Une mesure de similarité sémantique pour raisonner sur des documents

Rim Alhulou, Amedeo Napoli et Emmanuel Nauer
LORIA, Campus Scientifique,
B.P. 239, 54506 Vandoeuvre-les-Nancy,
Email: {alhulou,napoli,nauer}@loria.fr

Résumé *Dans cet article, nous proposons les premiers éléments pour une mesure de similarité s'appuyant sur le contenu des documents. Une telle mesure de similarité s'appuie sur une représentation arborescente des documents : les arbres sont considérés comme des annotations complexes sur lesquelles s'appuie le raisonnement sur les documents par l'intermédiaire de leur contenu. Un schéma d'algorithme de classification de documents est proposé sur la base de cette mesure de similarité.*

1 Introduction générale

Dans cet article, nous proposons une méthode fondée sur l'utilisation d'ontologies pour l'annotation sémantique de documents, permettant ainsi de mener des raisonnements et des recherches documentaires en s'appuyant sur le contenu des documents. L'annotation sémantique doit permettre de comparer les documents par rapport à leur contenu, de classifier les documents dans des catalogues, les organiser dans une hiérarchie de thèmes, les classer par rapport à des critères variés : "plus général", "plus spécifique", "similaire à", "fait partie de", vérifier la cohérence d'une requête avant de la satisfaire, et faire des requêtes par analogie et "similarité" sur le contenu des documents.

Dans ce cadre, un travail a été proposé dans [2], où les DTD (*Document Type Definition*) des documents XML sont représentées comme des concepts en logique de descriptions (LD). Par la suite, la subsomption est utilisée pour les classer et déterminer si deux DTD $\mathtt{D}_1$ et $\mathtt{D}_2$ sont équivalentes, ou si $\mathtt{D}_1$ est plus générale (spécifique) que $\mathtt{D}_2$. Ceci est utilisé pour la vérification de la validité des documents par rapport à une DTD donnée, pour la classification de documents et pour l'évaluation de requêtes. En particulier, la représentation de schéma de documents par des classes et de documents particuliers par des instances pose un certain nombre de problèmes difficiles non encore totalement résolus, comme cela est discuté dans [3] et ici au paragraphe 3.1.

Un autre travail sur l'utilisation des ontologies pour la catégorisation des méta-données a été proposé dans [6], qui s'appuie sur la similarité entre les méta-données présentes dans des documents du Web et représentées comme des instances des concepts de l'ontologie. Cette similarité est employée pour organiser ces méta-données dans des catégories (*clusters*). La similarité entre deux instances (deux méta-données) est mesurée par la similarité entre les concepts auxquels elles appartiennent (similarité taxonomique), celle entre les instances auxquelles elles sont liées (similarité relationnelle) et celle des attributs qui caractérisent leurs propriétés (similarité d'attributs). Les mêmes auteurs traitent de similarité sémantique pour comparer des ontologies dans [7].

L'idée de notre proposition s'inspire de ces deux travaux de recherche, et s'appuie sur les LD pour raisonner sur le contenu des documents (comme dans le travail présenté dans [2]), mais la comparaison se fait entre les documents annotés par rapport à une ontologie donnée. Le point commun avec le travail proposé dans [6] est que nous nous appuyons sur une ontologie du domaine pour mener à bien le raisonnement, mais le but est de comparer globalement des documents et pas seulement des données présentes dans ces documents. Cette proposition s'inspire aussi des recherches sur les chemins de similarité tels qu'ils sont introduits dans [5]. L'utilisation du *raisonnement par similarité* sur le contenu des documents peur permettre par exemple de :

- proposer à l'utilisateur un ensemble de documents similaires en réponse à sa requête, comme le font certains moteurs de recherche, la différence étant que la similarité ici est fondée sur le contenu des documents, et pas seulement sur des mots clés.
- proposer à l'utilisateur des requêtes similaires à sa requête, pour lesquelles des réponses sont déjà connues.

Le contenu d'un document peut être considéré comme un ensemble d'objets liés entre eux par des propriétés, où objets et propriétés sont dotés d'une sémantique. Plus précisément, un document `D` peut être vu comme un arbre enraciné (`N`,`B`), où `N` est un ensemble de nœuds et `B` est un ensemble de branches. Un nœud représente un objet qui peut être une instance d'une classe `C` définie dans une ontologie du domaine donnée. Un nœud est défini par un couple (`id`,`type`), où `id` est l'identifiant du nœud et `type` est son *type* : une classe définie dans l'ontologie. Les nœuds terminaux, ou feuilles, se voient associés une valeur dépendant du type `String` (seul type pris en compte actuellement). Une branche (n_1,b,n_2) représente un lien entre deux objets, où $b \in B$ est l'étiquette de la branche, $n_1 \in N$ est le nœud *origine* de la branche et $n_2 \in N$ est son nœud *extrémité*[1]

La *similarité sémantique* de documents peut être calculée en comparant les représentations arborescentes associées au contenu des documents. Ainsi, deux documents sont similaires sémantiquement si les représentations arborescentes de leur contenu possèdent un sous-arbre commun.

Par exemple, le contenu des documents `D1` et `D2` à la figure 1 est représenté par des arbres. Le document `D1` mentionne un atelier animé par une école nommée `ICN` et dont le thème est le `Commerce électronique`. `D2` mentionne

1. Toutefois, les branches ne sont pas considérées comme orientées.

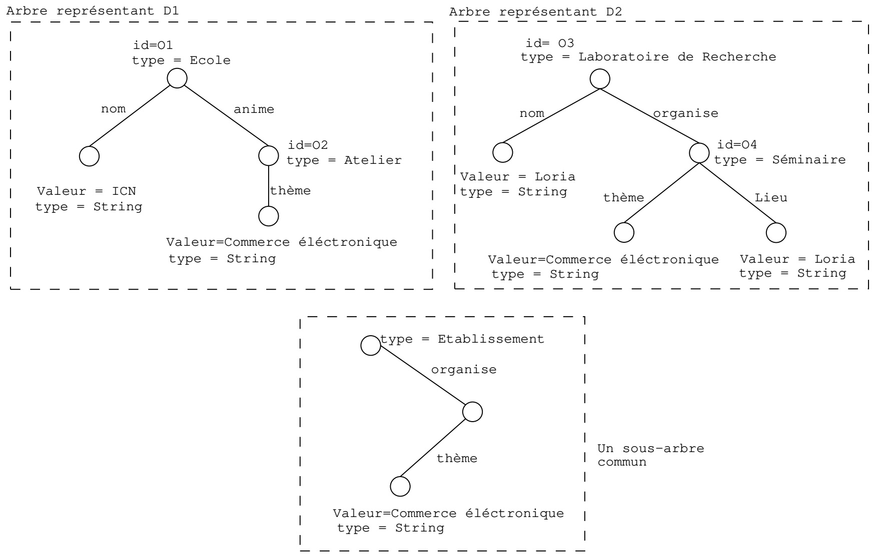

FIG. 1 – *Exemple de deux documents représentés par des arbres possédant un sous-arbre commun.*

un séminaire qui a lieu dans le laboratoire de recherche `Loria`, le laboratoire qui l'organise, et qui porte sur le même thème, le `Commerce électronique`. Ces deux documents peuvent être considérés comme similaires sémantiquement parce qu'ils décrivent sur un plan générique un `établissement` qui organise une `rencontre` qui a pour thème `Commerce électronique`. Cette similarité sémantique entre documents (ainsi qu'un degré de similarité) est définie rigoureusement dans la suite, après l'introduction du modèle de représentation dans lequel nous nous plaçons pour représenter et manipuler le contenu des documents.

2 Notion de similarité sémantique entre documents

Dans cette section, nous introduisons la similarité sémantique entre documents, puis la similarité sémantique entre nœuds et entre branches, et enfin la notion d'arbre de similarité.

Définition 1 *Une ontologie O_D du domaine des documents est définie par un couple (`C`,`P`), où `C` est un ensemble de classes et `P` est un ensemble de propriétés.*

Un exemple d'ontologie est donné à la figure 2, où les classes ne sont définies que par leur nom et sont organisées par une relation de subsomption. Deux propriétés sont définies dans cette ontologie, la propriété `animer` qui est subsumée par `organiser`.

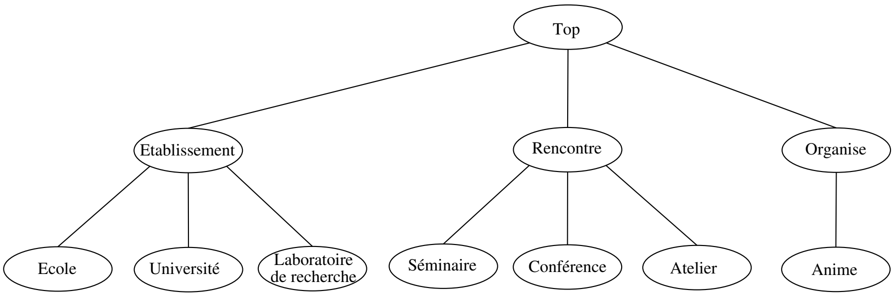

FIG. 2 – *L'ontologie des classes et des propriétés de l'exemple, où les classes ne sont définies que par leur nom.*

Une classe est définie par un nom et éventuellement un ensemble de propriétés. Une propriété est définie par un nom, un domaine et un co-domaine. Les ensembles des noms des classes et des noms de propriétés sont disjoints. Les classes sont organisées dans une hiérarchie par une relation de *subsomption*. Si $\mathtt{C}_1 \sqsubseteq \mathtt{C}_2$, alors tous les objets déclarés instances de $\mathtt{C}_1$ sont des instances de $\mathtt{C}_2$. Les propriétés peuvent aussi être organisées dans une hiérarchie par une relation de *subsomption*, si $\mathtt{P}_1 \sqsubseteq \mathtt{P}_2$ et s'il existe deux objets `a` et `b` tels que $\mathtt{P}_1\mathtt{(a,b)}$ alors $\mathtt{P}_2\mathtt{(a,b)}$ est vérifié. De plus, les propriétés peuvent être réflexives, symétriques, etc. La racine de la hiérarchie, `Top`, subsume toutes les classes et toutes les propriétés.

Définition 2 *Un d-arbre D=(N,B) est un arbre représentant le contenu de document, où N est un ensemble de nœuds et B un ensemble de branches. L'ensemble $N = N_I \cup N_F$ se compose des nœuds intermédiaires dans N_I et des feuilles dans N_F.*

Définition 3 *Des fonctions d'étiquetages sont associées à chaque d-arbre représentant un document. Une première fonction d'étiquetage $Etq_N : N \rightarrow ID_O$ est définie sur l'ensemble des nœuds N. Elle attribue à chaque nœud un identifiant. Tous les identifiants des nœuds appartiennent à un ensemble noté ID_O. Nous supposons que l'identifiant d'un nœud est unique. Les feuilles n'ont pas d'étiquette.*

Une seconde fonction d'étiquetage $Etq_B : B \rightarrow ID_P$ est définie sur l'ensemble B des branches. L'étiquette d'une branche est le nom de la propriété portée par cette branche.

Un nœud est défini par un couple (`id`, `type`), où `id` est l'identifiant du nœud et `type` le type du nœud. Les identifiants des nœuds sont uniques, et sont représentés par des séquences alphanumériques. Le type d'un nœud, donné par la fonction `type` : $N \rightarrow C$, correspond à la classe représentant le nœud dans l'ontologie. Chaque feuille est définie par une valeur de type `String`. Les branches, quant à elles, sont étiquetées par des propriétés définies dans l'ontologie du domaine des documents.

Définition 4 *Soient deux documents D_1 et D_2, et leurs d-arbres associés, $A_D(D_1)$ $=(N_1,B_1)$ et $A_D(D_2)=(N_2,B_2)$. D_1 et D_2 sont similaires sémantiquement s'ils possèdent un sous-arbre commun, autrement dit s'il existe un d-arbre $A_D=(N,B)$ tel que :*

$$A_D(D_1) = (N_1,B_1) \sqsubseteq_d A_D = (N,B) \text{ et}$$
$$A_D(D_2) = (N_2,B_2) \sqsubseteq_d A_D = (N,B),$$

où $\sqsubseteq_d$ représente la subsomption de d-arbres (défini ci-après).

En réalité, la relation $\sqsubseteq_d$ est une relation de comparaison de structures en fonction d'une ontologie de types [2].

La similarité sémantique de documents dépend de la similarité entre nœuds et entre branches des arbres qui représentent le contenu, comme cela va être précisé dans ce qui suit. Par la suite, s'il n'y a pas d'ambiguïté, nous utilisons de terme "similarité" pour "similarité sémantique".

2.1 Relation de similarité entre nœuds

Définition 5 *Deux nœuds $n_1 \in N_1$ et $n_2 \in N_2$ sont similaires sémantiquement si les conditions suivantes sont vérifiées :*

1. *Si n_1 et n_2 sont deux feuilles alors elles ont la même valeur atomique.*
2. *Si n_1 et n_2 sont des nœuds intermédiaires, alors il existe une classe $C \neq Top$, telle que : $C_1 \sqsubseteq C$ et $C_2 \sqsubseteq C$ dans l'ontologie, où $C_1 = type(n_1)$ et $C_2 = type(n_2)$ respectivement, et où C peut éventuellement être égale à C_1 ou C_2.*

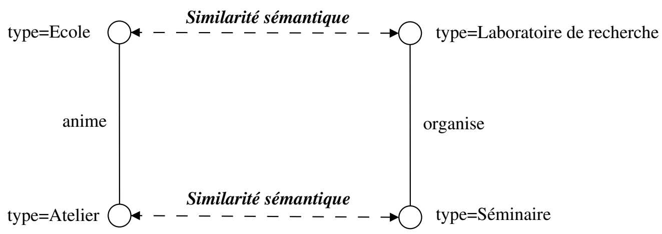

FIG. 3 – *Exemples de relation de similarité entre nœuds.*

Un exemple de relation de similarité définie entre deux nœuds sont représentés à la figure 3. Les nœuds `(id=01,type=Ecole)` et `(id=03,type=Laboratoire de recherche)` sont similaires parce que le type du premier nœud est `Ecole` et celui du second est `Laboratoire de recherche` ; deux classes qui possèdent la classe `Établissement` comme classe subsumante commune (voir l'ontologie présentée dans la figure 2).

2. La similarité sémantique de documents est comparable à la similarité (ou subsomption) de structure moléculaire introduite dans [9, 10]

2.2 Relation de similarité entre branches

Définition 6 *Deux branches $b_1 = (n_1, a_1, n'_1)$ et $b_2 = (n_2, a_2, n'_2)$ sont similaires sémantiquement si l'une des conditions suivantes est vérifiée :*

1. *$a_1 = a_2$, n_1 sémantiquement similaire à n_2 et n_1' sémantiquement similaire à n_2'.*
2. *$a_1 \neq a_2$, il existe une branche $b = (n, a, n')$ avec : $a \neq$ Top, $a_1 \sqsubseteq a$ et $a_2 \sqsubseteq a$, $n_1 \sqsubseteq n$, $n_2 \sqsubseteq n$, $n_1' \sqsubseteq n'$, et $n_2' \sqsubseteq n'$.*

Par exemple, une relation de similarité existe entre les deux branches dans la figure 3 parce que la propriété `anime` est subsumée par la propriété `organise` dans l'ontologie des propriétés.

2.3 Relation de similarité entre arbres

Définition 7 *: Deux documents D_1 et D_2, représentés par des d-arbres $A_D(D_1)$ et $A_D(D_2)$, sont dits similaires sémantiquement, s'il existe un sous-arbre d_1 de $A_D(D_1)$, et un sous-arbre d_2 de $A_D(D_2)$, tel que d_1 est isomorphe à d_2.*

L'*isomorphisme* est un isomorphisme d'arbres enracinés, si $A_D(D_1) = (N_1, B_1)$ et $A_D(D_2) = (N_2, B_2)$ sont deux arbres dont les racines sont r_1 et r_2, D_1 et D_2 sont isomorphiques s'il existe une bijection $f : N_1 \to N_2$ telle que :

1. $f(r_1) = r_2$ (correspondance entre les racines).
2. si $b_1 = (n_1, a_1, n_1')$ est une branche de D_1, alors il existe une branche $b_2 = (n_2, a_2, n_2')$ de D_2 telle que : $n_2 = f(n_1)$, $n_2' = f(n_1')$, avec n_1 sémantiquement similaire à n_2, n_1' sémantiquement similaire à n_2', et a_1 sémantiquement similaire à a_2.

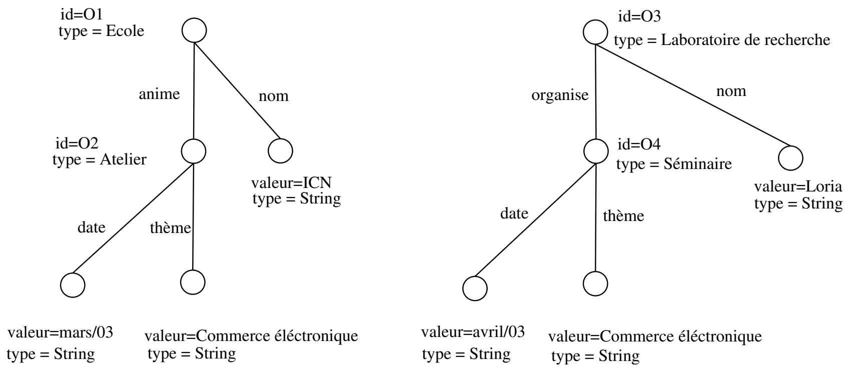

FIG. 4 – *Exemple de similarité entre deux arbres.*

La définition de la similarité sémantique entre documents dépend donc de la similarité entre les nœuds et de la similarité entre les branches. Ainsi, les deux arbres $\mathtt{A}_D\mathtt{(D_1)}$ et $\mathtt{A}_D\mathtt{(D_2)}$ de la figure 4 sont similaires puisque :

1. La racine de $\mathtt{A}_D\mathtt{(D_1)}$ qui est (`id=01`,`type = Ecole`) est similaire à la racine de $\mathtt{A}_D\mathtt{(D_2)}$ qui est (`id=03`,`type = Laboratoire de recherche`).
2. Les branches issues des deux racines, étiquetées par `anime` et `organise`, sont similaires.
3. Le nœud (`id=02`,`type = Atelier`) et le nœud (`id=04`,`type = Séminaire`) sont similaires car les classes `Atelier` et `Séminaire` sont toutes deux subsumées par la classe `Rencontre` dans l'ontologie.
4. Les étiquettes `date` et `thème` dans $\mathtt{A}_D\mathtt{(D_1)}$ sont identiques aux étiquettes correspondantes dans $\mathtt{A}_D\mathtt{(D_2)}$.
5. Les deux feuilles définies par (`valeur=Commerce électronique`,`type=String`) sont similaires (car identiques).

2.4 Arbre de similarité

Définition 8 *Deux documents D_1 et D_2 sont similaires sémantiquement s'il existe un sous-arbre $d_1=(N_1,B_1)$ de $A_D(D_1)$ et un sous-arbre $d_2=(N_2,B_2)$ de $A_D(D_2)$ tels que d_1 et d_2 peuvent être "unifiés" en un arbre de similarité $d=(N,B)$, dont la construction s'appuie sur deux substitutions σ_1 et σ_2 telles que $d=\sigma_1(d_1)=\sigma_2(d_2)$ telles que :*

1. *Pour deux nœuds similaires $n_1 \in N_1$ et $n_2 \in N_2$, un nœud n résultant de la généralisation de n_1 et de n_2 est ajouté à d tel que $\sigma_1(n_1)=\sigma_2(n_2)=n$. Une fonction GEN_NOEUD(n_1,n_2), présentée plus loin, est utilisée pour générer le nœud n.*
2. *Pour un nœud $n \in N_1$ qui n'a pas de nœud similaire dans N_2 (ou vice versa), la substitution n'est pas définie (et donc l'arbre de similarité n'existe pas).*
3. *Pour un couple de branches similaires $b_1=(n_1,a_1,n'_1) \in B_1$ et $b_2=(n_2,a_2,n'_2) \in B_2$, une branche $b=(n,a,n')$ résultant de la généralisation de b_1 et de b_2 est ajoutée à d avec $\sigma_1(a_1)=\sigma_2(a_2)=a$. Une fonction GEN_BRAN($a_1$,$a_2$), présentée plus loin, est utilisée pour générer la branche a.*
4. *Pour une branche b dans d_1 qui n'a pas de branche similaire dans d_2 (ou vice versa), la substitution n'est pas définie (et donc l'arbre de similarité n'existe pas).*

La fonction `GEN_NOEUD`$(\mathtt{n_1},\mathtt{n_2})$ prend comme arguments deux nœuds similaires $\mathtt{n_1} \in \mathtt{N_1}$ et $\mathtt{n_2} \in \mathtt{N_2}$, et elle génère un nœud `n` de la façon suivante. Soit $\mathtt{C_1}=\mathtt{type}(\mathtt{n_1})$ le type de $\mathtt{n_1}$ et $\mathtt{C_2}=\mathtt{type}(\mathtt{n_2})$ le type de $\mathtt{n_2}$, et `C` la classe telle que $\mathtt{C_1} \sqsubseteq \mathtt{C}$ et $\mathtt{C_2} \sqsubseteq \mathtt{C}$ ($\mathtt{n_1}$ et $\mathtt{n_2}$ sont similaires), alors `n` est défini comme une instance de la classe `C` avec l'étiquette $\mathtt{n}=\mathtt{n_1}.\mathtt{n_2}$, la concaténation des noms $\mathtt{n_1}$ et $\mathtt{n_2}$. Dans le cas où `C` est `String` et que les valeurs de $\mathtt{n_1}$ et $\mathtt{n_2}$ sont identiques, cette valeur est attachée au nœud `n`. De la même façon, la fonction `GEN_BRAN`$((\mathtt{b_1},\mathtt{a_1},\mathtt{b'_1}),(\mathtt{b_2},\mathtt{a_2},\mathtt{b'_2}))$ génère une branche $(\mathtt{b},\mathtt{a},\mathtt{b'})$ de la façon suivante : si $\mathtt{a_1} \sqsubseteq \mathtt{a}$ et $\mathtt{a_2} \sqsubseteq \mathtt{a}$ alors la branche engendrée porte la propriété `a` comme étiquette (`a` peut éventuellement être égale à $\mathtt{a_1}$ ou $\mathtt{a_2}$).

Un exemple d'arbre de similarité entre deux documents est donné à la figure 5. Les deux documents sont similaires parce que leurs *d-arbres* $\mathtt{d_1}$ et $\mathtt{d_2}$ sont

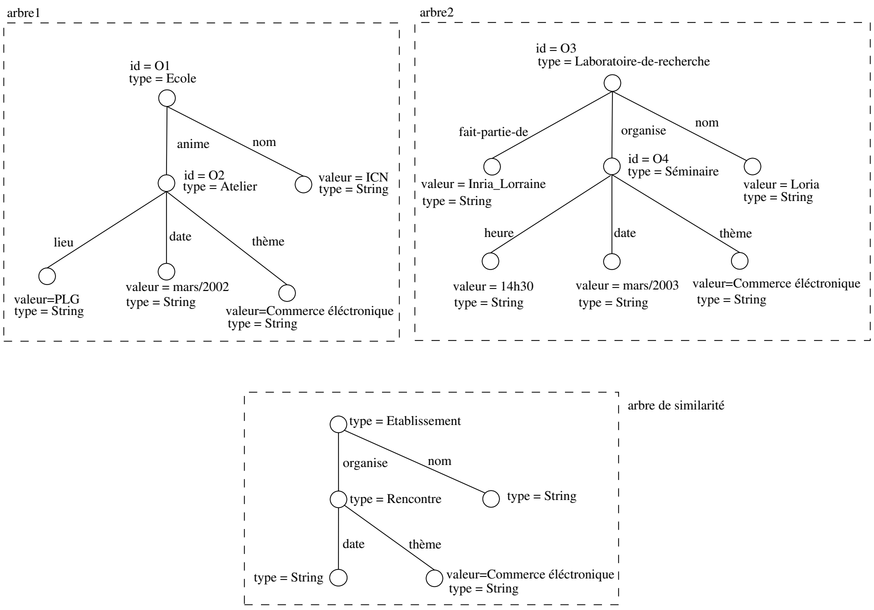

FIG. 5 – *Exemple d'arbre de similarité engendré à partir de deux arbres similaires sémantiquement.*

similaires : ils possèdent un sous-arbre commun généré à partir des sous-arbres similaires.

La racine de l'arbre de similarité `d` est générée à partir des racines de $\mathtt{d}_1$ et $\mathtt{d}_2$: le type de la racine de `d` est la classe `Établissement` qui est la classe subsumante commune des classes `Ecole` et `Laboratoire de recherche` (les types de $\mathtt{d}_1$ et $\mathtt{d}_2$). La branche sortant de la racine de l'arbre `d` étiquetée par `organise` est générée à partir des branches étiquetées par `organise` et `anime`. Un seul nœud intermédiaire est présent dans l'arbre `d` ; il possède la classe `Rencontre` comme type, qui subsume les classes `Atelier` et `Séminaire` dans l'ontologie. L'arbre `d` possède trois feuilles générées à partir des feuilles similaires dans $\mathtt{d}_1$ et $\mathtt{d}_2$, avec le même type `String`. L'une d'entre-elles possède de plus une valeur (`Commerce électronique`).

3 Degré de similarité et algorithme de classification

Soient deux documents $\mathtt{D}_1$ et $\mathtt{D}_2$ similaires sémantiquement, avec l'arbre de similarité $\mathtt{A}_S(\mathtt{D}_1,\mathtt{D}_2) = (\mathtt{N},\mathtt{B})$ engendré à partir des d-arbres $\mathtt{A}_D(\mathtt{D}_1) = (\mathtt{N}_1,\mathtt{B}_1)$ et $\mathtt{A}_D(\mathtt{D}_2) = (\mathtt{N}_2,\mathtt{B}_2)$. La *taille* d'un arbre `X`, notée $|\mathtt{X}|$, correspond au nombre de nœuds de l'arbre.

Le *degré de similarité sémantique* entre deux documents est défini en s'appuyant sur les relations de similarité existant entre leurs d-arbres. Les quatre cas ci-après suivent la construction d'un arbre de similarité présentée dans 5, et permettent de définir des *coûts*, qui traduisent l'*écart* existant entre l'annotation associée à un document et l'arbre de similarité :

1. Le coût d'un nœud $\mathtt{n}_1$ dans $\mathtt{A}_D\mathtt{(D}_1\mathtt{)}$ *par rapport* à un nœud `n` dans $\mathtt{A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{)}$ est donné par la longueur du chemin dans l'ontologie qui sépare $\mathtt{type(n}_1\mathtt{)}$ et `type(n)`. Dans le cas où $\mathtt{n}_1$ et `n` sont de type `String`, le coût est de 0 si les valeurs de $\mathtt{n}_1$ et `n` sont égales et de 1 sinon.
   Ainsi, dans la construction de l'arbre de similarité de la figure 5, le coût de `Séminaire` ou de `Atelier` est de 1, comme celui de `Ecole` ou `Laboratoire de recherche`. Le coût de (`type = String, valeur = Commerce électronique`) par rapport à (`type = String, valeur = Commerce électronique`) est de 0, alors que le coût de (`type = String, valeur = mars/03`) par rapport à (`type = String`) est de 1.
2. Le *coût* d'un nœud `n` de $\mathtt{N}_1$ qui ne possède pas de nœud similaire dans $\mathtt{N}_2$ est celui de la taille du sous-arbre contenant `n` qu'il faut supprimer pour avoir un appariement *exact* avec l'arbre se similarité. Ainsi le coût du nœud (`valeur = PLG, type = String`) de l'arbre de gauche de la figure 4 est de 1.
3. Le *coût* d'une branche $\mathtt{b}_1$ de $\mathtt{A}_D\mathtt{(D}_1\mathtt{)}$par rapport à une branche $\mathtt{b}_2$ de $\mathtt{A}_D\mathtt{(D}_2\mathtt{)}$est donné par la longueur du chemin dans l'ontologie qui sépare l'étiquette de $\mathtt{b}_1$ du subsumant commun avec l'étiquette de $\mathtt{b}_2$.
4. Le coût d'une étiquette d'une branche $\mathtt{b}=(\mathtt{n},\mathtt{a},\mathtt{n}')$ de $\mathtt{B}_1$ qui n'a pas de branche similaire dans $\mathtt{B}_2$ est pris en compte par l'intermédiaire des nœuds `n`et $\mathtt{n}'$qui composent la branche.

Le coût de la construction d'un arbre de similarité pour un document $\mathtt{D}_1$ est égal à la somme des coûts des nœuds et des branches ayant été substitués lors de la construction, augmenté des coûts des parties supprimées. Ainsi, le coût de la construction de l'arbre de similarité $\mathtt{A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{)}$(figure 4), noté $\mathtt{coût(D}_1\mathtt{|A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{))}$, est pour $\mathtt{D}_1$: 4 (pour les nœuds ayant un nœud similaire dans $\mathtt{A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{)}$) + 1 (pour les nœuds n'ayant pas de nœud similaire dans $\mathtt{A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{)}$) + 1 (pour les branches ayant une branche similaire dans $\mathtt{A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{)}$), soit 6. De la même façon, il vient $\mathtt{coût(D}_2\mathtt{|A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{))} = 4 + 2 + 0 = 6$.

Il est possible maintenant de définir le *degré de similarité* de deux documents de la façon suivante :

$$ds(\mathtt{D}_1,\mathtt{D}_2) = \min\Big(\frac{|\mathtt{A}_S(\mathtt{D}_1,\mathtt{D}_2)|}{|\mathtt{A}_D(\mathtt{D}_1)|} * \frac{1}{co\hat{u}t(\mathtt{D}_1|\mathtt{A}_S(\mathtt{D}_1,\mathtt{D}_2))+1}, \frac{|\mathtt{A}_S(\mathtt{D}_1,\mathtt{D}_2)|}{|\mathtt{A}_D(\mathtt{D}_2)|} * \frac{1}{co\hat{u}t(\mathtt{D}_2|\mathtt{A}_S(\mathtt{D}_1,\mathtt{D}_2))+1}\Big)$$

Ainsi, pour l'arbre $\mathtt{A}_S\mathtt{(D}_1\mathtt{,D}_2\mathtt{)}$de la figure 4, nous obtenons :

$$ds(\mathtt{D}_1,\mathtt{D}_2) = \min\Big(\frac{5}{6} * \frac{1}{6+1}, \frac{5}{7} * \frac{1}{6+1}\Big) = \min\Big(\frac{5}{42}, \frac{5}{49}\Big) = \frac{5}{49}$$

Le degré de similarité est compris entre 0 et 1, et il est d'autant meilleur qu'il est proche de 1 : le degré de similarité entre deux documents identiques est bien sûr 1.

3.1 Un schéma d'algorithme de classification

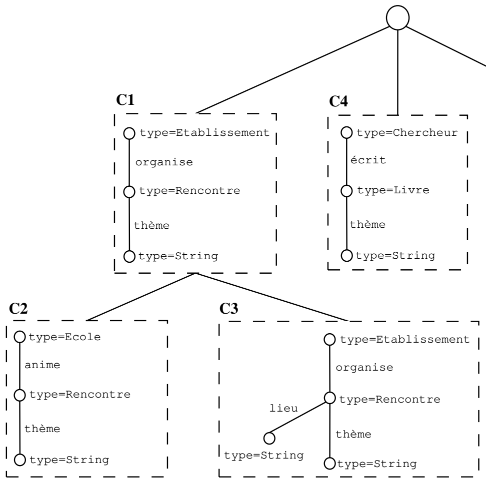

FIG. 6 – *Une hiérarchie de type de documents.*

Soient $\mathcal{D}$ un ensemble de documents, O_D une ontologie associée à $\mathcal{D}$ et A_D une hiérarchie de type de documents associée encore à $\mathcal{D}$ (comme celle de la figure 6). Les classes de A_D sont des classes de documents relatives à un ensemble de documents et à l'ontologie O_D. Grâce à la notion de degré de similarité, nous pouvons maintenant proposer un algorithme de classification, qui permet de retrouver la « meilleure classe d'appartenance » au sens de degré de similarité pour un document spécifique donné, sachant que dans cet article, nous ne considérons que la classification d'instances.

Cet algorithme va servir en particulier à classifier des documents par leur contenu, à retrouver des documents similaires sur la base de leur contenu, et à mettre en œuvre un raisonnement à partir de cas sur la base du contenu des documents.

Dans le cadre introduit ci-dessus, à savoir $\mathcal{D}$ un ensemble de documents, une ontologie $\mathtt{O}$ relative à $\mathcal{D}$, et une hiérarchie $\mathcal{A}_\mathcal{D}$ de types génériques de documents relatives encore à $\mathcal{D}$, le principe de l'algorithme de classification est le suivant. Pour un document $\mathtt{D}_1$de d-arbre $\mathtt{A}_D(\mathtt{D}_1)=(\mathtt{N}_1,\mathtt{B}_1)$ sont recherchées dans $\mathcal{A}_\mathcal{D}$ les classes subsumant $\mathtt{D}_1$, en suivant un parcours de $\mathcal{A}_\mathcal{D}$ classique en profondeur (voir par exemple [9]) :

- Une classe $\mathtt{C}$ dans $\mathcal{A}_\mathcal{D}$ *subsume* $\mathtt{A}_D(\mathtt{D}_1)$s'il existe un arbre de similarité entre $\mathtt{C}$ et $\mathtt{A}_D(\mathtt{D}_1)$.

– Zéro ou plusieurs classes subsumantes peuvent exister dans $\mathcal{A}_{\mathcal{D}}$. S'il n'existe aucune classe subsumante dans $\mathcal{A}_{\mathcal{D}}$, alors le type du document $\mathtt{D}_1$ n'est pas représenté dans $\mathcal{A}_{\mathcal{D}}$. Une nouvelle classe représentant $\mathtt{D}_1$ pourrait être créée comme dans un algorithme de classification incrémentale (voir par exemple [4]), mais cette éventualité n'est pas discutée plus avant ici. En réalité, lorsqu'aucune classe de $\mathcal{A}_{\mathcal{D}}$ ne subsume $\mathtt{A}_D(\mathtt{D}_1)$, cela signifie que le document $\mathtt{D}_1$est hors du thème de $\mathcal{D}$.
– Si une ou plusieurs classes subsument $\mathtt{A}_D(\mathtt{D}_1)$, alors des documents similaires à $\mathtt{D}_1$ ont été retrouvés : ce sont les documents qui sont des instances de toutes les classes subsumant $\mathtt{A}_D(\mathtt{D}_1)$. Les classes subsumant $\mathtt{A}_D(\mathtt{D}_1)$ sont alors classées selon leur degré de similarité, en classant en premier les classes de documents dont le contenu se rapproche le plus de $\mathtt{D}_1$.

Par exemple, si on cherche à classifier le d-arbre associée au document $\mathtt{D}_1$ dans la figure 1 dans la hiérarchie de types de documents de la figure 6, la classe `C2` qui subsume $\mathtt{D}_1$ est retrouvée : les instances de `C2` (mais aussi celles de `C1` qui est la superclasse de `C1`) sont des documents similaires à $\mathtt{D}_1$.

4 Implantation et tests

Cette approche a été implantée de façon simplifiée sur un type particulier d'annotations, établies à l'origine dans le cadre de l'ARC INRIA ESCRIRE[3] [1] [8], sur des résumés de documents de biologie issus de Medline[4]. L'annotation associée à un document décrit les interactions entre les gènes mentionnés dans les documents : une interaction met en relation un gène source et un gène cible, et se caractérise éventuellement par des informations complémentaires, comme l'effet, la localisation, etc. Une ontologie du domaine se compose de classes de gènes organisées hiérarchiquement par une relation de spécialisation, qui renferment des connaissances générales sur les gènes.

L'implantation expérimentale de la mesure de similarité et de l'algorithme de classification nous a permis de vérifier la faisabilité de l'approche introduite, ainsi que le bien-fondé des éléments introduit, mais il reste encore un volume important de travail de recherche à fournir, du point de vue théorique et pratique.

5 Conclusion

L'utilisation d'ontologies pour l'annotation du contenu de documents et la similarité sémantique entre documents peuvent être utiles dans de nombreux domaines, par exemple :

– La catégorisation de documents et la construction de catalogues, où les documents similaires sont regroupés dans la même catégorie : les documents qui décrivent des films *documentaires* dont le thème est *la guerre*

3. http://escrire.inrialpes.fr/
4. http://www.nlm.nih.gov/databases/databases_medline.html

et qui ont eu un *prix* peuvent être classés dans une même catégorie si besoin.
- La création de forums de discussion en comparant les documents où sont stockés les profils de visiteurs d'un certain site Web. Un abonnement dans un forum de discussion peut être proposé aux personnes ayant des profils similaires.
- La gestion d'un service en ligne pour la maintenance d'un outil acheté ou téléchargé sur un site Web. Une solution à un problème soumis par un client peut être trouvé en le comparant à des documents représentant d'autres problèmes similaires résolus auparavant.
- Le commerce électronique où les documents décrivant des produits ou des services similaires sont proposés pour répondre à une requête d'un client.

Établir un degré de similarité entre documents est donc une problématique importante dans des domaines variés. Il reste de nombreux points à étudier et à mettre au point, et l'un d'entre eux nous semble des plus importants, qui est la similarité de documents par rapport à un point de vue (qui peut être une conjonction d'attributs par exemple), et qui permet de généraliser le processus de classification des documents vu ci-dessus.

Références

[1] R. Al Hulou. *Les logiques de descriptions dans le traitement intelligent des données documentaires semi-structurées.* Thèse en Informatique, Université Henri Poincaré Nancy 1, Novembre 2003.

[2] D. Calvanese, Giuseppe De Giacomo, and M. Lenzerini. Representing and Reasoning on XML Documents: A Description Logic Approach. *Journal of Logic Computational*, 9(3):295–318, 1999.

[3] J. Euzenat. XML est-il le langage de représentation de connaissance de l'an 2000? In C. Dony and H. A. Sahraoui, editors, *Actes des 6émes journées Langages et Modèles à Objets (LMO'00)*, pages 75–90, Paris, janvier 2000. Hermes.

[4] P. Langley. *Elements of Machine Learning.* Morgan Kaufmann Publishers, San Francisco, California, 1996.

[5] J. Lieber and A. Napoli. Raisonnement à partir de cas et résolution de problèmes dans une représentation par objets. *Revue d'intelligence artificielle*, 13:9–35, 1999.

[6] A. Maedche, M. Ehrig, S. Handschuh, L. Stojanovic, and R. Volz. Ontology-focused crawling of documents and relational metadata. In *Proceedings of the 11th International WWW Conference*, 2002.

[7] A. Maedche, S. Staab, N. Stojanovic, R. Studer, and Y. Sure. SEmantic PortAL - The SEAL approach. In D. Fensel, J.A. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web*, pages 317–359. MIT Press, 2003.

[8] C. Medina-Ramirez. *Contribution tà la recherche d'informations sémantiques : Capitalisation de connaissances dans une mémoire d'interactions géniques.* PhD thesis, INRIA Sophia Antipolis, 2003.

[9] A. Napoli and C. Laurenço. Représentations à objets et classification – Conception d'un système d'aide à la planification de synthèses organiques. *Revue d'intelligence artificielle*, 7(2):175–221, 1993.

[10] P. Vismara, P. Jambaud, C. Laurenço, and J. Quinqueton. RESYN : objets, classification et raisonnement distribué en chimie organique. In R. Ducournau, J. Euzenat, G. Masini, and A. Napoli, editors, *Langages et modèles à objets — Ètat des recherches et perspectives*, Collection Didactique D-019, pages 397–419. INRIA, Le Chesnay, 1998.