Un cadre pour inclure et exploiter des informations probabilistes dans les rapports de validation SHACL

Rémi Felin, Catherine Faron, Andrea G. B. Tettamanzi

Un cadre pour inclure et exploiter des informations probabilistes dans les rapports de validation SHACL

Rémi Felin[1], Catherine Faron[1], Andrea G. B. Tettamanzi[1]

[1] Université Côte d'Azur, Inria, I3S, Sophia-Antipolis, France

Résumé

SHACL est une recommandation du W3C qui permet de représenter en RDF des contraintes appelées formes (shapes en anglais) et de valider des graphes de données RDF par rapport à ces contraintes. Un validateur SHACL produit un résultat booléen, faux pour une forme SHACL quand au moins un triple dans le graphe RDF n'est pas conforme à la forme, vrai sinon. Nous proposons un cadre probabiliste pour valider un graphe RDF avec une proportion réaliste de triplets qui ne se conforment pas à une shape.

Mots-clés

RDF, SHACL, Validation de données, Evaluation probabiliste

Abstract

SHACL is a W3C recommendation to represent constraints in RDF– shape graphs –, and validate RDF data against these constraints. A SHACL validator outputs boolean results, false for a shape as soon as there is at least one triple in the RDF data that does not conform to the shape, else true. In this paper, we propose a probabilistic framework to validate an RDF graph with a realistic proportion of triples that does not conform to a shape.

Keywords

RDF, SHACL, Data Validation, Probabilistic Assessment

1 Introduction

Le développement du Web sémantique a conduit à l'émergence de nouveaux domaines de recherche tels que la qualité des données RDF. SHACL est le langage recommandé par le W3C pour représenter des contraintes que les données RDF doivent respecter afin d'assurer la cohérence d'un jeu de données. Un validateur SHACL produit un résultat booléen, faux pour une *shape* quand au moins un noeud dans le graphe RDF n'est pas conforme à la *shape*, vrai sinon. En considérant par exemple un grand ensemble de données RDF construit de manière collaborative avec une augmentation massive et constante de triples RDF (par exemple, DBpedia), la violation de contraintes SHACL semble inévitable en raison de données incomplètes et/ou incorrectes. Dans la pratique, un examen plus approfondi des données semble nécessaire. Un expert pourrait élaborer une stratégie de mise à jour des données ou des *shapes* en fonction du taux et/ou de la nature des violations. Nous abordons la question de recherche suivante :

> *Comment concevoir un processus de validation prenant en compte les erreurs physiologiques dans les données de la vie réelle ?*

Notre contribution aborde le problème en suggérant un cadre basé sur un modèle probabiliste afin de considérer un taux de violations de contraintes autorisé p égal à la proportion d'erreurs que les données RDF contiennent. Nous définissons une *mesure de la probabilité* d'observer un nombre donné de violations. Nous évaluons un graphe RDF par rapport à un ensemble de *shapes* en tenant compte d'un taux d'erreur physiologique théorique.

Le présent document est organisé comme suit : Dans la section 2, nous résumons les travaux connexes et le positionnement de notre travail. Dans la section 3, nous présentons notre modèle probabiliste (3.1), notre extension du modèle de rapport de validation SHACL (3.2) et notre proposition d'un processus de validation SHACL étendu (3.3). Nous présentons les résultats de nos expériences dans la section 4. Nous concluons et discutons des recherches futures dans la section 5.

2 Travaux connexes

SHACL [14] étant une recommandation assez récente (2017), ses relations avec d'autres standards font l'objet de recherches en cours. En particulier, nous trouvons des travaux sur ses relations avec les règles d'inférence [21], avec OWL [2], le raisonnement en logiques de descripton [16] et les patrons de conception d'ontologies [19]. De plus, des extensions concernant la validation SHACL émergent, par exemple un moteur de validation SHACL basé sur l'étude de la connectivité d'un graphe RDF et la collecte de données dans ce même graphe [12]. L'expressivité et la sémantique de SHACL sont un sujet riche dans la littérature [1, 16] : ces travaux ont mis en évidence une sémantique basée sur $\mathcal{SROIQ}$, l'une des logiques de description les plus expressives.

La validation de données RDF avec SHACL est une question de recherche largement abordée dans la littérature [3, 8, 10, 13, 15, 20]. Tous ces travaux considèrent une utilisation standard de SHACL : un graphe RDF est valide par rapport à une *shape* s'il vérifie les contraintes exprimées. Notre approche étend le processus de validation SHACL standard

pour dépasser son caractère binaire en considérant un taux de violation de contrainte acceptable.

D'autres travaux s'intéressent à la génération de contraintes SHACL [11, 23, 24]. Différentes approches conduisent à différentes façons de traiter la validation de ces *shapes*. Certaines approches exploitent des données RDF et des statistiques, et nécessitent une analyse d'expert pour valider une *shape* candidate. Le profilage des graphes de connaissances [22] est une approche possible pour induire des contraintes à partir de grands graphes RDF. Une de ces approches [18] s'appuie sur des techniques d'apprentissage automatique pour générer automatiquement des *shapes* en utilisant des données RDF profilées comme caractéristiques. Certaines approches exploitent des ontologies pour générer des *shapes* [5] : notamment les signatures de propriétés ; dans ce cas les *shapes* générées peuvent être considérées valides si l'ontologie est de bonne qualité.

Le travail présenté dans cet article est axé sur la validation des données RDF par rapport à des *shapes* et vise à fournir une expertise sur la cohérence des données RDF par rapport à un ensemble de *shapes* (qui peuvent avoir été générées automatiquement ou fournies par un expert), et en acceptant un taux d'erreurs physiologique que les données peuvent contenir.

3 Un cadre probabiliste pour l'évaluation de *shapes* SHACL

3.1 Modèle probabiliste

Dans un contexte réel, les ensembles de données RDF sont imparfaits, incomplets (dans le sens où des données attendues sont manquantes) et contiennent des erreurs de différentes natures. Le contrôle de la qualité des données RDF et l'intégration efficace des données, garantissant la cohérence des données RDF, sont des cas d'utilisation qui peuvent être traités à l'aide de SHACL. Par ailleurs, l'extraction de *shapes* SHACL à partir de données RDF est une approche prometteuse pour apprendre la connaissance du domaine (contraintes du domaine). Les *shapes* candidates sont celles qui déclenchent quelques violations dans les données, mais cela est directement corrélé à la qualité (taux d'erreur, qui est cependant inconnu) de l'ensemble de données RDF considéré.

Nous proposons d'étendre l'évaluation des données RDF par rapport aux *shapes* en considérant une proportion d'erreur théorique physiologique p dans les données RDF réelles. Dans ce contexte, la modélisation mathématique du processus d'évaluation SHACL qui tient compte d'une proportion d'erreur p est basée sur un modèle probabiliste.

La cardinalité de référence. (ou cardinalité du support) d'une *shape* S, υ_s, est l'ensemble des triplets RDF concernés par S et testés durant la validation. On la note $\|\upsilon_s\|$.

Les confirmations et les violations. On note υ_s^+ et υ_s^- les ensembles disjoints, respectivement, des triplets qui sont conformes à S et des triplets qui violent S. Le support d'une *shape* S est l'union disjointe de ses confirmations et violations :

$$\upsilon_S = \upsilon_S^+ \cup \upsilon_S^- \tag{1}$$

Modélisation du processus de validation. Soit X une variable aléatoire qui conceptualise un ensemble d'observations provenant de la validation d'une *shape* S, c'est-à-dire un ensemble de triplets RDF υ_S où chaque triplet $t \in \upsilon_S$ peut être soit une *confirmation* ($t \in \upsilon_S^+$) soit une *violation* ($t \in \upsilon_S^-$)
Soit un triplet t, tiré au hasard dans υ_S ; nous pouvons définir, à partir de ce triplet, une variable aléatoire qui prend deux valeurs : **1** si $t \in \upsilon_S^-$ et **0** sinon. Nous en concluons qu'une loi binomiale peut modéliser cette approche probabiliste : $B(\|\upsilon_S\|, p)$ avec p la proportion d'erreur théorique.

Mesure de vraisemblance. On note L_k la vraisemblance d'obtenir k violations ($\|\upsilon_S^-\| = k$) parmi n triplets ciblés ($n = \|\upsilon_S\|$) : $L_k = P(X = k)$. En considérant que X suit la loi binomiale $B(\|\upsilon_S\|, p)$, on a :

$$L_{\|\upsilon_S^-\|} = P(X = \|\upsilon_S^-\|) = \binom{\|\upsilon_S\|}{\|\upsilon_S^-\|} \cdot p^{\|\upsilon_S^-\|} \cdot (1-p)^{\|\upsilon_S^+\|}. \tag{2}$$

3.2 Extension du modèle du rapport de validation SHACL

Nous proposons un modèle enrichi du rapport de validation SHACL afin d'exprimer des informations supplémentaires pour chaque *shape* considérée dans le rapport. Nous avons défini une extension du vocabulaire du rapport de validation SHACL, dans un espace de noms dénotée par le préfixe `psh` dans la suite. [1]. Pour chaque *shape* source considérée dans la validation d'un graphe RDF, nous générons des triplets supplémentaires : la propriété `psh:summary` relie le rapport de validation à un nœud blanc de type `psh:ValidationSummary`, qui est le sujet de plusieurs propriétés dont les valeurs sont le résultat du calcul de différentes mesures relatives à la *shape* source.

La *shape* ciblée. Il s'agit de la valeur de la propriété `psh:focusShape`. C'est la source de la *shape* de validation qui est ensuite décrite dans le résumé de validation.

La cardinalité de référence. $\|\upsilon_S\|$, est la valeur de la propriété `psh:referenceCardinality`.

Le nombre de confirmations et de violations. Respectivement $\|\upsilon_S^+\|$ et $\|\upsilon_S^-\|$, sont les valeurs des propriétés `psh:numConfirmation` et `psh:numViolation`.

La généralité $G(S) \in [0, 1]$ mesure la *représentativité* de S en considérant l'ensemble du graphe RDF υ :

$$G(S) = \frac{\|\upsilon_S\|}{\|\upsilon\|} \tag{3}$$

C'est la valeur de la propriété `psh:generality`.

La vraisemblance $L_{\|\upsilon_S^-\|}$ d'une *shape* S dans un graphe RDF υ telle que définie dans la Section 3.1 est la valeur de la propriété `psh:likelihood`.
La figure 1 présente un extrait d'un exemple de rapport de validation dans lequel :

1. `prefix psh: <http://ns.inria.fr/probabilistic-shacl/>`

```
[ a sh:ValidationReport ;
    sh:conforms boolean ;
    sh:result r ;
    # Probabilistic SHACL extension
    psh:summary [
        a psh:ValidationSummary ;
        psh:referenceCardinality ‖v_S‖ ;
        psh:numConfirmation ‖v_S^+‖ ;
        psh:numViolation ‖v_S^-‖ ;
        psh:generality G(S) ;
        psh:likelihood L_{‖v_S^-‖} ;
        psh:focusShape S^{‖v_S^-‖}
    ] ;
] .
```

FIGURE 1 – Structure du rapport de validation SHACL étendu

— l'URI `:s1` dénote une *shape* SHACL s_1 ;
— la cardinalité du graphe RDF en cours de validation est $\|v\| = 1000$;
— le paramètre de la distribution binomiale est $p = 0.1$.

3.3 Validation d'un graphe RDF par rapport à une *shape* SHACL comme un test d'hypothèse

Le processus de validation d'un graphe RDF par rapport à une *shape* donnée S est basé sur le modèle probabiliste proposé dans la section 3.1, qui repose sur l'hypothèse selon laquelle une observation donnée suit une distribution binomiale $X \sim B(\|v_S\|, p)$. Pour valider cette hypothèse, nous procédons à un test d'hypothèse.
La validation d'un graphe RDF par rapport à une *shape* S est basée sur la proportion observée de violations de S dans le graphe, notée $\hat{p}$: $\hat{p} = \frac{\|v_S^-\|}{\|v_S\|}$. Un graphe RDF est valide par rapport à S si la proportion observée de violation de S est inférieure à la proportion théorique :

$$\hat{p} \leq p \implies v \models S \tag{4}$$

Dans le cas où la proportion observée est supérieure à la proportion théorique, nous évaluons la distance de cette probabilité à partir des valeurs maximales de la fonction de masse de la distribution binomiale $B(\|v_S\|, p)$ en utilisant les tests d'hypothèses. La figure 3 montre la proportion du nombre de violations que nous acceptons par rapport au nombre que nous rejetons avec notre méthode.

L'hypothèse nulle (H_0) et l'hypothèse alternative (H_1). L'hypothèse nulle est que *les données suivent la distribution donnée*, c'est-à-dire que la fréquence des violations observées $\hat{p} = \frac{\|v_S^-\|}{\|v_S\|}$ est conforme aux proportions attendues de violations p et $X \sim B(\|v_S\|, p)$. L'hypothèse alternative H_1 est que *les données ne suivent pas la distribution donnée*.

Le test d'ajustement. Ce test d'hypothèse vérifie l'alignement de nos observations avec une distribution théorique : nous définissons X_s^2 la **statistique de test** pour une *shape* S qui suit $\chi^2_{k-1,\alpha}$ en supposant H_0, c'est-à-dire que $X_s^2 \sim \chi^2_{k-1,\alpha}$ (une distribution du chi-carré avec $k-1$ degrés de liberté et un seuil de signification de $1-\alpha$). Ce test est effectué au seuil α défini à 5%. Il considère k comme le nombre total de groupes, c'est-à-dire $k = 2$, n_i le nombre d'individus observés et T_i le nombre d'individus théoriques. La statistique de test X_s^2 est définie par

$$X_s^2 = \sum_{i=1}^{k} \frac{(n_i - T_i)^2}{T_i} \sim \chi^2_{k-1;\alpha}. \tag{5}$$

.
Le test d'ajustement (Formule 5) est applicable si $\forall i \in [1, k], T_i \geq 5$. Supposons une *shape* S pour laquelle nous observons un très faible support $\|v_S\|$ (supposons $\|v_S\| = 8$) implique une proportion de violations et/ou de confirmations inférieure à 5. Dans ce cas, le test d'hypothèse ne peut pas être réalisé car l'échantillon n'est pas suffisamment représentatif.

La valeur critique. La valeur à partir de laquelle on rejette l'hypothèse nulle H_0, est égale à $\chi^2_{k-1;\alpha}$. En prenant $\alpha = 0.05$ et $k = 2$, on a $\chi^2_{k-1;\alpha} = \chi^2_{1;\alpha=0.05} = 3.84$.
Une formule alternative considère l'intervalle d'acceptation I_a d'une distribution du chi-carré, c'est-à-dire $I_a = [0, \chi^2_{k-1;\alpha}]$ qui accepte H_0 si $X_S^2 \in I_a$.

L'acceptation de l'hypothèse nulle. Accepter H_0 et donc $X \sim B(\|v_S\|, p)$, implique que la valeur de notre statistique de test X_S^2 n'est pas incluse dans la zone de rejet de la distribution $\chi^2_{k=1}$:

$$X_S^2 \leq \chi^2_{k-1;\alpha}. \tag{6}$$

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix psh:
      <http://ns.inria.fr/probabilistic-shacl/> .
@prefix : <http://www.example.com/myDataGraph#> .

# SHACL Standard
:v1 a sh:ValidationResult ;
    sh:focusNode :n1 ;
    [...]
    sh:sourceShape :s1 .

:v2 a sh:ValidationResult ;
    sh:focusNode :n2 ;
    [...]
    sh:sourceShape :s1 .

[...]

[ a sh:ValidationReport ;
    sh:conforms false ;
    sh:result :v1 ;
    sh:result :v2 ;
    [...]
    # SHACL Extension
    # shape s1
    psh:summary [
        a psh:ValidationSummary ;
        psh:generality "0.2"^^xsd:decimal ;
        psh:numConfirmation 178 ;
        psh:numViolation 22 ;
        psh:likelihood "0.0806"^^xsd:decimal ;
        psh:referenceCardinality 200 ;
        psh:focusShape :s1
    ] ;
] .
```

FIGURE 2 – Exemple d'un rapport de validation SHACL étendu pour une *shape* `:s1`, calculé avec $\|v\| = 1000$ et $p = 0.1$

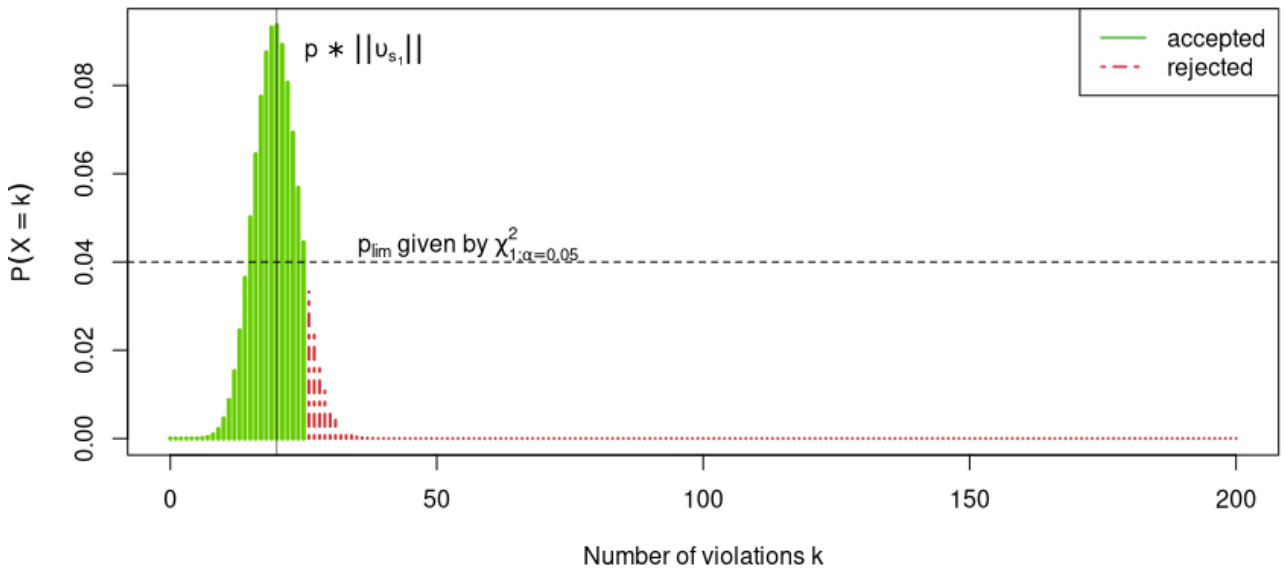


FIGURE 3 – Zone d'acceptation d'une *shape* s_1 considérant $X \sim B(\|\upsilon_{s_1}\|, p)$ où $\|\upsilon_{s_1}\| = 200$ et $p = 0.1$.

L'acceptation de H_0 implique la validation du graphe RDF par rapport à la *shape* S considérée, c'est-à-dire,

$$X_S^2 \leq \chi^2_{k-1;\alpha} \implies \upsilon \models S. \tag{7}$$

Prenons l'exemple de la Figure 2. Nous observons une proportion de violations légèrement supérieure à celle attendue, c'est-à-dire, $\hat{p} = \frac{\|\upsilon^-_{s_1}\|}{\|\upsilon_{s_1}\|} = 0.11$ et $\hat{p} > p$: le test d'hypothèse permet de déterminer si cette observation est compatible ou non avec l'hypothèse nulle, et dans cet exemple, nous rejetterions H_0 et ne validerions pas le graphe par rapport à la *shape* `:s1`. En prenant $\alpha = 5\%$ pour évaluer $X^2_{s_1}$, on obtient :

$$X^2_{s_1} = \frac{(22-20)^2}{20} + \frac{(178-180)^2}{180} \approx 0.222.$$

Ainsi, le test statistique a montré que $X^2_{s_1} \leq \chi^2_{1;\alpha=0.05}$ (c'est-à-dire 3.84) et donc $X^2_{s_1} \in I_a$. Nous acceptons H_0 i.e. l'hypothèse que nos observations sur la conformité des triplets à `:s1` suivent une distribution binomiale $X \sim B(200, 0.1)$.

4 Expériences

Nous avons implémenté le modèle proposé dans un moteur de validation probabiliste SHACL reposant sur le moteur sémantique *Corese* [6]. Le rapport de validation étendu fournit un degré de probabilité exprimé sous l'hypothèse que les échantillons suivent une distribution binomiale avec une cardinalité définie pour une *shape* S (c'est-à-dire $\|\upsilon_S\|$) et une probabilité p définie *empiriquement* correspondant à la proportion supposée de violations que nous acceptons pour certaines données RDF. En considérant un ensemble de contraintes SHACL représentatives d'un (large) graphe RDF donné, la recherche d'un taux d'erreur p pour lequel il est raisonnable de considérer l'acceptation de données est une manière d'évaluer ce travail. Cela implique une analyse détaillée des caractéristiques du graphe RDF considéré, des proportions de *shapes* acceptées ou rejetées et de l'impact des tests d'hypothèse sur l'acceptation.

4.1 Protocole expérimental

Nos expériences utilisent le jeu de données RDF *CovidOnTheWeb* [2] [17] et un ensemble de 377 *shapes* SHACL construites à partir de règles d'association issues de la fouille de *CovidOnTheWeb* [4] et considérées comme représentatives de ce graphe.
Nous effectuons une analyse du taux d'erreur théorique afin de trouver empiriquement un taux optimal : nous testons les 20 valeurs de $p \in \{0,05, 0,1, 0,15, \ldots, 0,95, 1\}$. Les expériences ont été effectuées sur un Dell Precision 3561 équipé d'un processeur Intel(R) Core i7-11850H de 11e génération, avec 32 Go de RAM fonctionnant sous le système d'exploitation Fedora Linux 35. Le code source est disponible dans un dépôt public. [3]

CovidOnTheWeb. Il s'agit d'un graphe de connaissances RDF produit à partir du *COVID-19 Open Research Dataset (CORD-19)*. Il décrit des articles scientifiques, identifiés par des URI et associés aux entités nommées extraites dans ces articles, désambiguïsées par *Entity-Fishing* et liées à des entités *Wikidata*. La figure 4 montre un extrait de description RDF dans CovidOnTheWeb au format *turtle* et le tableau 1 montre les caractéristiques du jeu de données RDF. Nous

2. https ://github.com/Wimmics/CovidOnTheWeb
3. https://github.com/RemiFELIN/RDFMining/tree/eswc_2023

TABLE 1 – Résumé du sous-graphe de données RDF *CovidOnTheWeb* considéré pour les expériences.

#triplets RDF	226,647
#articles distincts	20,912
#entités nommées distinctes	6,331
moyenne #entités nommées par article	10.52

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax
     -ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
     .
@prefix covid:    <http://ns.inria.fr/covid19/> .
@prefix entity:   <http://www.wikidata.org/entity/> .

covid:ec1[...]2c5   rdf:type    entity:Q4407  .
covid:fff[...]86d   rdf:type    entity:Q10876 .
[...]
entity:Q4407    rdfs:label  "methyl"@en .
entity:Q10876   rdfs:label  "bacteria"@en .
```

FIGURE 4 – Exemple d'un extrait de données RDF du sous-graphe *CovidOnTheWeb*

considérons un sous-ensemble contenant environ 18,79% des articles et 0,01% des entités nommées.

Les *shapes* candidates. Ces *shapes* représentent les règles d'association obtenues par Cadorel et al.[4] à partir d'un sous-ensemble du jeu de données *CovidOnTheWeb*. Ces règles ne sont pas nécessairement parfaites, nous nous intéressons donc à les utiliser dans notre approche probabiliste. À partir des résultats expérimentaux de Cadorel et al., nous avons extrait les entités nommées correspondant aux antécédents et conséquents de ces règles d'association. Nous avons effectué un traitement permettant la conversion de ces règles en *shapes* SHACL. Nous ciblons les articles appartenant à une entité nommée, représentant l'*antécédent*, avec la propriété `sh:targetClass`. Parmi les articles considérés, nous cherchons à déterminer l'affiliation à une autre entité nommée, représentant le *conséquent* : nous utilisons une contrainte appliquée sur le type d'article et ciblant une entité nommée avec la propriété `sh:hasValue`. Dans ce contexte, une violation invoquera une violation de type `sh:HasValueConstraintComponent` pour la *shape* courante. Un exemple de *shape* formée après traitement est présenté dans la figure 5.

4.2 Résultats

Le tableau 2 présente les premiers résultats expérimentaux, notamment le score de généralité qui est relativement faible : la cardinalité de référence moyenne est assez faible par rapport au nombre total de triplets RDF dans notre ensemble de données : environ 106 triplets RDF en moyenne sont ciblés par nos *shapes* (0,047% des triplets RDF). Le taux de violations est relativement élevé mais cela est nuancé par le taux de confirmations (33,19%).

La figure 6a montre une évolution croissante de la mesure de vraisemblance jusqu'à la valeur $p = 0.5$ puis une diminution. Il apparaît ainsi que le taux d'erreur le plus raisonnable est 50%, car il maximise la valeur moyenne de

```
@prefix : <http://www.example.com/myDataGraph#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix entity: <http://www.wikidata.org/entity/> .

:1 a sh:NodeShape ;
    sh:targetClass  entity:Q10295810 ;
    sh:property [
        sh:path rdf:type ;
        sh:hasValue entity:Q43656 ;
    ] .
```

FIGURE 5 – Exemple d'une *shape* SHACL représentant une règle d'association : `entity:Q10295810` (`"hypocholesterolemia"@en`) en tant qu' *antecedent* et `entity:Q43656` (`"cholesterol"@en`) en tant que *consequent*.

TABLE 2 – Résumé de la validation du graphe de *shapes* SHACL.

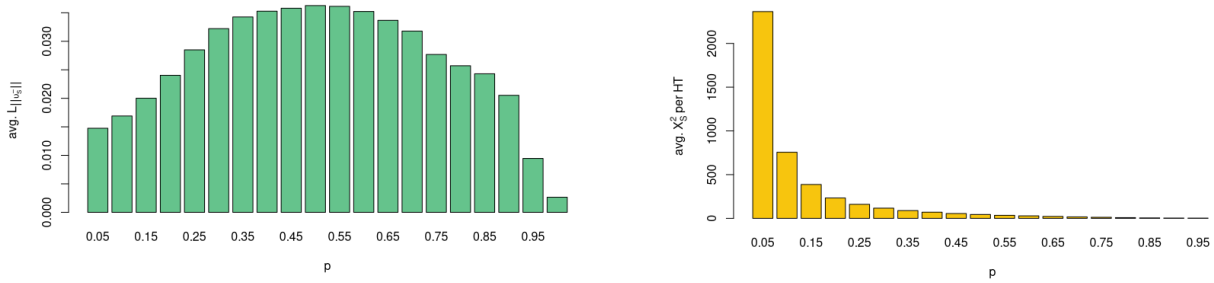
#entités nommées représentées	337 (*5.32%*)
moyenne $\|\upsilon_{\mathbf{S}}\|$	106.69 (*0.0470%*)
moyenne $\|\upsilon_{\mathbf{S}}^{+}\|$	33.19 (*31.11%*)
moyenne $\|\upsilon_{\mathbf{S}}^{-}\|$	73.50 (*70.89%*)
moyenne $\mathbf{G}(\mathbf{S})$ (Formule 3)	0.0005%

vraisemblance (0,0362%).

La figure 7 présente l'ensemble des décisions prises sur les *shapes* (acceptation, rejet) en fonction de la proportion théorique d'erreurs p et montre clairement l'importance des tests d'hypothèse. Le nombre de tests effectués augmente jusqu'à $p = 0.3$ puis diminue. De même, les tests d'hypothèse ont tendance à rejeter les *shapes* pour des valeurs "petites" de p et la tendance s'inverse à mesure que p augmente : le nombre de *shapes* acceptées augmente et la valeur de la statistique de test diminue (voir figure 6b). Une analyse plus poussée des résultats obtenus avec $p = 0.5$ montre que 63 *shapes* parmi les 187 *shapes* acceptées sont acceptées après avoir effectué un test d'hypothèse, c'est-à-dire 33,7% des *shapes* acceptées. Ces mêmes tests ont accepté 25,7% des *shapes* qui ont été testées, ce qui montre leur capacité à filtrer efficacement les *shapes* non valides avec un risque $\alpha = 0,05$ (5%) d'être incorrect.

La production des résultats au format HTML a été effectuée avec une transformation STTL [7], une extension du langage de requête SPARQL pour transformer RDF en n'importe quel format textuel Un extrait des rapports obtenus pour 20 *shapes* avec une proportion d'erreur théorique $p = 0.5$ est présenté dans la figure 8.

Nous avons comparé le temps de calcul de notre cadre de validation probabiliste proposé à celui de la validation standard. Pour notre base de 377 *shapes* et notre extraction de *CovidOnTheWeb* (226,647 triplets), nous avons observé un temps de calcul global de 1 minute et 35 secondes pour le cadre de validation probabiliste contre 1 minute et 29 secondes pour la validation standard : le cadre probabiliste prend 6.31% de temps supplémentaire par rapport à la validation standard et il est linéaire, ce qui le rend pratique et évolutif.



(a) $L_{\|\upsilon_S^-\|}$ average

(b) X_S^2 average

FIGURE 6 – Valeur moyenne (a) des mesures de vraisemblance et (b) des tests statistiques, en fonction de la proportion d'erreur théorique p.

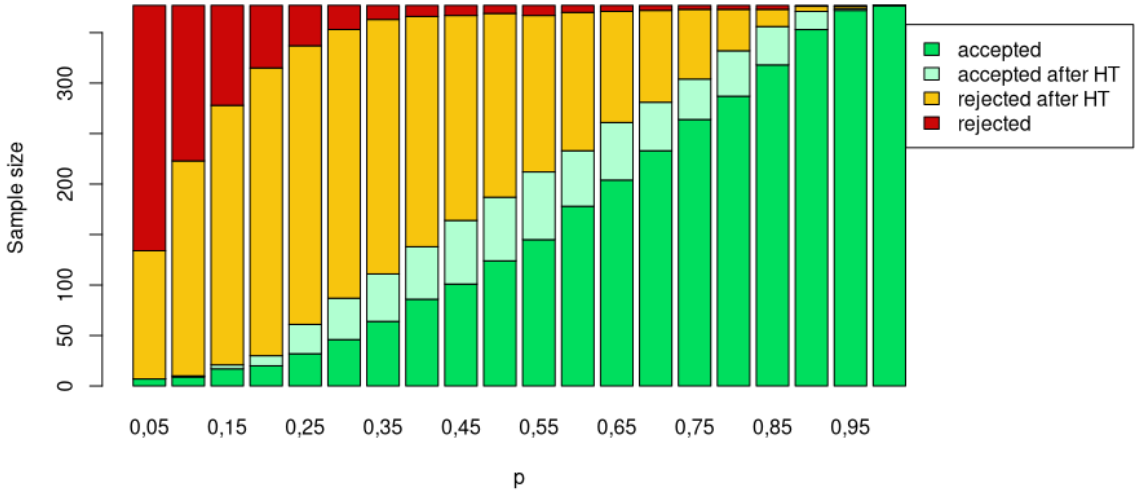


FIGURE 7 – Acceptation des *shapes* en fonction de la proportion d'erreur théorique p (HT = Test d'hypothèse)

5 Conclusion

Dans cet article, nous proposons un cadre probabiliste pour la validation SHACL, contribuant ainsi au contrôle de qualité des données RDF. Nous étendons le rapport de validation SHACL pour exprimer une mesure de vraisemblance pour le nombre de violations observées et proposons un modèle de décision pour une acceptation probabiliste des triplets RDF par rapport aux *shapes* SHACL. Nos expériences montrent les capacités de notre approche à valider un ensemble de *shapes* avec un taux d'erreur raisonnable p. Dans le cadre de travaux futurs, nous prévoyons d'étendre notre cadre proposé aux *shapes* complexes, en particulier les *shapes* récursives qui font l'objet de recherches en cours [3, 9, 20]. Nous prévoyons également d'étudier l'extraction automatique de *shapes* SHACL à partir de jeux de données RDF de référence, afin de capturer des connaissances de domaine sous forme de contraintes.

Remerciements

Ce travail a été partiellement financé par le projet 3IA Côte d'Azur "Investissements d'avenir" géré par l'Agence Nationale de la Recherche (ANR) avec le numéro de référence ANR-19-P3IA-0002.

Références

[1] Bart Bogaerts, Maxim Jakubowski, and Jan Van den Bussche. Expressiveness of shacl features. In *ICDT*, 2022.

[2] Bart Bogaerts, Maxime Jakubowski, and Jan Van den Bussche. Shacl : A description logic in disguise. 08 2021.

[3] Iovka Boneva, Jose G Labra Gayo, and Eric G Prud 'Hommeaux. Semantics and Validation of Shapes Schemas for RDF. In *ISWC2017 - 16th In-*

antecedent	consequent	referenceCardinality	#violation	likelihood	generality	X^2_S	Acceptance
two-hybrid screening	protein–protein interaction	48	19	0.041004880900459284	0.00021178308117910231		true
nidovirales	proteolysis	80	69	8.6669313322632E-12	0.00035297180196517053	42.05	false
intensive care medicine	acute respiratory distress syndrome	166	139	9.193409214822706E-20	0.0007324164890777288	75.56626506024097	false
astrocyte	central nervous system	70	34	0.09238587705330051	0.0003088503267195242		true
dopamine	serotonin	10	6	0.205078125	0.00004412147524564632	0.4	true
crystallography	crystal structure	20	7	0.0739288330078125	0.00008824295049129263		true
human parainfluenza	adenoviridae	237	133	0.00880821375320367	0.0010456789633218177	3.548523206751055	true
carbohydrate	lectin	114	75	2.4200572197826046E-4	0.000502984817800368	11.368421052631579	false
mycoplasma bovis	bovine coronavirus	12	6	0.2255859375	0.00005294577029477558		true
crystallization	diffraction	31	21	0.020653086248785257	0.00013677657326150358	3.903225806451613	false
membrane raft	methyl	32	19	0.08087921887636185	0.0001411887207860682	1.125	true
ifitm1	ifitm3	27	9	0.03491956740617752	0.00011912798316324504		true
multiple sclerosis	myelin	139	97	1.0209205741082355E-6	0.0006132885059144837	21.762589928057555	false
wheeze	asthma	85	44	0.08188889187584301	0.00037503253958799367	0.10588235294117647	true
influenza a virus subtype h5n1	avian influenza	277	165	2.969648471686876E-4	0.001222164864304403	10.140794223826715	false
hepatocellular carcinoma	liver cirrhosis	72	46	0.005843155895129734	0.00031767462176865343	5.555555555555555	false
diffraction	x-ray crystallography	16	7	0.174560546875	0.00007059436030930341		true
feline infectious peritonitis	feline coronavirus	130	46	2.605193913325792E-4	0.000573579178193402		true
aedes aegypti	culicidae	21	4	0.002853870391845703	0.00009265509801585726		true
monomer	oligomer	83	70	5.4692741602999564E-11	0.0003662082445388644	39.144578313253014	false

FIGURE 8 – Rapport de validation SHACL étendu en considérant $p = 0.5$.

ternational semantic web conference, Vienna, Austria, October 2017.

[4] Lucie Cadorel and Andrea Tettamanzi. Mining rdf data of covid-19 scientific literature for interesting association rules. *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 145–152, 2020.

[5] Andrea Cimmino, Alba Fernández-Izquierdo, and Raúl García-Castro. Astrea : Automatic generation of shacl shapes from ontologies. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web*, pages 497–513, Cham, 2020. Springer International Publishing.

[6] Olivier Corby, Rémi Ceres, Erwan Demairy, Fuqi Song, Virginie Bottollier, and Olivier Savoie. Corese : Semantic Web Factory. `https://project.inria.fr/corese/`.

[7] Olivier Corby and Catherine Faron Zucker. STTL : A SPARQL-based Transformation Language for RDF. In *11th International Conference on Web Information Systems and Technologies*, Lisbon, Portugal, May 2015.

[8] Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savkovic. Validating shacl constraints over a sparql endpoint. In *International Workshop on the Semantic Web*, 2019.

[9] Julien Corman, Juan L. Reutter, and Ognjen Savković. Semantics and validation of recursive shacl. In Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web – ISWC 2018*, pages 318–336, Cham, 2018. Springer International Publishing.

[10] Christophe Debruyne and Kris McGlinn. Reusable shacl constraint components for validating geospatial linked data (short paper). In *GeoLD@ESWC*, 2021.

[11] Daniel Fernandez-Álvarez, Jose Emilio Labra-Gayo, and Daniel Gayo-Avello. Automatic extraction of shapes using shexer. *Knowledge-Based Systems*, 238 :107975, 2022.

[12] Mónica Figuera, Philipp D. Rohde, and Maria-Esther Vidal. Trav-shacl : Efficiently validating networks of shacl constraints. In *Proceedings of the Web Conference 2021*, WWW '21, page 3337–3348, New York, NY, USA, 2021. Association for Computing Machinery.

[13] Ranjith K Soman. Modelling construction scheduling constraints using shapes constraint language (shacl). pages 351–358, 07 2019.

[14] Dimitris Kontokostas and Holger Knublauch. Shapes constraint language (SHACL). W3C recommendation, W3C, July 2017. https ://www.w3.org/TR/2017/REC-shacl-20170720/.

[15] Aljosha Köcher, Luis Miguel Vieira da Silva, and Alexander Fay. Constraint checking of skills using shacl. 07 2021.

[16] Martin Leinberger, Philipp Seifer, Tjitze Rienstra, Ralf Lämmel, and Steffen Staab. Deciding shacl shape containment through description logics reasoning. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 366–383, Cham, 2020. Springer International Publishing.

[17] Franck Michel, Fabien Gandon, Valentin Ah-Kane, Anna Bobasheva, Elena Cabrio, Olivier Corby, Raphaël Gazzotti, Alain Giboin, Santiago Marro, Tobias Mayer, Mathieu Simon, Serena Villata, and Marco

Winckler. Covid-on-the-Web : Knowledge Graph and Services to Advance COVID-19 Research. In *ISWC 2020 - 19th International Semantic Web Conference*, Athens / Virtual, Greece, November 2020.

[18] Nandana Mihindukulasooriya, Mohammad Rifat Ahmmad Rashid, Giuseppe Rizzo, Raúl García-Castro, Oscar Corcho, and Marco Torchiano. Rdf shape induction using knowledge base profiling. SAC '18, page 1952–1959, New York, NY, USA, 2018. Association for Computing Machinery.

[19] H.J. Pandit, D. O'Sullivan, and D. Lewis. Using ontology design patterns to define shacl shapes. In *WOP@ISWC*, pages 67–71, Monterey California, USA, 2018.

[20] Paolo Pareti and G. Konstantinidis. A review of shacl : From data validation to schema reasoning for rdf graphs. In *Reasoning Web*, 2021.

[21] Paolo Pareti, George Konstantinidis, Timothy J. Norman, and Murat Şensoy. Shacl constraints with inference rules. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web – ISWC 2019*, pages 539–557, Cham, 2019. Springer International Publishing.

[22] Renzo Principe, Andrea Maurino, Matteo Palmonari, Michele Ciavotta, and Blerina Spahiu. Abstat-hd : a scalable tool for profiling very large knowledge graphs. *The VLDB Journal*, 31, 09 2021.

[23] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Shacl and shex in the wild : A community survey on validating shapes generation and adoption. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 260–263, New York, NY, USA, 2022. Association for Computing Machinery.

[24] Jesse Wright, Sergio José Rodríguez Méndez, Armin Haller, Kerry Taylor, and Pouya G. Omran. Schímatos : A shacl-based web-form generator for knowledge graph editing. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 65–80, Cham, 2020. Springer International Publishing.