L'Apport Mutuel de la Combinaison des Tâches d'Interconnexion de Données et d'Alignement d'Ontologies pour l'Alignement Expressif

Chloé Khadija Jradeh[1], Jérôme David[2], Olivier Teste[1] et Cassia Trojahn[1]

[1]Université Toulouse 2, Jean Jaures
[2] Université Grenoble-Alpes

IC 2023

Mardi 4 juillet 2023

Plan de la présentation

Graphes de connaissances

Une graphe de connaissances est une ontologie généralisée définie par le paire $\langle \mathcal{A}, \mathcal{T} \rangle$ dont:

1. $\mathcal{A}$ est un ensembles de données, composés de:
   1. Entités: représentent des objets ou des concepts du monde réel,
   2. Relations: capturez les connexions et les associations entre les entités,
   3. Attributs: fournit des informations supplémentaires sur les entités ou les relations.
2. $\mathcal{T}$ est un schéma qui définissez la structure et la sémantique du graphe de connaissances.

L'interconnexion des données

1. L'interconnexion des données consiste à trouver différentes entités dans différents graphes de connaissances qui font référence à la même entité du monde réel,
2. Le résultat de l'interconnexion des données est un ensemble de relations owl:sameAs entre les entités identifiées,
3. Les clés de liage sont des axiomes permettant d'établir ces relations owl:sameAs.

Syntaxe d'une clé de liage

Étant donné une paire de graphes de connaissances $KG_1 = \langle \mathcal{A}_1, \mathcal{T}_1 \rangle$ et $KG_2 = \langle \mathcal{A}_2, \mathcal{T}_2 \rangle$, une clé de liage entre KG_1 et KG_2 a la forme suivante:

$$(\{\langle P_1, Q_1 \rangle, ..., \langle P_n, Q_n \rangle\} \text{ linkkey } \langle C, D \rangle)$$

où $\langle C, D \rangle$ sont un couple de concepts appartenant respectivement à $\mathcal{T}_1$ et $\mathcal{T}_2$ et $\langle P_1, Q_1 \rangle, ..., \langle P_n, Q_n \rangle$ est une séquence de paires non vides de propriétés où chaque propriété paire $\langle P_i, Q_i \rangle$ dans $\{\langle P_1, Q_1 \rangle, ..., \langle P_n, Q_n \rangle\}$, P_i appartient à $\mathcal{T}_1$ et Q_i appartient à $\mathcal{T}_2$.

Sémantique d'une clé de liage

$$(\{\langle P_1, Q_1\rangle, ..., \langle P_n, Q_n\rangle\} \text{ linkkey } \langle C, D\rangle)$$

stipule que si deux entités appartenant respectivement aux concepts C et D partagent au moins une valeur pour chaque paire de propriétés $\langle P_i, Q_i\rangle$ éventuellement multivaluées alors elles sont identiques.

Exemple:

$$(\{\langle \text{fname}, \text{prénom}\rangle, \langle \text{lname}, \text{nom}\rangle\} \text{ linkkey } \langle \text{Person}, \text{Personne}\rangle)$$

L'alignement d'ontologies

- Différents graphes de connaissances ont différentes schéma ontologique.
- La tâche de trouver des relations entre les différentes schéma est appelée l'alignement d'ontologies.
- Le résultat de l'alignement d'ontologies est un **alignement**, composé d'un **ensemble de correspondances**.

Correspondances

Étant donné une paire de graphes de connaissances $\mathsf{KG}_1 = \langle \mathcal{A}_1, \mathcal{T}_1 \rangle$ et $\mathsf{KG}_2 = \langle \mathcal{A}_2, \mathcal{T}_2 \rangle$, une correspondance entre $\mathcal{T}_1$ et $\mathcal{T}_2$ est un tuple $\langle e_{\mathcal{T}_1}, e_{\mathcal{T}_2}, r, n \rangle$ où

1. $e_{\mathcal{T}_1}$ et $e_{\mathcal{T}_2}$ sont des expressions *simples* ou *complexes*;
2. r est une relation entre $e_{\mathcal{T}_1}$ et $e_{\mathcal{T}_2}$, par exemple r peut être équivalence ($\equiv$), plus général ($\sqsubseteq$), plus spécifique ($\sqsupseteq$);
3. n est la valeur de confiance entre [0,1] indiquant le degré de confiance que la relation r existe entre $e_{\mathcal{T}_1}$ et $e_{\mathcal{T}_2}$.

Correspondances simples et complexes

- une correspondance est dite **simple** si $e_{\mathcal{T}_1}$ et $e_{\mathcal{T}_2}$ sont **simple**, exemple $\langle$:Paper, :Article, $\equiv$, 1$\rangle$
- une correspondance est dite **complex** si $e_{\mathcal{T}_1}$ et $e_{\mathcal{T}_2}$ sont **complex**, exemple $\langle$:Accepted_Paper, $\exists$:hasDecision.:Acceptance, $\equiv$, 1$\rangle$

Objectif

Dans ce travail, nous étudions les avantages mutuels de la combinaison des tâches l'alignement d'ontologies et d'interconnexion des données.

Découverte d'alignements expressifs

- Il existe différentes approches pour aborder la tâche d'alignement d'ontologies,
- les approches basées sur les instances font partie des ces méthodes, ils utilisent les entités réelles pour trouver des correspondances.
- **Canard** [1] est parmi eux, il trouve de correspondances simples et complexes entre un paire de graphes de connaissances.

[1] https://gitlab.irit.fr/melodi/ontology-matching/complex/canard

L'approche générale de **Canard**

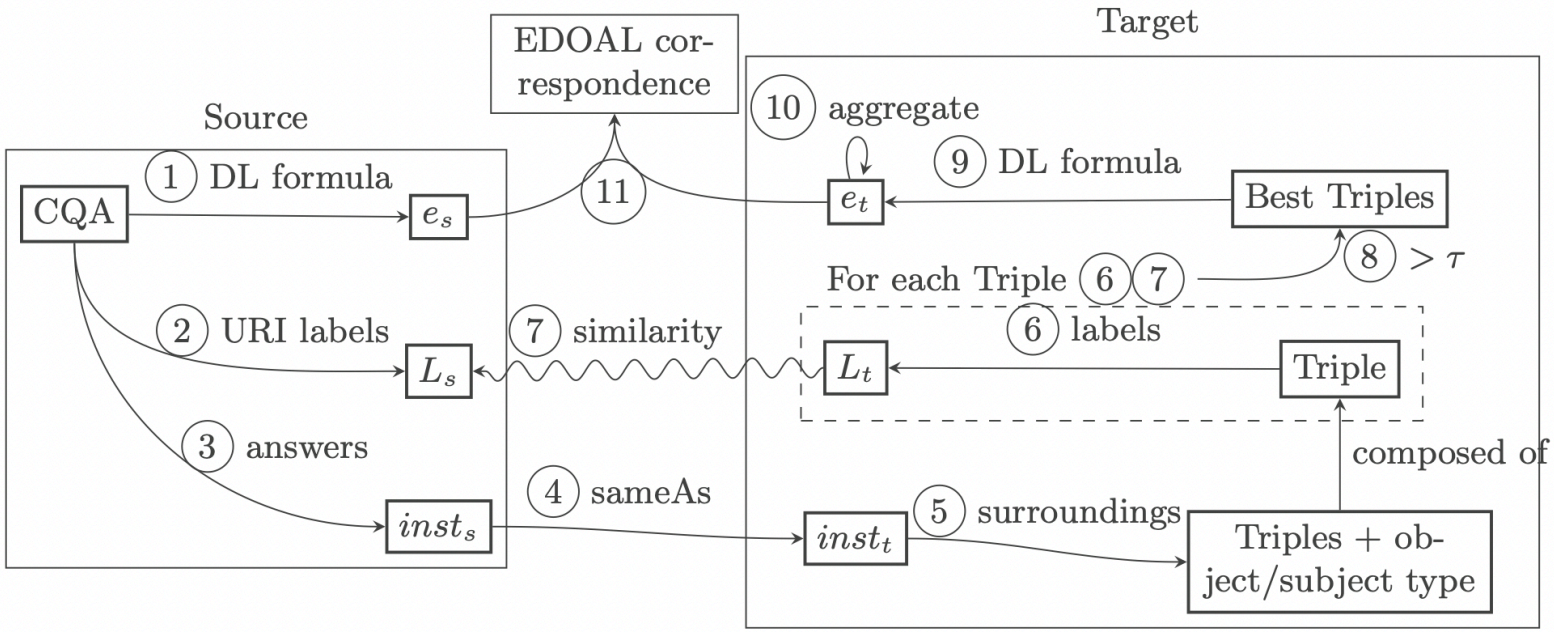


Schéma de l'approche générale de **Canard** [2], les CQAs définir la portée de l'alignement.

[2] https://oatao.univ-toulouse.fr/22677/1/thieblin_22677.pdf

Canard a besoin de sameAs!

- S'il n'y a pas de liens sameAs entre les graphes des connaissance, **Canard** ne peut établir aucun alignement.
- Nous faisons l'hypothèse qu'avoir plus de liens sameAs permet l'extraction de correspondances simples et complexes plus précises.

Linkex

1. **Linkex** est un logiciel open source [3],
2. Il permet d'extraire des clés de liage entre un paire de graphes de connaissances,
3. Ces clés de liage permettent la génération de les relations owl:sameAs.

[3] https://gitlab.inria.fr/moex/linkex

L'approche générale de **Linkex**

Graphe de connaissances source

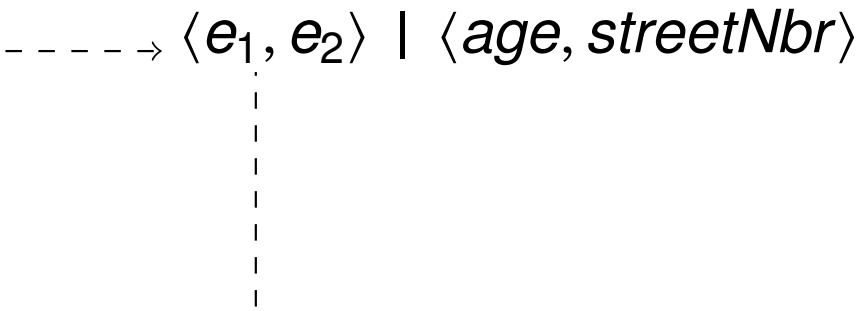
e_1 age "21";

e_1 rdf:type Person.

Graphe de connaissances cible

e_2 streetNbr "21";

e_2 rdf:type Address.



Les avantages d'utiliser des correspondances pour l'extraction de clés de liage

1. Permet de générer des clés de liage entre les classes équivalentes, permettant de générer des clés de liage discriminantes.
2. Réduit l'espace de recherche.

$\{\langle \text{age}, \text{streetNbr}\rangle\}$ linkkey $\{\langle \text{Person}, \text{Address}\rangle\}$

$\{\langle \text{fname}, \text{name}\rangle, \langle \text{lname}, \text{family\_name}\rangle\}$ linkkey $\{\langle \text{Person}, \text{Male} \sqcup \text{Female}\rangle\}$ parce que $\text{Person} \equiv \text{Male} \sqcup \text{Female}$

DICAP

- **DICAP** (**D**ata **I**interlinking and **C**omplex **A**lignment **P**ipeline) [4] est un logiciel open source,
- aborde simultanément les tâches d'interconnexion des données et d'alignement d'ontologies, permettant leur collaboration mutuelle.

[4] https://github.com/dace-dl-anr/DICAP

DICAP

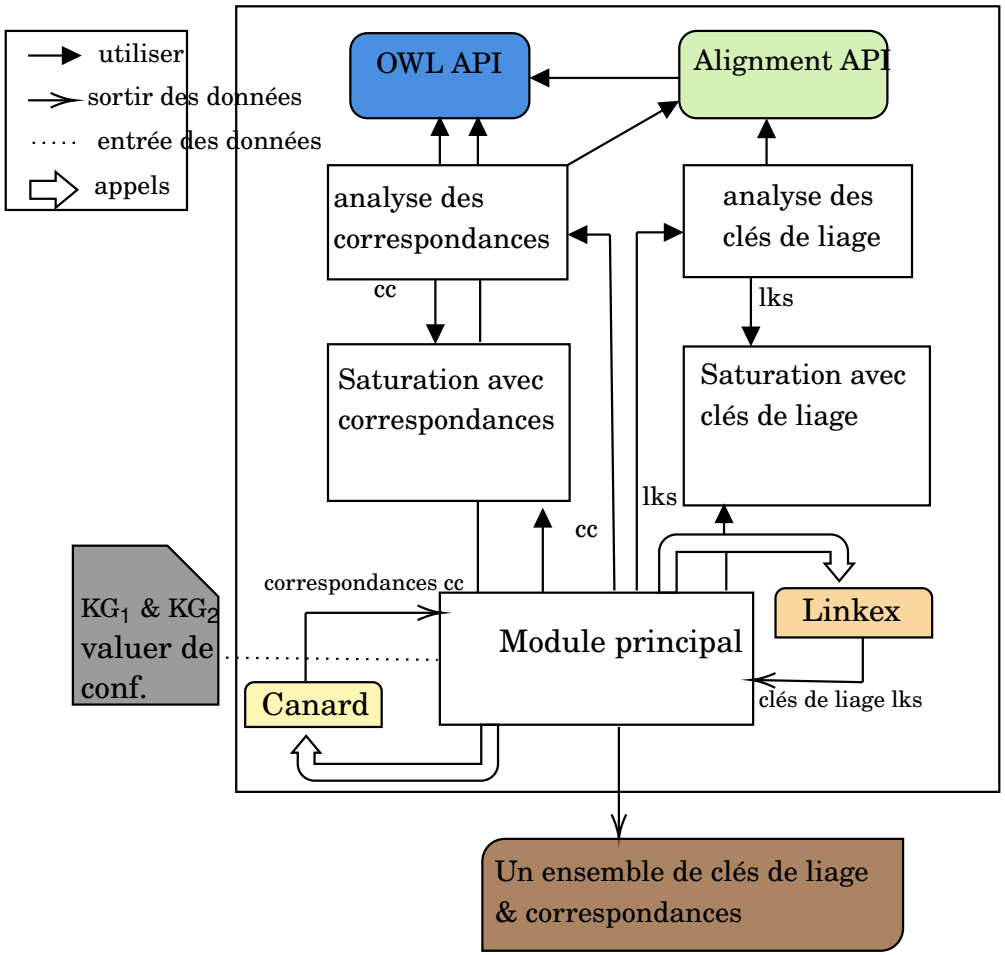
- Prend en entrée une paire de graphes de connaissances et un seuil de confiance [5],
- affiche un ensemble de correspondances et de clés de liage simples et complexes.

[5] utilisé par canard, degré de similitude entre les valeurs de propriété spécifiées par l'utilisateur

DICAP

- Appels respectivement **Canard** et **Linkex**,
- renvoie à chaque itération un ensemble de clés de liage et de correspondances,
- ces clés de liage et correspondances seront utilisées par les algorithmes de saturation,
- s'arrête lorsqu'il atteint une situation stationnaire, c'est-à-dire qu'aucune nouvelle clé de liage ou correspondance ne peut plus être obtenue.

L'architecture de **DICAP**.



Expérimentations

Nous avons lancé des expériences en utilisant les ensembles de données de conférence de l'OAEI [6].

	edas_100	**conference_100**
Taille (MB)	21.3	33.5
Individus	10 0517	21334
Axiomes	36 1087	248831
Ass. de classe	76 993	107394
DataProp.	8988	9314
ObjectProp.	196594	94020
Ann. prop.	21509	16361

Table 1: Les caractéristiques du jeu de données considéré.

[6] https://oaei.ontologymatching.org

Expérimentations

L'algorithme atteint la situation stationnaire après 2 itérations.

	LksSEq	**LksCEq**	**Corr.**
Etat initial	0	0	221
État intermédiaire state	23	18	291
État final	116	263	341

Table 2: Le nombre de correspondances et de clés de liage entre les classes équivalentes simples et complexes aux différentes itérations.

LksSEq sont les clés de liage entre les classes d'une correspondance simple.
LksCEq sont les clés de liage entre les classes d'une correspondance complexe.

Expérimentations

	DICAP	**CANARD**
précision	0.076	0.109
couverture	0.5	0.357

Table 3: La précision et la couverture des correspondances générées par **DICAP** et **CANARD**

La raison de la précision inférieure est l'utilisation du lien ($\{\langle$rdfs:label, rdfs:label$\rangle\}$ linkkey $\langle\top, \top\rangle$) dans la première itération.

Expérimentations

link key	hmean	disc.	cov.
{⟨rdfs:label,rdfs:label⟩} ⟨⊤, ⊤⟩	0.479	0.681	0.37
{⟨hasFirstName,has_the_First_Name⟩, ⟨hasLastName,has_the_Last_Name⟩} ⟨⊤,⊤⟩	0.173	0.999	0.094
{⟨hasLastName,has_the_Last_Name⟩} ⟨⊤, ⊤⟩	0.143	0.293	0.094

Table 4: Clés de liage obtenues par **Linkex**

link key	hmean	disc.	cov.
{⟨rdfs:label,rdfs:label⟩} ⟨edas:Review,conference:Review⟩	0.997	0.995	1
{⟨rdfs:label,conference:has_a_name⟩} ⟨edas:Workshop,conference:Workshop⟩	0.667	1	0.5
{⟨rdfs:label,rdfs:label⟩} ⟨edas:Review,∃ conference:has_authors.conference:Review⟩	0.567	0.995	0.397

Table 5: Clés de liage obtenues par **DICAP**

Conclusion et perspectives

1. Nous avons intégré les systèmes d'interconnexion de données (Linkex) et d'appariement d'ontologies (Canard),
2. Nous avons obtenu des clés de liage très discriminantes et amélioré la couverture des alignements.
3. Nous prévoyons de généraliser cette approche en intégrant d'autres systèmes d'appariement d'ontologies et d'interconnexion de données.

Saturation avec clés de liage

- Ajout d'assertions owl:sameAs entre individus satisfaisant les clés de liage,
  Exemple s'il y a deux individus $a \in KG_1$ et $b \in KG_2$ tels que a et b vérifient la condition d'une clé de liage λ alors on ajoute a owl:sameAs b à KG_1 et KG_2.

Saturation avec clés de liage

- Ajout d'assertions owl:sameAs entre individus satisfaisant les clés de liage,
  Exemple s'il y a deux individus $a \in KG_1$ et $b \in KG_2$ tels que a et b vérifient la condition d'une clé de liage λ alors on ajoute a owl:sameAs b à KG_1 et KG_2.
- propageant les valeurs des étiquettes, des assertions de classe et de rôle entre les individus owl:sameAs identifiés.

Saturation avec clés de liage

- Ajout d'assertions owl:sameAs entre individus satisfaisant les clés de liage,
  Exemple s'il y a deux individus $a \in KG_1$ et $b \in KG_2$ tels que a et b vérifient la condition d'une clé de liage λ alors on ajoute a owl:sameAs b à KG_1 et KG_2.
- propageant les valeurs des étiquettes, des assertions de classe et de rôle entre les individus owl:sameAs identifiés.
  Exemple s'il y a $C(a) \in \mathrm{KG}_1$ on ajoute $C(b)$ à KG_1.

Saturation avec correspondances

- ajouter $D(a)$ à KG_1 quand $C(a) \in KG_2$ et $C \equiv D$.