langage de requête permettant d'extraire de l'information temporelle à partir du graphe de connaissances temporel. Nous introduisons une classe de requêtes, appelées requêtes de relations tem-

porelles, ainsi que la sémantique associée de l'oracle.

Definition 7 (Requête de relation temporelle) *Soit $\mathcal{TKG}$ un graphe de connaissances temporel et P l'ensemble de ses propriétés temporelles. Une* requête de relation temporelle TR-QUERY*(p_i,p_j), avec $(p_i, p_j) \in P \times P$ et $i < j$, retourne la contrainte temporelle binaire C_{ij} en associant à chaque relation d'Allen $r \in \mathcal{L}$ une valeur de support $s(r, (p_i, p_j)) \in [0, 1] \cup \{\nu\}$, où ν indique qu'aucune entité du graphe de connaissances n'est décrite à la fois par p_i et p_j.*

Dans notre cadre, l'oracle est entièrement automatisé et ne nécessite aucune intervention humaine. Étant donnée une requête de relation temporelle TR-QUERY(p_i,p_j), l'oracle l'évalue en interrogeant $\mathcal{TKG}$ et en calculant, pour chaque relation d'Allen $r \in \mathcal{L}$, la valeur de support $s(r, (p_i, p_j))$ (Définition 2). Il retourne directement la contrainte temporelle binaire correspondante C_{ij} (Définition 3), sans risque de mauvaise classification : si p_i et p_j coexistent dans $\mathcal{TKG}$, chaque relation reçoit un support empirique dans $[0, 1]$; sinon, toutes les relations sont associées à ν.

Example 2 *Considérons le graphe de connaissances temporelles de la Figure 1(a) ainsi que les requêtes de relations temporelles suivantes :*

1. TR-QUERY*(*: `worksFor`,: `studiedAt`*),*
2. TR-QUERY*(*: `livesIn`,: `worksFor`*).*
3. TR-QUERY*(*: `livesIn`,: `studiedAt`*).*

Pour la première requête, deux entités (: `Alice` *et* : `Bob`*) sont associées aux deux propriétés dans $\mathcal{TKG}$. Pour* : `Alice`*, les deux intervalles sont liés par la relation* `after`*, et pour* : `Bob` *par* `met-by`*. L'oracle renvoie donc*

$$C_{\texttt{worksFor},\texttt{studiedAt}} = \{(\texttt{after}, 0.5),\ (\texttt{met-by}, 0.5),\ (r, 0) \text{ pour tout autre } r \in \mathcal{L}\}. \quad (2)$$

Pour la deuxième requête, seule l'entité : `Julie` *est associée aux deux propriétés dans $\mathcal{TKG}$, et les intervalles correspondants sont reliés par* `equals`*. L'oracle renvoie donc :*

$$C_{\texttt{livesIn},\texttt{worksFor}} = \{(\texttt{equals}, 1.0),\ (r, 0)\ \forall r \in \mathcal{L}\}. \quad (3)$$

Pour la troisième requête, aucune entité dans $\mathcal{TKG}$ n'est associée à la fois à : `livesIn` *et* : `studiedAt`*, donc $|\Omega_{\texttt{livesIn},\texttt{studiedAt}}| = 0$ et l'oracle renvoie $C_{\texttt{livesIn},\texttt{studiedAt}} = \{(r, \nu) \mid r \in \mathcal{L}\}$, puisqu'aucune entité dans $\mathcal{TKG}$ n'est associée simultanément à* : `livesIn` *et* : `studiedAt`*.*

4.2 Propagation des relations temporelles

Dans la propagation classique de contraintes qualitatives, une relation temporelle est valide ou non entre deux intervalles, et la propagation suit les règles de composition de l'algèbre des intervalles d'Allen Allen [1983], qui déterminent l'ensemble des relations cohérentes lorsque deux contraintes sont combinées dans un chemin. Dans notre cadre, chaque relation $r \in \mathcal{L}$ est associée à une valeur de support reflétant son niveau de confiance empirique dans $\mathcal{TKG}$. La propagation doit donc mettre à jour non seulement les relations cohérentes, mais aussi leurs supports associés dans le réseau. Nous décomposons cette procédure en deux étapes : composition des relations et propagation des supports.

Definition 8 (Composition des relations) *Soient C_{ij} et C_{jk} deux contraintes temporelles binaires, et soit $\circ$ la table de composition d'Allen. La composition de C_{ij} et C_{jk}, notée $C_{ij} \circ C_{jk}$, est l'ensemble des relations $r \in \mathcal{L}$ telles qu'il existe $r' \in C_{ij}$ et $r'' \in C_{jk}$ avec $r \in r' \circ r''$.*

Definition 9 (Propagation des supports) *Soient C_{ij} et C_{jk} deux contraintes temporelles binaires. Pour chaque relation $r \in \mathcal{L}$, le support propagé de p_i vers p_k via p_j est défini par :*

$$\tilde{s}(r, (p_i, p_k)) = \max_{\substack{r' \in C_{ij},\, r'' \in C_{jk} \\ r \in r' \circ r''}} \min\big(s(r', (p_i, p_j)),\ s(r'', (p_j, p_k))\big) \quad (4)$$

Le support de r dans C_{ik} est ensuite mis à jour comme suit :

$$s(r, (p_i, p_k)) \leftarrow \max\big(s(r, (p_i, p_k)),\ \tilde{s}(r, (p_i, p_k))\big) \quad (5)$$

Cette définition s'inspire du cadre des contraintes temporelles floues proposé par Godo et Vila Godo and Vila [1995]. Intuitivement, pour chaque chemin de p_i à p_k via p_j, l'opérateur $\min$ capture le maillon le plus faible, garantissant qu'une relation composée ne peut pas être plus fiable que sa composante la moins soutenue. Lorsque plusieurs chemins produisent la même relation, l'opérateur $\max$ conserve la preuve la plus forte, renforçant ainsi l'inférence la plus fiable. Pour des raisons de concision, l'opérateur $\circ$ désigne désormais à la fois la composition standard d'Allen et la propagation associée des supports.

Example 3 *Soient p_i, p_j, p_k trois propriétés temporelles avec les contraintes suivantes :*

$$C_{ij} = \{(\texttt{before}, 0.8)\},$$

$$C_{jk} = \{(\texttt{meets}, 0.6),\ (\texttt{overlaps}, 0.9)\}.$$

D'après la table de composition d'Allen : $\texttt{before} \circ \texttt{meets} = \texttt{before}$ *et* $\texttt{before} \circ \texttt{overlaps} = \texttt{before}$*. Les deux chemins produisent la même relation, avec les supports propagés respectifs :*

$$\min(0.8, 0.6) = 0.6, \qquad \min(0.8, 0.9) = 0.8.$$

Le support final correspond à la justification la plus forte parmi tous les chemins :

$$\tilde{s}(\texttt{before}, (p_i, p_k)) = \max(0.6, 0.8) = 0.8.$$

La contrainte résultante est donc :

$$C_{ik} = \{(\texttt{before}, 0.8)\}.$$

Nous décrivons maintenant l'algorithme *TCL* ainsi que son propagateur dédié.

Algorithme 1 : Apprentissage d'un réseau de contraintes temporelles (TCL)

Input : Graphe de connaissances temporel $\mathcal{TKG}$
Output : Réseau de contraintes temporelles $\mathcal{N} = (V, E)$

1 $\mathcal{N} \leftarrow \{C_{p,p'} = \{(r_1, \nu), \ldots, (r_{13}, \nu)\} \mid p, p' \in P, p \neq p'\}$; `// Initialisation du TCN`
2 **foreach** $(p_i, p_j) \in P \times P : i < j$ **do**
3 $\quad C_{ij} \leftarrow$TR-QUERY(p_i,p_j) ;
4 $\quad C_{ij} \leftarrow C_{ij} \setminus \{(r, 0) \in C_{ij}\}$;
5 $\mathcal{N} \longleftarrow$ REPAIRPC$(\mathcal{N})$; `// Propagation de la cohérence locale`
6 **return** $\mathcal{N}$;

4.3 Algorithme d'apprentissage *TCL*

L'Algorithme 1 présente TCL, notre approche pour apprendre un réseau de contraintes temporelles $\mathcal{N}$ à partir d'un graphe de connaissances temporel. L'algorithme prend en entrée un graphe de connaissances temporel $\mathcal{TKG}$ et retourne un réseau de contraintes temporelles qualitatives $\mathcal{N}$. L'algorithme commence par initialiser $\mathcal{N}$ de sorte que, pour chaque paire de propriétés temporelles distinctes $p, p' \in P$, la contrainte correspondante $C_{p,p'}$ contienne l'ensemble complet des relations d'Allen $\mathcal{L}$, chacune avec un support initial fixé à `none` (ligne 1). Ensuite, l'algorithme parcourt toutes les paires de propriétés distinctes (lignes 2–3). Pour chaque paire (p, p'), une requête qualitative est soumise à l'oracle (ligne 3). L'oracle retourne, pour chaque relation d'Allen $r \in \mathcal{L}$, une valeur de support s_r, ou ν si aucune entité n'est décrite par les deux propriétés. Les relations de support nul sont alors supprimées de $C_{p,p'}$ (ligne 4), puis une procédure de propagation est appliquée afin de diffuser cette réduction dans le réseau, réduisant davantage les domaines des contraintes (ligne 5).

4.4 Algorithme de propagation de *TCL*

L'Algorithme 2 implémente le propagateur REPAIRPC, qui impose une cohérence locale sur le réseau de contraintes temporelles tout en garantissant que la sortie est toujours cohérente. L'algorithme maintient une file Q contenant toutes les paires ordonnées à traiter, ainsi qu'une matrice booléenne `repaired` indiquant quelles contraintes ont subi une réparation. À chaque itération, une paire (i, j) est extraite de la file et, pour chaque propriété intermédiaire p_k, deux directions de propagation sont considérées : le chemin direct $p_i \to p_j \to p_k$ et le chemin inverse $p_k \to p_i \to p_j$. Pour chaque direction (a, b, c), si C_{ab} et C_{bc} possèdent un support différent de ν, la contrainte composée $C_{ab} \circ C_{bc}$ est intersectée avec la contrainte courante C_{ac}. Si l'intersection est non vide et différente de C_{ac}, la contrainte est restreinte et la paire (a, c) est réinsérée dans Q pour propagation ultérieure. Si l'intersection est vide, une *réparation minimale* est appliquée : C_{ac} est directement défini comme $C_{ab} \circ C_{bc}$, et la paire est marquée dans `repaired` afin d'empêcher toute modification ultérieure de cette contrainte. Ce mécanisme de réparation constitue la principale différence

Algorithme 2 : Réparation de la cohérence de chemin (REPAIRPC)

Input : Un réseau de contraintes temporelles $\mathcal{N} = (V, E)$
Output : Un réseau de contraintes temporelles cohérent $\mathcal{N}' = (V, E')$

1 $Q \leftarrow \{(i, j) \mid 1 \leq i, j \leq |V|,\ i \neq j\}$
$\quad$ `repaired`$[i][j] \leftarrow false$ **pour tout** $i \neq j$
2 **while** $Q \neq \emptyset$ **do**
3 $\quad$ Choisir $(i, j) \in Q$;
4 $\quad$ **foreach** $k \in \{1, \ldots, |V|\}$ *tel que* $(i \neq k \neq j)$ **do**
5 $\quad\quad$ **foreach** $(a, b, c) \in \{(i, j, k),\ (k, i, j)\}$ **do**
6 $\quad\quad\quad$ **if** $\neg$`repaired`$[a][c]$ ***et*** C_{ab}, C_{bc} *ont un support différent de* ν **then**
7 $\quad\quad\quad\quad C'_{ac} \leftarrow C_{ac} \cap (C_{ab} \circ C_{bc})$;
8 $\quad\quad\quad\quad$ **if** $C'_{ac} \neq C_{ac}$ **then**
9 $\quad\quad\quad\quad\quad$ **if** $C'_{ac} = \emptyset$ **then**
10 $\quad\quad\quad\quad\quad\quad C_{ac} \leftarrow C_{ab} \circ C_{bc}$;
11 $\quad\quad\quad\quad\quad\quad$ `repaired`$[a][c] \leftarrow true$;
12 $\quad\quad\quad\quad\quad\quad$ `repaired`$[c][a] \leftarrow true$;
13 $\quad\quad\quad\quad\quad$ **else**
14 $\quad\quad\quad\quad\quad\quad C_{ac} \leftarrow C'_{ac}$;
15 $\quad\quad\quad\quad\quad\quad Q \leftarrow Q \cup \{(a, c)\}$;
16 $\quad\quad\quad\quad\quad C_{ca} \leftarrow C_{ac}^{-1}$;
17 **return** $\mathcal{N}' = (V, E)$;

avec la cohérence de chemin classique, garantissant qu'aucune contrainte n'est réduite à $\perp$ et que le réseau retourné est toujours cohérent.

4.5 Évaluation de la complexité

Dans cette section, nous présentons la complexité temporelle dans le pire cas de l'algorithme *TCL*. Cet algorithme étant composé de deux fonctions (T-QUERY$_{\mathcal{TKG}}$ et REPAIRPC), nous commençons par les analyser séparément.

Complexité temporelle de T-QUERY$_{\mathcal{TKG}}$. Dans le pire cas, T-QUERY$_{\mathcal{TKG}}$ s'exécute en $O(|\mathcal{TKG}|^2 \cdot |\mathcal{L}|)$, lorsque tous les faits sont associés à une seule entité et que chaque paire est évaluée pour toutes les relations.

Complexité temporelle de REPAIRPC. L'algorithme 2 commence par initialiser la file Q avec toutes les paires ordonnées de variables distinctes (Ligne 1). Cela nécessite $|V|(|V| - 1)$ éléments, soit un temps $O(|V|^2)$. La matrice booléenne `repaired` est également initialisée pour toutes les paires de variables, ce qui requiert $O(|V|^2)$ temps (Ligne 1). La boucle principale traite les paires selon une stratégie de liste de travail (Ligne 2). Chaque paire (i, j) est extraite de la file (Ligne 3) et peut y être réinsérée (Ligne 15) chaque fois que sa contrainte est mise à jour. Chaque contrainte C_{ij} est un sous-ensemble de l'algèbre des intervalles d'Allen, qui contient 13 relations. Chaque étape de raffinement réduit strictement $|C_{ij}|$, et donc chaque paire (i, j) peut être raffinée au plus 13 fois. Comme il y a $|V|(|V| - 1)$ paires, le nombre total de mises

à jour effectives de contraintes est $13 \cdot |V|(|V|-1)$, ce qui se simplifie en $O(|V|^2)$. Pour chaque paire extraite (i, j), l'algorithme parcourt toutes les variables intermédiaires $k \in V \setminus \{i, j\}$ (Ligne 4), ce qui correspond à $O(|V|)$ itérations. Chaque itération considère un nombre constant de triplets (Ligne 5) et effectue un nombre constant d'opérations sur les contraintes, notamment intersection, composition et inversion (Lignes 7–16), chacune en temps $O(1)$. Bien que le mécanisme de réparation (Lignes 9–12) puisse localement enrichir les ensembles de contraintes, toutes les contraintes restent des sous-ensembles de l'algèbre relationnelle finie $\mathcal{L}$. Ainsi, la réparation ne modifie pas la borne asymptotique du nombre de mises à jour. Chaque mise à jour de contrainte peut entraîner une réinsertion dans la file, mais comme le nombre total de mises à jour est borné par $O(|V|^2)$, le nombre total de paires traitées est également $O(|V|^2)$.
Chaque paire nécessite $O(|V|)$ opérations. Comme il existe $O(|V|^2)$ paires, la complexité temporelle dans le pire cas est :

$$O(|V|^3) \tag{6}$$

Complexité globale de *TCL*. L'algorithme 1 commence par initialiser le réseau de contraintes temporelles (Ligne 1) en parcourant toutes les paires de propriétés $(p, p') \in P \times P$ avec $p \neq p'$, ce qui nécessite $O(|P|^2)$ temps. Ensuite, l'algorithme itère à nouveau sur toutes les paires de propriétés (Ligne 2), ce qui donne $O(|P|^2)$ itérations. Pour chaque paire (p_i, p_j), une requête est formulée vers l'oracle via TR-QUERY (Ligne 3), dont la complexité dans le pire cas est $O(|\mathcal{TKG}|^2 \cdot |\mathcal{L}|)$. L'étape de filtrage (Ligne 4) supprime les relations à support nul et nécessite $O(|\mathcal{L}|)$ temps. Enfin, l'algorithme REPAIRPC est appliqué à chaque itération de la boucle (Ligne 5), avec une complexité $O(|V|^3)$. Ainsi, la complexité globale est :

$$O\big(|P|^2 + |P|^2 \cdot \big(|\mathcal{TKG}|^2 \cdot |\mathcal{L}| + |V|^3\big)\big) \tag{7}$$

5 Évaluation expérimentale

Cette section évalue les performances de notre approche en répondant aux questions de recherche suivantes :

1. Quels sont les taux de réduction des domaines de contraintes temporelles obtenus à chaque étape de *TCL* ?
2. La propagation permet-elle d'inférer des relations pour lesquelles l'oracle ne fournit aucune information ?
3. Quelle est la distribution des valeurs de support parmi les relations du TCN ?

5.1 Jeux de données et protocole expérimental

Pour évaluer notre approche, nous avons construit trois nouveaux jeux de données extraits de Wikidata Vrandecic and Krötzsch [2014], l'un des graphes de connaissances les plus complets actuellement disponibles. Nous nous sommes concentrés sur trois types d'entités distincts : les pays (*Country*), les groupes musicaux (*Musical Group*)et les politiciens (*Politician*). Les jeux de données temporels existants, tels que TemporalFC Qudus et al. [2023], ne sont pas adaptés à notre étude, car ils ne contiennent pas certaines caractéristiques essentielles, notamment les types d'entités, les attributs littéraux et des intervalles temporels correctement annotés.

Classe	# Entités	# Propriétés temporelles	# Faits temporels
Country	205	242	183 249
Musical Group	55 507	168	131 476
Politician	658 445	446	2 085 232

TABLE 1 – Vue d'ensemble des jeux de données construits.

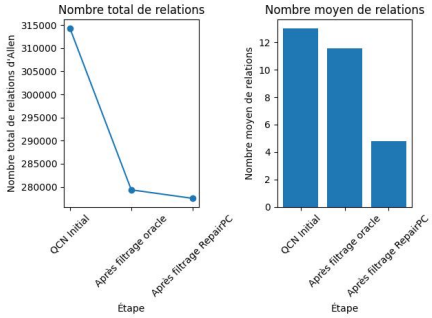
La Table 1 présente une vue d'ensemble des jeux de données construits, en mettant en évidence leurs caractéristiques en termes de nombre d'entités, de propriétés et de faits temporels (représentés sous forme de quadruplets). Toutes les expériences ont été réalisées sur un processeur Intel Xeon Silver 4112 (2.6 GHz, 4 cœurs / 8 threads) avec 64 Go de RAM, sous Ubuntu 24.04.

5.2 Taux de réduction des relations temporelles valides

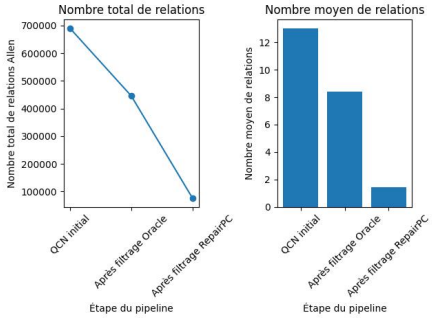
Cette section répond à la question de recherche 1 en analysant la réduction des domaines de contraintes dans le TCN. La Figure 2 présente, à gauche de chaque sous-figure, le nombre cumulé de relations sur l'ensemble des domaines de contraintes à chaque étape de l'approche.
Nous observons que l'étape de filtrage par l'oracle entraîne des réductions de domaines de 7.60% pour *Politician*, 11.13% pour *Musical Group* et 35.48% pour *Country*. Ces résultats indiquent que l'oracle élimine une proportion significative de relations non observées par le graphe de connaissances.
Les réductions produites par l'étape de propagation (cf. Algorithme 2) sont significativement plus importantes, atteignant 58.51% pour *Groupe musical*, 72.99% pour *Politician* et 83.15% pour *Country*. Cela met en évidence des incohérences structurelles dans le graphe de connaissances sous composition. Un mécanisme de cohérence locale permet de réduire significativement les domaines des contraintes temporelles valides.
La partie droite de chaque sous-figure présente la taille moyenne des domaines. En partant des 13 relations d'Allen par contrainte, l'oracle réduit les domaines à des moyennes de 12.01, 11.55 et 8.39 relations pour *Politician*, *Musical Group* et *Country*, respectivement. Après application de notre algorithme de cohérence de chemin revisité, les domaines diminuent davantage pour atteindre des moyennes de 4.75, 3.25 et 1.41 relations par contrainte. Ce résultat confirme la puissance du filtrage de cohérence locale sur les domaines des relations temporelles.

5.3 Inférence de nouvelles contraintes temporelles

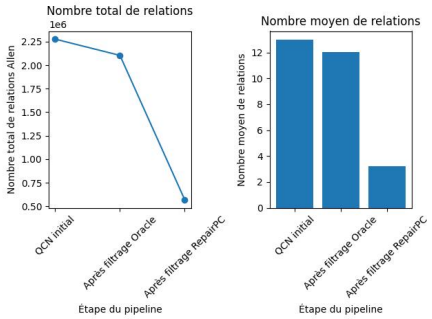
Cette section répond à la question de recherche 2 en analysant la capacité d'inférence de notre propagateur sur des



(a) Musical group



(b) Country



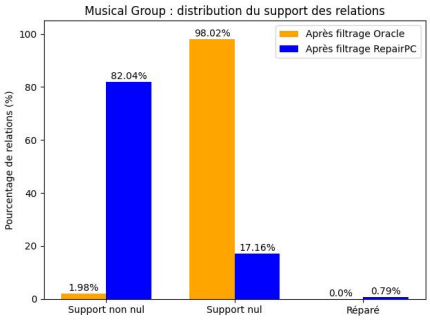
(c) Politician

FIGURE 2 – Taux de réduction des domaines

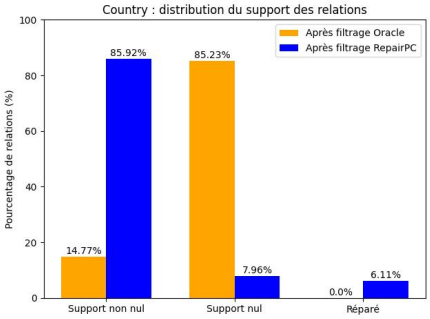
paires de propriétés temporelles pour lesquelles l'oracle ne fournit aucune information. Ce cas apparaît lorsque deux propriétés ne sont jamais décrites dans une même entité.

Avant l'application de l'algorithme de propagation, nous observons que la proportion de relations ayant un support non nul dans l'ensemble du TCN est de 1.87% pour *Politician*, 1.98% pour *Musical Group* et 14.77% pour *Country*. Ces valeurs indiquent que, pour la majorité des paires de propriétés distinctes, aucune information temporelle explicite n'est disponible dans le graphe de connaissances. Après l'application de l'Algorithme 2, la tendance s'inverse. La proportion de relations avec un support non nul augmente à 85.23% pour *Politician*, 85.92% pour *Musical Group* et 98.13% pour *Country*. Cela montre la capacité de l'Algorithme 2 à inférer de nouvelles connaissances temporelles à partir des relations observées dans le graphe, même lorsque le taux d'information initiale est limitée.

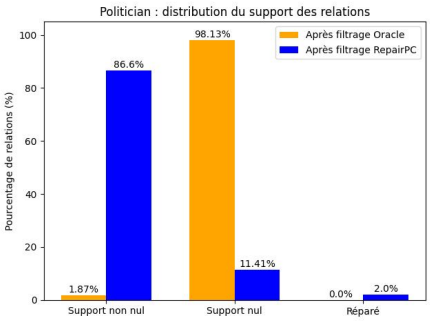
Concernant le taux de réparation induit par notre approche, nous observons des proportions de réparation non nulles dans tous les cas. Cela confirme la présence d'incohérences temporelles dans les graphes de connaissances. Néanmoins, la proportion de relations issues de réparations reste limitée, allant de 0.79% pour *Musical Group* à 6.11% pour *Country*. Ce résultat s'explique par notre choix de ne pas propager les relations réparées, ce qui privilégie les informations observées et inférées par rapport aux contraintes introduites artificiellement.



(a) Musical group



(b) Country



(c) Politician

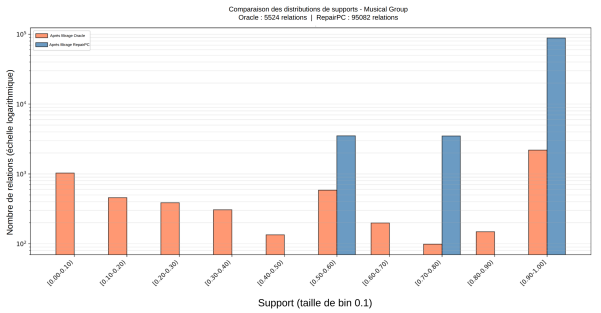
FIGURE 3 – Proportion de relations selon leur type de support

5.4 Distribution des supports des relations

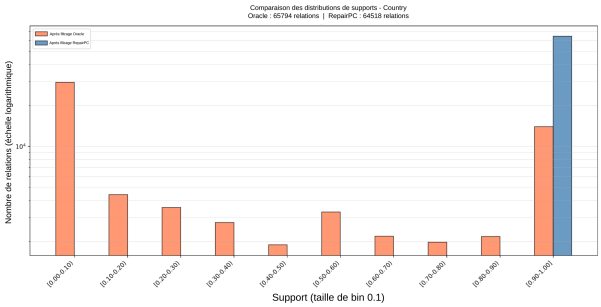
Cette section répond à la question de recherche 3 en analysant la distribution des supports pour l'ensemble des relations du TCN, avant et après l'application de l'algorithme de propagation (cf. Algorithme 2).

Comme illustré dans la Figure 4, avant l'application de l'algorithme de propagation (orange), la distribution des sup-

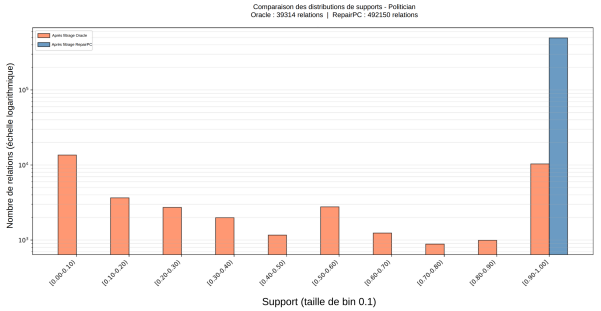
ports est presque uniforme pour toutes les instances. Après propagation (bleu), la distribution évolue, avec des supports compris entre 0.5 et 1.0 pour *Musical Group*, et entre 0.9 et 1.0 pour *Country* et *Politician*. Cette augmentation résulte de la capacité du propagateur à raisonner à partir des relations observées et à en inférer de nouvelles. Le support de ces relations inférées est déterminé par les justifications fournies par les explications sous-jacentes. Chaque relation se voit attribuer une valeur correspondant à l'explication contributive la plus faible, tandis que plusieurs explications pour une même relation la renforcent en retenant le support le plus élevé (cf. Définition 9).



(a) Musical group



(b) Country



(c) Politician

FIGURE 4 – Distribution du support des relations à travers les domaines

6 Conclusion

Dans cet article, nous avons introduit *TCL*, la première méthode dédiée à l'apprentissage de contraintes temporelles à partir de graphes de connaissances temporels. *TCL* s'appuie sur le paradigme de l'acquisition qualitative de contraintes en modélisant les relations temporelles d'un graphe de connaissances via un réseau de contraintes temporelles. Grâce à un oracle automatisé, *TCL* extrait des informations sur la présence ou l'absence de relations temporelles entre les faits du TKG, permettant un premier filtrage des relations jamais observées par le graphe de connaissances. Ensuite, via une procédure de propagation dédiée, *TCL* infère de nouvelles relations temporelles à partir de celles observées par l'oracle. Expérimentalement, nous avons montré l'intérêt de combiner les informations observées avec les connaissances inférées.

Ce travail ouvre plusieurs perspectives de recherche. Une extension naturelle de notre approche concerne son application à la détection et à la réparation des incohérences temporelles dans les graphes de connaissances temporels. Le réseau de contraintes temporelles appris pourrait servir de schéma temporel de référence pour identifier des violations structurelles et guider des stratégies de révision visant à rétablir la cohérence temporelle. De plus, le TCN appris pourrait être utilisé pour la vérification et la validation de nouveaux faits temporels. Les assertions candidates pourraient être évaluées au regard des contraintes apprises, permettant une évolution du graphe de connaissances tenant compte de la cohérence.

Une autre perspective concerne le passage à un monde ouvert et dynamique. Adapter *TCL* à des mises à jour incrémentales nécessiterait de maintenir et de réviser le TCN appris à mesure que des faits temporels sont ajoutés ou modifiés. Cela soulève des défis liés à la scalabilité, à la stabilité des contraintes apprises dans un contexte évolutif, ainsi qu'à l'application incrémentale de la cohérence dans des graphes de connaissances temporels à grande échelle.

Remerciements. Le premier auteur est financé par le projet DeMik dans le cadre du programme UPS Excellences Springboard (subvention SPATTR_2025-12).

Références

Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. *Data Profiling*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2018. ISBN 978-3-031-00737-8. doi : 10.2200/S00878ED1V01Y201810DTM052. URL https://doi.org/10.2200/S00878ED1V01Y201810DTM052.

James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, November 1983. ISSN 0001-0782. doi : 10.1145/182.358434. URL https://doi.org/10.1145/182.358434.

Mohamed-Bachir Belaid, Nassim Belmecheri, Arnaud Gotlieb, Nadjib Lazaar, and Helge Spieker. GEQCA : generic qualitative constraint acquisition. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*,

pages 3690–3697. AAAI Press, 2022. doi : 10.1609/AAAI.V36I4.20282. URL https://doi.org/10.1609/aaai.v36i4.20282.

Mohamed-Bachir Belaid, Nassim Belmecheri, Arnaud Gotlieb, Nadjib Lazaar, and Helge Spieker. Query-driven qualitative constraint acquisition. *J. Artif. Intell. Res.*, 79 :241–271, 2024. doi : 10.1613/JAIR.1.14752. URL https://doi.org/10.1613/jair.1.14752.

Jianhao Chen, Junyang Ren, Wentao Ding, and Yuzhong Qu. Patecon : A pattern-based temporal constraint mining method for conflict detection on knowledge graphs. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 4166–4172. AAAI Press, 2023. doi : 10.1609/AAAI.V37I4.25533. URL https://doi.org/10.1609/aaai.v37i4.25533.

Jean-François Condotta, Issam Nouaouri, and Michael Sioutis. A SAT approach for maximizing satisfiability in qualitative spatial and temporal constraint networks. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning : Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pages 432–442. AAAI Press, 2016.

Cristian Consonni, Paolo Sottovia, Alberto Montresor, and Yannis Velegrakis. Discovering order dependencies through order compatibility. In Melanie Herschel, Helena Galhardas, Berthold Reinwald, Irini Fundulaki, Carsten Binnig, and Zoi Kaoudi, editors, *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, pages 409–420. OpenProceedings.org, 2019. doi : 10.5441/002/EDBT.2019.36. URL https://doi.org/10.5441/002/edbt.2019.36.

Lluís Godo and Lluís Vila. Possibilistic temporal reasoning based on fuzzy temporal constraints. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, pages 1916–1923. Morgan Kaufmann, 1995. URL http://ijcai.org/Proceedings/95-2/Papers/114.pdf.

Ali Mensi, Jean-François Condotta, Issam Nouaouri, Michael Sioutis, and Lamjed Ben Saïd. A hybrid evolutionary algorithm for maximizing satisfiability in temporal or spatial qualitative constraints. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence, SETN 2018, Patras, Greece, July 09-12, 2018*, pages 19 :1–19 :9. ACM, 2018. doi : 10.1145/3200947.3201021. URL https://doi.org/10.1145/3200947.3201021.

Anastasia Paparrizou and Michael Sioutis. Learning to resolve inconsistencies in qualitative constraint networks. *Inf. Syst.*, 133 :102557, 2025. doi : 10.1016/J.IS.2025.102557. URL https://doi.org/10.1016/j.is.2025.102557.

Umair Qudus, Michael Röder, Sabrina Kirrane, and Axel-Cyrille Ngonga Ngomo. Temporalfc : A temporal fact checking approach over knowledge graphs. In Terry R. Payne, Valentina Presutti, Guilin Qi, María Poveda-Villalón, Giorgos Stoilos, Laura Hollink, Zoi Kaoudi, Gong Cheng, and Juanzi Li, editors, *The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part I*, volume 14265 of *Lecture Notes in Computer Science*, pages 465–483. Springer, 2023. doi : 10.1007/978-3-031-47240-4_25. URL https://doi.org/10.1007/978-3-031-47240-4_25.

Michael Sioutis and Jean-François Condotta. Efficiently enforcing path consistency on qualitative constraint networks by use of abstraction. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1262–1268. ijcai.org, 2017. doi : 10.24963/IJCAI.2017/175. URL https://doi.org/10.24963/ijcai.2017/175.

Thibaut Soulard, Fatiha Saïs, and Joe Raad. Explainable temporal fact validation through constraints discovery in knowledge graphs. In Edward Curry, Maribel Acosta, María Poveda-Villalón, Marieke van Erp, Adegboyega K. Ojo, Katja Hose, Cogan Shimizu, and Pasquale Lisena, editors, *The Semantic Web - 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1-5, 2025, Proceedings, Part I*, volume 15718 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2025. doi : 10.1007/978-3-031-94575-5_13. URL https://doi.org/10.1007/978-3-031-94575-5_13.

Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. YAGO 4.5 : A large and clean knowledge base with a rich taxonomy. pages 131–140, 2024. doi : 10.1145/3626772.3657876. URL https://doi.org/10.1145/3626772.3657876.

Denny Vrandecic and Markus Krötzsch. Wikidata : a free collaborative knowledgebase. *Commun. ACM*, 57(10) : 78–85, 2014. doi : 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Renjie Xiao, Zijing Tan, Haojin Wang, and Shuai Ma. Fast approximate denial constraint discovery. *Proc. VLDB Endow.*, 16(2) :269–281, 2022. doi : 10.14778/3565816.3565828. URL https://www.vldb.org/pvldb/vol16/p269-tan.pdf.