Distance sémantique entre concepts définis en ALE

Mohamed Zied Maala, Alexandre Delteil, Amedeo Napoli

Distance sémantique entre concepts définis en $\mathcal{ALE}$

Mohamed Zied MAALA* — **Alexandre DELTEIL*** — **Amedeo NAPOLI****

* *FT R&D, 38 rue du General Leclerc, Issy-les-Moulineaux*
{zied.maala, alexandre.delteil}@orange-ftgroup.com

** *LORIA-UMR 7503*
Équipe Orpailleur- Bâtiment
B - B.P. 239
54506 Vandoeuvre-les-Nancy
napoli@loria.fr

RÉSUMÉ. *Cet article présente une approche permettant d'évaluer la similarité entre deux concepts décrits avec la logique de descriptions $\mathcal{ALE}$. Une telle approche peut être utilisée dans de nombreuses situations, et spécialement pour le classement de réponses à une requête. Dans plusieurs applications pratiques, en particulier pour le Web sémantique, des concepts peuvent être organisés dans une hiérarchie. Une originalité de notre travail est de compléter une hiérarchie de concepts donnée éventuellement en entrée en un treillis de concepts complet. Par la suite, un chemin entre concepts dans le treillis est utilisé pour évaluer la similarité entre deux concepts.*

ABSTRACT. *This paper presents an approach for evaluating likeness between two concepts represented within the $\mathcal{ALE}$ description logics. This can be used in a number of situations, and mainly for retrieving documents according to a given point of view (e.g. the content of the document). In many practical applications, especially for Semantic web applications, concepts are organized within a hierarchy. One originality of the present work on likeness is first to complete the input concept hierarchy into a complete concept lattice. Then, a path between concepts in the lattice is established and is used for setting the likeness between two concepts. Contrasting other existing similarity measures, the present distance takes advantage of the complete lattice organization of concepts, providing a reified path between concepts. This kind of path is based on a set of intermediate concepts that are members of the complete lattice, and that, in case, can give a qualitative rather than numerical explanation on the similarity between two concepts.*

MOTS-CLÉS : *Logique de descriptions, mesure de similarité, distance sémantique, classement de concepts, treillis,...*

KEYWORDS: *Description logics, similarity measure, semantic distance, concepts ranking, lattice,...*

1re soumission à *LMO*, le 27 Octobre 2006.

1. Introduction

Le classement de réponses est un élément important dans la qualité d'un moteur de recherche sur le Web. Les utilisateurs ne veulent pas seulement obtenir des ressources qui correspondent à leurs requêtes, mais aussi que les documents les plus pertinents soient classés dans les premières positions. Ainsi, la position de leader de Google est principalement due à l'utilisation de sa formule de classement des pages Web ("PageRank") dans le moteur de recherche. Cette formule utilise les liens hypertextes pour déterminer l'importance des pages Web et pour les classer dans un ordre plus ou moins pertinent par rapport à la requête utilisateur. Les méthodes de classement dans les moteurs de recherche actuels exploitent une combinaison de critères tels que : la présence de mots clés, les liens hypertextes, la position des mots clés dans les titres, la fréquence des mots clés dans un document et dans l'ensemble de tous les documents... Cependant, ces méthodes avant tout syntaxiques ou structurelles ne sont pas réellement suffisantes pour des applications du Web Sémantique, où l'accès par le contenu est nécessaire. Pour cela, des ontologies (qui regroupent en quelque sorte les connaissances d'un domaine) pour définir la sémantique des concepts et leurs relations sont utilisées, et les ressources sont annotées avec des descriptions de concepts. Une méthode de recherche qui s'appuie sur le contenu des documents qui sont considérés fournira un classement des ressources plus pertinent par rapport à la requête d'un utilisateur. Une telle méthode de classement, fonction des connaissances d'un domaine, permet à un moteur de recherche de classer les résultats d'une requête pour tout type de ressources, par exemple les documents textuels, les images et les vidéos. Elle peut être exploitée dans tout type d'application offrant une fonction de recherche comme les sites d'enchères, d'offres d'emploi, de vente en ligne, etc.

Les processus peuvent s'appuyer sur une mesure de similarité (dont la valeur augmente avec la ressemblance entre les concepts) ou bien sur une distance (dont la valeur décroît avec la ressemblance entre les concepts) pour classer les ressources par rapport à une requête tenant compte d'un ensemble de contraintes. Plusieurs mesures de similarité ou de distance ont été proposées pour évaluer les ressemblances entre concepts organisés dans une hiérarchie. La plupart permettent d'évaluer la similarité entre des concepts atomiques organisés dans une hiérarchie arborescente. Par exemple, sur la figure 1, la similarité entre deux aspects du concept "engineer" peut être évaluée en fonction des concepts les liant dans une telle hiérarchie. Une approche qui exploite aussi les connaissances d'un domaine pourrait utiliser un chemin comprenant les concepts de la hiérarchie actuelle, mais aussi des concepts supplémentaires générés si c'est nécessaire. Les éléments de connaissances liés aux concepts peuvent ainsi être utilisés pour expliquer la similarité entre deux concepts (par exemple "aeronautical engineer" et "aerospace engineer" sont deux concepts proches).

Cet article introduit une "distance sémantique" entre des concepts définis avec la logique de descriptions $\mathcal{ALE}$ [BAA 03], qui tire parti des connaissances d'un domaine incluses dans une ontologie ou un thésaurus. Cette distance sémantique est calculée grâce à une structure de treillis complet généré automatiquement et contenant l'ensemble de toutes les constructions de concepts possibles avec $\mathcal{ALE}$ et un ensemble

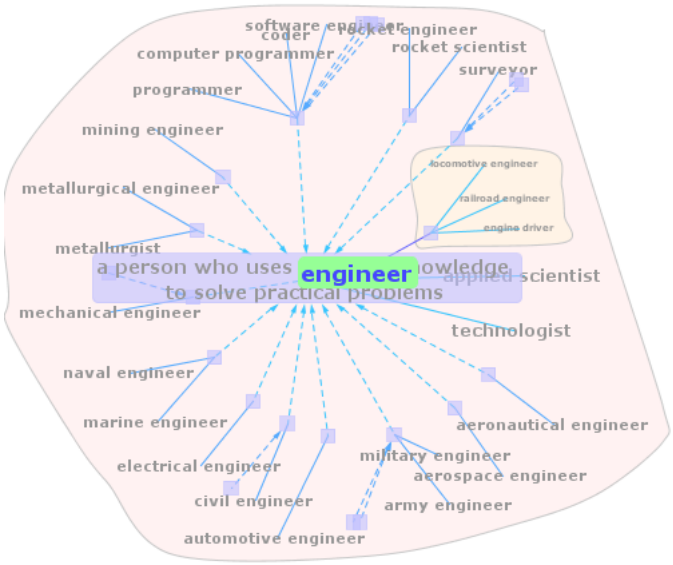

Figure 1. *Fragment de Wordnet*

donné de concepts et de rôles primitifs. La mesure de ressemblance entre concepts permet d'évaluer la "proximite"' entre concepts et peut être considérée soit comme mesure de similarité (croissance avec la ressemblance) soit comme une distance (décroissance avec la ressemblance).

Cet article est organisé comme suit : le paragraphe 2 présente les logiques de descriptions ainsi que les notions de concept subsumant le plus spécifique et de concept subsumé le plus général. Le paragraphe 3 donne un aperçu sur quelques mesures de similarité existantes. Le paragraphe 4 définit la notion de treillis de concepts complet. Le paragraphe 5 présente des propriétés souhaitables pour une distance sémantique. Une nouvelle distance est introduite dans le paragraphe 6. Un algorithme implémentant cette distance est présenté dans le paragraphe 7.

2. Quelques rappels sur les logiques de descriptions

Ce paragraphe n'introduit que brièvement le formalisme des logiques de descriptions et les notions de concept subsumant le plus spécifique (SPS) et de concept subsumé le plus général (SPG). Pour plus de détails, le lecteur peut se référer à [BAA 03]. Les notions de SPS et SPG sont introduites pour évaluer la distance entre deux concepts définis en $\mathcal{ALE}$. Cette distance se voit attribuer une valeur qui dépend

de la longueur du chemin entre les deux concepts, chemin qui passe soit par les SPS soit par les SPG.

2.1. *Définitions de base*

Les logiques de descriptions (LDs) forment une famille de langages de représentation des connaissances [BAA 03]. Les éléments importants du domaine sont décrits par des descriptions de concepts, c'est-à-dire, des expressions établies à partir de concepts atomiques (relation unaire) et des rôle atomiques (relations binaires) et d'un ensemble de constructeurs comme : $\top$ (concept le plus général), $\bot$ (concept le plus spécifique), $\neg\mathtt{C}$ (négation), $\forall\mathtt{r}.\mathtt{C}$ (quantification universelle), $\exists\mathtt{r}.\mathtt{C}$ (quantification existentielle), $\mathtt{C}\sqcap\mathtt{D}$ (conjonction), $\mathtt{C}\sqcup\mathtt{D}$ (disjonction), ($\geq\mathtt{n}\,\mathtt{r}$) et ($\leq\mathtt{n}\,\mathtt{r}$) (cardinalités sur les rôles).

À l'instar de la logique classique, une sémantique est associée aux descriptions des concepts et des rôles : les concepts sont interprétés comme des sous-ensembles d'un domaine d'interprétation Δ^I et les rôles comme des sous-ensembles du produit $\Delta^I\times\Delta^I$. Une interprétation *I*= (Δ^I,.^{I}) est la donnée d'un ensemble Δ^I appelé domaine de l'interprétation et d'une fonction d'interprétation .^{I} qui fait correspondre à un concept un sous-ensemble C^I de Δ^I et à un rôle un sous-ensemble r^I de $\Delta^I\times\Delta^I$.

Les LDs introduisent la notion de satisfiabilité d'un concept : un concept C est satisfiable si et seulement s'il existe une interprétation *I* telle que $C^I\neq\emptyset$; C est non satisfiable sinon. La subsomption est la relation d'ordre partiel qui permet d'organiser les concepts en hiérarchie : C est subsumé par D (noté C $\sqsubseteq$ D) si l'interprétation C^I de C est incluse dans l'interprétation D^I de D pour toute interprétation *I* (ou si toutes les instances de C sont nécessairement des instances de D) [BAA 03]. Les LDs offrent des modes de raisonnement comme la classification et l'instanciation. L'algorithme de classification permet de déterminer les concepts les plus généraux et les concepts les plus spécifiques d'un concept. Le test d'instantiation permet de vérifier qu'un individu *a* est instance d'un concept C dans une base de connaissances. Ce test permet, d'une manière plus précise, de trouver les concepts les plus spécifiques dont l'individu est une instance. Les axiomes sont de la forme $\mathtt{E}_1\sqsubseteq\mathtt{E}_2$ où $\mathtt{E}_1$ et $\mathtt{E}_2$ sont des expressions conceptuelles. Un axiome sert à introduire un concept ou encore à établir une "règle" entre concepts ou expressions conceptuelles.

Dans ce papier, notre étude se limite à l'étude des concepts définis avec la logique de descriptions $\mathcal{ALE}$ [BAA 03], qui repose sur les constructeurs $\top, \bot, \mathtt{C}\sqcap\mathtt{D}, \forall\mathtt{r}.\mathtt{C}, \exists\mathtt{r}.\mathtt{C}, \neg\mathtt{A}$ (la dernière négation ne porte que sur les concepts atomiques).

2.2. *Les SPS et les SPG de concepts dans les logiques descriptions*

Ce paragraphe introduit les notions du concept subsumant le plus spécifique (SPS, en anglais *LCS* pour *least common subsumer*) et du concept subsumé le plus général (SPG, en anglais *MGS* pour *most general subsumee*) pour une collection de concepts.

Le SPS d'une collection de concepts C_1,...,C_n est le concept le plus spécifique D qui subsume C_1,...,C_n tel que $C_i \sqsubseteq D$ pour i=1,..,n ; et si $C_i \sqsubseteq E$ pour i=1,..,n alors $D \sqsubseteq E$.

Le SPG d'une collection de concepts C_1,...,C_n est le concept le plus général D qui est subsumé par C_1,...,C_n tel que $D \sqsubseteq C_i$ pour i=1,..,n ; et si $E \sqsubseteq C_i$ pour i=1,..,n alors $E \sqsubseteq D$.

Dans [BAA 04b], la borne supérieure de deux concepts pour la logique de descriptions $\mathcal{ALE}$ se définit comme suit : le concept $\mathtt{C}$ est la *borne supérieure* des concepts $\mathtt{C_1}$ et $\mathtt{C_2}$ si $\mathtt{C_1} \sqsubseteq \mathtt{C}$ et $\mathtt{C_2} \sqsubseteq \mathtt{C}$, et si $\mathtt{C}$ est le plus petit concept vérifiant cette propriété, c'est-à-dire que si un concept $\mathtt{C'}$ vérifie $\mathtt{C_1} \sqsubseteq \mathtt{C'}$ et $\mathtt{C_2} \sqsubseteq \mathtt{C'}$, alors $\mathtt{C} \sqsubseteq \mathtt{C'}$.

Un algorithme élémentaire pour calculer la borne supérieure de deux concepts $\mathtt{lcs}(\mathtt{C_1}, \mathtt{C_2})$ dans $\mathcal{ALE}$ est le suivant :

– si $\mathtt{C} = \mathtt{A_1} \sqcap \mathtt{A_2}$ et $\mathtt{D} = \mathtt{A_3} \sqcap \mathtt{A_4}$ alors $\mathtt{lcs}(\mathtt{C},\mathtt{D}) = \top$, où les $\mathtt{A_i}$ sont primitifs et (tous) différents,

– si $\mathtt{C} = \mathtt{A_1} \sqcap \mathtt{A_2}$ et $\mathtt{D} = \mathtt{A_1} \sqcap \mathtt{A_3}$ alors $\mathtt{lcs}(\mathtt{C},\mathtt{D}) = \mathtt{A_1}$, où les $\mathtt{A_i}$ sont primitifs et différents,

– si $\mathtt{C} = \forall \mathtt{r}.\mathtt{C'}$ et $\mathtt{D} = \forall \mathtt{r}.\mathtt{D'}$ alors $\mathtt{lcs}(\mathtt{C},\mathtt{D}) = \forall \mathtt{r}.\mathtt{lcs}(\mathtt{C'},\mathtt{D'})$,

– si $\mathtt{C} = \exists \mathtt{r}.\mathtt{C'}$ et $\mathtt{D} = \exists \mathtt{r}.\mathtt{D'}$ alors $\mathtt{lcs}(\mathtt{C},\mathtt{D}) = \exists \mathtt{r}.\mathtt{lcs}(\mathtt{C'},\mathtt{D'})$.

Le SPG de deux concepts A et B définis en $\mathcal{ALE}$ est le concept A⊓B.

Par exemple dans la figure 2, le SPS et le SPG des concepts $\mathtt{A}$ et $\mathtt{B}$ calculés selon l'algorithme de [BAA 04b] sont :

$\mathtt{A} \equiv \mathtt{Technician} \sqcap \exists \mathtt{competency}.(\mathtt{Java} \sqcap \mathtt{Network})$
$\mathtt{B} \equiv \mathtt{Engineer} \sqcap \exists \mathtt{competency}.\mathtt{Java}$
$\mathtt{SPS}(\mathtt{A},\mathtt{B}) = \exists \mathtt{competency}.\mathtt{Java}$
$\mathtt{SPG}(\mathtt{A},\mathtt{B}) = \mathtt{Technician} \sqcap \exists \mathtt{competency}.(\mathtt{Java} \sqcap \mathtt{Network}) \sqcap \mathtt{Engineer}$

Après avoir introduit la logique de descriptions $\mathcal{ALE}$ qui est le formalisme utilisé pour décrire les concepts dont on va calculer la proximité, nous présentons dans la section suivante un bref état de l'art sur les mesures de similarité.

3. Un bref état de l'art sur les mesures de similarité

3.1. *La mesure de Wu et Palmer*

La longueur de chemin entre deux concepts dans une hiérarchie est une mesure intuitive pour calculer la similarité. C'est une mesure utile et facile à implémenter en même temps. Une mesure qui est fonction de la longueur de chemin est proposée par

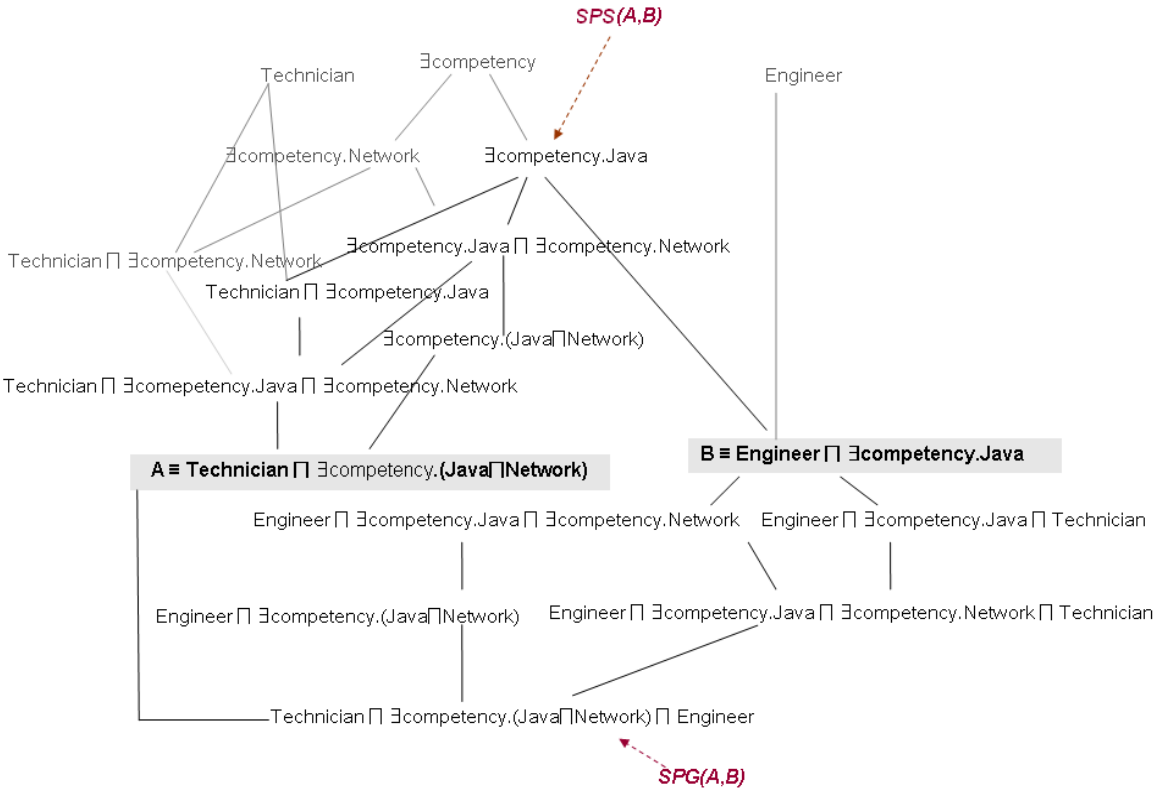

Figure 2. *Exemple de chemins entre deux concepts avec leur SPS et leur SPG*

[WU 94], qui s'appuie sur la longueur du chemin entre deux concepts d'une même hiérarchie. Cette mesure est définie comme suit :

$$\mathtt{sim}(\mathtt{C}_1, \mathtt{C}_2) = \frac{2 * \mathtt{depth}(\mathtt{C})}{\mathtt{depth}_{\mathtt{C}}(\mathtt{C}_1) + \mathtt{depth}_{\mathtt{C}}(\mathtt{C}_2)}$$

où C est le subsumant commun le plus spécifique, depth(C) est la longueur du chemin entre C et la racine de la hiérarchie, $depth_C(C_i)$ est le nombre d'arcs entre C_i et la racine en passant par C. Cette mesure est facile à implémenter mais elle ne tient pas compte des descriptions de concepts.

3.2. *La mesure de Resnik*

La notion de contenu informationnel (CI) a été utilisée par [RES 99], qui définit la pertinence d'un concept dans un corpus. La fréquence d'un concept est calculée pour déterminer le CI (les fréquences des concepts dans une hiérarchie sont estimées en utilisant la fréquence des termes dans un corpus) :

$$\begin{aligned} \mathtt{CI}(\mathtt{c}) &= -\mathtt{log}(\mathtt{P}(\mathtt{c})) \\ \mathtt{P}(\mathtt{c}) &= \frac{\mathtt{frequency(c)}}{\mathtt{N}} \end{aligned}$$

où P(c) est la probabilité d'un concept $\mathtt{c}$ et où $\mathtt{N}$ est le nombre total de concepts). [RES 99] définit une mesure de similarité entre deux concepts par la quantité d'information qu'ils partagent. Cette similarité est calculée par :

$$\mathtt{sim}(\mathtt{c}_1, \mathtt{c}_2) = \mathtt{CI}(\mathtt{lcs}(\mathtt{c}_1, \mathtt{c}_2))$$

où $\mathtt{lcs}(\mathtt{C}_1, \mathtt{C}_2)$ est le subsumant commun le plus spécifique de $\mathtt{C}_1$ et $\mathtt{C}_2$ dans la hiérarchie.

3.3. *La mesure de Lin*

[LIN 98] propose une définition théorique d'une mesure de similarité applicable à partir du moment où l'on dispose d'un modèle de probabilité. Cette similarité est définie comme le rapport des informations partagées par A et B sur les informations nécessaires pour décrire d'une façon complète A et B. Cette mesure est définie par :

$$\mathtt{sim}(\mathtt{A}, \mathtt{B}) = \frac{\mathtt{logP(common(A,B))}}{\mathtt{logP(description(A,B))}}$$

où $\mathtt{common}(\mathtt{A}, \mathtt{B})$ est l'ensemble des caractéristiques communes de $\mathtt{A}$ et $\mathtt{B}$, et $\mathtt{description}(\mathtt{A}, \mathtt{B})$ est l'union de toutes les caractéristiques de $\mathtt{A}$ et $\mathtt{B}$. Cette mesure est applicable dans plusieurs domaines et elle a montré sa bonne correspondance avec les jugements humains.

3.4. *Une mesure reposant sur l'interprétation de concepts*

Les auteurs dans [D'A 06] ont proposé une mesure de similarité pour des concepts décrits avec la logique de descriptions $\mathcal{ALC}$ [BAA 03] définie de la façon suivante :

$\mathtt{s} : \ell \times \ell \rightarrow [\mathtt{0..1}]$:

$$\mathtt{s}(\mathtt{C}, \mathtt{D}) = \frac{|(\mathtt{C} \sqcap \mathtt{D})|^{\mathtt{I}}}{|\mathtt{C}|^{\mathtt{I}} + |\mathtt{D}|^{\mathtt{I}} - |(\mathtt{C} \sqcap \mathtt{D})|^{\mathtt{I}}} * \mathtt{max}(\frac{|(\mathtt{C} \sqcap \mathtt{D})|^{\mathtt{I}}}{|\mathtt{C}|^{\mathtt{I}}}, \frac{|(\mathtt{C} \sqcap \mathtt{D})|^{\mathtt{I}}}{|\mathtt{D}|^{\mathtt{I}}})$$

où $(.)^{\mathtt{I}}$ est une fonction d'interprétation et $|.|$ représente le cardinal d'un ensemble.

Cette mesure est proche de la mesure de similarité proposée par [TVE 77] à un coefficient près. Cette mesure est intéressante parce qu'elle vérifie des propriétés sémantiques comme : la similarité entre deux concepts équivalents ($\mathtt{C} \equiv \mathtt{D}$) est égale à 1, la similarité entre deux concepts incompatibles est nulle.

3.5. *Discussion*

La plupart des mesures existantes calculent la proximité pour des concepts appartenant à une même hiérarchie prédéfinie (hiérarchie de concepts atomiques donnée comme entrée) ([WU 94], [LIN 98],[JIA 97]). [D'A 06] calcule la similarité entre concepts définis avec la logique de descriptions $\mathcal{ALC}$. Les mesures de similarité développées pour des logiques de descriptions expressives nécessitent des informations supplémentaires : un modèle de probabilité ou une base d'instances. Dans plusieurs applications pratiques, un modèle de probabilité (ou bien des probabilités d'apparitions de concepts) n'est pas fourni avec une hiérarchie de concepts. Les mesures de similarité utilisant une base d'instances pour le calcul de similarité, comme [D'A 06], présentent certains inconvénients : par exemple s'il n'existe pas une instance satisfaisant un concept requête, la similarité entre une requête et les ressources proches sera nulle et le moteur de recherche ne sera pas capable de retourner des résultats approchés.

Pour surmonter ce genre de limitations, cet article propose une nouvelle distance permettant d'évaluer les ressemblances entre deux concepts définis en $\mathcal{ALE}$. De plus, cette distance ne nécessite pas obligatoirement une hiérarchie prédéfinie de concepts atomiques (mais peut la prendre en compte si elle existe) ni d'autres informations supplémentaires (modèle de probabilité, base d'instances,...). Le principe de la distance proposée est de générer automatiquement un treillis de concepts complet à partir de la donnée d'un ensemble de concepts et de rôles primitifs ou bien en complétant une hiérarchie prédéfinie de concepts atomiques si elle existe. La distance entre deux concepts est évaluée en calculant la longueur de chemins entre les concepts dans le treillis complet généré. De plus, la structure d'un treillis complet permet de résoudre le problème des hiérarchies ayant une densité hétérogène à cause de la régularité de sa structure.

La nouvelle distance que nous proposons peut être vue comme une distance d'édition entre deux concepts définis en $\mathcal{ALE}$. Il existe des mesures de similarité ou des distances qui sont développées pour les arbres avec des distances d'édition : [BUN 94]. L'approche est similaire à la notre mais s'appuie sur un formalisme différent. Une distance d'édition évaluant la proximité entre deux éléments dans un formalisme donné sur la base d'un chemin minimal de transformations entre éléments est intuitive et naturelle. Cependant, à notre connaissance cette approche n'a pas encore été appliquée entre concepts définis dans une logique de descriptions comme $\mathcal{ALE}$.

Dans le paragraphe suivant nous introduisons la notion de treillis de concepts complet, structure qui nous permet de calculer la distance entre deux concepts.

4. Treillis de concepts

Un treillis $(\mathcal{L}, \leq)$ est un ensemble ordonné par une relation d'ordre partiel tel que pour tout couple d'éléments de $\mathtt{v}_1, \mathtt{v}_2$ de $\mathcal{L}$ une borne inférieure $\mathtt{inf}(\mathtt{v}_1, \mathtt{v}_2)$ et une borne supérieure $\mathtt{sup}(\mathtt{v}_1, \mathtt{v}_2)$ existent et sont uniques [DAV 02], [BAR 70].

Pour une LD donnée $\mathcal{L}$ munie d'un ensemble de constructeurs κ, un ensemble C de concepts primitifs et un ensemble R de rôles primitifs, soit $S_{\mathcal{L}}(C, R)$ l'ensemble de toutes les descriptions qui peuvent être construites avec κ, C et R, qui est appelé *ensemble complet de concepts*. Nous définissons un "treillis de concepts complet" (ou "treillis complet") comme le treillis obtenu en classifiant les concepts de $S_{\mathcal{L}}(C, R)$ en prenant compte les relations d'équivalence et de subsomption.

Pour certaines LDs, l'ensemble complet de concepts est un treillis. Si une LD contient le constructeur $\sqcap$ alors la borne inférieure $inf(C_1, C_2) = C_1 \sqcap C_2$ est unique. Pour la LD $\mathcal{ALE}$, la borne supérieure $sup(C_1, C_2)$ est unique est unique pour un couple de concepts, comme cela est démontré dans [BAA 04b] : $sup(C_1, C_2) = SPS(C_1, C_2)$ et $inf(C_1, C_2) = C_1 \sqcap C_2$. par suite, l'ensemble complet de concepts $S_{\mathcal{L}}(C, R)$ dans $\mathcal{ALE}$ est un treillis.

La figure 2 introduit l'exemple d'un fragment de treillis complet construit entre les concepts $\mathtt{A}$ et $\mathtt{B}$ de la façon suivante :

$$\mathtt{A} \equiv \mathtt{Technician} \sqcap \exists \mathtt{competency}.\mathtt{Java} \sqcap \mathtt{Network}$$
$$\mathtt{B} \equiv \mathtt{Engineer} \sqcap \exists \mathtt{competency}.\mathtt{Java}$$

Cet extrait de treillis complet contient l'ensemble complet de tous les concepts possibles entre $\mathtt{A}$ et $\mathtt{B}$, $\mathtt{SPS}(\mathtt{A}, \mathtt{B})$ et et $\mathtt{SPG}(\mathtt{A}, \mathtt{B})$, lorsque l'ensemble des concepts primitifs est $\{\mathtt{Technician}, \mathtt{Engineer}, \mathtt{Java}, \mathtt{Network}\}$, et l'ensemble des rôles est $\{\mathtt{competency}\}$, et les constructeurs sont $\{\sqcap, \exists\}$.

Il faut encore remarquer que dans un treillis complet de concepts représentés en $\mathcal{ALE}$, les longueurs de tous les plus courts chemins entre deux concepts A et B sont égales. En outre, le SPS et le SPG de deux concepts dans un treillis de concepts complet définis en $\mathcal{ALE}$ sont uniques. Les preuves de ces propriétés sont données en détail dans le rapport [MAA 07] (voir aussi [BAA 04a]).

5. Propriétés souhaitées pour une distance sémantique

Notre objectif est de fournir une distance sémantique *d* applicable dans une LD expressive comme $\mathcal{ALE}$, qui ne nécessite pas la donnée d'informations supplémentaires (modèle de probabilité, base d'instances...). La donnée d'une hiérarchie de concepts atomiques si elle existe, ou encore la simple donnée d'un ensemble de concepts et de rôles primitifs, peuvent être suffisantes pour calculer la distance entre n'importe quel couple de concepts définis en $\mathcal{ALE}$.

Une distance *d* est dite *sémantique* si et seulement si elle vérifie les propriétés suivantes, appelées *propriétés sémantiques* :

1) si A≡B alors d(A,B) = 0 : la distance entre deux concepts équivalents est nulle.

2) si A $\sqsubseteq$ B $\sqsubseteq$ C alors d(A,B)$\leq$ d(A,C) : la distance entre un concept avec un subsumant direct est inférieure à sa distance avec n'importe quel autre subsumant.

3) si A $\sqcap$ B $\sqsubseteq$ $\bot$ alors d(A,B) = ∞ : la distance entre deux concepts incompatibles est infinie.

Cette distance *d* est sémantique car elle permet d'exprimer certaines relations sémantiques intuitives entre les concepts, relatives à la ressemblance entre les concepts, comme le fait que la distance entre deux concepts équivalents est nulle. Nous voulons qu'une telle distance sémantique soit aussi une distance mathématique et elle doit alors vérifier les axiomes suivants :

– d(A,B) = 0 $\Leftrightarrow$ A = B (séparation)

– d(A,B) = d(B,A) (symétrie)

– d(A,C) $\leq$ d(A,B) + d(B,C) (inégalité triangulaire)

6. Une proposition de distance sémantique pour $\mathcal{ALE}$

Dans ce paragraphe, nous introduisons une distance *d* dans $\mathcal{ALE}$ vérifiant toutes les propriétés sémantiques et les axiomes d'une distance mathématique, avec :

$$\texttt{v\_arc}(\texttt{A}, \bot) = \infty \text{ où } \texttt{A subsume directement } \bot$$
$$\begin{aligned}\texttt{d}(\texttt{A},\texttt{B}) &= \texttt{n\_arc}(\texttt{A}, \texttt{SPS}(\texttt{A},\texttt{B})) + \texttt{n\_arc}(\texttt{B}, \texttt{SPS}(\texttt{A},\texttt{B})) \\ &= \texttt{n\_arc}(\texttt{A}, \texttt{SPG}(\texttt{A},\texttt{B})) + \texttt{n\_arc}(\texttt{B}, \texttt{SPG}(\texttt{A},\texttt{B}))\end{aligned}$$

où `v_arc` est la valeur d'un arc séparant deux concepts (`v_arc(A,B)` est égale à 1 sauf pour les concepts subsumant directement $\bot$) et `n_arc(A,B)` représente le nombre d'arcs entre A et B dans un treillis de concepts complet à travers leur SPS, c'est à dire le chemin minimal de A à B en passant par leur SPS, ou à travers leur SPG (de même c'est le chemin minimal entre A et B en passant par leur SPG) en prenant en compte de la valeur des arcs entre eux (le SPS et le SPG de deux concepts dans un treillis de concepts complet définis en $\mathcal{ALE}$ sont uniques).

Pour calculer la longueur d'un chemin entre deux concepts, une méthode simple consiste à compter le nombre d'arcs intermédiaires qui existent entre eux dans un treillis de concepts complet. L'algorithme dans la section 7 explique la façon de trouver les concepts intermédiaires entre deux concepts. Cette distance qui permet de calculer la proximité entre deux concepts définis en $\mathcal{ALE}$ possède l'avantage d'être à la fois une distance sémantique et une distance mathématique.

La longueur de chemin entre A et B calculée par notre distance est analogue à la mesure proposée par [WU 94]. Cependant, ici il est possible d'évaluer la similarité entre deux concepts définis en $\mathcal{ALE}$ et non seulement entre concepts atomiques d'une

hiérarchie prédéfinie. La distance peut ne pas être calculée en tenant compte du SPS de deux concepts, à cause de la complexité de son calcul dans $\mathcal{ALE}$, mais elle peut être calculée en prenant en compte leur SPG.

Les preuves que notre distance est sémantique et mathématique sont données en détail dans le rapport [MAA 07].

7. Une distance entre concepts définis en $\mathcal{ALE}$

Nous présentons un algorithme permettant de générer les concepts intermédiaires entre deux concepts A et B codés en $\mathcal{ALE}$ et leur SPG. Le calcul des concepts intermédiaires entre (A, SPG(A,B)) et (B, SPG(A,B)) est utile pour pouvoir les compter et déterminer ainsi la distance entre eux. Comme SPG(A,B) dans $\mathcal{ALE}$ est équivalent à (A $\sqcap$ B) alors les concepts entre A et (A $\sqcap$ B) sont de la forme (A $\sqcap$ D) où D subsume B : le concept (A $\sqcap$ D) est plus spécifique que A et plus général que (A $\sqcap$ B). De même, les concepts intermédiaires entre B et (A $\sqcap$ B) sont de la forme (B $\sqcap$ E) où E subsume A. Par exemple, dans la figure 2, nous obtenons les résultats suivants, pour les concepts $\texttt{A}$ et $\texttt{B}$:

$\texttt{A} \equiv \texttt{Technician} \sqcap \exists \texttt{competency}.(\texttt{Java} \sqcap \texttt{Network})$
$\texttt{B} \equiv \texttt{Engineer} \sqcap \exists \texttt{competency}.\texttt{Java}$
$\texttt{SPG}(\texttt{A},\texttt{B}) = \texttt{Technician} \sqcap \exists \texttt{competency}.(\texttt{Java} \sqcap \texttt{Network}) \sqcap \texttt{Engineer}$
$\texttt{Y} = \texttt{Engineer} \sqcap \exists \texttt{competency}.\texttt{Java} \sqcap \exists \texttt{competency}.\texttt{Network}$

Tous les concepts intermédiaires entre $\texttt{A}$ et $\texttt{B}$ sont des la forme $\texttt{B} \sqcap \texttt{D}$ où $\texttt{D}$ est un subsumant de $\texttt{A}$. En effet, le concept $\texttt{Y}$ est plus spécifique que $\texttt{B}$ et plus général que $\texttt{SPG}(\texttt{A},\texttt{B})$, et $\texttt{Y}$ est de la forme $\texttt{B} \sqcap \exists \texttt{competency}.\texttt{Network}$ où $\exists \texttt{competency}.\texttt{Network}$ est un subsumant de $\texttt{A}$.

Le calcul des concepts intermédiaires entre $\texttt{A}$, $\texttt{B}$ et leur SPG, se fait selon l'algorithme suivant :

– calculer les subsumants de A et les subsumants de B,

– construire le concept A $\sqcap$ B (concept correspondant à SPG(A,B)),

– construire les concepts de la forme (A $\sqcap$ D) où D subsume B et les classer entre A et SPG(A,B) (ceci permet de construire tous les concepts intermédiaires entre A et SPG(A,B)),

– construire les concepts de la forme (B $\sqcap$ E) où E subsume A et les classer entre B et SPG(A,B) (après cette étape les concepts intermédiaires entre A, B et leur SPG sont alors tous construits).

Le calcul des subsumants d'un concept défini en $\mathcal{ALE}$ (nécessaires pour la construction de concepts intermédiaires entre deux concepts) se fait comme suit. Soit A un concept défini en $\mathcal{ALE}$ et $\gamma(A)$ l'ensemble des subsumants de A. $\gamma(A)$ est construit

de la façon suivante :

– $\gamma(A) \leftarrow \{A, \top\}$ si A est un concept atomique ou la négation d'un concept atomique et $A \neq \bot$.

– $\gamma(A) \leftarrow \{C' \sqcap D'\}$ où $C' \in \gamma(C)$ et $D' \in \gamma(D)$ si $A = C \sqcap D$ et $A \neq \bot$.

– $\gamma(A) \leftarrow \{\bigsqcap_i \exists r.C_i\}$ où $C_i \in \gamma(C)$ si $A = \exists r.C$ et $A \neq \bot$.

– $\gamma(A) \leftarrow \{\forall r.C'\}$ où $C' \in \gamma(C)$ si $A = \forall r.C$ et $A \neq \bot$.

Ces règles sont appliquées jusqu'à saturation (il n'est plus possible d'appliquer aucune règle). Par exemple, l'application de cet algorithme sur le concept `A` permet d'engendrer l'ensemble `S` de tous les subsumants de `A` :

$$\texttt{A} \equiv \texttt{Technician} \sqcap \exists \texttt{competency}.(\texttt{Java} \sqcap \texttt{Network})$$

$$\begin{aligned}\texttt{S} = \{&\texttt{Technician},\\ &\exists\texttt{competency.Java},\\ &\exists\texttt{competency.Network},\\ &\exists\texttt{competency.Java} \sqcap \texttt{Network},\\ &\exists\texttt{competency.Java} \sqcap \exists\texttt{competency.Network},\\ &\texttt{Technician} \sqcap \exists\texttt{competency.Java},\\ &\texttt{Technician} \sqcap \exists\texttt{competency.Network},\\ &\texttt{Technician} \sqcap \exists\texttt{competency.Java} \sqcap \exists\texttt{competency.Network},\\ &\exists\texttt{competency}.\top,\\ &\texttt{Technician} \sqcap \exists\texttt{competency}.\top,\\ &\top\}\end{aligned}$$

L'algorithme général implémentant notre distance est très simple et fait appel à l'algorithme permettant de générer les concepts intermédiaires entre deux concepts A et B, puis compte le nombre d'arcs entre les concepts, tout en tenant compte des valeurs des poids des arcs (voir section 6).

8. Conclusion et perspectives

Cet article a proposé une distance sémantique entre deux concepts définis avec la logique de descriptions $\mathcal{ALE}$. La distance entre deux concepts correspond à la longueur d'un chemin dans un treillis de concepts complet, généré automatiquement, à partir des SPS et des SPG des concepts considérés. Contrairement au travaux existants, la distance entre deux concepts ne nécessitent pas des informations supplémentaires (modèle de probabilité, base d'instances,...).

Dans des futurs travaux, nous voulons étendre notre distance pour prendre en compte une logique de descriptions plus expressive. Dans un premier temps, nous voulons introduire le constructeur "$\sqcup$". Cependant, l'ajout de ce constructeur en-

traîne un nombre infini de subsumants pour tout concept et donc un nombre infini de concepts entre chaque couple de concepts A et B tel que A$\sqsubseteq$B. Par exemple, $\exists r.A \sqsubseteq \ldots \sqsubseteq \exists r(A \sqcup \exists r) \sqsubseteq \ldots \sqsubseteq \exists r.A \sqcup (\exists r.A \sqcup (\exists r.A \sqcup (\exists r \ldots))) \sqsubseteq \exists r.\top$.

9. Bibliographie

[BAA 03] BAADER F., CALVANESE D., MCGUINNESS D., NARDI D., PATEL-SCHNEIDER P., Eds., *The Description Logic Handbook*, Cambridge University Press, Cambridge, UK, 2003.

[BAA 04a] BAADER F., SERTKAYA B., « Applying Formal Concept Analysis to Description Logics », EKLUND P., Ed., *Second International Conference on Formal Concept Analysis, Sydney (ICFCA 2004)*, Lecture Notes in Artificial Intelligence 2961, p. 261–286, Springer, Berlin, 2004.

[BAA 04b] BAADER F., SERTKAYA B., TURHAN A.-Y., « Computing the Least Common Subsumer w.r.t. a Background Terminology », ALFERES J., LEITE J., Eds., *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, vol. 3229 de *Lecture Notes in Computer Science*, Lisbon, Portugal, 2004, Springer-Verlag, p. 400–412.

[BAR 70] BARBUT M., MONJARDET B., *Ordre et classification – Algèbre et combinatoire (2 tomes)*, Hachette, Paris, 1970.

[BUN 94] BUNKE H., MESSMER B., « Similarity Measures for Structured Representations », WESS S., ALTHOFF K.-D., RICHTER M., Eds., *Topics in Case-Based Reasoning – First European Workshop (EWCBR'93), Kaiserslautern*, Lecture Notes in Artificial Intelligence 837, p. 106–118, Springer Verlag, berlin, 1994.

[D'A 06] D'AMATO C., FANIZZI N., ESPOSITO F., « A dissimilarity measure for ALC concept descriptions », *SAC'06 : Proceedings of the 2006 ACM symposium on Applied computing*, New York, NY, USA, 2006, ACM Press, p. 1695–1699.

[DAV 02] DAVEY B., PRIESTLEY H., *Introduction to Lattices and Order*, Cambridge University Press, 2002.

[JIA 97] JIANG J. J., CONRATH D. W., « Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy », 1997.

[LIN 98] LIN D., « An Information-Theoretic Definition of Similarity », *Proceedings of the 15th International Conference on Machine Learning (ICML'98), Madison, Wisconsin*, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1998.

[MAA 07] MAALA M., DELTEIL A., NAPOLI A., « Distance sémantique entre concepts définis en ALE », Rapport de recherche, 2007, LORIA, Nancy.

[RES 99] RESNIK P., « Semantic Similarity in a Taxonomy : An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language », *Journal of Artificial Intelligence Research*, vol. 11, 1999, p. 95-130.

[TVE 77] TVERSKY A., « Features of Similarity », *Psychological Review*, vol. 84, n° 4, 1977, p. 327–352.

[WU 94] WU Z., PALMER M., « Verb Semantics and Lexical Selection », *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics (ACL-94), Las Cruces, New Mexico*, 1994, p. 133–138.