Rapport Réalisation du Projet Tutoré

Exploring analogical proportions between and within Knowledge Graphs (KG)

Master 1 Sciences Cognitives
IDMC - Université de Lorraine
2022-2023

Auteurs :
Adrien Chassaing-Monjou

Superviseurs :
Miguel Couceiro
Pierre Monnin
Esteban Marquer

Organisme d'accueil : LORIA

Table des matières

1 Introduction

Depuis l'introduction du Web Sémantique par Tim Berners Lee au tout début des années 2000, celui-ci s'est rapidement développé et est aujourd'hui un sujet d'étude particulièrement actif. Un peu plus de 10 ans après, les graphes de connaissances ont été introduit. Depuis, le nombre de données écrits dans le modèle RDF n'a fait qu'exploser, comme le montre Fig.1 représentant différentes données et la manière dont elles sont inter-connectées [1].

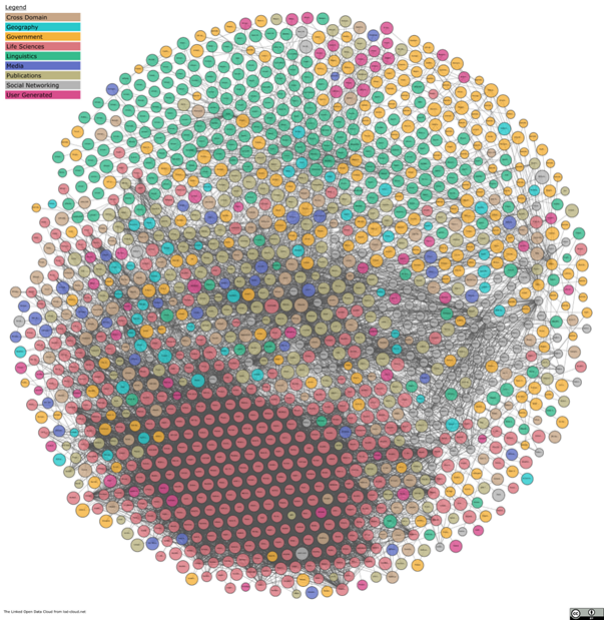

FIGURE 1 – Lod-cloud diagram

A partir de ces données il est possible d'obtenir des ontologies et leurs représentations graphiques, les graphes de connaissances. Par conséquent, au vu du nombre exponentiel de données existantes il est nécessaire de se poser les questions suivantes : Comment gérer tant de données ? Comment les connecter les unes aux autres comme en Fig.1 ? Comment est-il possible de réutiliser des données présentes dans une ontologie dans une autre du même domaine ? C'est ici qu'intervient l'alignement d'entités ou bien d'ontologies qui consiste à dire que telle entité dans une ontologie correspond à telle autre entité dans la seconde ontologie. De cette volonté d'aligner différentes ontologies entre elles est né il y a quelques années l'OAEI [2] (Ontology Alignment Evaluation Initiative) qui a pour but de proposer des jeux de données de diverses natures afin de tester différentes manières de faire des alignements. Ce projet s'insère donc dans la continuité de cette initiative.

En effet, ce projet a pour but d'étudier les graphes de connaissances, et plus particulièrement d'explorer l'utilisation de proportions analogiques pour l'alignement d'entités ou d'ontologies. Il a été réalisé sous la supervision de Miguel Couceiro (Loria), Pierre Monnin (Orange) ainsi que d'Esteban Marquer (Loria) et s'inscrit dans la continuité de la recherche bibliographique faite au semestre dernier. Il s'agit ici de la réalisation du projet

1. https ://lod-cloud.net/
2. https ://oaei.ontologymatching.org/

qui est la mise en place plus concrète d'un algorithme utilisant les proportions analogiques pour étudier des graphes de connaissances à travers le problème de l'alignement.

Dans ce rapport se trouve d'abord le contexte du projet comprenant l'explication du cadre théorique dans lequel nous nous plaçons ainsi que la définition d'un graphe de connaissances et d'une proportion analogique. Puis vient un exposé sur le travail réalisé durant ce projet. Nous continuons en expliquant les résultats obtenus. Nous finissons en discutant sur l'intérêt que nous avons trouvé dans ce sujet et l'adéquation entre le projet et notre parcours universitaire.

2 Contexte

2.1 Le modèle RDF et les Graphes de connaissances

Qui dit Web Sémantique dit bien évidemment « sémantique », c'est-à-dire que l'information disponible sur le web doit être structurée de manière à être compréhensible par l'humain ainsi que par la machine. Ceci est possible à travers un modèle nommé RDF (Resource Description Framework). Ce dernier consiste à structurer l'information, ou connaissance, sous forme de triplets, souvent appelés triplet SPO puisqu'il s'agit de triplet ayant comme forme (Sujet, Prédicat, Objet). Pour ce modèle RDF il existe différentes syntaxes (trig, n-triples, n-quads, turtle, etc.) et dans le cadre de ce projet nous travaillons avec la syntaxe turtle dont un exemple simple peut être le suivant [3] :

Exemple 1.

```
@prefix ex:<http://example.org/> .
@prefix foaf:<http://xmlns.com/foaf/0.1/> .


ex:Chica a ex:Dog ;
         ex:hasBreed ex:Chug ;
         ex:hasSize ex:Small ;
         ex:belongsTo ex:Mathieu d'Aquin .

ex:Mathieu d'Aquin a foaf:Person ;
                   a ex:Researcher ;
                   ex:gives ex:SWLecture .
```

Ici nous avons 7 triplets qui sont (`ex:Chica, a, ex:Dog`), (`ex:Chica, ex:hasBreed, ex:Chug`), (`ex:Chica, ex:hasSize, ex:Small`), (`ex:Chica ex:belongsTo ex:Mathieu d'Aquin`), (`ex:Mathieu d'Aquin a foaf:Person`), (`ex:Mathieu d'Aquin a ex:Researcher`), (`ex:Mathieu d'Aquin ex:gives ex:SWLecture`). Ceci signifie que Chica est un chien qui a pour race Chug, qui est de petite taille et qui appartient à Mathieu d'Aquin où Mathieu d'Aquin est une personne, qui est un chercheur donnant des cours de Web Sémantique. Nous voyons donc ici que nous pouvons écrire un fait (ou plusieurs) de manière systématique dans une forme particulière qu'aussi bien l'humain que la machine peut comprendre et traiter. Plus particulièrement, nous avons donc à partir de cette syntaxe une

3. Des exemples plus complexes seront étudiés dans la section suivante.

manière d'écrire le contenu d'une ontologie. Remarquons que le fait d'écrire des triplets SPO revient également à pouvoir décrire l'information contenue dans ces triplets sous forme d'un multigraphe dirigé étiqueté. Si nous reprenons notre Exemple 1 avec Chica nous obtenons :

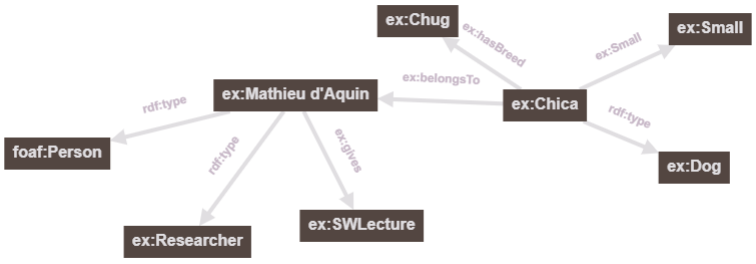

FIGURE 2 – Graphe de connaissances impliquant Chica

Cette représentation sous forme de graphe est ce qui est appelée un graphe de connaissances. Comme indiqué dans le rapport bibliographique, il en existe plusieurs définitions dont une discussion peut être trouvée dans [10] ou encore [18]. Ici nous prenons la définition qu'un graphe de connaissance est une représentation d'une ontologie sous forme d'un graphe portant sur des entités et relations (ou connaissances) comme vu précédemment. Ceci est donc l'un des deux objets au centre de ce projet.

2.2 Proportion Analogique

Une proportion analogique est un énoncé de la forme « A est à B ce que C est à D ». Elle est habituellement représentée de manière formelle de la façon suivante : « $A : B :: C : D$ » ceci mettant en avant aussi bien leurs similarités que leurs dissimilarités. Selon les auteurs (par exemple [1], [4], [28]), elle respecterait principalement les postulats suivants :

- $A : B :: A : B$ (réflexivité)
- $A : B :: C : D \rightarrow C : D :: A : B$ (symétrie)
- $A : B :: C : D \rightarrow A : C :: B : D$ (permutation centrale)

Certains auteurs rajoutent d'autres postulats tels que par exemple le déterminisme ($A : A :: A : D \Rightarrow D = A$). Ces postulats suivent les travaux de Y. Lepage [21] sur les proportions analogiques appliquées à la linguistique. Il est possible que des auteurs ne considèrent pas certains postulats comme cela est notre cas. Par exemple, dans [2], la permutation centrale n'est plus un axiome à respecter mais la symétrie interne ($A : B :: C : D \Leftrightarrow B : A :: D : C$) ou la réflexivité interne ($A : A :: C : C$) en sont. Comme dit précédemment et puisque nous souhaitons appliquer ces proportions analogiques aux graphes de connaissances, nous ne prenons pas en compte le postulat de permutation centrale qui n'a pas grand sens dans le cas des graphes de connaissances, ce que nous verrons plus loin dans le rapport.

Nous avons donc ainsi vu les deux objets sur lesquels nous travaillons dans ce projet. Nous pouvons donc à présent exposer le travail fait durant ce semestre.

3 Travail réalisé

3.1 Objectif

Nous rappelons que l'objectif du projet est d'explorer l'utilisation de la proportion analogique lors de l'alignement d'entités provenant de différents graphes. Lors de ce projet nous avons principalement travaillé avec quatre jeux de données différents, qui seront présentés dans la suite de cette section, dont nous pouvons prendre l'exemple suivant, Fig.3, pour illustrer la tâche d'alignement que nous souhaitons réaliser (ici il s'agit d'aligner « AT7-21 » avec « deployment/57490 »).

FIGURE 3 – Exemple d'une tâche d'alignement

Pour réaliser notre objectif le projet a été divisé en plusieurs étapes. La première était de réfléchir à comment transformer la formalisation mise au point lors de la partie bibliographique en problème d'optimisation afin d'avoir une base plus concrète pour l'algorithme à écrire. La seconde a été une première implémentation afin de voir si la théorie était applicable et afin de voir vers où se diriger, c'est ici que l'idée principale sur laquelle repose l'algorithme a été conçue. La troisième consiste en une modification de l'algorithme et aux premiers résultats.

3.2 Première étape : Problème d'optimisation

La première étape, comme dit ci-dessus, a été une première réflexion sur la manière de traduire en terme de problème d'optimisation la formalisation mise au point durant la période de projet portée sur la bibliographie. Tout ce qui a été fait dans cette sous-section n'a pas encore été utilisé et nous ne savons pas encore si tout le sera. Malgré cela nous commençons par rappeler les différentes définitions de notre formalisation (il s'agit de la représentation matricielle des entités, prédicats et proportions analogiques) nécessaire pour comprendre la dernière définition portant sur les proportions analogiques qui elle a été importante pour notre algorithme.

Lors de la partie bibliographique, nous avons donc mis au point une formalisation inspirée par les travaux de C. Antić [2]. Nous avons commencé par mettre au point une notation pour définir le fait que deux entités dans un même graphe de connaissances sont en relation.

Definition 3.1. *Soit $\mathcal{K} = \{\mathcal{E}, \mathcal{P}, \mathcal{F}\}$ un graphe de connaissance où $\mathcal{E}$ est l'ensemble des entités de $\mathcal{K}$, $\mathcal{P}$ est l'ensemble des prédicats de $\mathcal{K}$ et $\mathcal{F}$ l'ensemble des faits de $\mathcal{K}$, alors*

pour deux éléments $a, b \in \mathcal{E}$, nous dirons qu'ils sont en relation, dénoté $\mathcal{K} \models a : b$, si il existe $p \in \mathcal{P}$ tel que $p(a, b) \in \mathcal{F}$ que l'on dénotera $\mathcal{K} \models p(a, b)$.

Si nous reprenons notre Exemple 1 nous avons par exemple que $p =$ `ex:hasSize`, $a =$ `ex:Chica` et $b =$ `ex:Small` appartiennent au graphe de connaissances.

Comme nous voulons une représentation matricielle des entités et des prédicats, nous pouvons considérer $\mathcal{E}$ comme étant $\mathbb{R}^n$ et puisqu'un prédicat lie deux entités nous avons $\mathcal{P} = M_n(\mathbb{R})$. En particulier, en considérant un prédicat comme étant une matrice, nous avons bien que la composition de prédicats, donc le produit de matrices, reste un prédicat puisque les matrices sont stables par multiplication. Nous avons trouvé dans [22] une manière adéquate de représentation des entités/prédicats selon notre formalisation. Dans cet article, le modèle du nom d'ANALOGY considère comme nous le souhaitons les entités comme étant des vecteurs de $\mathbb{R}^n$ et les prédicats comme étant des matrices. Ils prennent cependant pour les prédicats des matrices normales de taille n dans $\mathbb{R}$, dénoté $\mathcal{N}_n(\mathbb{R})$, à savoir les matrices P telles que $PP^T = P^TP$ où P^T est la transposée de P. Ces matrices représentent de manière adéquate les relations car dans les matrices normales nous avons :

- les matrices symétriques respectant $PP^T = P^TP = P^2$ permettant de représenter les relations symétriques ;
- les matrices antisymétriques respectant $PP^T = P^TP = -P^2$ permettant de représenter les relations asymétriques ;
- les matrices de rotations respectant $PP^T = P^TP = I_n$ représentant les relations $1-1$.

Leur problème d'optimisation consiste à trouver les matrices minimisant la distance des représentations matricielles du produit entre la matrice du sujet (dénotée A) et la matrice du prédicat (dénotée P) avec la matrice de l'objet (dénotée B). Cette distance s'écrit $\langle A^TP, B\rangle = A^TPB$ où $\langle \cdot, \cdot \rangle$ est le produit scalaire.

Puisque nous voulons comparer deux graphes de connaissances il nous faut une correspondance entre ces derniers. Ceci est formulé de la façon suivante :

Definition 3.2. *Soit $\mathcal{K}_1 = \{\mathcal{E}_1, \mathcal{P}_1, \mathcal{F}_1\}$ (resp. $\mathcal{K}_2 = \{\mathcal{E}_2, \mathcal{P}_2, \mathcal{F}_2\}$) un graphe de connaissance où $\mathcal{E}_1$ (resp. $\mathcal{E}_2$) est l'ensemble des entités de $\mathcal{K}_1$ (resp. $\mathcal{K}_2$) et $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) est l'ensemble des prédicats de $\mathcal{K}_1$ (resp. $\mathcal{K}_2$). Nous appelons une application $t : \mathcal{K}_1 \to \mathcal{K}_2$ une correspondance si pour $p \in \mathcal{P}_1, q \in \mathcal{P}_2$ nous avons $(\mathcal{K}_1, \mathcal{K}_2) \models t(p) = q$ et pour $a, b \in \mathcal{E}_1$, $c, d \in \mathcal{E}_2$ avec $\mathcal{K}_1 \models p(a, b)$ et $\mathcal{K}_2 \models q(c, d)$ nous avons $(\mathcal{K}_1, \mathcal{K}_2) \models t(p(a, b)) = q(c, d)$.*

Ceci signifie que la correspondance t permet de mettre en relation aussi bien les entités du premier graphe de connaissances avec le second que les prédicats de ces graphes de connaissances.

Nous pouvons traduire ceci en problème d'optimisation de la même manière que précédemment. Nous voulons réduire la distance entre les matrices $T(P)$ et Q de telle manière que $t(p(a, b))$ soit proche de $q(c, d)$. Nous voulons donc que t respecte :

$$T^* = \underset{TP=Q}{argmin} \sum_{TA=C} \langle((TP)(TA), Q(TA))\rangle,$$

où A, C sont les matrices représentant a et c et T représente la matrice de t.

Maintenant que nous avons rappelé ce qui nous est nécessaire pour comprendre notre formalisation d'une proportion analogique nous en donnons la définition :

Definition 3.3. *Soit $\mathcal{K}_1 = \{\mathcal{E}_1, \mathcal{P}_1, \mathcal{F}_1\}$ (resp. $\mathcal{K}_2 = \{\mathcal{E}_2, \mathcal{P}_2, \mathcal{F}_2\}$) un graphe de connaissance où $\mathcal{E}_1$ (resp. $\mathcal{E}_2$) est l'ensemble des entités de $\mathcal{K}_1$ (resp. $\mathcal{K}_2$) et $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) est l'ensemble des prédicats de $\mathcal{K}_1$ (resp. $\mathcal{K}_2$). Pour $a, b \in \mathcal{E}_1$ tel qu'il existe $p \in \mathcal{P}_1$ avec $\mathcal{K}_1 \models p(a, b)$, $c \in \mathcal{E}_2$ et $q \in \mathcal{P}_2$ tel qu'il existe une correspondance t avec $(\mathcal{K}_1, \mathcal{K}_2) \models t(p) = q$, une équation analogique est une équation de la forme :*

$$(\mathcal{K}_1, \mathcal{K}_2) \models a : b :: c : x$$

où x est une variable dans $\mathcal{E}_2$. De manière équivalente, cette équation peut s'écrire sous la forme

$$(\mathcal{K}_1, \mathcal{K}_2) \models t(p(a, b)) = q(c, x).$$

Nous avons donc qu'une solution d d'une équation analogique est un élément du second graphe de connaissance qui permet à a, b, c et à d d'être en proportion analogique. Si nous reformulons cela en terme de problème d'optimisation nous voulons trouver une matrice représentant d tel que $TB = D$, où B est la matrice représentant b, qui réduit la distance entre $T(PA)$ et D. Si nous reprenons les notations du problème d'optimisation de Définition 3.2 nous obtenons :

$$D^* = \underset{TP=Q}{argmin} \sum_{D=TB} \langle T(PA), D \rangle.$$

Comme pour que les entités a, b, c, d soit en proportion analogique il faut également que leur représentation matricielle vérifie :

$$(\mathcal{K}_1, \mathcal{K}_2) \models T(P(A, B)) = T(Q(X, D))$$

pour maximiser la similartié de PA avec C et $T(PA)$ avec D, nous obtenons le problème d'optimisation suivant pour déterminer des proportions analogiques :

$$T^* = \underset{TP=Q}{argmin} \sum_{D=TB} \langle T(A), C \rangle + \underset{TP=Q}{argmin} \sum_{C=TA} \langle T(PA), D \rangle. \quad (1)$$

Ceci signifie que pour avoir une proportion analogique avec deux entités a, b en relation dans un premier graphe de connaissances, il nous faut déterminer une correspondance entre deux entités c, d en relation dans un autre graphe de connaissances de telle manière que la similartié entre a et c et entre b et d soit maximisée.

3.3 Deuxième étape : Implémentation

Une fois que le passage de la formalisation aux problèmes d'optimisation a été fait (comme ci-dessus), il a fallu implémenter les différentes idées obtenues. C'est durant cette étape que nous nous sommes rendu compte que nous n'allions pas utiliser tout de suite cette reformulation. En effet, il s'agit d'une approche matricielle, qui certes est désirée à la fin, mais qui pour l'instant n'est pas (nous pensons) réalisable au vu des compétences en algorithmie que nous possédons. Nous avons alors opté pour une approche symbolique qui est plus intuitive dans un premier temps. Cependant tout ce qui a été fait précédemment n'a pas été inutile puisque cela nous a permis de prendre connaissance d'un article [30] ayant une approche similaire à celle que nous souhaitions, et plus particulièrement de la définition de proportion analogique (Définition 3.3) que nous avons donné plus haut.

3.3.1 Inspiration

Dans cette section nous allons résumer l'article [30] de Sultan et Shahaf qui nous a permis de formuler différemment l'équation 1 qui est au coeur de notre algorithme. En particulier, cela nous a donné l'idée de définir différentes similarités entre entités (que ce soit dans un même graphe ou dans le produit cartésien de deux graphes) afin de chercher les correspondances T^*.

Dans cet article, Oren Sultan et Dafna Shahaf, expliquent le fonctionnement de leur algorithme qui a pour but de détecter des analogies entre des situations ou des processus. Pour ce faire, ils se placent dans le cadre du Structure Mapping Theory de Gentner, c'est-à-dire trouver un mapping entre un ensemble d'entités de départ et d'arrivé (comme vu dans le rapport bibliographique, se référer à [14] pour un exposé sur la Structure Mapping Theory de Gentner).

Puisque leur algorithme a pour but de miner des analogies entre des processus provenant de texte, il leur faut d'abord individualiser les entités sur lequel l'algorithme travaillera. Les entités représentent dans leur cadre de travail des phrases nominales et les relations des verbes liant ces entités. Leur procédé d'extraction de ces entités et de ces verbes prend en compte leur sémantique, par exemple l'entité « cellules animales » devrait être similaire à « cellule » et les relations doivent être capable de capturer la similarité sémantique entre deux entités liées par une relation et deux autres entités liées par une autre relation.

Pour extraire les entités et résoudre le problème de différentes occurrences d'un même nom comme dans l'exemple cité ci-dessus, ils réalisent des clusters d'entités selon l'apparition d'un nom commun aux différentes phrases nominales puis ne considèrent qu'un représentant pour chaque cluster, le mot le plus court qui n'est pas un pronom ou un verbe. Ceci leur permet de ne récupérer qu'un nom sans pronom.

En ce qui concerne les relations, donc les verbes, ils les extraient en utilisant un QA-SRL, c'est-à-dire qu'ils posent des questions utilisant un verbe cible ainsi qu'une entité, par exemple « Qu'est-ce qui *verbe entité* ? » ou encore « Qui *verbe entité* ? ». Ceci leur permet par conséquent d'obtenir des ensembles de relations pour chaque paire d'entités (la seconde entité est la réponse à la question). Avec un exemple plus concret : Qu'est-ce qui synthétise les protéines ? Les cellules. Qu'est-ce qui utilise les protéines ? Les cellules. Par conséquent $\mathcal{R}$(cellules, protéines) = {synthétise, utilise}.

Leur objectif est de trouver des analogies entre des entités liées par une relation provenant du premier texte avec des entités liées par une relation dans le second texte. Cela signifie qu'ils veulent trouver une similarité entre ces entités à travers leurs relations. Ils définissent pour cela une métrique sur leur similarité relationnelle. Celle-ci est définit de telle manière que des ensembles de relations sont similaires s'ils possèdent des relations différentes mais ayant une similarité sémantique proche. Plus exactement, ils cherchent à maximiser la similarité entre l'ensemble des relations liants deux entités du premier texte avec l'ensemble des relations liant deux autres entités provenant du second texte (dans les

deux ordres). Plus formellement cela correspond à

$$sim^*(b_i, b_j, t_k, t_l) = sim(\mathcal{R}(b_i, b_j), \mathcal{R}(t_k, t_l)) + sim(\mathcal{R}(b_j, b_i), \mathcal{R}(t_l, t_k))$$

Une fois cette métrique définie, leur problème d'optimisation pour trouver un mapping entre les entités est le suivant :

$$\mathcal{M}^* = \underset{\mathcal{M}}{argmax} \sum_{\substack{j \in [1, n-1] \\ i \in [j+1, n]}} sim^*(b_j, b_i, \mathcal{M}(b_j), \mathcal{M}(b_i)) \tag{2}$$

Cela signifie qu'un mapping maximise la relation de similarité entre une paire d'entités du premier texte avec une paire d'entités du second texte. Ou autrement dit, cela signifie que plus les ensembles de relations entre b_i et b_j et entre $\mathcal{M}(b_i)$ et $\mathcal{M}(b_j)$ sont similaires selon leur métrique, plus $\mathcal{M}$ est un bon candidat. Nous faisons remarquer ici que leur problème d'optimisation est très proche du nôtre, d'où le fait que nous nous soyons inspiré de cet article pour la suite.

Si nous regardons Equation 1, nous voyons que celle-ci est exprimée en terme de $argmin$ tandis que Equation 2 est exprimée en terme de $argmax$. Cela s'explique par le fait que dans notre approche matricielle nous souhaitons réduire une distance tandis que pour l'autre équation il s'agit de maximiser une relation. Nous reformulons Equation 1 dans la suite pour qu'elle corresponde à Equation 2.

3.3.2 L'idée centrale

Comme nous venons de voir, si nous souhaitons trouver des proportions analogiques entre entités de deux graphes de connaissances, il nous faut définir des relations de similarité entre les entités des différents graphes. En particulier, puisqu'une proportion analogique considère un couple d'éléments dans un premier graphe et un couple d'éléments dans un second graphe il nous faut définir à minima une relation de similarité entre les entités d'un même graphe et une relation de similarité entre les entités de deux graphes.

Puisque nous souhaitions adapter l'approche de l'article [30] à la nôtre, il a fallu trouver une manière de remplacer leur approche sémantique utilisant le QA-SRL avec une manière d'étudier la structure du voisinage d'une entité dans un graphe de connaissances. Pour ce faire nous avons repris l'idée du courant structuraliste en philosophie des mathématiques qui est de dire que ce n'est pas parce qu'une entité est quelque chose qu'elle a des propriétés particulières mais qu'au contraire, c'est parce que cette entité possède ces propriétés qu'elle est ce quelque chose. Autrement dit, les propriétés des entités (au sens philosophique d'être quelque chose ayant des propriétés) sont intrinsèques et non extrinsèques et les entités ne sont que des instances d'une structure qui elle possède des propriétés [7].

A partir de cette idée nous avons donc pensé à regarder la similarité entre deux entités faisant parties d'un même graphe de connaissances grâce aux prédicats pour lesquelles elles sont des sujets ou bien des objets. Plus précisément, nous regardons indirectement le schema de l'ontologie en nous focalisant sur les domaines de départ (i.e. les `rdfs:domain`) et d'arrivé des prédicats (i.e. les `rdfs:range`).

En effet, si nous définissons notre similarité en nous basant sur l'intersection des ensembles de prédicats pour lesquels deux entités sont sujets ou objets, plus ils ont des prédicats en commun, donc plus l'intersection des domaines des prédicats est petit (sauf si les domaines sont égaux) et plus les entités partagent des attributs en commun. Il s'agit ainsi d'une manière d'éviter de travailler avec la sémantique des mots comme dans [30].

Bien évidemment, cette similarité n'est pas suffisante pour pouvoir faire des alignements en utilisant les proportions analogiques. Comme nous l'avons expliqué plus haut, une proportion analogique implique une relation entre entités d'un même graphe mais également entre entités de différents graphes. Puisqu'ici nous ne regardons une similarité qu'entre les paires au sein d'un graphe de connaissances et que nous ne croisons pas les similarités, il nous faut d'autres indices sur leur similarité. Nous avons eu alors l'idée de regarder les URI des entités et les littéraux présents dans leur voisinage. La raison de cela est qu'une ontologie est construite par l'Homme et que comme nous l'avons expliqué dans la section 2.1 celle-ci doit être interprétable par celui-ci tout comme par la machine. Par conséquent, bien que cela ne soit pas le cas tout le temps, pour que l'Homme puisse comprendre et connaître ce qui est présent dans une ontologie, il lui faut écrire l'information présente dans celle-ci avec des termes qui sont interprétables. Nous obtenons donc des informations à partir de la façon dont l'ontologie est construite. Par exemple, dans l'un de nos jeux de données, nous avions un morceau de l'URI des entités dans la première ontologie qui était présente comme littéral dans la seconde ontologie. Par conséquent, nous avons une première similarité dans un même graphe qui regarde les prédicats des entités et une seconde similarité qui fait le lien entre ces entités provenant de différents graphes de connaissances. Si deux entités sont similaires dans un même graphe, alors des entités qui sont syntaxiquement proches à ces entités auront tendance à être similaire de la même façon dans le second graphe de connaissances.

Nous donnons à présent une description de l'algorithme dans ce qui suit.

3.4 Troisième étape : L'algorithme

Durant cette période de projet il y a eu 3 versions différentes de l'algorithme. La première version était la plus naïve et malheureusement était biaisée comme nous allons le présenter.

Quelque soit la version de l'algorithme, la première étape était de créer une liste de triplet SPO pour chaque graphe de connaissances à partir d'un fichier écrit en turtle en utilisant la librairie rdflib de Python puis de se débarrasser des nœuds vides qui empêchaient l'algorithme de fonctionner correctement. Ensuite, un score était calculé pour chaque couple d'éléments à partir de différentes fonctions que nous allons présenter. Nous pouvons trouver le pseudo-code de ces différentes étapes en Annexe.

Dans la version 1, la première fonction (sim_pred) calculait la similarité entre un élément du premier graphe de connaissances et un second élément du deuxième graphe de connaissances en regardant l'intersection de l'ensemble des prédicats qui étaient liés à ces entités. Il s'agit donc de la similarité basée sur l'idée présentée plus haut concernant la

structure d'un voisinage. La seconde fonction (jaccard) calculait le Jaccard [4] de l'ensemble des prédicats utilisés dans la fonction précédente. Ce Jaccard était calculé pour pondérer la similarité liée aux prédicats (il a été supprimé par la suite car comme nos superviseurs nous l'ont fait remarquer, cela était redondant avec notre similarité basée sur les prédicats). Ensuite une troisième fonction (sim_litt) comparait l'URI transformée en string d'une entité dans un graphe avec les labels dans le voisinage d'une seconde entité provenant du second graphe. La quatrième fonction (sim_type) comparait le type qu'avait chaque entité des graphes de connaissances pour chaque combinaison possible de couple où les entités pouvaient provenir du même graphe ou non. Finalement, nous cherchions à maximiser la somme des scores pour chaque couple possible afin de trouver le meilleur mapping (cf. pseudo-code 5) comme expliqué plus haut dans ce rapport. Il s'agit ici de notre Equation 1 reformulée pour nos similarités. Plus explicitement celle-ci est la suivante :

$$T^* = \underset{T}{argmax} \sum_{a,b \in \mathcal{E}_1} sim^*(a, b, T(a), T(b)),$$

où sim^* est la somme des similartiés décrites plus haut (voir pseudo-code 1, 2, 3, 4 et 5).

Plusieurs problèmes nous ont été signalés. L'un était que les scores retournés par les fonctions n'étaient pas normalisés et surtout que selon les fonctions, le score variait entre 5 et 10 (ce problème nous a été par la suite évident lorsqu'en cours d'Agents Intelligents nous avons appris qu'il ne fallait pas mettre de scores « intermédiaires » car ceci forçait l'algorithme à avoir un comportement particulier). De plus, le plus gros problème était que le calcul final du score que nous cherchions à maximiser n'était pas optimisé et comprenait 4 boucles *for* imbriquées donnant une complexité de $O(n^4)$. Ce calcul pouvait par conséquent prendre plusieurs heures pour des jeux de données d'une centaine d'entité (alors que dans la dernière version nous travaillons avec des jeux de données d'environ 1000 entités).

La seconde version de l'algorithme n'a eu que quelques modifications (cf. pseudo-code 9). En particulier nous avons commencé à normaliser les résultats et nous avons rajouté un calcul supplémentaire dans notre fonction sim_litt qui considérait les URI et les labels (voir pseudo-code 7). Ce nouveau calcul comparait les littéraux (dont les labels) présents dans le voisinage d'une entité avec les littéraux dans le voisinage d'une seconde entité. L'idée ici était d'utiliser un peu plus le principe que les littéraux sont présents dans une ontologie pour donner des informations interprétables par l'Homme pour l'identification de l'entité. De plus, sous les conseils de nos superviseurs nous avons rajouté des « mémoisation », une astuce que nous ne connaissions pas. Autrement aucune modification dans cette version n'a été apporté.

En ce qui concerne la troisième version de l'algorithme, celle-ci est une réécriture complète de la version 2 sous les conseils de l'un de nos superviseurs, Esteban Marquer. En effet, comme dit précédemment, le calcul du score final pouvait prendre plusieurs heures pour des jeux de données de petite taille. Avec cette nouvelle version, le calcul pour ces jeux de données ne prenait qu'une ou deux secondes ce qui nous a permis de tester notre algorithme sur des jeux de données 10 fois plus gros. Le calcul dans ce cas là ne prend

4. Le Jaccard de deux ensembles, ou indice de Jaccard, est défini comme étant le cardinal de l'intersection des ensembles divisé par le cardinal de l'union de ces deux ensembles.

qu'une vingtaine ou trentaine de minutes environ.

La réécriture sous les conseils de Esteban Marquer à grandement changé la manière dont les calculs sont effectués. Dans cette nouvelle version, ces derniers sont fait à partir de « one-hot vectors » plutôt que des strings comme dans notre version d'origine, ce qui bien évidemment optimise grandement notre algorithme. Également, les outils proposés par la bibliothèque rdflib ont été utilisés de manière plus adéquate, ce qui optimise également les calculs. Il s'agit d'une version heuristique qui ne considère, au fil des calculs, que les paires ayant potentiellement le plus de chance d'être dans le top K, où K est le nombre de proportions analogiques que nous souhaitons retourner. Une fois notre algorithme réécrit, nous avons fait des modifications mineures en suivant toujours ses conseils. Nous avons par exemple enlever la pondération de notre fonction de similarité à partir du Jaccard. Nous avons également changé les autres pondérations qui n'étaient pas adéquates (cf. les pseudo-codes 10, 11, 12, 13). Une fois cela fait nous avons donc testé notre algorithme sur plusieurs jeux de données provenant de l'OAEI, ce que nous exposons maintenant.

4 Expériences et discussion

Dans cette section nous allons commencer par décrire les différents jeux de données utilisés provenant de l'OAEI puis présenter nos résultats. Nous finirons par une discussion sur nos résultats et les perspectives à venir.

4.1 Présentation des jeux de données

4.1.1 Geolink

Lors de ce projet nous avons testé notre algorithme sur des jeux de données qui, comme dit précédemment, proviennent de l'OAEI. En particulier nous avons utilisé quatre jeux de données provenant des compagnes OAEI de 2020 et 2022.

Les deux premiers proviennent du « GeoLink Cruise Instance Matching Track » [5] et sont nommés « gbor2r » et « gbobcodmo ». Différentes tâches sont présentées dans le Geolink project, la première consistant à retrouver des alignements d'*entités* provenant de ces deux ontologies. Ces alignements « gold » – ceux que nous souhaitons retrouver – sont disponibles dans un fichier mis à disposition par l'OAEI.

Les deux ontologies consistent en des données décrivant des croisières liées à des expéditions scientifiques en géologie (d'où leur nom). Nous donnons en Fig.4 et Fig.5 un exemple de graphe de connaissances provenant chacun d'une des deux ontologies, chaque figure ne représente que le voisinage direct de ces entités (il s'agit d'une entité que nous devons aligner comme vu en Fig.3).

Dans ces deux ontologies se trouvent 40 classes. Pour « gbobcodmo » il y a 1061 individus tandis que pour « gbor2r » il y a 5320 individus. Malgré le fait que notre dernière version est bien plus rapide, le nombre d'individus dans les deux ontologies est trop grand pour celle-ci rendant les calculs trop longs et ces derniers n'arrivent pas à terme sur notre

5. http ://oaei.ontologymatching.org/2020/geolinkcruise/index.html

FIGURE 4 – Graphe de connaissances de l'entité « AT7-21 » provenant de gbo2r2

FIGURE 5 – Graphe de connaissances de l'entité « deployment 57490 » provenant de gbobcodmo

ordinateur. Nous avons donc créé manuellement deux ontologies restreintes en prenant des entités de manière aléatoire. Celle qui est une restriction de « gbo2r2 » contient 664 individus et celle provenant de « gbobcodmo » contient 442 individus. Nous avons fait attention à ce que le nombre d'individus ne soient pas trop grand pour un premier test afin que l'algorithme tourne assez vite. Dans le fichier de référence pour les alignements golds, 491 alignements sont recensés mais dans les jeux de données artificiels il n'y en a plus que 225.

4.1.2 Anatomy

Les autres ontologies que nous avons utilisé proviennent de l'OAEI 2022 et sont nommés « human » et « mouse ». Elles font parties de « Anatomy track » [6]. La tâche ici est d'aligner l'*ontologie* « human » décrivant l'anatomie humaine avec la seconde *ontologie* « mouse » décrivant elle l'anatomie de la souris adulte. La première ontologie contient 3304 classes et la seconde 2744 classes. Nous avons donc, comme pour le Geolink project, créé deux ontologies artificielles. La première remplaçant celle sur l'anatomie humaine contenait 937 classes et celle remplaçant « mouse » contenait 1041 classes. A l'origine le fichier de référence pour les alignements gold en contenait 1516 mais nous sommes passés à seulement 68 alignements gold présent dans les jeux de données artificiels.

4.2 Résultats

Dans cette section nous exposons nos résultats, c'est-à-dire que nous donnons les alignements gold trouvés ainsi que les proportions analogiques obtenues avec une discussion sur ces derniers.

En ce qui concerne les ontologies provenant de Geolink, nous rappelons qu'il y a dans le fichier de référence 491 alignements gold mais que dans nos jeux de données ils s'élèvent au nombre de 225. Comme nous avons travaillé sur des ontologies créées manuellement, il est normal qu'il y en ait moins dedans. Notre algorithme nous a renvoyé 1170 alignements alors que nous lui avons demandé 3000 proportions analogiques, ce qui implique une répétition des alignements dans celles-ci. De plus, nous faisons remarquer que notre algorithme nous renvoie des alignements aussi bien de la forme $A \to C$ que $C \to A$. Nous

6. http ://oaei.ontologymatching.org/2020/anatomy/index.html

pouvons par conséquent considérer qu'il n'y a en réalité que 585 alignements. Sur ces 585 alignements, les 202 premiers sont des alignements gold, les 23 autres alignements gold sont compris entre les alignements numéro 502 et 525. Nous avons mis en Annexe la Table A.1 représentant les premiers alignements trouvés.

Un fait intéressant est que les autres alignements détéctés sont tous des alignements reliants des entités de type « award », comme illustré en Table A.2, ce qui n'est en soit pas incohérent puisque cela signifie que nous avons des proportions analogiques de la forme « une croisière est à une croisière ce qu'un prix est à un prix » ou encore « un prix est à un prix ce qu'un prix est à un prix ». De plus ce fait est intéressant parce que notre algorithme est censé privilégier à travers la similarité sur les littéraux les alignements d'entités dont l'une possède dans son voisinage un littéral qui correspond (au moins en parti) à l'URI de l'autre entité comme nous pouvons voir Fig. 4 et Fig. 5. Il est donc normal que les alignements gold soient trouvés. Nous voyons ici que notre algorithme détecte et aligne correctement à minima les labels mais qu'il peut faire plus que cela.

Pour les ontologies artificielles provenant de Anatomy, étrangement, malgré environ 1000 classes dans chacune d'entre elles, il n'y a que 68 alignements gold tandis que dans le fichier de référence nous en avons 1516. Puisque ces ontologies sont plus grandes que celles de Geolink, nous avons demandé à ce que l'algorithme nous retourne 5000 proportions analogiques plutôt que 3000. Ceci nous a retourné 1658 alignements, donc 784 alignements modulo la permutation de A et C. Sur ces 784 alignements, seulement 44 sont des alignements gold mais les 11 premiers sont des alignements gold. Nous avons mis en Annexe la Table A.3 représentant les 40 premiers alignements gold trouvés par notre algorithme. Leur URI a été remplacé par leur label pour une meilleure lisibilité des résultats (autrement les URI des entités ressemblent à `http://mouse.owl#MA_0000796` pour l'ontologie mouse ou encore à `http://human.owl#NCI_C12299` pour l'ontologie human).

Bien que nous n'ayons pas trouvé beaucoup d'alignements gold, il faut cependant faire attention car nous sommes dans ce projet intéressés par les proportions analogiques qui leurs sont associées. De plus comme il n'y avait pas beaucoup d'alignements à trouver ce résultat est à relativiser. Nous pouvons voir dans la Table A.3 que les premiers alignements ayant un score maximal concernent les classes dont les labels ne contiennent que peu de mots. Au vu de notre algorithme, ils sont effectivement privilégiés. Ce qui est intéressant vis-à-vis de nos résultats n'est pas les alignements en tant que tels mais les proportions analogiques obtenues. Bien évidemment, les premières d'entre elles concernent les classes apparaissant dans la Table A.3 puisqu'elles ont le score le plus grand. Ces proportions analogiques sont représentées dans la Table A.4 présent en Annexe.

Parmi les proportions analogiques obtenues, il est donc plus intéressant de regarder celles qui viennent un peu plus tard dans le classement comme nous pouvons voir dans la Table A.5.

Dans ces proportions analogiques nous pouvons remarquer, par exemple avec la 58ième proportion analogique « nipple : lower lip : : Nipple : Lip » et la 59ième proportion analogique « nipple : upper lip : : Nipple : Lip » que notre algorithme permet de faire des alignements complexes, c'est-à-dire aligner deux entités différentes mais similaires sur une seule

entité. Ici nous avons que « upper lip » et « lower lip » ont une même similarité/dissimilarité avec « nipple » dans le premier jeu de données que « Lip » avec « Nipple » dans le second jeu de données, ce qui est un résultat intéressant. De plus, dans le jeu de données « mouse » il n'y a pas de classe (ou sous-classe) représentant « Lip », par conséquent nous voyons que les classes « upper lip » et « lower lip » dans le premier jeu de données se sont retranchées sur la classe « Lip » dans le second jeu de données qui est la classe la plus proche de la leur. En ayant ce genre de résultat il est possible de penser que si nous fixons dans notre algorithme une classe dans un jeu de données, il nous serait possible de trouver les sous-classes de l'équivalent de cette classe dans le second jeu de données, c'est-à-dire en apprendre plus sur le schema de la seconde ontologie. Dans cet exemple il n'y a que 2 sous-classes présentes mais si nous regardons la 55ième proportion analogique (nipple : midbrain meninges : : Nipple : Meninges) ainsi que la 56ième (nipple : hindbrain meninges : : Nipple : Meninges) et la 57ième (nipple : telecephalon meninges : : Nipple : Meninges) nous avons la présence de 3 sous-classes de Meninges.

5 Conclusion

Dans ce rapport, nous avons commencé par rappeler le contexte dans lequel se place ce projet, c'est-à-dire que nous avons exposer rapidement les outils théoriques nécessaires pour la bonne compréhension de ce dernier. En particulier nous avons expliqué ce qu'est le modèle RDF ainsi que ce que sont les graphes de connaissances et les proportions analogiques.

Dans un second temps, nous avons détaillé les différentes étapes que nous avons suivi lors de ce projet afin de répondre à notre objectif rappelé au début de cette partie. Les trois étapes consistaient en : *i)* reformuler la formalisation créée au premier semestre pour pouvoir la rendre plus concrète dans le but d'écrire un algorithme la respectant ; *ii)* compléter et corriger cette formalisation à partir de [30], ce qui nous a permis de dégager l'idée centrale sur laquelle se base notre algorithme ainsi que notre formule finale pour le problème d'optimisation que nous souhaitons résoudre ; *iii)* écrire, tester et corriger empiriquement notre algorithme.

Dans un dernier temps, nous avons présenté les jeux de données sur lesquels nous avons travaillé pour ensuite exposer nos résultats. Ces derniers sont intéressants mais montrent que nous devons certainement modifier légèrement notre algorithme et peut-être également travailler avec des jeux de données plus complexes. En particulier, il a été discuté de travailler avec les labels des entités des voisinages plutôt que les URI de ces derniers. Nous pourrions également mettre un peu plus en avant l'utilisation des prédicats dans notre approche en appliquant notre similarité sim_pred entre les graphes de connaissances. Différentes tâches telles que celles discutées plus haut sur trouver les sous-classes ou encore aligner des ontologies dans différentes langues seront à tester pour notre algorithme afin de voir son efficacité. Ceci pourra être fait dans un travail futur, par exemple lors du stage qui nous a été proposé cet été afin de continuer notre travail. Ce stage, en colaboration avec l'IDMC, le Loria et l'INRAE (Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement) nous permettra d'utiliser notre algorithme sur des données dans le domaine de l'agro-alimentaire et par conséquent cela nous permettra également de raffiner notre algorithme.

Ce projet s'inscrit bien dans le parcours M1 Sciences Cognitives que nous suivons car celui-ci repose sur aussi bien l'EC Algorithmes de l'UE 701 que sur l'EC Semantic Web de l'UE 801. De plus, comme dit dans ce rapport, certains cours impliquant de la programmation comme l'EC Agents Intelligents & Collectifs de l'UE 803 nous ont été utiles dans la réflexion qu'elle nous a amené à produire. Puisque notre projet nous a demandé un certain nombre de compétence en programmation dans le langage Python, il aurait été agréable d'avoir un cours supplémentaire au second semestre.

Références

[1] Safa Alsaidi et al. « A Neural Approach for Detecting Morphological Analogies ». In : *8th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2021, Porto, Portugal, October 6-9, 2021*. IEEE, 2021, p. 1-10. doi : 10.1109/DSAA53316.2021.9564186. url : https://doi.org/10.1109/DSAA53316.2021.9564186.

[2] Christian Antic. « Analogical proportions ». In : *Ann. Math. Artif. Intell.* 90.6 (2022), p. 595-644. doi : 10.1007/s10472-022-09798-y. url : https://doi.org/10.1007/s10472-022-09798-y.

[3] Christian Antic. « Boolean proportions ». In : *CoRR* abs/2109.00388 (2021). arXiv : 2109.00388. url : https://arxiv.org/abs/2109.00388.

[4] Nelly Barbot, Laurent Miclet et Henri Prade. « Analogy between concepts ». In : *Artif. Intell.* 275 (2019), p. 487-539. doi : 10.1016/j.artint.2019.06.008. url : https://doi.org/10.1016/j.artint.2019.06.008.

[5] Antoine Bordes et al. « A semantic matching energy function for learning with multi-relational data - Application to word-sense disambiguation ». In : *Mach. Learn.* 94.2 (2014), p. 233-259. doi : 10.1007/s10994-013-5363-6. url : https://doi.org/10.1007/s10994-013-5363-6.

[6] Hongyun Cai, Vincent W. Zheng et Kevin Chen-Chuan Chang. « A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications ». In : *IEEE Trans. Knowl. Data Eng.* 30.9 (2018), p. 1616-1637. doi : 10.1109/TKDE.2018.2807452. url : https://doi.org/10.1109/TKDE.2018.2807452.

[7] Adrien Chassaing-Monjou. « Le concept de structure en mathématiques ». Mém. de mast. Université Bordeaux-Montaigne - Master Epistémologie, Histoire des Sciences et des Techniques, 2021, p. 1-43.

[8] Zhe Chen et al. « Knowledge Graph Completion: A Review ». In : *IEEE Access* 8 (2020), p. 192435-192456. doi : 10.1109/ACCESS.2020.3030076. url : https://doi.org/10.1109/ACCESS.2020.3030076.

[9] Aleksandr Drozd, Anna Gladkova et Satoshi Matsuoka. « Word Embeddings, Analogies, and Machine Learning: Beyond king - man + woman = queen ». In : *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. Sous la dir. de Nicoletta Calzolari, Yuji Matsumoto et Rashmi Prasad. ACL, 2016, p. 3519-3530. url : https://aclanthology.org/C16-1332/.

[10] Lisa Ehrlinger et Wolfram Wöss. « Towards a Definition of Knowledge Graphs ». In : *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016*. Sous la dir. de Michael Martin, Martı Cuquet et Erwin Folmer. T. 1695. CEUR Workshop Proceedings. CEUR-WS.org, 2016. url : https://ceur-ws.org/Vol-1695/paper4.pdf.

[11] Chris Eliasmith et Paul Thagard. « Integrating structure and meaning: a distributed model of analogical mapping ». In : *Cogn. Sci.* 25.2 (2001), p. 245-286. doi : 10.1016/S0364-0213(01)00036-2. url : https://doi.org/10.1016/S0364-0213(01)00036-2.

[12] Brian Falkenhainer, Kenneth D. Forbus et Dedre Gentner. « The Structure-Mapping Engine: Algorithm and Examples ». In : *Artif. Intell.* 41.1 (1989), p. 1-63. doi : 10.1016/0004-3702(89)90077-5. url : https://doi.org/10.1016/0004-3702(89)90077-5.

[13] Kenneth D. Forbus et al. « CogSketch: Sketch Understanding for Cognitive Science Research and for Education ». In : *Top. Cogn. Sci.* 3.4 (2011), p. 648-666. doi : 10.1111/j.1756-8765.2011.01149.x. url : https://doi.org/10.1111/j.1756-8765.2011.01149.x.

[14] Dedre Gentner. « Structure-Mapping: A Theoretical Framework for Analogy ». In : *Cogn. Sci.* 7.2 (1983), p. 155-170. doi : 10.1207/s15516709cog0702_3. url : https://doi.org/10.1207/s15516709cog0702%5C_3.

[15] Dedre Gentner et Arthur B. Markman. « Analogy - Watershed or Waterloo? Structural Alignment and the Development of Connectionist Models of Cognition ». In : *Advances in Neural Information Processing Systems 5, [NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992]*. Sous la dir. de Stephen Jose Hanson, Jack D. Cowan et C. Lee Giles. Morgan Kaufmann, 1992, p. 855-862. url : http://papers.nips.cc/paper/624-analogy-watershed-or-waterloo-structural-alignment-and-the-development-of-connectionist-models-of-cognition.

[16] Graeme Halford et al. « Connectionist Implications for Processing Capacity Limitations in Analogies ». In : *Advances in Connectionist and Neural Computation Theory: Analogical Connections, Vol. 2, Chapter 7*. T. 2. Journal Abbreviation: Advances in Connectionist and Neural Computation Theory: Analogical Connections, Vol. 2, Chapter 7. Jan. 1994, p. 363-415.

[17] Felix Hill et al. « Learning to Make Analogies by Contrasting Abstract Relational Structure ». In : *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. url : https://openreview.net/forum?id=SylLYsCcFm.

[18] Aidan Hogan et al. « Knowledge Graphs ». In : *ACM Comput. Surv.* 54.4 (2022), 71:1-71:37. doi : 10.1145/3447772. url : https://doi.org/10.1145/3447772.

[19] Shaoxiong Ji et al. « A Survey on Knowledge Graphs: Representation, Acquisition, and Applications ». In : *IEEE Trans. Neural Networks Learn. Syst.* 33.2 (2022), p. 494-514. doi : 10.1109/TNNLS.2021.3070843. url : https://doi.org/10.1109/TNNLS.2021.3070843.

[20] Yuval Kirstain, Ori Ram et Omer Levy. « Coreference Resolution without Span Representations ». In : *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*. Sous la dir. de Chengqing Zong et al. Association for Computational Linguistics, 2021, p. 14-19. doi : 10.18653/v1/2021.acl-short.3. url : https://doi.org/10.18653/v1/2021.acl-short.3.

[21] Yves Lepage. « Analogy and Formal Languages ». In : *Proceedings of the joint meeting of the 6th Conference on Formal Grammar (FG) and the 7th Conference on Mathematics of Language (MOL), FGMOL 2001, Helsinki, Finland, August 10-12, 2001*. Sous la dir. de Lawrence S. Moss et Richard T. Oehrle. T. 53. Electronic Notes in Theoretical Computer Science. Elsevier, 2001, p. 180-191. doi : 10.1016/S1571-0661(05)82582-4. url : https://doi.org/10.1016/S1571-0661(05)82582-4.

[22] Hanxiao Liu, Yuexin Wu et Yiming Yang. « Analogical Inference for Multi-relational Embeddings ». In : *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Sous la dir. de Doina Precup et Yee Whye Teh. T. 70. Proceedings of Machine Learning Research. PMLR, 2017, p. 2168-2178. url : http://proceedings.mlr.press/v70/liu17d.html.

[23] Pierre Monnin et Miguel Couceiro. « Interactions Between Knowledge Graph-Related Tasks and Analogical Reasoning: A Discussion ». In : *Workshop ATA@ICCBR 2022 - Analogies: from Theory to Applications*. Nancy (FR), France, sept. 2022. url : https://hal.science/hal-04057391.

[24] Pierre Monnin et al. « Discovering alignment relations with Graph Convolutional Networks: A biomedical case study ». In : *Semantic Web* 13.3 (2022), p. 379-398. doi : 10.3233/SW-210452. url : https://doi.org/10.3233/SW-210452.

[25] Maximilian Nickel et al. « A Review of Relational Machine Learning for Knowledge Graphs ». In : *Proc. IEEE* 104.1 (2016), p. 11-33. doi : 10.1109/JPROC.2015.2483592. url : https://doi.org/10.1109/JPROC.2015.2483592.

[26] Heiko Paulheim. « Knowledge graph refinement: A survey of approaches and evaluation methods ». In : *Semantic Web* 8.3 (2017), p. 489-508. doi : 10.3233/SW-160218. url : https://doi.org/10.3233/SW-160218.

[27] Henri Prade et Gilles Richard. « Analogical Proportions and Analogical Reasoning - An Introduction ». In : *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*. Sous la dir. de David W. Aha et Jean Lieber. T. 10339. Lecture Notes in Computer Science. Springer, 2017, p. 16-32. doi : 10.1007/978-3-319-61030-6_2. url : https://doi.org/10.1007/978-3-319-61030-6%5C_2.

[28] Henri Prade et Gilles Richard. « Analogical Proportions: Why They Are Useful in AI ». In : *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Sous la dir. de Zhi-Hua Zhou. ijcai.org, 2021, p. 4568-4576. doi : 10.24963/ijcai.2021/621. url : https://doi.org/10.24963/ijcai.2021/621.

[29] Michael Sejr Schlichtkrull et al. « Modeling Relational Data with Graph Convolutional Networks ». In : *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*. Sous la dir. d'Aldo Gangemi et al. T. 10843. Lecture Notes in Computer Science. Springer, 2018, p. 593-607. doi : 10.1007/978-3-319-93417-4_38. url : https://doi.org/10.1007/978-3-319-93417-4%5C_38.

[30] Oren Sultan et Dafna Shahaf. « Life is a Circus and We are the Clowns: Automatically Finding Analogies between Situations and Processes ». In : *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Sous la dir. d'Yoav Goldberg, Zornitsa Kozareva et Yue Zhang. Association for Computational Linguistics, 2022, p. 3547-3562. url : https://aclanthology.org/2022.emnlp-main.232.

[31] Bishan Yang et al. « Embedding Entities and Relations for Learning and Inference in Knowledge Bases ». In : *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Sous la dir. d'Yoshua Bengio et Yann LeCun. 2015. url : http://arxiv.org/abs/1412.6575.

[32] Lu Zhou et al. « A Complex Alignment Benchmark: GeoLink Dataset ». In : *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II*. Sous la dir. de Denny Vrandecic et al. T. 11137. Lecture Notes in Computer Science. Springer, 2018, p. 273-288. doi : 10.1007/978-3-030-00668-6_17. url : https://doi.org/10.1007/978-3-030-00668-6%5C_17.

A Tables des résultats

indice	entité 1	entité 2	score
1	deployment/58003	AT11-17	8.505
2	AT11-17	deployment/58003	8.505
3	AT11-30	deployment/58004	8.505
4	deployment/58004	AT11-30	8.505
5	deployment/59025	BH10-14	8.481
6	BH10-14	deployment/59025	8.481
7	deployment/58719	BH10-03	8.454
8	BH10-03	deployment/58719	8.454
9	AE1103	deployment/543356	8.453
10	deployment/543356	AE1103	8.453
11	AE1114	deployment/543358	8.450
12	deployment/543358	AE1114	8.450
13	AE1224	deployment/543346	8.446
14	AE1215	deployment/543348	8.446
15	deployment/543346	AE1224	8.446

Table 1 – 15 premiers alignements Geolink

indice	entité 1	entité 2	score
1001	award/100274	award/546818	8.252
1002	award/100362	award/546818	8.252
1003	award/546818	award/100362	8.252
1004	deployment/563697	EN538	8.252
1005	EN538	deployment/563697	8.252
1006	deployment/563687	EN532	8.252
1007	EN532	deployment/563687	8.252
1008	HRS1402	deployment/565457	8.252
1009	deployment/550442	AT26-23	8.252
1010	AT26-23	deployment/550442	8.252
1011	deployment/565616	HRS1422	8.252
1012	HRS1422	deployment/565616	8.252
1013	deployment/565535	HRS1416	8.252
1014	HRS1416	deployment/565535	8.252
1015	deployment/565457	HRS1402	8.252
1016	HRS110714EN	deployment/565814	8.252
1017	deployment/514970	HRS100709BC	8.252
1018	HRS100709BC	deployment/514970	8.252
1019	deployment/474285	NBP1002	8.252
1020	NBP1002	deployment/474285	8.252

Table 2 – Alignements contenant « awards » Geolink

indice	entité 1	entité 2	score
0	nipple	Nipple	6.0
1	prepuce	Prepuce	6.0
2	areola	Areola	6.0
3	Artery	artery	6.0
4	Areola	areola	6.0
5	artery	Artery	6.0
6	Prepuce	prepuce	6.0
7	epididymis	Epididymis	6.0
8	Epididymis	epididymis	6.0
9	myocardium	Myocardium	6.0
10	Myocardium	myocardium	6.0
11	Nipple	nipple	6.0
12	cartilage	Cartilage	6.0
13	Cartilage	cartilage	6.0
14	subiculum	Subiculum	6.0
15	Subiculum	subiculum	6.0
16	Trunk	trunk	6.0
17	trunk	Trunk	6.0
18	main bronchus	Main_Bronchus	5.5
19	Wrist_Joint	wrist joint	5.5
20	Main_Bronchus	main bronchus	5.5
31	wrist joint	Wrist_Joint	5.5
46	terminal bronchiole	Terminal_Bronchiole	5.5
48	Terminal_Bronchiole	terminal bronchiole	5.5
72	Ventricle_Brain	brain ventricle	5.5
78	brain ventricle	Ventricle_Brain	5.5
81	Arteriole_Endothelium	arteriole endothelium	5.5
82	aorta endothelium	Aorta_Endothelium	5.5
83	Aorta_Endothelium	aorta endothelium	5.5
86	limb digit	Digit	5.5
87	Digit	limb digit	5.5
94	arteriole endothelium	Arteriole_Endothelium	5.5
95	Artery_Endothelium	artery endothelium	5.5
96	thymus medulla	Thymus_Medulla	5.5
97	Thymus_Medulla	thymus medulla	5.5
98	thymus cortex	Thymus_Cortex	5.5
99	Thymus_Cortex	thymus cortex	5.5
100	capillary endothelium	Capillary_Endothelium	5.5
101	Capillary_Endothelium	capillary endothelium	5.5
102	artery endothelium	Artery_Endothelium	5.5
103	Sweat_Gland	sweat gland	5.5

TABLE 3 – 40 premiers alignements gold détectés pour Anatomy

indice	proportion analogique
0	nipple : subiculum : : Nipple : Subiculum
1	areola : subiculum : : Areola : Subiculum
2	areola : nipple : : Areola : Nipple
3	prepuce : subiculum : : Prepuce : Subiculum
4	prepuce : nipple : : Prepuce : Nipple
5	prepuce : areola : : Prepuce : Areola
6	epididymis : subiculum : : Epididymis : Subiculum
7	epididymis : nipple : : Epididymis : Nipple
8	epididymis : areola : : Epididymis : Areola
9	epididymis : prepuce : : Epididymis : Prepuce
10	myocardium : subiculum : : Myocardium : Subiculum
11	myocardium : nipple : : Myocardium : Nipple
12	myocardium : areola : : Myocardium : Areola
13	myocardium : prepuce : : Myocardium : Prepuce
14	myocardium : epididymis : : Myocardium : Epididymis
15	cartilage : subiculum : : Cartilage : Subiculum
16	cartilage : nipple : : Cartilage : Nipple
17	cartilage : areola : : Cartilage : Areola
18	cartilage : prepuce : : Cartilage : Prepuce
19	cartilage : epididymis : : Cartilage : Epididymis
20	cartilage : myocardium : : Cartilage : Myocardium
21	artery : subiculum : : Artery : Subiculum
22	artery : nipple : : Artery : Nipple
23	artery : areola : : Artery : Areola
24	artery : prepuce : : Artery : Prepuce
25	artery : epididymis : : Artery : Epididymis
26	artery : myocardium : : Artery : Myocardium
27	artery : cartilage : : Artery : Cartilage
28	trunk : subiculum : : Trunk : Subiculum
29	trunk : nipple : : Trunk : Nipple
30	trunk : areola : : Trunk : Areola
31	trunk : prepuce : : Trunk : Prepuce
32	trunk : epididymis : : Trunk : Epididymis
33	trunk : myocardium : : Trunk : Myocardium
34	trunk : cartilage : : Trunk : Cartilage
35	trunk : artery : : Trunk : Artery
36	subiculum : palpebral conjunctiva : : Subiculum : Conjunctiva
37	subiculum : auricular cartilage : : Subiculum : Cartilage
38	subiculum : tensor tympani : : Subiculum : Tensor_Tympani
39	subiculum : tectorial membrane : : Subiculum : Tectorial_Membrane
40	subiculum : midbrain meninges : : Subiculum : Meninges

TABLE 4 – 40 premières proportions analogiques pour Anatomy

indice	proportion analogique
50	nipple : palpebral conjunctiva : : Nipple : Conjunctiva
51	nipple : auricular cartilage : : Nipple : Cartilage
52	nipple : tensor tympani : : Nipple : Tensor_Tympani
53	nipple : tectorial membrane : : Nipple : Tectorial_Membrane
54	nipple : midbrain meninges : : Nipple : Meninges
55	nipple : hindbrain meninges : : Nipple : Meninges
56	nipple : telencephalon meninges : : Nipple : Meninges
57	nipple : lower lip : : Nipple : Lip
58	nipple : upper lip : : Nipple : Lip
59	nipple : forebrain meninges : : Nipple : Meninges
60	nipple : diencephalon meninges : : Nipple : Meninges
61	nipple : choroid plexus : : Nipple : Choroid
62	nipple : brain ventricle : : Nipple : Ventricle_Brain
63	nipple : brain meninges : : Nipple : Meninges
64	areola : palpebral conjunctiva : : Areola : Conjunctiva
65	areola : auricular cartilage : : Areola : Cartilage

TABLE 5 – 50-65ième proportions analogiques pour Anatomy

B Les pseudo-codes des différentes versions de l'algorithme

Algorithm 1: sim_pred V1

Data: e_1, e_2 in the same KG
Result: score_pred
create $P_1 = \{$predicates p for which e_1 is either subject or object$\}$
create $P_2 = \{$predicates p for which e_2 is either subject or object$\}$
for *p in* P_1 **do**
  if *p in* P_2 **then**
    score $\leftarrow$ score_pred + 1
  end
end

Algorithm 2: Jaccard V1

Data: e_1, e_2 in the same KG
Result: score_jacc
create $P_1 = \{$predicates p for which e_1 is either subject or object$\}$
create $P_2 = \{$predicates p for which e_2 is either subject or object$\}$
for *p in* P_1 **do**
  if *p in* P_2 **then**
    score_jacc $\leftarrow$ score_jacc + 1
  end
end
score $\leftarrow$ score/(len($P_1 \bigcup P_2$))

Algorithm 3: sim_type V1

Data: e_1, e_2 entities of the same KG or of two KGs
Result: score_type
create $T_1 =$ { types t_1 of e_1 }
create $T_2 =$ { types t_2 of e_2 }
for *t in T_1* **do**
  if *t in T_2* **then**
    score_type $\leftarrow$ score_type + 10
  end
end

Algorithm 4: sim_litt V1

Data: e_1, e_2 entities of the same KG or of two KGs
Result: score_litt
create $L_1 =$ {litterals l_1 that are linked to e_1 }
create $L_2 =$ {litterals l_2 that are linked to e_2 }
for *l_1 in L_1* **do**
  if *a part of the string of the URI of e_2 in l_1* **then**
    score_litt $\leftarrow$ score_litt + 5
  end
end
for *l_2 in L_2* **do**
  if *a part of the string of the URI of e_1 in l_2* **then**
    score_litt $\leftarrow$ score_litt + 5
  end
end

Algorithm 5: Version 1

Data: entities $(e_1, e_2), (fe_1, fe_2)$ of two different KGs
Result: analogical proportion $e_1 : e_2 :: fe_1 : fe_2$
Remove blank nodes
for *all possible pairs of entities* **do**
  compute sim_pred
  compute Jaccard
  compute sim_litt
  compute *sim*_type
  score $\leftarrow$ score_pred*score_jacc + score_litt + score_type
end
Look for the quadruplets $(e_1, e_2, fe_1, fe_2) \in KG_1^2 \times KG_2^2$ that maximise the score

Algorithm 6: sim_pred V2

Data: e_1, e_2 in the same KG
Result: score_pred
create $P_1 = \{$predicates p for which e_1 is either subject or object$\}$
create $P_2 = \{$predicates p for which e_2 is either subject or object$\}$
for *p in P_1* **do**
  if *p in P_2* **then**
    score $\leftarrow$ score_pred + 1/len(P_1)
  end
end

Algorithm 7: sim_litt V2

Data: e_1, e_2 entities of the same KG or of two KGs
Result: score_litt
create $L_1 = \{$litterals l_1 that are linked to e_1 $\}$
create $L_2 = \{$litterals l_2 that are linked to e_2 $\}$
for *l_1 in L_1* **do**
  if *a part of the string of the URI of e_2 in l_1* **then**
    score_litt $\leftarrow$ score_litt + 1
  end
end
for *l_2 in L_2* **do**
  if *a part of the string of the URI of e_1 in l_2* **then**
    score_litt $\leftarrow$ score_litt + 1
  end
end
for *l_1 in L_1* **do**
  for *l_2 in L_2* **do**
    for *each word of l_1* **do**
      if *word in l_2* **then**
        score_litt $\leftarrow$ score_litt + 1/(len(l_1)*len(l_2))
      end
    end
  end
end

Algorithm 8: sim_type V2

Data: e_1, e_2 entities of the same KG or of two KGs
Result: score_type
create $T_1 = \{$ types t_1 of e_1 $\}$
create $T_2 = \{$ types t_2 of e_2 $\}$
for *t in T_1* **do**
  if *t in T_2* **then**
    score_type $\leftarrow$ score_type + 1/(len(T_1))
  end
end

Algorithm 9: Version 2

Data: entities $(e_1, e_2), (fe_1, fe_2)$ of two different KGs
Result: analogical proportion $e_1 : e_2 :: fe_1 : fe_2$
Remove blank nodes
for *all possible pairs of entities* **do**
  compute sim_pred
  compute sim_litt
  compute *sim*_type
  score ← score_pred + score_litt + score_type
end
Look for the quadruplets $(e_1, e_2, fe_1, fe_2) \in KG_1^2 \times KG_2^2$ that maximise the score

Algorithm 10: sim_pred V3

Data: Array of one-hot vectors of e_1, e_2 two entities of the same KG
Result: score_pred
score_pred ← sum of the elementwise multiplication of one-hot vectors / sum of the scalar product of the one-hot vectors

Algorithm 11: sim_litt V3

Data: e_1, e_2 entities of the same KG or of two KGs
Result: score_litt
for *lit a list of lists of substrings of litterals of* e_2 **do**
  for *substrings of the string URI of* e_1 **do**
    if *substrings in lit* **then**
      score_litt ← score_litt + 1/len(lit)
    end
  end
end
for *lit a list of lists of substrings of litterals of* e_1 **do**
  for *substrings of the string URI of* e_2 **do**
    if *substrings in lit* **then**
      score_litt ← score_litt + 1/len(lit)
    end
  end
end
for lit_1 *a list of lists of substrings of litterals of* e_1 **do**
  for lit_2 *a list of lists of substrings of litterals of* e_2 **do**
    for *word in* lit_1 **do**
      if *word in* lit_2 **then**
        score_litt ← score_litt+1/((len(lit_1)*len(lit_2))
      end
    end
  end
end

Algorithm 12: sim_type V3

Data: e_1, e_2 two entities of the same KG
Result: score_type
Create $T_1 = \{$ types of $e_1\}$
Create $T_2 = \{$ types of $e_2\}$
score_type $\leftarrow$ len$(T_1 \bigcap T_2)$

Algorithm 13: Version 3

Data: Entities $(e1, e2), (fe_1, fe_2)$ of two different KGs
Result: Analogies $(e1, fe_1), (e_2, fe_2)$ and Analogical Proportions
$e_1 : e_2 :: fe_1 : fe_2$ with higher score
Create sorted lists of subjects, predicates and objects for each KG whithout blank nodes
Encode as a boolean vector the predicates for each subject
Compute sim_pred with the boolean vectors in both KGs
Compute sim_type in both KGs
Compute sim_lit in the cartesian product of the KGs
for e_1 *in subject of* KG_1 **do**
  for fe_1 *in subject of* KG_2 **do**
    Look for the maximal possible score to consider only potential better solutions e_1, fe_1
    for e_2 *in subject of* KG_1 **do**
      Look for the maximal possible score to consider only potential better solution e_2
      for fe_2 *in subject of* KG_2 **do**
        score $\leftarrow$ sim_pred + sim_type + sim_litt
      end
    end
  end
end