Partitionnement d'ontologies pour le passage à l'échelle des techniques d'alignement

Fayçal Hamdi *, Brigitte Safar*
Haïfa Zargayouna*,**, Chantal Reynaud*

*LRI, Université Paris-Sud, Bât. G, INRIA Futurs
2-4 rue Jacques Monod, F-91893 Orsay, France
{Faycal.Hamdi, safar, reynaud}@lri.fr,
http://www.lri.fr
**LIPN, Université Paris 13 - CNRS UMR 7030,
99 av. J.B. Clément, 93440 Villetaneuse, France.
Haifa.Zargayouna@lipn.univ-paris13.fr

Résumé. L'alignement d'ontologies est une tâche importante dans les systèmes d'intégration puisqu'elle autorise la prise en compte conjointe de ressources décrites par des ontologies différentes, en identifiant des appariements entre concepts. Avec l'apparition de très grandes ontologies dans des domaines comme la médecine ou l'agronomie, les techniques d'alignement, qui mettent souvent en œuvre des calculs complexes, se trouvent face à un défi : passer à l'échelle. Pour relever ce défi, nous proposons dans cet article deux méthodes de partitionnement, conçues pour prendre en compte, le plus tôt possible, l'objectif d'alignement. Ces méthodes permettent de décomposer les deux ontologies à aligner en deux ensembles de blocs de taille limitée et tels que les éléments susceptibles d'être appariés se retrouvent concentrés dans un ensemble minimal de blocs qui seront effectivement comparés. Les résultats des tests effectuées avec nos deux méthodes sur différents couples d'ontologies montrent leur efficacité.

1 Introduction

Le développement rapide des technologies internet a engendré un intérêt croissant dans la recherche sur le partage et l'intégration de sources dispersées dans un environnement distribué. Le Web sémantique (Berners-Lee et al., 2001) offre la possibilité à des agents logiciels d'exploiter des représentations du contenu des sources. Les ontologies ont été reconnues comme une composante essentielle pour le partage des connaissances et la réalisation de cette vision. En définissant les concepts associés à des domaines particuliers, elles permettent à la fois de décrire le contenu des sources à intégrer et d'expliciter le vocabulaire utilisable dans des requêtes par des utilisateurs. Toutefois, il est peu probable qu'une ontologie globale couvrant l'ensemble des systèmes distribués puisse être développée. Dans la pratique, les ontologies de différents systèmes sont développées indépendamment les unes des autres par des communautés différentes. Ainsi, si les connaissances et les données doivent être partagées, il est essentiel d'établir des correspondances sémantiques entre les ontologies qui les décrivent. La tâche

d'alignement d'ontologies (recherche de mappings ou appariements entre concepts) est donc particulièrement importante dans les systèmes d'intégration puisqu'elle autorise la prise en compte conjointe de ressources décrites par des ontologies différentes. Ce thème de recherche a donné lieu à de très nombreux travaux (Shvaiko et Euzenat, 2005).

Les techniques actuelles d'alignement s'appuient en général sur des mesures calculant la similarité de couples de concepts issus des deux ontologies. Ces mesures sont pour la plupart fondées sur les caractéristiques lexicales des labels des concepts et/ou les caractéristiques structurelles des ontologies (Rahm et Bernstein, 2001), (Noy et Musen, 2000), (Reynaud et Safar, 2007) ce qui implique la comparaison de chaque description de concept d'une ontologie avec les descriptions de tous les concepts de l'autre ontologie. Elles sont souvent testées sur des ontologies de petite taille (quelques centaines de concepts).

Quand les ontologies sont de très grande taille, par exemple en Agronomie ou en Médecine, des ontologies comportent plusieurs dizaines de milliers de concepts (AGROVOC : 28 439, NALT : 42 326, NCI : 27 652)[1], l'efficacité des méthodes d'alignement automatique diminue considérablement que ce soit en terme de temps d'exécution, de taille mémoire utilisée ou de la précision des mappings obtenus du fait de l'augmentation du bruit.

Une solution possible pour résoudre ce problème est d'essayer de limiter la taille des ensembles de concepts en entrée de l'outil d'alignement, et pour cela de partitionner les deux ontologies à aligner en plusieurs blocs, afin de n'avoir à traiter que des blocs de taille raisonnable.

Nous proposons deux méthodes de partitionnement orientées par la tâche d'alignement. Elles sont en partie inspirées des techniques de co-clustering, qui consistent à exploiter, en plus des informations exprimées par les relations entre les concepts au sein d'une même ontologie, celles qui correspondent aux relations pouvant exister entre les concepts des deux ontologies. Le fait que des concepts des deux ontologies puissent avoir exactement le même label et puissent être reliés par une relation d'équivalence est un exemple de relation facile à calculer même sur des ontologies de grande taille, et dont nous allons tirer parti dans notre proposition. Nos méthodes commenceront donc par identifier avec une mesure de similarité stricte et peu coûteuse à calculer, les couples de concepts issus des deux ontologies dont le label est identique, et s'appuieront sur ces concepts, appelés des *ancres*, pour effectuer les partitions.

Le reste de cet article est organisé comme suit. Dans la section suivante, nous présentons le contexte de travail et quelques travaux dans le domaine du partitionnement, puis nous détaillons plus précisément l'algorithme de partionnement FALCON (Hu et al., 2006) sur lequel s'appuyent nos propositions. La section 3 détaille nos deux méthodes de partitionnement et la section 4 présente et analyse les résultats expérimentaux qui démontrent l'efficacité de ces méthodes. Enfin, nous concluons et donnons quelques perspectives en section 5.

2 Contexte et état de l'art

Le problème qui nous intéresse ici est celui du passage à l'échelle des méthodes d'alignement d'ontologies.

2.1 Contexte

[1] http ://www4.fao.org/agrovoc/, http ://agclass.nal.usda.gov/agt/, http ://www.mindswap.org/2003/CancerOntology/

Une ontologie correspond à une description d'un domaine d'application en terme de concepts caractérisés par des attributs et reliés par des relations. La tâche d'alignement d'ontologies consiste à générer le plus automatiquement possible des relations ou appariements entre les concepts de deux ontologies, les types de relations établies entre les concepts par appariements pouvant être des relations d'équivalence *isEq*, de subsomption *isA* ou de proximité *isClose*. Quand les ontologies sont de très grande taille, l'efficacité des méthodes d'alignement automatique diminue considérablement. La solution que nous envisageons est de limiter la taille des ensembles de concepts en entrée de l'outil d'alignement, et pour cela de partitionner les deux ontologies à aligner en plusieurs blocs, afin de n'avoir à traiter que des blocs de taille raisonnable. Les deux ensembles de blocs obtenus devront ensuite être alignés par paires comprenant un bloc de chacun des deux ensembles et l'objectif consiste à minimiser le nombre de paires de blocs à aligner.

Notre contribution est l'élaboration d'un algorithme de partitionnement adapté au contexte de l'alignement et applicable à toutes ontologies contenant une hiérarchie de concepts munis de labels puisqu'il n'exploitera que les relations de subsomption entre concepts et leurs labels.

Partitionner un ensemble E consiste à trouver des sous ensembles disjoints $E_1, E_2, ..., E_n$, d'éléments sémantiquement proches c.à.d. liés par un nombre de relations important. La réalisation de cet objectif consiste à maximiser les relations à l'intérieur d'un sous-ensemble et à minimiser les relations entre les différents sous-ensembles.

La qualité du résultat d'un partitionnement sera appréciée selon les critères suivants :

- La taille des blocs générés : les blocs doivent avoir une taille inférieure au nombre d'éléments que peut traiter l'outil d'alignement.
- Le nombre de blocs générés : ce nombre doit être le plus faible possible pour limiter le nombre de paires de blocs à aligner.
- Le degré de cohésion des blocs : un bloc aura une forte cohésion si les relations structurelles sont fortes à l'intérieur du bloc et faibles à l'extérieur, de façon à ce que les éléments des deux ontologies susceptibles d'être appariés se retrouvent concentrés dans un ensemble minimal de blocs qui seront effectivement comparés.

Le fait que l'algorithme de partitionnement n'exploite, dans un traitement léger, que les liens de subsomption entre concepts, permet le partitionnement d'ontologies de très grande taille et le passage à l'échelle.

2.2 Etat de l'art

Dans les domaines d'applications réelles, les ontologies devenant de plus en plus volumineuses, de nombreux travaux (Stuckenschmidt et Klein, 2004), (Grau et al., 2005) et (Hu et al., 2006) se sont intéressés au problème de leur partitionnement.

Ainsi les travaux de Stuckenschmidt et Klein (2004) visent la décomposition d'une ontologie en sous-blocs (ou *îlots*) indépendants les uns des autres, de façon à faciliter en toute généralité différentes opérations sur les ontologies comme la maintenance, la visualisation, la validation ou le raisonnement. Cette méthode n'est pas adaptée à notre problème car le processus de génération des blocs impose une contrainte sur la taille minimale des blocs générés qui n'est pas appropriée pour l'alignement. De plus, elle construit beaucoup trop de petits blocs, ce qui a un impact négatif sur l'étape d'alignement finale. Les travaux présentés dans la conférence ModularOntology-ISWC (2006) se concentrent plus particulièrement sur les problèmes

de raisonnement et cherchent à construire des modules centrés autour de sous-thématiques cohérentes et auto-suffisantes pour raisonner. Par exemple, les travaux de Grau et al. (2005) très représentatifs de cette problématique, garantissent que tous les concepts reliés par des liens de subsomption sont regroupés dans un seul module. Pour des ontologies comportant des dizaines de milliers de relations de subsomption (comme AGROVOC et NALT) ce type de contrainte peut conduire à la création de modules de tailles très mal réparties, inutilisables pour l'alignement. Seul le système FALCON (Hu et al., 2006) est réalisé dans un contexte d'alignement d'ontologies, mais nous verrons que sa méthode de décomposition ne prend pas complètement en compte toutes les contraintes imposées par ce contexte, en particulier le fait de travailler sur deux ontologies.

2.3 Méthode FALCON

La méthode proposée dans FALCON (Hu et al., 2006) consiste à décomposer en blocs chaque ontologie indépendamment l'une de l'autre, en utilisant la méthode de clustering ROCK (Guha et al., 2000), puis à mesurer la proximité de chacun des blocs d'une ontologie avec chaque bloc de l'autre ontologie de façon à n'effectuer l'alignement qu'entre les concepts des paires de blocs les plus proches.

Pour construire la partition, alors que ROCK considère que les liens entre les concepts ont tous la même valeur, FALCON introduit la notion de *liens pondérés*[2] qui s'appuie principalement sur une mesure structurelle de similarité entre concepts.

2.3.1 Liens pondérés

Soient c_i, c_j deux concepts d'une même ontologie O, c_{ij} leur plus petit ancêtre commun et $depthOf(c)$ la distance en nombre d'arcs entre le concept c et la racine de O. FALCON mesure la valeur du lien reliant c_i et c_j notée $Link_s(c_i, c_j)$ en utilisant la mesure de Wu et Palmer (1994) :

$$Link_s\left(c_i, c_j\right) = \frac{2 * depthOf\left(c_{ij}\right)}{depthOf\left(c_i\right) + depthOf\left(c_j\right)}$$

Dans une ontologie de grande taille, ce calcul peut prendre beaucoup de temps. En considérant que seuls les concepts de profondeurs adjacentes auront des liens de valeurs élevées, FALCON ne compare que les concepts qui satisfont la relation suivante :

$$\left|depthOf\left(c_i\right) - depthOf\left(c_j\right)\right| \leq 1$$

2.3.2 Algorithme de Partitionnement

L'algorithme permettant à FALCON de partitionner une ontologie en blocs s'appuie sur deux notions essentielles : la *cohésion* au sein d'un bloc et le *couplage* entre deux blocs distincts. La cohésion est une mesure du poids de l'ensemble des liens reliant les concepts appartenant à un même bloc et le couplage, du poids de l'ensemble des liens reliant les concepts

[2]La description de l'algorithme de partionnement de FALCON présenté ici s'appuie sur l'implémentation disponible à l'adresse suivante http ://iws.seu.edu.cn/projects/matching/

de deux blocs différents. Les formules de calcul de la cohésion et du couplage s'expriment à partir d'une même mesure appelée *goodness* :

$$goodness\,(B_i, B_j) = \frac{\sum_{c_i \in B_i, c_j \in B_j} link\,(c_i, c_j)}{sizeOf\,(B_i) \cdot sizeOf\,(B_j)}$$

$Cohésion(B_i) = goodness(B_i, B_i)$, $Couplage(B_i, B_j) = goodness(B_i, B_j)$ où $B_i \neq B_j$.

Etant donnée une ontologie O, l'algorithme prend en entrée l'ensemble B des n blocs à partitionner, où chaque bloc est réduit au départ à un unique concept de O, et le nombre k de blocs souhaité en sortie. Il initialise tout d'abord la valeur de cohésion de chaque bloc ainsi que les valeurs de couplage. A chaque itération, l'algorithme choisit le bloc qui a la cohésion maximale et le bloc qui a la valeur de couplage maximale avec ce premier bloc. Il remplace ces deux blocs par celui résultant de leur fusion et met à jour les valeurs de couplage de tous les blocs avec ce nouveau bloc. L'algorithme s'arrête quand il a atteint le nombre de blocs souhaité ou quand tous les blocs fusionnables ont atteint une taille limite.

2.3.3 Identification des paires de blocs à aligner

Une fois réalisé de façon séparée le partitionnement des deux ontologies, l'évaluation de la proximité des blocs s'effectue en s'appuyant sur des *ancres*, i.e. des appariements préalablement connus entre les termes des deux ontologies, définis par des techniques de comparaison de chaînes de caractères ou par un expert. Plus deux blocs contiennent d'ancres communes, plus ils sont jugés proches.

Soient k (resp. k') le nombre de blocs générés par le partitionnement d'une ontologie O (resp. O') et B_i (resp. B'_j) un de ces blocs. Soit la fonction $anchors(B_u, B'_v)$ qui calcule le nombre d'ancres partagées par les deux blocs B_u et B'_v et soit $\sum_{v=1}^{k'} anchors(B_i, B'_v)$ le nombre d'ancres contenues par un bloc B_i. La relation de *Proximité* entre deux blocs B_i et B'_j est définie comme suit :

$$Proximity(B_i, B'_j) = \frac{2.anchors(B_i, B'_j)}{\sum_{u=1}^{k} anchors(B_u, B'_j) + \sum_{v=1}^{k'} anchors(B_i, B'_v)}$$

Les paires de blocs alignées sont toutes les paires dont la proximité est supérieure à un seuil donné $\epsilon_2 \in [0, 1]$. Un bloc pourra donc être aligné avec plusieurs blocs de l'autre ontologie ou avec aucun suivant la valeur choisie pour ce seuil.

Cette méthode permet à FALCON de décomposer des ontologies volumineuses, mais la décomposition est faite a priori, sans prendre en compte l'objectif d'alignement, en s'appliquant sur chaque ontologie indépendamment l'une de l'autre. Le partitionnement étant fait à l'aveugle, certaines ancres pourront ne pas se trouver dans des blocs finalement alignés et l'alignement résultant ne comprendra pas forcément tous les appariements souhaités. Enfin, le calcul des blocs pertinents à aligner est coûteux (en temps de traitement).

Malgré ces critiques, l'algorithme de décomposition de FALCON est le plus adapté à la tâche d'alignement des algorithmes de partitionnement existants puisqu'il permet de contrôler la taille maximale des blocs construits.

Les deux méthodes que nous proposons le réutilisent en modifiant sa façon de générer les blocs. Notre idée est de prendre en considération au plus tôt lors du partitionnement, toutes les données relatives à l'alignement existant entre les concepts des deux ontologies et d'essayer de faire, au moins dans la deuxième méthode, du co-clustering.

3 Méthodes de partitionnement orientés alignement

Pour prendre en compte au plus tôt l'objectif d'alignement, nos méthodes vont s'appuyer sur deux données : d'une part les couples de concepts issus des deux ontologies qui ont exactement le même label et peuvent être reliés par une relation d'équivalence et d'autre part, l'éventuelle dissymétrie structurelle des 2 ontologies à aligner.

Même sur des ontologies de grande taille, il est possible d'identifier avec une mesure de similarité stricte et peu coûteuse à calculer, les concepts dont un des labels est identique à l'un des labels d'un concept de l'autre ontologie. Comme dans FALCON, nous appellerons ces couples de concepts des *ancres* mais nous les utiliserons dès la construction des partitions.

La dissymétrie structurelle des 2 ontologies est exploitée pour ordonner leur partitionnement : si l'une des deux ontologies est mieux structurée que l'autre, elle sera plus facile à décomposer en blocs ayant une forte cohésion interne et sa décomposition pourra servir de guide à celle de l'autre ontologie. Dans ce qui suit, l'ontologie la plus structurée sera appelée la *cible*, O_T et la moins structurée, la *source*, O_S.

3.1 Méthode 1

La première méthode consiste à commencer par décomposer la cible O_T, puis à forcer le partitionnement de O_S à suivre celui réalisé pour O_T. Pour cela, la méthode identifie pour chacun des blocs B_{Ti} construits à partir de O_T, l'ensemble des ancres lui appartenant. Chacun de ces ensembles constituera le noyau ou *centre* CB_{Si} d'un futur bloc B_{Si} à générer à partir de la source O_S. L'alignement des paires de blocs ainsi constituées permet de retrouver dans la phase d'alignement finale, toutes les relations d'équivalence entre les ancres. La première méthode comprend donc quatre étapes en plus du calcul des ancres :

Partitionner la cible O_T en plusieurs blocs B_{Ti} Le partitionnement est effectué conformément à l'algorithme FALCON.

Identifier les centres CB_{Si} des futurs blocs de O_S Les centres de O_S sont déterminés en se basant sur deux critères : les couples d'ancres identifiés entre O_S et O_T, et les blocs B_{Ti} construits à partir de l'ontologie cible O_T.

Soit la fonction $Ancre(E, E')$, dont les arguments E et E' peuvent être chacun, soit une ontologie, soit un bloc, et qui retourne l'ensemble des concepts de E qui ont le même label qu'un concept de E'. Pour chaque bloc B_{Ti} construit à l'étape précédente, les centres des futurs blocs correspondants de O_S sont calculés comme suit :

$$CB_{Si} = Ancre(O_S, B_{Ti})$$

Partitionner la source O_S autour des centres CB_{Si} Après l'identification des centres des futurs blocs de O_S, nous appliquons l'algorithme FALCON avec la différence suivante. Au

lieu d'introduire en entrées l'ensemble des m concepts de l'ontologie comme m blocs réduits chacun à un unique concept, nous introduisons les n centres identifiés à l'étape précédente, comme autant de blocs distincts mais regroupant plusieurs concepts, puis les autres concepts de O_S qui n'ont pas d'équivalents dans O_T, chacun dans un bloc individuel. La cohésion des blocs représentant les centres de O_S est initialisée avec la valeur maximale.

Identifier les paires de blocs à aligner Chacun des blocs B_{Si} construits à partir d'un centre n'est aligné qu'avec le bloc B_{Ti} correspondant. L'algorithme peut mener à la constitution de blocs B_{Sj} indépendants des centres, i.e. ne contenant pas d'ancres et qui, dans l'état courant de notre implémentation, ne sont pas pris en compte dans le processus d'appariement. Le traitement de ces blocs sans ancres est une des perspectives de ce travail, encore à l'étude.

3.2 Méthode 2

L'idée de cette méthode est de partitionner les deux ontologies en même temps, c.à.d. de faire du co-clustering. Le problème est que nous ne pouvons pas traiter réellement ces ontologies en parallèle du fait de leur grande taille. Pour simuler le parallélisme, nous partitionnons l'ontologie cible en favorisant la fusion des blocs partageant des ancres avec la source, et nous partitionnons la source en favorisant la fusion des blocs partageant des ancres avec un même bloc généré pour la cible. Prendre en compte les relations d'équivalence identifiées entre les ontologies dès le partitionnement de O_T, devrait permettre par la suite de faciliter la recherche des paires de blocs les plus proches et d'améliorer les résultats de l'alignement. Nous pouvons ainsi dire que ce partitionnement, contrairement à celui de FALCON ou à celui implémenté dans la méthode 1, est orienté alignement pour les deux ontologies à partitionner. La deuxième méthode comprend trois étapes :

Déterminer les blocs de O_T Pour construire les blocs de la cible O_T, nous utilisons l'algorithme FALCON en modifiant la définition de la mesure de *goodness* pour prendre en compte les relations d'équivalence entre les deux ontologies. Nous lui ajoutons un coefficient qui représente la proportion d'ancres partagées qui sont présentes dans un bloc B_j de O_T. Plus un bloc contient d'ancres, plus ce coefficient augmente sa cohésion ou sa valeur de couplage à d'autres blocs. De ce fait, au cours de la génération des blocs, le choix du bloc qui a la valeur maximale de cohésion ou de couplage ne dépend pas seulement des relations des concepts à l'intérieur ou à l'extérieur des blocs de O_T, mais aussi des ancres partagées avec O_S.

Soient $\alpha \in [0,1]$, B_i et B_j 2 blocs de O_T, $|Ancre(B_j, O_S)|$ représentant le nombre d'ancres présentes dans B_j et $|Ancre(O_T, O_S)|$, le nombre d'ancres total, l'équation de *goodness* devient :

$$goodness\,(B_i, B_j) = \alpha \left(\frac{\sum_{c_i \in B_i, c_j \in B_j} link\,(c_i, c_j)}{sizeOf\,(B_i) \cdot sizeOf\,(B_j)} \right) + (1-\alpha)\,\frac{|Ancre(B_j, O_S)|}{|Ancre(O_T, O_S)|}$$

Déterminer les blocs de O_S Là aussi nous modifions la mesure de *goodness* pour qu'elle prenne en compte à la fois les valeurs des liens entre les concepts de O_S, les ancres partagées entre les deux ontologies et les blocs construits pour O_T. Soit le bloc B_i de O_S ayant la valeur de cohésion maximale, soit B_k le bloc de O_T qui partage le plus d'ancres avec B_i, le nouveau calcul de *goodness* favorisera la fusion de B_i avec le bloc B_j qui contient le plus d'ancre

partagées avec B_k, de façon à regrouper dans un même bloc de la source, les ancres partagées avec un même bloc de la cible.

Soient $\alpha \in [0,1]$, B_i et B_j, 2 blocs distincts de O_S, B_k le bloc de O_T qui partage le plus d'ancres avec B_i, l'équation de *goodness* devient :

$$goodness\left(B_i, B_j\right) = \alpha \left(\frac{\sum_{c_i \in B_i, c_j \in B_j} link\left(c_i, c_j\right)}{sizeOf\left(B_i\right) \cdot sizeOf\left(B_j\right)} \right) + \left(1 - \alpha\right) \frac{|Ancre(B_j, B_k)|}{|Ancre(O_T, O_S)|}$$

Identification des paires de blocs à aligner L'alignement se fait entre les blocs partageant le plus d'ancres, un bloc de O_S ne s'alignant qu'avec un seul bloc de O_T.

4 Expérimentations

Nous avons implémenté les deux méthodes présentées précédemment et des expérimentations ont été faites sur différentes ontologies afin de comparer les méthodes de partitionnement au travers de leur efficacité pour l'alignement. Les blocs constitués ont été alignés par paires à l'aide du logiciel d'alignement développé au sein de l'équipe, *TaxoMap*.

Les expérimentations ont tout d'abord été réalisées sur des ontologies dans le domaine géographique, fournies par le COGIT[3]. Ces ontologies sont de tailles limitées ce qui permet de les aligner directement sans avoir besoin de les partitionner et d'obtenir ainsi des appariements de référence. Elles sont de plus bien connues dans l'équipe ce qui nous a permis d'analyser la pertinence sémantique des blocs générés. D'autres expérimentations ont été faites ensuite sur deux ontologies de grandes tailles que notre outil ne parvient pas à aligner en l'état.

4.1 Expérimentations sur les ontologies géographiques

L'ontologie *Cible* BDTopo, est composée de 612 concepts reliés par des liens de subsomption au sein d'une hiérarchie qui compte 7 niveaux de profondeur. L'ontologie *Source* BDCarto comprend 506 concepts dans une hiérarchie de profondeur 4. Les résultats de l'alignement direct effectué sans partionnement préalable des ontologies sont présentés dans le tableau 1.

Ontologies	Taille de la Cible	Taille de la Source	isEq	isClose	isA	$\Sigma =$
Topo-Carto	612	505	197	13	95	305

Tab.1. Les relations identifiées par l'alignement de BDCarto vers BDTopo.

Pour effectuer les partitions, la taille maximale des blocs fusionnables a été fixée à 100 concepts, i.e. un bloc dépassant cette taille ne peut plus être fusionné mais deux blocs comprenant au plus 99 concepts chacun peuvent l'être. Les blocs générés contiennent donc au plus 198 concepts. Le tableau 2 donne le nombre de blocs générés par méthode pour chaque ontologie. Remarquons que le nombre d'ancres identifiées, 191, est plus faible que le nombre de concepts jugés équivalents dans l'alignement de référence, 197. En effet, la mesure de similarité utilisée pour calculer les ancres est plus stricte (mais plus rapide à calculer), que celle de notre outil.

[3]Le laboratoire COGIT (Conception Objet et Généralisation de l'Information Topographique), Institut Géographique National

		Ontologie Cible BDTopo			Ontologie Source BDCarto		
Méthodes	ancres	blocs générés	concepts isolés	plus grand bloc	blocs générés	concepts isolés	plus grand bloc
FALCON	191	5	0	151	25	22	105
Méthode 1	191	5	0	151	10	16	143
Méthode 2	191	6	0	123	10	16	153

Tab.2. Partitionnement de BDTopo et BDCarto par méthode

L'ontologie cible BDTopo est l'ontologie de référence du COGIT. Elle est bien construite, compacte et très structurée. La racine ne comporte que deux fils directs de profondeur 1, eux-mêmes pères directs d'un nombre limité de noeuds. Elle est ainsi facile à partitionner en blocs sémantiquement pertinents, que ce soit par la méthode FALCON qui s'appuie essentiellement sur les relations structurelles entre concepts, par la méthode 1 qui reprend exactement FALCON pour le partitionnement de la cible ou par la méthode 2. Les deux décompositions proposées, composées de 5 ou 6 blocs, sont toutes les deux pertinentes.

A l'inverse, l'ontologie source BDCarto est moins structurée, très dispersée. La racine est reliée à presque une trentaine de fils directs, et de très nombreux sous-arbres ne contiennent pas plus d'une dizaine d'éléments. Sa décomposition est plus délicate. L'algorithme FALCON génère un nombre important de petits blocs ne comprenant pas plus de 5 ou 6 concepts, 19 blocs ne contiennent pas d'ancres, et 22 blocs ne contiennent qu'un seul concept. En utilisant l'information sur les ancres partagées, nos méthodes permettent en revanche d'agréger à des blocs plus importants, plus de la moitié de ces petits blocs ainsi que de nombreux concepts isolés, tout en gardant la cohérence sémantique de ces derniers. Les partitions construites, moins dispersées, sont donc plus lisibles pour l'humain et plus efficaces pour la phase suivante d'alignement des blocs.

Pour effectuer l'alignement, le choix des paires de blocs à aligner diffère suivant les méthodes :

FALCON : parmi les 25 blocs générés, seuls 6 blocs sources contiennent des ancres. Ces 6 blocs sont alignés avec le ou les blocs cibles pour lesquels le ratio d'ancres partagées sur la somme des ancres présentes dans les deux blocs, est supérieur à un seuil donné ϵ, ici fixé à 0.1. Ce seuil étant atteint par 9 paires de blocs, 9 alignements sont effectués.

Méthode 1 : Les 5 blocs sources construits à partir des 5 blocs de la cible contenant des ancres conduisent à 5 alignements en tout.

Méthode 2 : les 7 paires choisies sont celles qui maximisent le nombre d'ancres partagées des 7 blocs source contenant des ancres et qui ne participent chacun qu'à un seul alignement.

Les résultats présentés dans le tableau 3 montrent que même en appariant moins de paires de blocs que dans la méthode FALCON, l'appariement des blocs générés par nos méthodes donnent de meilleurs résultats en nombre de mappings identifiés. Si l'on analyse les résultats[4] des deux mesures classiquement utilisées en alignement pour comparer l'efficacité des techniques, la *précision* (le nombre d'appariements corrects identifiés après partition par rapport

[4]Ces résultats ont été calculés automatiquement par l'API d'évaluation d'alignements disponible sur le Web, http ://oaei.ontologymatching.org/2008/align.html, en fournissant en référence le fichier généré par l'alignement direct sans partition.

au nombre total d'appariements trouvés après partition) et le *rappel* (le nombre d'appariements corrects identifiés après partition par rapport au nombre d'appariements de référence), on voit que nos méthodes ont un bien meilleur rappel. En effet, en prenant en considération les relations d'équivalence entre les labels dans le processus de partitionnement, nos méthodes permettent de regrouper les concepts qui ont des relations entre eux dans des blocs qui seront considérés par la suite comme des paires à aligner, alors que la méthode FALCON partitionne les ontologies indépendamment l'une de l'autre et ne considère l'alignement qu'a posteriori. La méthode 1 permet en particulier, par construction, de retrouver tous les appariements correspondants aux ancres et donc un rappel supérieur.

Méthodes	Nb paires Alignées	isEq	isClose	isA	Σ =	Précision	Rappel
FALCON	9	118	13	52	183	0.96	0.57
Méthode 1	5	192	10	55	257	0.97	0.81
Méthode 2	7	147	11	61	219	0.97	0.69

Tab.3. Les relations identifiées par l'alignement des blocs générés par les différentes méthodes

Le fait que les différentes méthodes aient une précision inférieure à 1 signifie qu'elles trouvent toutes les trois des appariements qui n'avaient pas été identifiés par l'alignement des deux ontologies non partitionnées. Bien que comptabilisés ici comme incorrects, ces appariements ne sont pas forcément faux. En effet, pour chaque concept de la source, notre outil ne produit qu'un unique appariement avec un seul concept de l'ontologie cible, celui qu'il juge le meilleur, même si plusieurs concepts de la cible pouvaient en être rapprochés. Si les deux concepts intervenant dans un appariement de référence ne sont plus comparés entre eux parce qu'ils sont répartis dans des blocs non alignés, un autre appariement, qui ne sera pas forcément inintéressant, peut être trouvé pour le concept source. L'étude de la qualité de ces nouveaux mappings ainsi qu'une analyse plus poussée des qualités relatives de nos deux méthodes seront effectuées dans des travaux complémentaires.

4.2 Expérimentations sur les ontologies de grande taille

Nous avons testé les différentes méthodes sur deux ontologies de grande taille, AGROVOC et NALT, qui sont composées respectivement de 28 439 et 42 326 concepts. Ces ontologies servent de test dans la campagne d'évaluation OAEI (*Ontology Alignment Evaluation Initiative*) qui fait concourir chaque année les outils d'alignement sur des couples d'ontologies de taille et de domaine variés. AGROVOC est une ontologie multilingue construite par la FAO (Food and Agriculture Organization). Elle couvre les domaines de l'agriculture, des forêts, de la pêche, de l'environnement et de l'alimentation. NALT est le thésaurus de la NAL (National Agricultural Library) sur les mêmes domaines.

L'ontologie NALT, la plus importante, est utilisée comme cible, l'ontologie AGROVOC comme source, et la taille maximale des blocs fusionnables est fixée à 2 000 concepts.

Les résultats présentés dans le tableau 4 ci-dessous, montre que dans cette expérimentation comme dans la précédente, le partitionnement conduit par nos méthodes permet de minimiser le nombre de blocs générés ainsi que les concepts isolés. Ne disposant pas des appariements de référence qui permettraient d'analyser la qualité des alignements produits sur ces partitions, nous ne présentons pas ici les résultats des alignements obtenus.

		Ontologie Cible NALT			Ontologie Source AGROVOC		
Méthodes	ancres	blocs générés	concepts isolés	plus grand bloc	blocs générés	concepts isolés	plus grand bloc
FALCON	14 787	47	4	3 356	318	492	2 830
Méthode 1	14 787	47	4	3 356	252	199	2 939
Méthode 2	14 787	47	4	3 118	95	199	3 534

Tab.4. Partitionnement de AGROVOC et NALT

5 Conclusion

Les outils d'alignement d'ontologies perdant leur efficacité sur des ontologies de grandes tailles, l'objectif de ce travail était d'étudier des techniques de partitionnement d'ontologies orientées pour la tâche d'alignement.

Les deux méthodes que nous proposons reprennent l'algorithme de partitionnement d'ontologies développé pour le système d'alignement FALCON, mais au lieu d'appliquer l'algorithme, comme FALCON, successivement sur chaque ontologie indépendamment l'une de l'autre, elles essayent de prendre en compte au plus tôt dans le processus de partitionnement, le contexte de la tâche d'alignement. Nos méthodes vont ainsi exploiter le fait de travailler sur deux ontologies à la fois et des données relatives à l'alignement, faciles à identifier au préalable même sur des ontologies de grande taille, comme l'existence de couples de concepts issus des deux ontologies et ayant un label identique ou l'éventuelle dissymétrie structurelle des deux ontologies à aligner.

La première méthode commence par décomposer l'ontologie la mieux structurée puis force la décomposition de la deuxième ontologie à suivre celle de la première. La deuxième méthode favorise la constitution de blocs comprenant des concepts reliés par des relations d'équivalence avec l'autre ontologie.

Nos méthodes, de complexité comparable à celle de FALCON, ont été testées sur différents couples d'ontologies. Les résultats présentés montrent qu'elles permettent de construire des partitions moins dispersées, en limitant le nombre de blocs générés et les concepts isolés. Pour l'expérimentation où nous disposions d'appariements de référence, nous avons aussi pu vérifier que nos partitions permettaient de perdre moins d'appariements.

Nous poursuivons actuellement les expérimentations pour analyser les qualités relatives de nos deux méthodes quand les deux ontologies sont fortement déséquilibrées (en terme de taille et de structure) ou quand le nombre de concepts de label identique est limité.

Références

Berners-Lee, T., J. Hendler, et O. Lassila (2001). The semantic web. In *The Scientific American 284(5)*, pp. 34–43.

Grau, B. C., B. Parsia, E. Sirin, et A. Kalyanpur (2005). Automatic partitioning of owl ontologie using e-connections. In *DL 2005, Proceedings of 18th International Workshop on Description Logics, Edinburgh, UK*.

Guha, S., R. Rastogi, et K. Shim (2000). ROCK : A robust clustering algorithm for categorical attributes. *Information Systems 25*, 345–366.

Hu, W., Y. Zhao, et Y. Qu (2006). Partition-based block matching of large class hierarchies. In *International Semantic Web Conference- ISWC*, pp. 72–83.

ModularOntology-ISWC, W. (2006). Proceedings of the workshop on modular ontologies. In *International Semantic Web Conference- ISWC*.

Noy, N. F. et M. A. Musen (2000). PROMPT : Algorithm and tool for automated ontology merging and alignment. In *AAAI/IAAI*, pp. 450–455.

Rahm, E. et P. A. Bernstein (2001). A survey of approaches to automatic schema matching. *VLDB Journal : Very Large Data Bases 10*(4), 334–350.

Reynaud, C. et B. Safar (2007). Techniques structurelles d'alignement pour portail web. In *RNTI, Revue des Nouvelles Technologies de l'Information*.

Shvaiko, P. et J. Euzenat (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV*, 146–171.

Stuckenschmidt, H. et M. Klein (2004). Structured-based partitioning of large concept hierarchies. In *International Semantic Web Conference- ISWC*, pp. 289–303.

Wu, Z. et M. Palmer (1994). Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pp. 133 –138.

Summary

Ontology alignment is an important task for information integration systems that can make different resources described by various and heterogeneous ontologies interoperate. However very large ontologies have been built in some domains like medicine or agronomy and the challenge is now to scale up alignment techniques that often perform complex tasks. In this paper, we propose two partitioning methods which have been designed to take the alignment objective into account in the partitioning process as soon as possible. These methods transform two ontologies to be aligned into two sets of blocks of a limited size. Furthermore, elements of the two ontologies that might be aligned are grouped in a minimal set of blocks that will be then really compared. Results of experiments performed by the two methods on various pairs of ontologies are promising.