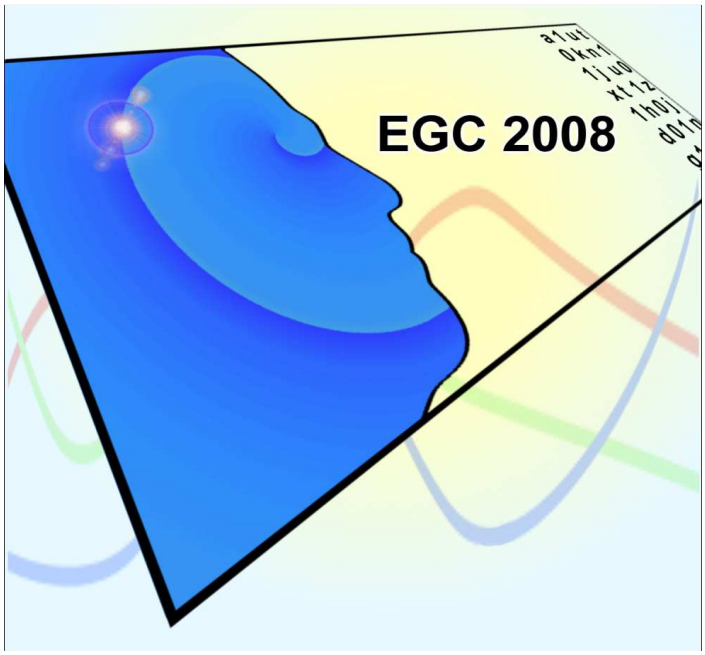
8èmes Journées Francophones
Extraction et Gestion des Connaissances
Sophia Antipolis 29 janvier 2008

Atelier



Mesures de similarité sémantique

Organisateurs :

- Marie-Aude Aufaure (SUPELEC & INRIA)
- Omar Boussaid (ERIC, Univ. Lyon 2)
- Pascale Kuntz (LINA, Univ. de Nantes)

Responsables des Ateliers EGC :

Alzennyr Da Silva (INRIA, Rocquencourt)
Alice Marascu (INRIA, Sophia Antipolis)
Florent Masseglia (INRIA, Sophia Antipolis)

http://www-sop.inria.fr/axis/egc08

EGC

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SOPHIA ANTIPOLIS - MÉDITERRANÉE

Tables des matières

Avant- propos

1er atelier sur les Mesures de Similarité Sémantique – 29 janvier 2008 – Sophia-Antipolis

Préoccupation centrale des taxonomistes du siècle précédent, le problème de la construction de mesures de similarité – ou de dissimilarité – permettant de comparer des entités distinctes, connaît un regain d'intérêt majeur inhérent à l'évolution des nouvelles technologies de traitement de l'information. La complexification croissante des données (données textuelles, données multimédia, données spatiales, données structurées, …) nécessite le développement de mesures capables d'intégrer des informations hétérogènes tout en gardant une pertinence sémantique eu égard au domaine applicatif. Cette complexification est également associée à un changement d'échelle qui requiert pour le traitement des algorithmes performants dont les résultats dépendent souvent étroitement des choix préalables des mesures de similarité.

Certaines mesures, celle de Jaccard qui remonte au tout début du XXème siècle étant un exemple paradigmatique, sont utilisées, parfois sous des appellations différentes, dans des domaines variés. D'autres ont été spécifiquement construites pour des objectifs bien spécifiques et semblent a priori plus difficilement adaptables.

La variété des champs applicatifs rend donc plus que jamais nécessaire la confrontation des points de vue sur la construction de ces mesures. Citons à titre illustratif, la recherche d'information avec les problématiques d'indexation de documents et de résumé automatique de textes, l'ingénierie des connaissances avec les problématiques d'alignement et d'extraction d'ontologie, ou la bio-informatique avec les problématiques majeures de comparaison de séquences ou de molécules. Les informations prises en compte dans la construction des mesures de similarité peuvent être de différentes natures : textuelles ou spatiales mais aussi combinatoires lorsque les données sont reliées explicitement par des relations, comme par exemple les concepts dans une ontologie décrite par une hiérarchie de subsomption ou les molécules décrites par des graphes. Elles peuvent également intégrer la notion de contexte souvent nécessaire pour améliorer la qualité des résultats obtenus. Par exemple, dans le cas de l'extraction d'ontologies à partir de

pages web, le contexte qui peut être représenté par la structure des pages, permet de pondérer les concepts selon leur position dans le texte.
L'objectif de cet atelier est donc de faire le point sur les recherches en cours dans ce domaine. Il se veut un lieu ouvert de rencontres et d'échanges permettant à la fois de présenter des travaux aboutis et d'exposer des réflexions sur le domaine ou des travaux préliminaires.

Cet atelier est composé de 11 communications réparties dans trois sessions. Elles présentent des problématiques variées où la notion similarité sémantique joue un rôle clé.
La *première session* présente des travaux autour des schémas XML, des entrepôts de données, de l'enrichissement de requêtes dans des systèmes distribués, ainsi qu'une approche de contextualisation de distances sémantiques.
La *seconde session* débute par un aperçu de différentes approches permettant de calculer une distance entre ontologies. La seconde présentation est consacrée à l'extraction de relations de généralités entre noms provenant d'un corpus web. Enfin, le dernier article propose une évaluation comparative d'une mesure de degré de relation sémantique avec d'autres mesures classiques en ingénierie des connaissances.
Dans la *troisième session*, le premier article est consacré à un retour d'expériences sur les distances pour le web sémantique, ainsi que des propositions d'extensions. La seconde présentation s'articule autour de la comparaison « sémantique » d'objets et notamment la comparaison des éléments de base constitutifs des langages de modélisation d'entreprise. Une nouvelle mesure de similarité permettant de catégoriser des molécules chimiques dans le but de faciliter la recherche de molécules structurellement similaires est ensuite proposée. La présentation qui clôt cet atelier s'articule autour de la détection de similarité sémantique entre pages visitées dans un contexte de E-learning.

Responsables :

Marie-Aude Aufaure (Supelec & Inria projet Axis) : Marie-Aude.Aufaure@inria.fr
Omar Boussaid (Laboratoire ERIC, Université Lyon 2) : Omar.Boussaid@univ-lyon2.fr
Pascale Kuntz-Cosperec (Equipe COD, Laboratoire LINA) : Pascale.Kuntz-Cosperec@univ-nantes.fr

Comité de lecture :

Guillaume Cleuziou (LIFO, Université d'Orléans)
Sylvie Després (LIPN, Université Paris 13)
Jérôme Euzenat (INRIA Rhône-Alpes / Exmo)
Frédéric Furst (LARIA, Université de Picardie).
Fabien Gandon (INRIA Sophia-Antipolis / Edelweiss)
Amédeo Napoli (LORIA Nancy / Orpailleur)
Chantal Reynaud (LRI Université Paris-Sud & INRIA Futurs/ Gemo)
Franky Trichet (LINA, Université de Nantes)
Brigitte Trousse (INRIA Sophia-Antipolis / Axis)
Djamel Zighed (ERIC, Université Lyon 2)

Matching of enhanced XML schemas with a measure of context similarity

Myriam Lamolle*, Amar Zerdazi*

*LINC-IUT de Montreuil, Université Paris8
140, rue de la Nouvelle France, 93100 Montreuil
{m.lamolle, a.zerdazi}@iut.univ-paris8.fr
http://www.iut.univ-paris8.fr

Abstract. Schema matching is a critical step in integration of heterogeneous data sources. Recent integration work has mainly focused on developing matching techniques to find equivalent elements among the different XML sources. In this paper we propose a new approach to structural similarity measure based on the notion of context, between entities of the Enhanced XML Schemas, called EXS. In our approach, the set of the EXS schemas, are considered like a federation of XML schemas descended of different heterogeneous sources schemas (relational, object, XML, etc.) and enriched by the semantic metaknowledge. We present here the major problems bound to this crucial task, notably with regard to the semantic of schemas. So, we propose a structural matching algorithm. The algorithm takes two schema graphs as input, and produces as output a mapping between corresponding nodes of the schema graphs. After our algorithm runs, we expect a human to check and adjust the results.

1 Introduction

Schema matching is a schema manipulation process that takes as input two heterogeneous schemas and possibly some auxiliary information, and returns a set of dependencies, so called mappings that identify semantically related schema elements (Rahm, 2001). In practice, schema matching is done manually by domain experts (Miller and al., 2000), and it is time consuming and error prone. As a result, much effort has been done toward automating schema matching process. This is challenging for many fundamental reasons. According to (Drew and al., 1993), schema elements are matched based on their semantics. Semantics can be embodied within few information sources including designers, schemas, and data instances. Hence schema matching process typically relies on purely structure in schema and data instances (Doan and al., 2001). Schemas developed for different applications are heterogeneous in nature i.e. although the data they describe are semantically similar, the structure and the employed syntax may differ significantly (Abiteboul and al., 1997). To resolve schematic and semantic conflicts, schema matching often relies on element names, element datatypes, structure definitions, integrity constraints, and data values. However, such clues are often unreliable and incomplete. Schema matching cannot be fully automated and

thus requires user intervention, it is important that the matching process not only do as much as possible automatically but also identify when user input is necessary and maximally used (Boukottaya and al., 2004).

Consequently, a lot of work on schema matching tried to automate this process. The main goal of this paper is to propose a novel approach for structural matching based on the notion of structural node context. We propose a structural algorithm that can be used for matching of Enhanced XML Schema, called EXS. The EXS schemas, are considered like a federation of XML schemas descended of different heterogeneous schema sources (relational, object, XML, etc.) and enriched by the set of semantic metaknowledge. The algorithm that we suggest to perform structural matching is based on the following idea. The first step assigns for each node in source and target schema a context. After what, such context is compared to produce a node similarity coefficient that reflects structural similarity between schema nodes. The second one uses produced node similarity to derive a set of mappings.

The rest of paper is organized as follows. In section 2, we summarize some examples of recent schema matching algorithms that incorporate XML structural matching. Section 3 gives a brief overview of the Enhanced XML Schema (EXS), with its schema graph (EXS graph). This graph is used in the matching process for the measure of node context similarity. Section 4 presents and discuses our algorithms for structural contexts. Section 5 concludes the paper.

2 Related works

Schema matching is not a recent problem for the community of databases. Castano and De Antonellis (1999) developed the ARTEMIS system employ rules that compute the similarity between schemas as a weighted sum of similarities of elements names, data types, and structural position. With the growing use of XML, several matching tools take into consideration the hierarchical and deal essentially with DTDs. In the following, we present some examples of recent schema matching algorithms that incorporate XML structural matching. We do not present here of exhaustive manner all existing systems for schema matching, but those that appeared us interesting for the problematic that they raise or for the considered solutions.

2.1 Cupid

Cupid is a hybrid matcher combining several matching methods (Madhavan and al., 2001). It is intended to be generic across data models and has been applied to XML and relational data sources. Cupid is based on schema comparison without the use of instances. Despite these extensions, Cupid does not exploit all XML schema features such as substitution groups, abstract types, etc that could give a significant clue in solving XML schema matching problem.

2.2 LSD

The LSD (Learning Source Description) system (Doan and al., 2001) uses machine-learning techniques to match a new data source against a previously defined global schema. LSD is based on the combination of several match result obtained by independent learners.

This approach presents several limitations since it does not fully exploit XML structure. Besides, the only structural relationship considered within the LSD system is the parent-child relationship, which is not sufficient to describe the context of elements to matcher.

2.3 Similarity Flooding

In (Melnik and al., 2002), authors present a structure matching algorithm called Similarity Flooding (SF). The SF algorithm is implemented as part of a generic schema manipulation tool that supports, in addition to structural SF matcher, a name matcher, schema converters and a number of filters of choosing the best match candidates from the list of ranked map pairs returned by the SF algorithm. SF ignores all type of constraints while performing structural matching. Constraints like typing and integrity constraints are used at the end of the process to filter mapping pairs with the help of user.

2.4 SemInt

SemInt (Ly, 1994, 2000) represents a hybrid approach exploiting both schema and instance information to identify corresponding attributes between relational schemas. The schema-level constraints, such as data type and key constraints are derived from the DBMS catalog. Instance data are exploited to obtain further information, such as actual value distributions, numerical averages, etc. For each attribute, SemInt determines a signature consisting of values in the interval [0,1] for all involved matching criteria. The signatures are used first to cluster similar attributes from the first schema and then to find the best matching cluster for attributes from the second schema. The clustering and classification process is performed using neural networks with an automatic training, hereby limiting pre-match effort. The match result consists of clusters of similar attributes from both input schemas, leading to m:n local and global match cardinality.

3 Our data model

As we already mention in section 2, up to now few existent XML schema matching algorithms focus on structural matching exploiting all W3C XML schemas features (XML Schema, 2001). In this section, we propose an abstract model that serves as a foundation to represent conceptually W3C XML schemas and potentially other schema languages. We model XML schemas as a directed labelled graph with constraint sets; so-called schema graph. Schema graph consists of series of nodes that are connected to each other through directed labelled links. In addition, constraints can be defined over nodes and links. In (Zerdazi and Lamolle, 2005), we detail the proposed model for XML schemas in order to define a formal framework for solving matching problem. Figure 1 illustrates a schema graph example.

3.1 Enhanced XML Schema (EXS)

The first step in our integration methodology is the data sources representation in our data model which includes a minimal number of entities to manipulate but sufficient to translate schemas (Lamolle, 2003), (Lamolle, 2005). These entities are:

- *Concept :* a concept in the EXS schema is equivalent to the notion of entity in an ER diagram, a class in an object-oriented diagram or an element in the semi-structured data model. He can generate properties and/or include other concepts and having relations with some concepts ;
- *Relation :* two concepts are connected via a relationship. A relationship in EXS schema represents a nesting relationship. Each relationship has a degree and some constraints, and possesses also a predefined category and other metaknowledge ;
- *Property :* a property in the EXS schema is equivalent to the notion of attribute in the relational, object-oriented or the semi-structured data model. A property can be a property of a concept or property of a relationship.

Once this transformation made, the semantic part of the created *XSDi* is refined by the addition of metaknowledge (semantic modelling) (Zerdazi, 2005), which are deducted (for instance, the catalogue of data in the case of relational databases), or are specified by the expert of the domain. The semantic enrichment phase follows the phase of structural modelling of the EXS schemas. The semantic dimension of entities (concepts, relations, and properties) is enriched by the contribution of metaknowledge at the time of a survey more deepened of the entity state (structure, completeness, level of encapsulation, type of association, constraints on properties, cardinality, etc.). These metaknowledge are used at the time of the integration phase in order to get more precise correspondences between EXS schema. The semantic enrichment consists in spreading the structure of entities (concepts, relations, properties) via attributes and facets.

3.2 EXS schema graph

We model an EXS schema as a directed labelled graph with constraint sets. An EXS schema graph consists of series of nodes that are connected to each other through directed labelled links. In addition, constraints can be defined over nodes and links (Zerdazi, 2006). Figure 1 illustrates a schema graph example.

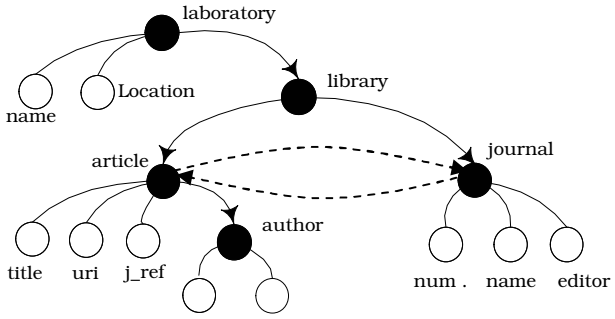


FIG. 1 – *An EXS schema graph example.*

3.2.1 Schema graph nodes

We categorize nodes into *atomic nodes* and *complex nodes*. Atomic nodes have no edges emanating from them. They are the leaf nodes in the schema graph. Complex nodes are the internal nodes in the schema graph. Each atomic node has a simple content, which is either an atomic value from the domain of basic data types (e.g., string, integer, date, etc.). The content of a complex node, called complex content, refers to some other nodes through directed labelled edges. In figure 1, nodes *laboratory* and *publication* are complex nodes, while nodes *name* and *location* are atomic nodes.

3.2.2 Schema graph edges

Each edge in the schema graph links two nodes capturing the structural aspects of XML schemas. We distinguish two kinds of edges: (i) *implicit edges* (e.g. the *parent/child* relationships between elements), they are depicted with a solid line edges in figure 1 and (ii) *explicit edges* defined in XML schema by means of *xs:key* and *xs:keyref* pairs or similar mechanisms.

They are represented using a pair of reverse parallel edges (generally bidirectional, specifying that both nodes are conceptually at the same level: *association relationship*). In figure 1, an implicit edge links the two nodes *laboratory* and *publication*. An explicit edge between *journal* and *article* specifies a key/keyref relation.

3.2.3 Schema graph constraints

Different constraints can be specified with XML Schema language. These constraints can be defined over both nodes and edges. Typical constraints over an edge are cardinality constraints. Cardinality constraints over a containment edge specify the cardinality of a child with respect to its parent. Cardinality constraints over an implicit edge imply for example an optional or mandatory attribute for a given node. The default cardinality specification is [1,1]. We also distinguish three kinds of constraints over a set of edges: (i) *ordered composition,* defined for a set of containment relationships and used for modelling XML Schema "sequences" and *all* mechanisms; (ii) *exclusive disjunction,* used for modelling the XML Schema *choice* and applied to containment edges; and (iii) *referential constraint,* used to model XML schema referential constraints. Referential constraints are applied to association edges. Other constraints are furthermore defined over nodes. Examples include *uniqueness* and *domain constraints.* Domain constraints are very broad. They essentially concern the content of atomic nodes. They can restrict the legal range of numerical values by giving the maximal/minimal values; limit the length of string values, or constraint the patterns of string values.

4 Structural matching algorithms

In this paper, we focus on understanding, modelling and formalising the problem of structural XML schema matching. The scope of this paper, in the context of our research, is indicated in figure 2.

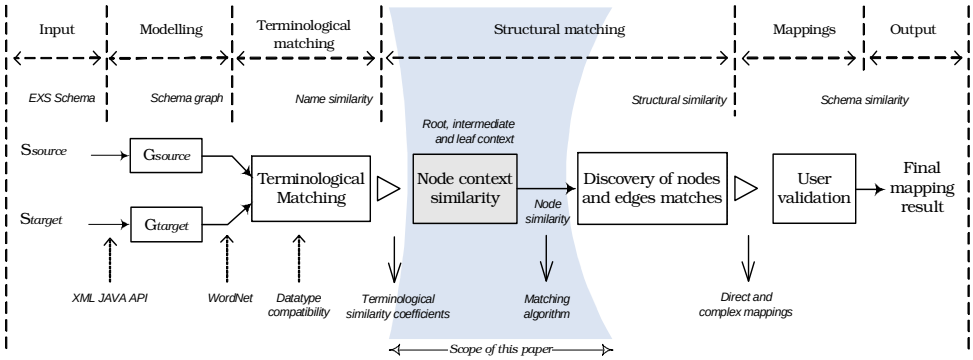


FIG. 2 – *The matching process.*

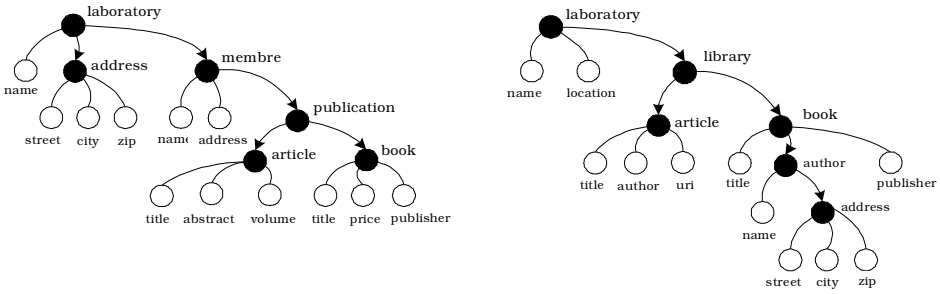
The first phase of our matching process concerns the **terminological matching (T.M)**. The aim of this phase is to compute the similarity between schema nodes based on the similarity of their labels. The proposed matching method takes as input a matrix of terminological similarity coefficients (ranging in [0,1]) between source and target nodes as well as semantic relationships. Terminological similarity uses WordNet (Miller, 1995), (Fellbum, 1998) as auxiliary information.

Techniques of terminological matching compare only nodes between a source schema and target schema. These matching techniques may provide incorrect match candidates. Structural matching is used to correct such match candidates based on their **structural context**. For example, assume that we let the schema graph in figure X be the source schema graph, denoted G_{source} and the target schema graph, denoted G_{target}.

Based on the two terminological matching and datatype compatibility techniques, we obtain a match between node *laboratory/address* (of G_{source}) and node *Author/Address* (of G_{target}), while the first is a laboratory address and the second is an author address. The structural matching (following phase in the matching process) compares the contexts in which nodes appear and can deduce that the two nodes *address* do not match, instead the node *address* in the source schema match the node *location* in the target schema and source relationship between *laboratory* and *address* match target relationship between *laboratory* and *location*. In this paper we only present our structural matching relies on the notion of *node context*.

4.1 Nodes context definition

The aim of structural matching is the comparison of the structural contexts in which nodes in the schema graph appear. Thus, we need a precise definition on what we mean by node context. We distinguish three kinds of node contexts depending on its position in the schema graph.



(a) A schema graph source *(Gsource)*

(b) A schema graph target *(Gcible)*

FIG. 3 – *EXS schema graphs (source and target).*

The root-context of a node n_i is defined by the *root path* having n_i as its ending node and the root of the schema tree as its starting node. For instance, the root-context of node *publication* in figure 3(a) is given by the path *laboratory/member/publication* which describes the publications of a member belonging to a laboratory. If the root-context of the root node is empty, it is assigned a *null* value.

The intermediate-context of node n_i includes its immediate subnodes. The intermediate-context of node reflects its basic structure and its local composition. For instance, the intermediate-context of node *Laboratory* of figure 3(a) is given by *(name, address, and member)*. The intermediate-context of an atomic node is assigned a *null* value.

The leaf-context of node n_i includes the leaves[1] of the subtree rooted at n_i. For instance, the leaf-context of node *Publication* in the schema graph of figure 3(a) is given by the set *(title, abstract, volume, title, price, and publisher)*. The leaf-context of an atomic node is assigned a *null* value.

Finally, the context of a node is defined as the union of its root-context, its intermediate-context and its leaf-context. Two nodes are structurally similar if they have similar contexts. The notion of context similarity has been used in Cupid and SF; however none of them relies on the three kinds of contexts. To measure the structural similarity between two nodes, we compute respectively the similarity of their root, intermediate and leaf contexts. In the following we describe the basis needed to compute such similarity.

4.2 Nodes context similarity

4.2.1 Root-context similarity

The root context similarity, *root_ctxSim* captures the similarity between two nodes based on their root context. Since the root context of a given node n_i is described by the root path (the path from the root to n_i), computing root contexts similarity is equivalent to comparing two paths. The *root_ctxSim* between two nodes n_1 and n_2 is given by the path resemblance measure between the two paths $(root_1, n_1)$ and $(root_2, n_2)$ weighted by the terminological similarity *(TSim)* between n_1 and n_2.:

[1] Leaves in the XML tree represent the atomic data that the schema describes.

$$\textbf{root\_ ctxSim } (n_1, n_2) \leftarrow \textbf{pathSim } ((root_1, n_1), (root_2, n_2)) \times \textbf{TSim}(n_1, n_2)$$

4.2.2 Intermediate-context similarity

The intermediate-context similarity *(inter_ctxSim)* is obtained by comparing nodes immediate descendents (children) sets including subtree. Given a node n_1 having m immediate children represented by the set $(n_{11}, \ldots, n_{1m})$ and node n_2 having k immediate children represented by $(n_{21}, \ldots, n_{2k})$. To compute the similarity between these two sets, we (i) compute the terminological similarity between each pair of children in the two sets, (ii) select the matching pairs with maximum similarity values and (iii) take the average of best similarity values. Algorithm 1 illustrates how we compute the intermediate-context similarity.

Algorithm 1 – INTERMEDIATE-CONTEXT SIMILARITY

1. ***Input*:** n_1, n_2, TSimMat // having respectively m and k children
2. ***Output*:** Inter_ctxSim
3. ***Begin***
4. inter_sim $\leftarrow$ MAX $_{i\in[1,m],\, j\in[1,k]}$ $\{(n_{1i}, n_{2j}, \text{TSim}) \in \text{TSimMat}\}$
5. sim_pairs $\leftarrow (n_{1i}, n_{2j}, \text{Sim\_Inter})$
6. inter_ctxSim $\leftarrow \dfrac{\sum_{(n1i,\, n2j,\, \text{inter\_sim}) \in \text{sim-pairs}} (\text{inter\_sim})}{\text{MAX}\,(m, k)}$
7. return inter_ctxSim
8. ***End.***

4.2.3 Leaf-context similarity

Since the effective content of a node is often captured by the leaf nodes of the subtree rooted at that node, we compute leaf context similarity of two nodes n_1 and n_2 by comparing their respective leaves sets, $n_1\{leaves\}$ and $n_2\{leaves\}$. Thus to compute the similarity between two leaves $l_1 \in n_1\{leaves\}$ and $l_2 \in n_2\{leaves\}$, we propose to compare the contexts in which these leaves appear. If a leaf node $l \in n_1\{leaves\}$, then the context of l is given by the path from n_1 to l. The context similarity of two leaves is then obtained by comparing such paths, and the similarity between two leaf nodes is obtained by comparing their context similarities and their terminological similarity. Algorithm 2 illustrates how we compute the leaf-context similarity.

Algorithm 2 – LEAF-CONTEXT SIMILARITY

1. ***Input:*** n_1, n_2 *// having respectively m and k leaves*
2. ***Output:*** leaf_ctxSim
3. ***Begin***
4. for each $l_{1i} \in n_1\{\text{leaves}\}$
5. for each $l_{2j} \in n_2\{\text{leaves}\}$
6. leaf_sim $(l_{1i}, l_{2j}) \leftarrow \text{pathSim}((n_1, l_{1i}), (n_2, l_{2j}) \times \text{TSim}(l_{1i}, l_{2j})$
7. temp_sim $\leftarrow$ MAX $_{i\in[1,m],\, j\in[1,k]}$ $(l_{1i}, l_{2j}, \text{leaf\_sim}\}$
8. sim_pairs $\leftarrow (n_{fk}, n_{fh}, \text{temp\_Sim})$
9. leaf_ctxSim $\leftarrow \dfrac{\sum_{(l1i,\, l2j,\, \text{leaf\_sim}) \in \text{sim-pairs}} (\text{leaf\_sim})}{\text{MAX}\,(m, k)}$

10. return leaf_ctxSim
11. ***End***

The goal of this measure of structural-context similarity, it easier to optimise and to automatically generate transformation scripts expressed in XSL language between EXS schemas.

5 Conclusions

In this paper we have interested on schema matching, and focused on structural context matching for enhanced XML schemas. We began by an analysis of problems involved in the matching, and we proposed a new solution taking into account of heterogeneity of the schema sources. For the structural similarity measure, we recovered a matrix of terminological similarity coefficients between schema nodes based on the similarity of their labels.

We outlined the limitations of current solutions through the study of Cupid and Similarity Flooding systems. Then we proposed a structural matching technique that considers the context of schemas nodes (defined by their roots, intermediates and leafs contexts in schema graph). By the way, we suggest a simple structural algorithm based on the previous ideas and exploit the three types of contexts. We refer to the result produced by the algorithm as a mapping. The user validates this mapping in order to produce a final mapping result that serves to generate transformation scripts.

For future work, we would like to improve the matching process, while taking into account the optimisation of the process in order to determine a set of semantic equivalences between schemas (source and target). That will facilitate the generation of operators based on the primitive of transformations between entities of EXS schemas. The second axis to land concerns the efficiency and the time of human interaction. The key is then to discover how to minimize user interaction but maximizing the impact of the feedback.

References

Abiteboul, S., Cluet, S., Milo, T., 1997. Correspondence and Translation for heterogeneous data. In *Proceeding of The international Conference on Database Theory (ICDT)*. 351-363.

Boukottaya, A., Vanoirbeek, C., Paganelli, F., Abou-Khaled, O., 2004. Automating XML documents transformations: a conceptual modelling based approach. *In Proceedings of the first Asian-Pacific conference on Conceptual modelling*. ACM, 81-90.

Castano, S., De Antonellis, V., 1999. A schema analysis and Reconciliation Tool Environment For Heterogeneous Databases. In *Proceedings of International Database Engineering and Applications Symposium*.

Doan, A., Madhavan, J., Domingos, P., Halevey, A., 2001. Reconciling schemas of disparate data sources: A machine Learning Approach. In *Proceedings ACM SIGMOD conference*. 509-520.

Drew, P., King, R., McLeod, D., Rusinkiewicz, M., Silberschatz, A., 1993. Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems. In *Proceedings ACM SIGMOD record*, 47-56.

Fellbum, C., 1998. WordNet: An Electronic Lexical Database. *MIT press*.

Lamolle, M., Mellouli, N., 2003. Intégration de bases de données hétérogènes via XML, *EGC'2003.*

Lamolle, M., Zerdazi, A., 2005. Intégration de Bases de données hétérogènes par une modélisation conceptuelle XML, *COSI'05.* 216-227.

Li, W.S., Clifton, C., 1994, Semantic Integration in Heterogeneous Databases Using Neural Networks. *VLDB.*

Li, W.S., Clifton C., 2000, SemInt: A Tool for Identifying Attribute Correspondences in Heterogeneous Databases Using Neural Network. Data and Knowledge Engineering. 49-84.

Madhavan, J., Bernstein, P., Rahm, E., 2001. Generic schema matching with cupid. *VLDB*

Melnik, S., Garcia-Molina, H., Rahm, E., 2002. Similarity Flooding: A versatile Graph Matching and its Application to Schema Matching. *Data Engineering.*

Miller, A.G., 1995. WordNet: A lexical Database for English. *ACM.* 39-41.

Miller, A.G., Hass, L., Hernandez, M.A., 2000. Schema mapping as query discovery. *VLDB*. 77-88.

Rahm, E., Bernstein, P., 2001 A survey of approaches to automatic schema matching. In *VLDB Journal*. 334-350.

XML Schema, W3C Recommendation, 2001. XML-Schema Primer, W3 Consortium, 2001. *Available at http://www.w3.org/TR /xmlschema-0.*

Zerdazi, A., Lamolle, M., 2005. Modélisation des schémas XML par adjonction de métaconnaissances sémantiques. *ASTI'05.* 29-32.

Zerdazi, A., Lamolle, M., 2006. Intégration de sources hétérogènes par matching semi-automatique de schémas XML étendus. *INFORSID'2006*. 991-1006.

Résumé

L'appariement de schémas est une étape critique lors de l'intégration de sources de données hétérogènes. Les récents travaux de recherche sur l'intégration se sont focalisés sur le développement de techniques d'appariement pour trouver les éléments équivalents dans divers schémas XML à comparer. Dans cet article, nous proposons une nouvelle approche de mesure de similarité sémantique basée sur le calcul de trois contextes différents entre des noeuds de schémas XML étendus (dits EXS). La première étape consiste à transformer et ajouter des métaconnaissances aux sources de données sous la forme de schémas EXS. Puis, nous appliquons un algorithme d'appariement sémantique qui, à partir de deux graphes de schémas en entrée, produit en sortie un mapping entre les nœuds considérés similaires dans les deux graphes. Une fois ce travail réalisé, le mapping est contrôlé et ajusté manuellement.

Fusion automatique des ontologies par classification hiérarchique pour la conception d'un entrepôt de données

Nora Maiz*, Omar Boussaid**
Fadila Bentayeb*
Laboratoire ERIC, University Lumière Lyon2
5, avenue Pierre Mendès France
69676, Bron Cedex, France
http://eric.univ-lyon2.fr/

* nmaiz, bentayeb@eric.univ-lyon2.fr,
** omar.boussaid@univ-lyon2.fr

Résumé. Dans cet article, nous présentons une nouvelle approche de fusion des ontologies-OWL par l'utilisation de la technique de classification hiérarchique. Ce travail s'insère dans le cadre de l'entreposage de données par médiation pour la construction des contextes d'analyse à la volée afin de faire l'analyse en ligne. Notre approche est constituée de quatre étapes : la première consiste à utiliser un algorithme de classification hiérarchique pour construire des classes de concepts synonymes. Pour cela, nous définissons une mesure de similarité entre concepts. Cette étape permet de résoudre les conflits de synonymie et d'homomymie. Ensuite, nous construisons pour chaque ontologie, l'ensemble des paires de concepts dont la première composante de la paire subsume la deuxième et nous fusionnons ces ensembles. Dans la troisième étape, nous utilisons les classes des synonymes trouvées dans la première étape avec l'ensemble des paires de concepts de la deuxème étape pour résoudre les conflits sémantiques entre les classes dans ce dernier. La transformation de l'ensemble des paires de concepts en un arbre est fait dans la quatrième étape.

1 Introduction

les outils d'aide à la décision sont de plus en plus utilisés dans les entreprises modernes. Ces dernières manipulent des donnèes stokées dans des sources réparties et hétérogènes. L'intégration de ces données est devenue une nécessité cruciale pour pouvoir construire des contextes d'analyse nommés cube de données, qui sont modélisés comme une vue multidimentionnelle sur les données distribuèes dans des différentes sources. En effet, il existe deux méthodes parallèlles pour l'intégration de données : l'entrepôt (Inmon, 1992; Kimball, 1998) et le médiateur (Goasdoué et al., 2000; Lamarre et al., 2004; Huang et al., 2000). La première consiste à construire une base de données réelle et centralisée, selon un schéma particulier. Celle-ci contient les données intègrèes à partir des différentes sources de données et elle est prête à supporter le processus d'analyse en ligne (OLAP). Cette approche est caractérisée par sa per-

formance en termes de temps de réponse des requêtes mais elle présente un ensemble d'inconvénients. L'inconvénient majeur de cette approche est le problème de mise à jour, du dait que les données manipulées changent souvent. Ces changements doivent être propagés dans l'entrepôt ce qui implique l'utlisation des techniques de rafraichissement qui engendre un coût de maintenance supplémentaire pour l'entrepôt.

Pour remédier à ces problèmes, nous proposons d'utiliser la deuxième approche qui est basée sur la médiation le médiateur pour modéliser un entrepôt de données et construire le cube de données à la volée. L'approche médiateur consiste à définir trois éléments : les schémas des sources locales, le schéma global et les correspondances entre le schéma global et les schémas locaux. Comme nous sommes dans le domaine décisionnel, nous sommes intéressés beaucoup plus par la pertinence des données résultantes de la consultation des sources à travers le médiateur par des requêtes basées sur le schéma global et éventuellement les schémas locaux. Une simple recherche par des mots clés n'est donc pas suffisante. Il faut procéder à une recherche basée sur la sémantique du vocabulaire utilisé dans le schéma global et la requête.

Pour cela, nous utilisons les ontologies comme support de représentation de la sémantique et de partage de la connaissance entre plusieurs utilisateurs ou bien entre les différentes sources de données. Ces dernières sont décrites par leurs propres sources sémantiques (ontologies). Nous proposons d'utiliser ces sources sémantiques pour concevoir le schéma global. Dans ce cadre, nous proposons une méthode de fusion des ontologies locales, afin de construire une ontologie globale qui contient toutes les connaissances réparties dans les différentes ontologies locales. Notre méthode est réalisée en plusieurs étapes qui sont basées sur une technique de classification hiérarchique pour la découverte de connaissances implicites dans les concepts des ontologies et la construction des classes des concepts synonymes. Nous définissons pour la classification une mesure de similarité qui prend en compte la terminologie, la structure et la sémantique du concept. Les classes des concepts vont être utilisées par la suite pour résoudre les conflits sémantiques entre les concepts afin de construire l'ontologie globale.

Nous avons organisé notre article comme suit : la section 2 présente les travaux existant dans le domaine de fusion des ontologies. La section 3 dèfinit des notions de base utilisèes dans l'article. La section 4 présente notre approche de fusion avec ses différentes étapes. La conclusion et les perspectives sont discutées dans la section 5.

2 Etat de l'art

Dans cette section, nous allons présenter les approches principales de fusion d'ontologies existantes. Au fait, peu de méthodes de fusion on été proposées. Nous citons par exemple Anchor-PROMPT (Noy et Musen, 2003) et FCA-MERGE (Stumme et Maedche, 2001). Dans la première, les auteurs proposent une méthode de comparaison des ontologies de domaine par la définition des correspondances entre leurs concepts. Ils choisissent deux paires de concepts équivalents comme référence. Chaque paire appartient à une ontologie. Ensuite, ils sélectionnent tous les concepts intermédiaires deux à deux qui occupent les mêmes positions dans deux chemins de même longueur, reliant les deux concepts de la même paire. Cela permet aux auteurs de juger si ces paires de concepts sont équivalents ou pas. Selon les auteurs, deux concepts se trouvant dans la même position entre deux concepts équivalents, sont également équivalents. Anchors-Prompt suppose que les deux ontologies sont construites de la même façon. Ce n'est pas le cas dans la réalité. Dans FCA-Merge, les auteurs définissent une méthode

formelle et ascendante de fusion des ontologies en se basant sur un ensemble de documents. Ils appliquent des techniques de traitement du langage naturel et d'analyse formel de concepts pour dériver le treillis des concepts. Ce dernier est exploré et transformé en une ontologie par l'intervention de l'être humain.

Par contre, plusieurs travaux sur l'alignement des ontologies ont été développés. Ils traitent une étape du processus de fusion qui est la découverte des cerrespondances entre les entité des ontologies à aligner. Quelques uns comme Aleksovski et al. (2006) ont montré l'avantage d'utiliser la connaissance du domaine dans des cas définis. D'autres approches comme ASCO (Bach et al., 2004), GLUE (Doan et al., 2004), QOM (Ehrig et Staab, 2004) et OLA (Euzenat et al., 2005) ont été développées pour supporter le processus d'alignement des ontologies. OLA par exemple décrit les ontologies comme deux graphes-OWL et utilise la mesure de similarité de Valtchev (1999) pour comparer les entités appartenant à la même catégorie (propriété, instance). Cet algorithme est conçu pour l'alignement des ontologies.

Les approches précédentes utilisent des ontologies en format XML *(Extensible Markup Language)*, RDF *(Resource Description Framework)* ou bien OWL-Lite *(Ontology Web Language)*. D'un autre côté, la mjorité d'entre elles utilisent des mesures de similarité qui couvrent plus au moins toute la structure des ontologies à aligner.

Les approches de fusion et d'alignement précédentes utilisent un seuil de stabilisation pour arrêter le processus d'alignement. Ce qui limite la propagation de la similarité et par conséquent réduit la précision de la méthode. En plus, ces méthodes sont construites pour aligner deux ontologies. En réalité, il existe plusieurs ontologies qui décrivent le même domaine et qui nécessitent d'être alignées pour devenir réutilisables, d'oû la nécessité de proposer une méthode permettant le passage à l'échelle pour supporter plusieurs ontologies à la fois. C'est le cas de notre approche que nous détaillons dans la suite.

3 Définitions et concepts de base

Ontologie

Le concept d'ontologie peut avoir plusieurs définitions selon le type de l'ontologie et son utilisation. Dans notre cas, nous définissons une ontologie comme un triplet (C, R, I), où Cest l'ensemble des concepts de l'ontologie, R est l'ensemble des relations entre les concepts et I est l'ensemble des instances.

Concept

Un concept est l'abstration d'une réalité. Il est défini par un vecteur d'attributs $V_i = (T_i, P_1, ..., P_k, R_1, ..., R_j)$ tel que T_i est le terme qui décrit le concept, les P_i représentent les propriétés du concept et les R_i sont les relations du concept avec les concepts voisins. Ces différents attributs vont être utilisés pour comparer la sémantique des différents concepts.

Mesure de similarité

Le calcul de la similarité entre deux concepts permet de déterminer s'ils sont équivalents ou indépendants sémantiquement. Cette mesure est basée sur la terminologie du concept, ses propriétés et ses relations avec son voisinage. En fait, deux concepts qui possèdent la même

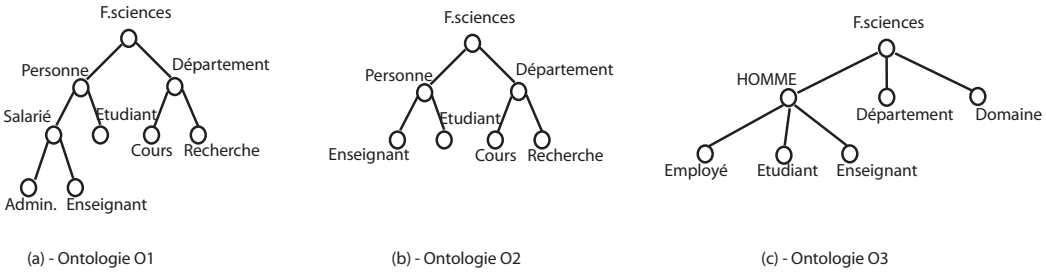


FIG. 1 – *Exemple de trois ontologies du même domaine.*

terminologie et les mêmes propriétés et s'ils ont des relations identiques avec des voisins similaires, il y a une forte chance qu'ils soient identiques. Pour calculer la similarité entre deux concepts, on doit d'abord calculer la similaritéentre les différents paires des attributs $Attribut_i$ et$Attribut_j$, le premier attribut appartient au premier concept et le deuxième appartient au deuxième concept.

La similarité entre deux attributs $Attribut_i$ et$Attribut_j$ notée Sim($Attribut_i$, $Attribut_j$), est une similarité terminologique, on peut utiliser le thésaurus *WordNet*[1] (Miller, 1995) pour la calculer. Le principe du calcul de similarité avec *WordNet* est montré dans l'algorithme **??**. Pour trouver la similarité entre deux termes (attributs) At_1 et At_2, on doit procéder à une recherche en largeur à partir de l'ensemble *Synset*[2] de At_1, jusqu'aux *Synsets* du *Synset* de At_2, et ainsi de suite, jusqu'à ce que At_2 soit trouvé. Si le terme recherché n'est pas trouvé, alors l'algorithme retourne 0. Sinon, la similarité est définie par 0.8^{depth}. Une fois que les similarité des attributs sont calculées, on doit fixer un seuil afin de ne garder que les paires d'attributs qui sont similaires et qui vont rentrer dans le calcul de la similarité des deux concepts considérés. On définit l'ensemble *A* comme l'ensemble de toutes les paires d'attributs sélectionnés. La mesure de similarité entre deux concepts C_i et C_j est définie ainsi :

$$GSim(C_i, C_j) = \sum_{(k=1,\ldots,Card(A))} \Pi_{i_k}\ Sim(Attribut_{i_k}, Attributs_{j_m})$$

Tel que $Attribut_{i_k}$ ($Attributs_{j_m}$) est le k^{ieme} attribut du concept C_i (C_j) et qui peut être un terme, une propriété ou une relation ; Π_{i_k} est le poids du k^{ieme} attribut fixé par l'utilisateur dès le départ.

4 Présentation de la démarche de fusion automatique des ontologies

Dans cette section, nous allons présenter notre méthode de fusion automatique des ontologies (voir figure 2. Elle est constituée de quatre étapes principales :

[1] http ://wordnet.princeton.edu/.

[2] *Synset* est l'ensemble des synonymes d'un mot donné

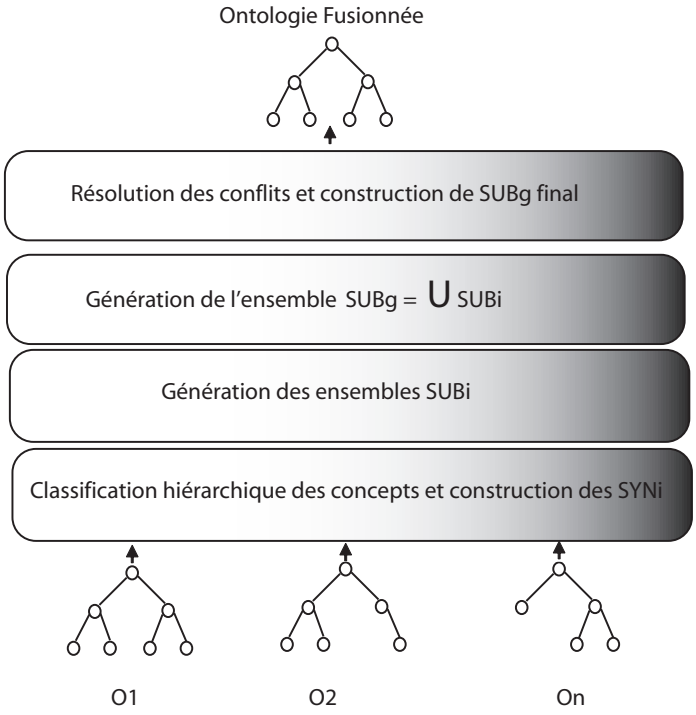


FIG. 2 – *Schéma général de l'approche de fusion d'ontologies.*

1. Classification hiérarchique des concepts ;
2. Construction de l'ensemble des paires *(Père, Fils)* SUB_g des différentes ontologies ;
3. Utilisation des classes SYN_i pour générer l'ensemble des paires *(Père, Fils)* SUB_g ;
4. Construction de l'ontologie fusionnée.

La première étape utilise un algorithme de classification hiérarchique afin de classifier l'ensemble de tous les concepts des différentes ontologies à fusionner. À la sortie de l'algorithme, nous obtenons *N* classes SYN_i où chacune représente les concepts synonymes appartenant aux différentes ontologies. Cette étape a comme objectif de trouver tous les synonymes dans les différentes ontologies des sources locales. Pour effectuer cette classification, nous aurons besoin d'une mesure de similarité ou distance pour pouvoir calculer la similarité entre chaque paire de concepts.

Après la construction des différentes classes SYN_i, la deuxième étape, consiste à générer l'ensemble *SUB* qui est défini comme l'ensemble de toutes les paires *(Père, fils)* des différentes ontologies. Pour cela, nous commençons par la génération de chaque ensemble SUB_i, correspondant à une ontologie O_i. Ensuite, nous fusionnons les SUB_i pour avoir l'ensemble *SUB* total. L'objectif de construire l'ensemble *SUB* est de garder la hiérarchie des différentes ontologies pour pouvoir déduire la hiérarchie de l'ontologie fusionnée dans les étapes suivantes.

La troisième étape consiste à utiliser les classes SYN_i de la première étape pour construire l'ensemble *SUB* de la deuxième étape. Nous remplaçons chaque concept dans l'ensemble *SUB* par le concept général correspondant à la classe qui le contienne. Par conséquent, nous obtenons des paires de concepts redendants dans l'ensemble *SUB*. Nous procédons donc à enlever la redondance par la suppression des différentes copies d'une même paire de concepts et ne garder qu'une seule. À la fin de cette étape, nous obtenons l'ensemble *SUB* qui contient les concepts généraux ainsi que la hiérarchie de l'ontologie globale fusionnée.

L'étape suivante utilise directement l'ensemble *SUB* pour construire l'ontologie globale. Dans la suite, nous détaillerons chaque étape du processus de fusion des ontologies. Pour cela, nous commençons d'abord par la définition d'une ontologie et la structure du concept que nous adoptons.

4.1 Classification hiérarchique des concepts

Dans cette section, nous allons détailler le processus de classification hiérarchique des concepts appartenants aux différentes ontologies à fusionner. La tâche de classification dans notre travail a comme objectif de construire un ensemble de *M* classes disjointes de concepts dont chacune ne contient que des concepts synonymes appartenant aux différentes ontologies. Pour cela, nous appliquons un algorithme inspiré de la définition de la classification hiérarchique qui utilise notre mesure pour calculer la similarité entre les différents paires de concepts. Cette étape contient deux phases :

Application de l'algorithme de classification : l'algorithme de classification utilise une matrice de similarités dont la première ligne et la première colonne de la matrice reprèsentent tous les concepts des différentes ontologies. Les autres cases de la matrice représentent les similarités entre les différentes paires de concepts. Ensuite, en se basant sur la matrice de similarité, l'algorithme sélectionne d'abord la paire de concepts dont la similarité est maximale et construit une première classe qui va contenir ces deux concepts sémantiquement équivalents. Dans l'itération suivante, l'algorithme va considérer la classe construite dans l'itération précédente comme étant un seul individu et va calculer de nouveau sa similarité avec les autres concepts. La similarité entre une classe SYN_i contenant les éléments $(C_1..C_i)$, et un individu C_j est définie comme suit :

$$Sim(SYN_i, C_j) = Min(Sim(C_1, C_j), ..., Sim(C_i, C_j)$$

L'algorithme continue à itérer jusqu'à ce qu'il obtienne une classe qui contient tous les concepts. Ensuite, on fait la meilleure coupe ou bien on fixe un seuil de similarité entre classes pour que l'algorithme s'arrête. Le résultat de cette étape est un ensemble de classe nomées SYN_i dont chacune contient tous les concepts synonymes appartenant aux ontologies différentes.

Exemple : Considérons les trois ontologies montrées dans la figure 1, Ces ontologies représentent le même domaine sauf qu'elles sont définies de manières différentes. Pour fusionner ces ontologies on doit d'abord, extraire l'ensemble C.

$C = \{F.sciences, Personne, Dpartement, Salari, Etudiant, Cours, Recherche, Admin, Enseignant, F.sciences, personne, Dpartement, Etudiant, Cours, Recherche, Enseignant, F.sciences, HOMME, Dpartement, Employ, Etudiant, Cours, Recherche, Domaine, Enseignant\}$.

Après l'application de l'algorithme de classification sur cette population, nous obtenons l'ensemble des SYN_i suivant :

$$SYN = \bigcup SYN_i$$

$SYN = \{\{F.Sciences, F.sciences, F.sciences\}, \{Personne, Personne, HOMME\}, \{Dpartement, Dpartement, Dpartement\}, \{Etudiant, Etudiant, Etudiant\}, \{Enseignant, Enseignant, Enseignant\}, \{Employ, Salari\}, \{Cours, Cours\}, \{Admin\}, \{Domaine\}\}$

Ensuite, nous généralisons chaque sous ensemble *Fusion des concepts de chaque classe* SYN_i.

La tâche suivante après la classification, consiste à attribuer à chaque classe un nom de concept représentatif de ses différents éléments et qui peut être l'un des objets de la classe considérée. À chaque fois qu'on affecte un nom de concept à une classe, on vérifie qu'il n'est pas déjà affecté précédemment à une autre classe. Si c'est le cas, on le change pour la classe en cours. Ainsi, en plus de la résolution du problème des conflits sémantiques en termes de synonymie, on résoud le problème des conflits sémantiques en terme d'antonymie et de définir des tables de correspondances pour garder celles entre le nouveau mot et tous les éléments de la classe. Cela sera utile pour des tâches antérieures comme la réécriture des requêtes par exemple.

Exemple : Une fois qu'on a construit l'ensemble SYN qui contient des sous ensembles des concepts synonymes SYN_i, nous remplaçons chaque SYN_i par un autre concept. Ce dernier, peut être l'un des concepts de l'ensemble SYN_i. Dans notre exemple, l'ensemble SYN devient donc :

$SYN = \{F.Sciences, Personne, Dpartement, Etudiant, Enseignant, Salari, Cours, Admin, Domaine\}$

Les attributs d'un nouveau concept C_g, qui généralise un ensemble SYN_i, est l'union des attributs de tous les concepts C_i appartenant à SYN_i. On garde le lien entre le concept C_g et les concepts C_i de SYN_i dans une table de correspondances que nous utilisons dans les étapes suivantes.

4.2 Construction de l'ensemble SUB_g

Après la construction des classes de synonymes SYN_i, on entamme la troisième étape qui consiste à construire, à partir des hiérarchies des ontologies à fusionner, l'ensemble des paires

$(C_i, C_j) \in O_i,\ (i = 1..P)$ (P est le nombre d'ontologies) dont C_i est le père de C_j dans la hiérarchie de l'ontologie. La construction de cet ensemble, et l'utilisation des ensembles des synonymes construits précédemment va nous donner la structure de l'ontologie fusionnée. Pour construire l'ensemble SUB, nous procédons en deux temps :

1. Gènèration des ensembles SUB_i : La première phase consiste à définir les ensembles $SUB_i,\ (i = 1, ..., p)$, où chaque SUB_i correspond à une ontologie O_i. La détermination des ensembles SUB_i se fait par un simple parcours d'hiérarchies des différentes ontologies, et à chaque noeud, on le prend avec son fils et ainsi de suite. À la fin de cette phase, on obtient P ensembles SUB_i où chacun correspond à une ontologie O_i. Ces ensembles SUB_i contiennent des conflits sémantiques qu'on va résoudre en utilisant les résultats de la première étape qui sont les SYN_i lors de la quatrième étape. Une fois que nous avons construit les SUB_i, nous abordons la deuxième phase de cette ètape.

   Exemple : Dans notre exemple, les trois ensembles SUB_1, SUB_2 et SUB_3 correspondant aux trois ontologies sont ainsi :

   $SUB_1 = \{(F.sciences, Personne), (F.sciences, D\acute{e}partement), (Personne, Salarie\acute{}), (Personne, Etudiant), (D\acute{e}partement, Cours), (D\acute{e}partement, Recherche), (Salarie\acute{}, Admin.), (Salarie\acute{}, Enseignant)\}$

   $SUB_2 = \{(F.sciences, Personne), (F.sciences, D\acute{e}partement), (Personne, Enseignant), (Personne, Etudiant), (D\acute{e}partement, Cours), (D\acute{e}partement, Recherche)\}$

   $SUB_3 = \{(F.sciences, Homme), (F.sciences, D\acute{e}partement), (F.sciences, Domaine), (Homme, Employe\acute{}), (Homme, Etudiant), (Homme, Enseignant)\}$

2. Fusion des ensembles des paires SUB_i :

   Nous prenons les ensembles SUB_i construits lors de la phase précédente. On procède à les fusionner pour avoir un seul ensemble SUB qui contient toustes les paires des concepts (Père, Fils) de toutes les ontologies candidates pour le processus de fusion. L'ensemble SUB est défini donc comme suit :

   $$SUB_g = \bigcup_{i=1,...,P} SUB_i$$

   l'opération d'union définie dans la formule ci-dessus est l'union classique des ensembles qui ne garde qu'une seule occurence de chaque élément. Sauf que dans notre cas, ce n'est pas possible de comparer ces éléments et de trouver la similarité entre deux couples de concepts. On va trouver donc dans l'ensemble SUB_g toutes les paires appartenant aux différents ensembles SUB_i avec redondance. Pour pouvoir détecter ces dernières, on utilise les classe SYN_i, définies précédemment, pour pouvoir distinguer les paires similaires (équivantentes) et du coup éliminer la redondance. C'est le rôle de l'étape qui suit.

Exemple : L'union des trois ensembles SUB_i précédents est l'ensemble SUB_g défini comme suit :

$SUB_g = \{(F.sciences, Personne), (F.sciences, De^{'}partement), (Personne, Salarie^{'}), (Personne, Etudiant), (De^{'}partement, Cours), (De^{'}partement, Recherche), (Salarie^{'}, Admin.), (Salarie^{'}, Enseignant), (F.sciences, Personne), (F.sciences, De^{'}partement), (Personne, Enseignant), (Personne, Etudiant), (De^{'}partement, Cours), (De^{'}partement, Recherche), (F.sciences, HOMME), (F.sciences, De^{'}partement), (F.sciences, Domaine), (Homme, Employe^{'}), (HOMME, Etudiant), (HOMME, Enseignant)\}$.

4.3 Utilisation des classes SUN_i pour générer l'ensemble SUB_g

Comme nous avons expliqué précédemment, l'ensemble SUB_g contient des structures redondantes à éliminer. Pour pouvoir éliminer cette redondance, Nous utilisons donc notre réservoir de connaissances extraites de la population des concepts des différentes ontologies. L'extraction de ces connaissances se fait par l'application de l'algorithme de classification de concepts selon leur rapprochement sémantique. L'objectif de cette classification est de faire apparaitre les structures similaires dans cet ensemble. Pour cela nous procédons dans deux phases :

Remplacer les concepts dans SUB_g par leur généralisant :

Dans l'ensemble SUB, nous parcourons les paires de concepts, paire par paire, et pour chaque composant de la paire en cours, nous cherchons la classe à laquelle ce composant appartient. Une fois la classe correspondante est trouvée, nous remplacons le composant de la paire par le nom de la classe. On réitère jusqu'à ce qu'on ait parcouru toutes les paires de l'ensemble SUB. A la fin de cette phase, nous obtenons un ensemble SUB des paires de concepts qu'on peut comparer entre elles.

Exemple : L'ensemble SUB_g dans nore exemple devient ainsi :

$SUB_g = \{(F.sciences, Personne), (F.sciences, De^{'}partement), (Personne, Salarie^{'}), (Personne, Etudiant), (De^{'}partement, Cours), (De^{'}partement, Recherche), (Salarie^{'}, Admin.), (Salarie^{'}, Enseignant), (F.sciences, Personne), (F.sciences, De^{'}partement), (Personne, Enseignant), (Personne, Etudiant), (De^{'}partement, Cours), (De^{'}partement, Recherche), (F.sciences, Personne), (F.sciences, De^{'}partement), (F.sciences, Domaine), (Personne, Salarie^{'}), (Personne, Etudiant), (Personne, Enseignant)\}$

Supprimer les redandances Cette phase consiste à parcourir l'ensemble SUB_g et à comparer les paires de concepts deux à deux. Les paires similaires redondantes vont être supprimées afin de ne garder qu'une seule paire dans l'ensemble SUB_g et qui va servir à construire l'ontologie fusionnée.

Exemple : Après la suppression des éléments redondants dans l'ensemble SUB_g, nous obtenons :

$SUB_g = \{(F.sciences, Personne), (F.sciences, D\acute{e}partement), (Personne, Salarie\acute{}), (Personne, Etudiant), (D\acute{e}partement, Cours), (D\acute{e}partement, Recherche), (Salarie\acute{}, Admin.), (Salarie\acute{}, Enseignant), (Personne, Enseignant), (F.sciences, Domaine)\}$

4.4 Construction de l'ontologie fusionnée

Dans cette phase, nous utilisons l'ensemble SUB_g, construit auparavant et on construit l'arbre défini par les différentes paires de concepts dans SUB_g. Pour la construction de l'arbre, nous commençons par parcourir l'ensemble SUB_g jusqu'à trouver le concept qui n'est pas *fils* dans aucune paire de concepts. C'est ce concept-là qui représente la racine de l'arbre. La deuxiéme composante de la paire qui contient le concept racine va être le *fils* direct de la racine dans l'arbre. Cette paire va être marquée pour ne pas être parcourue une prochaine fois. On cherche ensuie s'il y a une autre paire contennant la racine comme première composante. Si c'est le cas, la deuxième composante de cette paire va représenter le deuxième *fils* du concept racine et on continnue ainsi jusqu'à qu'il n'y ait plus de paires contennant la racine. Ensuite, on prend le premier *fils* de la racine et on fait la même chose pour chercher ses *fils* à partir des paires des concepts dans l'ensemble SUB_g. On continnue de la même façon jusqu'à marquer toutes les paires de concepts dans l'ensemble SUB_g. le résultat de cette étape sera alors, l'ontologie fusionnée.

5 Conclusion et perspectives

Dans cet article, nous avons présenté notre travail sur la fusion automatique des ontologies et cela dans le cadre de l'entreposage virtuel des données réparties et hétérogènes. Notre approche est constituée de quatre étapes : la première consiste à appliquer un algorithme de classification hiérarchique pour pouvoir construire des classes des concepts synonymes. Pour cela, nous définissons une mesure de similarité qui prend en compte la terminologie, la structure et la sémantique du concept. Après avoir construit toutes les classes possibles, nous généralisons chaque classe par un nouveau concept. Nous construisons dans la deuxième étape pour chaque ontologies, l'ensemble des paires de concepts dont la première composante de la paire subsume la deuxième. Ensuite, nous fusionnons ces ensembles. Dans la troisième étape, nous utilisons les classes des synonymes trouvées dans la première étape avec l'ensemble des paires de concepts SUB_g pour pouvoir résoudre les conflits sémantiques dans ce dernier. La transformation de l'ensemble des paires SUB_g en un arbre est fait dans la quatrième étape.

Dans un travail proche, nous envisageons dans un premier temps, de compléter l'implémentation de l'approche sur un cas réel pour pouvoir comparer sa précision avec les approches existantes. Ensuite nous, envisageons á considérer des cas plus complexes là où l'union de deux concepts par exemple peut être équivalente à un autre concept. Ensuite, inclure le pouvoir de raisonnement de OWL-DL dans le calcul de la similarité et la résolution des conflits sémantiques. Dans un deuxième temps, adapter l'ontologie globale fusionnée pour supporter

des requêtes décisionnelles. Ensuite, définir la stratégie de réécriture de ces requêtes afin d'accéder aux données des sources locales. Les résultats des requêtes vont être utilisés dans la suite pour construire les cubes de données à la volée.

Références

Aleksovski, Z., M. Klein, W. ten Katen, et F. van Harmelen (2006). Matching unstructured vocabularies using a background ontology. In *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW'06)*, Lecture Notes in Artificial Intelligence, pp. 182–197. Springer-Verlag.

Bach, T. L., R. Dieng-Kuntz, et F. Gandon (2004). On ontology matching problems - for building a corporate semantic web in a multi-communities organization. In *ICEIS (4)*, pp. 236–243.

Doan, A., J. Madhavan, P. Domingos, et A. Y. Halevy (2004). Ontology matching : A machine learning approach. In *Handbook on Ontologies*, International Handbooks on Information Systems, pp. 385–404. Springer.

Ehrig, M. et S. Staab (2004). Qom - quick ontology mapping. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004), GI Jahrestagung (1), Hiroshima, Japan*, pp. 356–361.

Euzenat, J., P. Guégan, et P. Valtchev (2005). Ola in the oaei 2005 alignment contest. In *Integrating Ontologies*.

Goasdoué, F., V. Lattès, et M.-C. Rousset (2000). The use of carin language and algorithms for information integration : The picsel system. *Int. J. Cooperative Inf. Syst. 9*(4), 383–401.

Huang, H.-C., J. M. Kerridge, et S.-L. Chen (2000). A query mediation approach to interoperability of heterogeneous databases. In *Australasian Database Conference*, pp. 41–48.

Inmon, W. H. (1992). *Building the Data Warehouse*. New York, NY, USA : John Wiley & Sons, Inc.

Kimball, R. (1998). The operational data warehouse. *DBMS 11*(1), 14–16.

Lamarre, P., S. Cazalens, S. Lemp, et P. Valduriez (2004). A flexible mediation process for large distributed information systems. In *CoopIS/DOA/ODBASE (1)*, Volume 3290 of *Lecture Notes in Computer Science*, pp. 19–36. Springer.

Miller, G. A. (1995). Wordnet : A lexical database for english. *Commun. ACM 38*(11), 39–41.

Noy, N. F. et M. A. Musen (2003). The prompt suite : interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud. 59*(6), 983–1024.

Stumme, G. et A. Maedche (2001). FCA-MERGE : Bottom-up merging of ontologies. In *IJCAI*, pp. 225–234.

Valtchev, P. (1999). *Construction automatique de taxonomies pour l'aide à la représentation de connaissance par objets*. Thèse de doctorat, Université de Grenoble 1.

Enrichissement sémantique de requêtes utilisant un ordre sur les concepts

Anthony Ventresque*, Sylvie Cazalens*, Philippe Lamarre* et Patrick Valduriez**

*Laboratoire d'Informatique de Nantes Atlantique (LINA)
2 rue de la Houssinière, 44322 Nantes
Prenom.Nom@univ-nantes.fr,
**INRIA et LINA
2 rue de la Houssinière, 44322 Nantes
Patrick.Valduriez@inria.fr,

Résumé. L'interopérabilité sémantique dans les systèmes distribués est assez problématique : les matchings sont partiels et l'hétérogénéité demeure souvent. Nous essayons de dépasser cette hétérogénéité en exprimant requêtes et documents sur les parties "communes" entre ontologies mais en tenant compte des différences. Côté utilisateur posant une requête, nous mettons en place une expansion de requête, et côté fournisseur de document, une interprétation de la requête. Lors des deux étapes, nous avons besoin de classer les concepts des ontologies grâce à une mesure de similarité sémantique. Parmi les mesures disponibles, nous avons remarqué que beaucoup satisfont l'inégalité triangulaire, et un certain nombre la symétrie. Cela nous semblait étonnant pour notre travail, et effectivement, des travaux en psychologie nous ont donné raison. Nous avons donc choisi une mesure qui nous satisfait au niveau des propriétés et des résultats.

1 Introduction

Dans les systèmes d'information distribués, les pairs ne partagent pas toujours la même ontologie. Il est même connu que la création et la maintenance de grosses ontologies est un problème (trop) difficile et qu'il est préférable d'utiliser des ontologies locales plus petites, plus faciles à mettre à jour Rousset (2006). Souvent, ce sont les mappings locaux entre ontologies qui sont utilisés Ives et al. (2003). La plupart des travaux prennent en compte ce que les pairs partagent (leurs concepts-relations-axiomes) mais pas ce en quoi ils sont différents. Selon nous, ce qui n'est pas consensuel peut aussi avoir une importance pour la recherche d'information.

Nous proposons donc deux processus nouveaux lors de la recherche d'information. Du côté de l'utilisateur initiant la requête, une expansion de requête, qui consiste à indiquer virtuellement tous les liens entre les concepts de sa requête et les autres concepts de son ontologie. Notre méthode est différente des expansions ou extensions de requêtes classiques (Voorhees (1994)) en ce qu'elle maintient séparées les différentes expansions en travaillant avec plusieurs vecteurs sémantiques. Du côté du fournisseur d'information, nous mettons en place une

interprétation de la requête selon ses propres connaissances. C'est-à-dire qu'il va déduire des vecteurs sémantiques reçus des pondérations sur des concepts qui lui sont propres.

Lors de ces deux processus, il est nécessaire de classer les concepts d'une ontologie. Plus précisément, étant donné un concept central, pivot pour l'organisation des autres concepts de l'ontologie, il nous faut une mesure de similarité sémantique qui donne une valeur à tous les concepts. Il existe de nombreuses mesures de similarité sémantique, avec des propriétés et des résultats différents. La plupart considèrent cependant la similarité entre deux concepts. Or, nous voulons classer tous les concepts par rapport à un concept central. Il s'agit pour nous de choisir la meilleure mesure de similarité sémantique, étant données ces contraintes et eu égard aux résultats des différentes mesures.

Dans cet article, nous commençons par présenter notre système (section 2) ; puis nous étudions quelques mesures de similarité sémantique en pointant leurs limites, grâce à des considérations de psychologie, et en présentant une mesure qui nous convient partiellement et que nous modifions quelque peu (section 3) ; finalement, nous présentons les résultats des mesures décrites précédemment (section 4).

2 Présentation générale

Nous utilisons un modèle vectoriel sémantique, comme dans Woods (1997) qui est basé sur le modèle vectoriel de Berry et al. (1999), en utilisant des concepts plutôt que des termes. Un *vecteur sémantique* $\overrightarrow{v_\Omega}$ est alors défini comme une application sur un ensemble de concepts $\mathcal{C}_\Omega$ d'une ontologie $\Omega : \forall c \in \mathcal{C}_\Omega, \overrightarrow{v_\Omega} : c \rightarrow [0..1]$. Généralement on mesure la proximité entre documents et requêtes grâce au cosinus, comme chez Salton et MacGill (1983). Le problème du cosinus est qu'il considère comme indépendantes des dimensions proches. Il est alors classique d'utiliser une expansion de requête pour exprimer ces liens entre dimensions, en *propageant* les poids initiaux sur d'autres concepts, et trouver d'autres documents pertinents. Pour ce faire, il est nécessaire de disposer d'une *fonction de similarité* sim_c entre concepts : $sim_c : \mathcal{C}_\Omega \rightarrow [0, 1]$, est une fonction de similarité ssi $sim_c(c) = 1$ et $0 \leq sim_c(c_j) < 1$ for all $c_j \neq c$ in $\mathcal{C}_\Omega$. La propagation à partir d'un concept donne alors un poids à chaque valeur de similarité.

Definition 1 (Fonction de Propagation) *Soit c un concept de Ω pondéré par v ; et soit sim_c une fonction de similarité. Une fonction* $\mathcal{P}f_c : \begin{array}{ccc} [0..1] & \mapsto & [0..1] \\ sim_c(c') & \rightarrow & \mathcal{P}f_c(sim_c(c')) \end{array}$

est une fonction de propagation de c *ssi*

- $\mathcal{P}f_c(sim_c(c)) = v$, *et*
- $\forall c_k, c_l \in \mathcal{C}_\Omega \; sim_c(c_k) \leq sim_c(c_l) \Rightarrow \mathcal{P}f_c(sim_c(c_k)) \; \leq \; \mathcal{P}f_c(sim_c(c_l))$

Parmi les différentes possibilités de fonctions de propagation, les fonctions d'appartenance utilisées en logique floue sont bien adaptées (cf. figure 1). Elles sont définies par trois paramètres : v, le poids du concept central, l_1, la valeur de similarité jusqu'à laquelle les concepts ont aussi la valeur v, l_2, la valeur de similarité jusqu'à laquelle les concepts ont un poids non nul, $\forall x = sim_c(c'), c' \in c_\Omega$,

$$\mathcal{P}f_c(x) = f_{v,l_1,l_2}(x) = \left| \begin{array}{ll} v & if\ x \geq l_1 \\ \frac{v}{l_1 - l_2}x + \frac{l_2 \times v}{l_1 - l_2} & if\ l_1 > x > l_2 \\ 0 & if\ l_2 \geq x \end{array} \right.$$

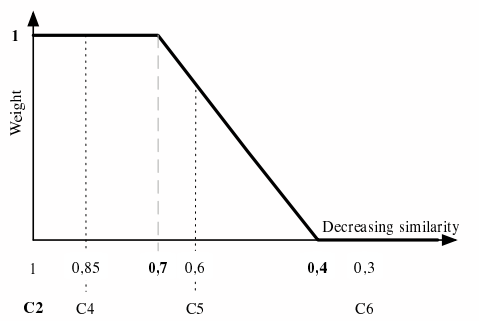


FIG. 1 – *Exemple d'une fonction de propagation* $f_{1,0.7,0.4}$ *avec le concept central* c_2 .

2.1 Expansion de requête

Comme chez Nie et Jin (2002), nous pensons qu'une expansion ne doit pas "bruiter" la requête en y ajoutant des concepts. C'est pourquoi nous proposons de maintenir séparées les propagations issues des concepts principaux de la requête et de générer un vecteur sémantique différent pour chacun d'entre eux. Nous appelons ces vecteurs sémantiques les dimensions sémantiquement enrichies et l'ensemble de ces dimensions, l'expansion de la requête. Soit $\mathcal{C}_{\overrightarrow{q}}$ l'ensemble des concepts centraux de la requête $\overrightarrow{q}$, c'est-à-dire ceux qui la représentent le mieux.

Definition 2 (Dimension sémantiquement enrichie) *Soit* $\overrightarrow{q}$ *une requête et* c *un concept appartenant à* $\mathcal{C}_{\overrightarrow{q}}$*. Un vecteur sémantique* $\overrightarrow{sed}_c$ *est une* dimension sémantiquement enrichie, *ssi* $\forall c' \in \mathcal{C}_\Omega, \overrightarrow{sed}_c[c'] \leq \overrightarrow{sed}_c[c]$.

Definition 3 (Expansion de requête) *Soit* $\overrightarrow{q}$ *un vecteur requête. Une* expansion *de* $\overrightarrow{q}$*, notée* $\mathcal{E}_{\overrightarrow{q}}$ *est un ensemble défini par :*
$\mathcal{E}_{\overrightarrow{q}} = \{\overrightarrow{sed}_c : c \in \mathcal{C}_{\overrightarrow{q}}, \forall c' \in \mathcal{C}_\Omega, \overrightarrow{sed}_c[c'] = \mathcal{P}f_c(c'),$ *où* $\mathcal{P}f_c$ *est une fonction de propagation*$\}$.

La figure 2 illustre le processus décrit : les deux concepts (centraux) de la requête, c_4 and c_7 donnent deux dimensions sémantiquement enrichiesdifférentes, des pondérations étant mises sur les concepts les plus proches des concepts centraux dans chacun d'entre eux.

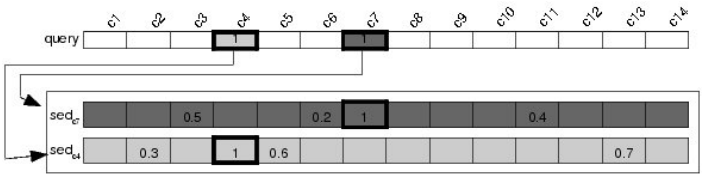


FIG. 2 – *Une expansion de requête composées de deux dimensions sémantiquement enrichies.*

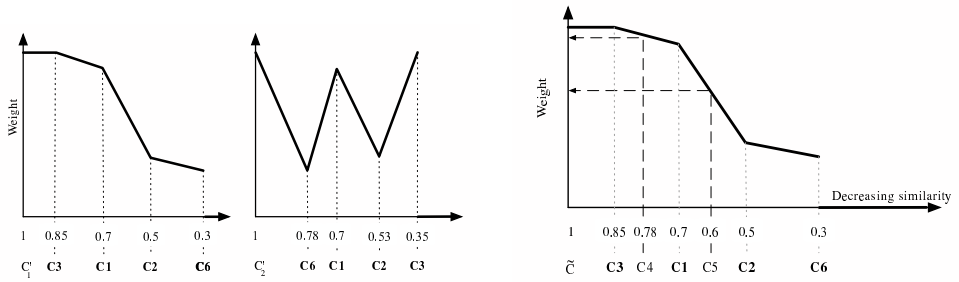
2.2 Interprétation

Du côté du fournisseur, nous voulons qu'il soit possible d'adapter l'expansion à sa propre ontologie. En effet, les concepts mis en jeu dans les DSES peuvent n'être que partiellement partagés. Du coup, il doit être intéressant de laisser au fournisseur la liberté d'indiquer ce qui selon lui et grâce aux informations apportées par les DSES, est pertinent pour la requête. C'est pourquoi nous proposons une étape *d'interprétation de la requête* du côté du fournisseur d'information. Le résultat est un ensemble de DSES interprétées sur $\mathcal{C}_{\Omega_2}$, l'ontologie du pair p_2, i.e. le fournisseur d'information, et une requête interprétée. Chaque DSE est interprété séparément. L'interprétation d'une DSE $\overrightarrow{sed}_c$ se fait en deux étapes :

– trouver un concept dans $\mathcal{C}_{\Omega_2}$ qui corresponde à c, noté $\tilde{c}$;

– attribuer des pondérations aux concepts non partagés de $\mathcal{C}_{\Omega_2}$ qui sont liés à la DSE $\overrightarrow{sed}_c$.

Nous ne pouvons pas décrire ici les deux étapes. Tout d'abord il s'agit d'utiliser une fonction de similarité sur chacun des concepts candidats de $\mathcal{C}_{\Omega_2}$ pour trouver le plus adéquat ; c'est-à-dire celui qui minimise le désordre dans la fonction définie par la DSE d'origine : dans la figure 3 (a), nous choisissons $\tilde{c_1}$ plutôt que $\tilde{c_2}$. Ensuite, nous pondérons les concepts non partagés dans les DSES qui sont maintenant des DSES interprétées : figure 3 (b). Pour plus de détails, voir Ventresque et al. (2007).



(a) (b)

FIG. 3 – *Deux moments de l'interprétation : (a) choix du concept candidat ($\tilde{c_1}$ plutôt que $\tilde{c_2}$) et (b) pondération des concepts non partagés.*

2.3 Image d'un document et pertinence

Une fois la requête étendue et interprétée, nous mettons en place l'image des documents par rapport à cette requête. Il s'agit de synthétiser les différentes DSES en un seul vecteur, qui donne une valeur (possiblement nulle) pour chacun des concepts centraux de la requête. L'objectif étant de n'utiliser qu'un seul espace lors de la mesure de pertinence d'un document par rapport à une requête, qui pour nous est le cosinus entre la requête $\overrightarrow{q}$ et l'image $\overrightarrow{i}_d$ du document $\overrightarrow{d}$. Pour chacun des DSES, nous prenons la valeur maximale des produits entre chacun des concepts du DSEet ceux du document, et nous pondérons dans l'image du document l'indice du concept central du DSEavec cette valeur maximale. Voir par exemple la figure 4.

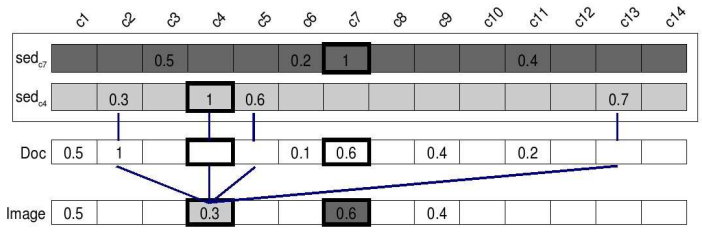


FIG. 4 – *Obtention de l'image d'un document à partir d'une requête étendue.*

3 Similarité sémantique d'un concept par rapport à un autre

3.1 Mesures de similarité sémantique dans la littérature

Nous sommes dans le cadre d'un graphe dont les nœuds sont des concepts. Il paraît donc évident d'utiliser les chemins (suite d'arcs du graphe) pour mesurer la distance entre les concepts. Selon Rada et al. (1989) il s'agit même de la démarche la plus intuitive. Il présente ainsi une mesure utilisant une métrique, $dist(c_1, c_2)$, qui indique le nombre d'arcs minimum à parcourir pour aller d'un concept c_1 à un concept c_2 :

$$sim_{rada}(c_1, c_2) = \frac{1}{1 + dist(c_1, c_2)}$$

D'autres mesures utilisent la notion de plus petit généralisant commun, c'est-à-dire le généralisant commun à $c1$ et $c2$ le plus éloigné de la racine. Ainsi la mesure de WU et PALMER :

$$sim_{W\&P}(c_1, c_2) = \frac{2 \times prof(c)}{prof(c_1) + prof(c_2)}$$

avec $prof(c_i)$ la profondeur du concept c_i, c'est-à-dire la distance à la racine de c_i ; et c le plus petit ancêtre commun à c_1 et c_2. Certaines autres prennent en compte la profondeur de la hiérarchie, comme avec Leacok et Chodorow (1998), ou encore le type de relation entre les concepts (Hirst et St-Onge (1998)).

Tout à fait différemment, des approches "basées sur les nœuds", cherchent le *contenu informatif* des nœuds. Deux versions existent. La première utilise un corpus d'apprentissage et mesure la probabilité de trouver un concept ou un de ses descendants dans ce corpus. Soit c un concept, et $p(c)$ la probabilité de le trouver lui ou un de ses descendants dans le corpus. Le contenu informatif associé à c est alors défini par $IC(c) = -log(p(c))$. Si nous cherchons la proximité entre les concepts c_1 et c_2, il nous faut alors trouver l'ensemble des concepts qui les subsument tous les deux. Soit $S(c_1, c_2)$ cet ensemble. Selon Resnik (1995), nous avons alors par exemple :

$$sim_{resnik}(c1, c2) = \max_{c \in S(c1,c2)} [IC(c)]$$

La seconde version refuse l'utilisation d'un corpus et essaie de calculer le contenu informatif des nœuds à partir de WordNet (Felbaum (1998)) uniquement. La thèse de Seco et al.

(2004) est que, plus un concept a de descendants, moins il est informatif. Ils utilisent donc les hyponymes des concepts pour calculer le contenu informatif de ceux-ci.

$$ic_{wn}(c) = \frac{log(\frac{hypo(c)+1}{max_{wn}})}{log(\frac{1}{max_{wn}})} = 1 - \frac{log(hypo(c)+1)}{log(max_{wn})}$$

avec $hypo(c)$ qui indique le nombre d'hyponymes dont dispose le concept c, et max_{wn}, une constante qui indique le nombre de concepts de la taxonomie. Les différentes mesures de similarité sémantique utilisant le contenu informationnel de Resnik (1995) peuvent donc être redéfinies en utilisant celui de Seco et al. (2004).

Les deux grandes approches définies précédemment peuvent être combinées. Souvent, il s'agit de réutiliser le contenu informatif et le plus petit ancêtre commun (c), comme avec Lin (1998) :

$$sim_{lin}(c_1, c_2) = \frac{2 \times logP(c)}{logP(c_1) + logP(c_2)}$$

ou encore avec Jiang et Conrath (1997)

$$sim_{jiang-conrath}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \times (IC(c))$$

3.2 Psychologie et choix d'une approche

Dans le tableau 1, nous récapitulons quelques propriétés de certaines mesures précédentes. Nous remarquons que la plupart d'entre elles vérifient la symétrie, et de nombreuses sont des distances. Or, des travaux en psychologie éclairent la question de la similarité. Les résultats

	Wu & Palmer	Resnik	Seco	Lin	Bidault
symétrie	oui	oui	oui	oui	**non**
inégalité triangulaire	non	non	non	non	non

TAB. 1 – *Propriétés de quelques mesures des similarité sémantique.*

les plus intéressants sont ceux de Tversky (1977), indiquant que la similarité sémantique n'est pas une distance, parce qu'elle ne satisfait pas la symétrie et l'inégalité triangulaire. Quand nous comparons deux entités, par exemple les rugbymen anglais et les lions, nous disons "les rugbymen anglais se battent comme des lions", et non pas "les lions se battent comme des rugbymen anglais". Parce qu'il y a un sens dans le jugement de similarité. Ici, comme les lions sont le référent (pour leur esprit de combativité), ils ne peuvent pas être le sujet du jugement. De la même façon, nous comprenons de façon très différente "les hommes ressemblent à des arbres" et "les arbres ressemblent à des hommes", parce que la similarité entre deux entités n'est pas symétrique. De plus, si la Martinique et les Bahamas sont similaires parce que ce sont des îles des Caraïbes, et les Bahamas et le Canada sont similaires parce que ce sont d'anciennes colonies britanniques, nous ne pouvons pas en déduire que la similarité entre la Martinique et le Canada est plus grande que la somme des deux premières. La similarité sémantique ne valide pas non plus l'inégalité triangulaire.

Tversky (1977) propose alors un modèle qui tient compte des parties communes et des différences entre deux entités.D'autre part, il faut noter que nous cherchons à mettre en place un classement de tous les concepts par rapport à un concept central. Il ne s'agit pas comme dans la plupart des situations où sont utilisées les mesures de similarité classiques de donner une valeur de proximité[1] entre des concepts. Chez nous la symétrie et l'inégalité triangulaire sont donc d'autant moins justifiées, car il existe un "effet de perspective" dans le classement suivant le concept central choisi.

La solution de Bidault (2002) ne vérifie pas les propriétés de symétrie et d'inégalité triangulaire (cf. tableau 1). Elle met aussi en place un classement des concepts d'une ontologie par rapport à un concept central dans le but d'étendre des requêtes (les "réparer", les "affiner" dans la terminologie de Bidault (2002)). Pour ces deux raisons, elle a attiré notre attention et sert de cadre à notre solution.

3.3 Solution de BIDAULT et améliorations

Bidault (2002) propose une numérotation de tous les concepts de l'ontologie, en partant du principe que descendre, se spécialiser, c'est acquérir des caractéristiques (cf. figure 5). Ainsi, en regardant le ou les numéros[2] d'un concept, on peut facilement savoir non seulement quelle est sa profondeur, mais aussi quels sont ses ancêtres, leur nombre, etc. Nous présentons des formules quelques peu modifiées par rapport à celles de Bidault (2002), car les siennes ne sont pas "normalisées" et ne permettent pas de "ventiler" les concepts sur tout l'intervalle des valeurs de similarité. Soient deux descripteurs m_j et n_i, nous avons la note de proximité de m_j centré sur n_i :

$$R_{m_j \to n_i} = 1 - \frac{2^{P_h - P_{com_{ij}}+1} - 2^{P_h - P_{n_i}+1}}{P_h} - M \times (|m_j| - |com_{ij}|)$$

avec com_{ij} la partie commune aux deux descripteurs, $P_{com_{ij}}$ qui est la profondeur du descripteur commun à n_i et m_j, P_h la profondeur de la hiérarchie, P_{n_i} la profondeur d'un descripteur et M un malus. Selon nous, le malus vaut $\frac{1}{(P_h)^2}$ pour permettre de "ventiler" tous les descripteurs selon leur proximité au descripteur pivot, c'est-à-dire les répartir sur tout l'intervalle de valeurs. Nous avons ensuite les fonctions permettant de noter la proximité d'un concept c centré sur un descripteur n_i, puis d'un concept c centré sur un autre c' :

$$\begin{aligned} R_{c \to n_i} &= \max\{R_{m_j^p \to n_i}, p \in [1..q]\} \\ R_{c \to c'} &= moy\{R_{c \to n_i^p}, p \in [1..q]\} \end{aligned}$$

avec m_j^p, $p \in [1..q]$ l'ensemble des descripteurs pour le concept c . De même pour n_i^p, $p \in [1..q]$ et le concept c' .

[1] Nous utilisons sans grande distinction similarité et proximité, tout en étant conscient que ces notions sont différentes bien que réductibles l'une à l'autre.

[2] Bidault (2002) appelle *descripteur* le numéro d'un concept. Un concept peut évidemment avoir plusieurs descripteurs, s'il a plusieurs hypéronymes.

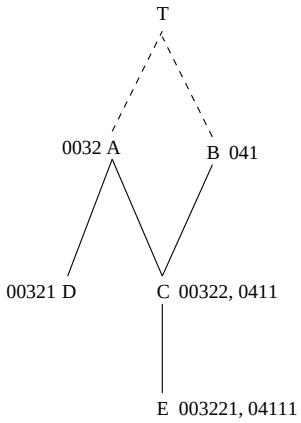


FIG. 5 – *Numérotation d'une ontologie selon le principe de Bidault (2002). T est le concept racine (top). Une numérotation est mise en place depuis ce dernier et est incrémentée d'un digit à chaque niveau de la hiérarchie.*

	human	Wu & P.	Resnik	Seco	Lin	Ventresque et alii.
car - automobile	3.92	0.89	6.11	0.68	1.00	0.937
journey - voyage	3.84	0.92	5.82	0.66	0.69	0.937
asylum - madhouse	3.61	0.82	11.50	0.94	0.98	0.875
bird - crane	2.97	0.84	7.74	0.40	*	O.937
brother - monk	2.82	0.92	10.99	0.18	0.25	0.469
coast - hill	0.87	0.67	6.57	0.50	0.71	0.687
chord - smile	0.13	0.60	2.80	0.18	0.27	0.00
rooster - voyage	0.08	0.00	0.00	0.00	0.00	0.00
corrélation	1.0	0.74	0.77	0.77	0.80	0.82

TAB. 2 – *Valeurs de différentes mesures de similarité sémantique sur quelques exemples du test de Miller et Charles (1991).*

4 Résultats

Pour commencer, nous avons comparé plusieurs mesures de similarité sémantique, grâce au test de Miller et Charles (1991). Ils ont proposé une étude sur des humains (un groupe d'étudiants à qui on demande de noter la similarité entre couples de concepts). Le résultat complet de notre étude se trouve en Ventresque (2004). Nous présentons ici (figure 2) quelques exemples de couples de concepts et le coefficient de corrélation entre les différentes mesures et celle sur les humains : plus la valeur est élevée, plus on est proche du résultat témoin. Nous remarquons que notre mesure obtient de bons résultats, meilleurs que ceux que nous proposons ici.

5 Conclusion

Mesurer la similarité sémantique d'un concept par rapport aux autres dans une ontologie, dans le but de classer ceux-ci par rapport à celui-là n'est pas la même chose qu'évaluer la proximité entre deux concepts comme cela est fait classiquement par les mesures de similarité sémantique. Nous avions besoin d'une mesure qui ne soit pas une distance, et des recherches en psychologie nous on conforté dans notre choix. Nous avons alors choisi dans la littérature la mesure qui nous convenait, puis nous l'avons modifiée à notre convenance.

Cette mesure a été validée avec succès par le test de Miller et Charles (1991). Utilisée dans notre système, au moment de l'expansion et de l'interprétation, elle nous permet aussi d'obtenir de très bons résultats dans un cadre hétérogène.

Références

Berry, M. W., Z. Drmac, et E. R. Jessup (1999). Matrices, vector spaces, and information retrieval. *SIAM Rev. 41*(2), 335–362.

Bidault, A. (2002). *Affinement de requêtes posées à un médiateur*. Ph. D. thesis, University Paris XI, Orsay, Paris, France.

Felbaum, C. (1998). *WordNet : an electronic lexical database*. Bradford Books.

Hirst, G. et D. St-Onge (1998). Lexical chains as representation of context for the detection and correction malapropisms. In C. Fellbaum (Ed.), *WordNet : An electronic lexical database*, Chapter 13, pp. 305–332. The MIT Press.

Ives, Z. G., A. Y. Halevy, P. Mork, et I. Tatarinov (2003). Piazza : mediation and integration infrastructure for semantic web data. *Journal of Web Semantics*.

Jiang, J. et D. Conrath (1997). Semantic similarity based on corpus statistics. In *International Conference on Research in Computational Linguistics*.

Leacok, C. et M. Chodorow (1998). Combining local context and and wordnet similarity for word sense identification. In C. Fellbaum (Ed.), *WordNet : An electronic lexical database and some of its applications*. The MIT Press.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pp. 296–304. Morgan Kaufmann, San Francisco, CA.

Miller, G. A. et W. G. Charles (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes 6*(1), 1–28.

Nie, J.-Y. et F. Jin (2002). Integrating logical operators in query expansion invector space model. In *SIGIR workshop on Mathematical and Formal methods in Information Retrieval*.

Rada, R., H. Mili, E. Bicknell, et M. Blettner (1989). Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man, and Cybernetics 19*(1), 17–30.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pp. 448–453.

Rousset, M.-C. (2006). Somewhere : a scalable p2p infrastructure for querying distributed ontologies. In *CoopIS/DOA/ODBASE*.

Salton, G. et M. MacGill (1983). *Introduction to Modern Information Retrieval*. MacGraw-Hill.

Seco, N., T. Veale, et J. Hayes (2004). An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of ECAI'2004, the 16th European Conference on Artificial Intelligence*.

Tversky, A. (1977). Features of similarity. *Psychological Review 84*(4), 327–352.

Ventresque, A. (2004). Focus et ontologie pour la recherche d'information. Mémoire de DEA d'informatique, Université de Nantes, France.

Ventresque, A., S. Cazalens, P. Lamarre, et P. Valduriez (2007). Query expansion and interpretation to go beyond semantic interoperability. In *ODBASE : Proceedings of the The 6th International Conference on Ontologies, DataBases, and Applications of Semantics*.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *Research and Development on Information Retrieval - ACM-SIGIR*, Dublin, pp. 61–70.

Woods, W. (1997). Conceptual indexing : A better way to organize knowledge. Technical report, Sun Microsystems Laboratories.

Summary

Semantic interoperability in distributed systems is a great problem : mappings are partials and heterogeneity remains. We try to go beyond this problem in expressing queries and documents on shared parts between ontologies, but considering the "unshared parts". At the query initiator side, we set up a query expansion step, and at the provider side, a query interpretation step. During this two steps, we need a ranking of concepts of ontologies thanks to a semantic similarity measure. Most of measures in the literature satisfy the triangle inequality, and some the symmetry. Work in the field of psychology show that it is counterintuitive. So we chose a measure without this properties, but with good results.

Enhancing Semantic Distances With Context Awareness

Ahmad El Sayed *, Hakim Hacid *, Abdelkader Djamel Zighed*

*Université Lyon 2, Laboratoire ERIC
Bat. L, 5 Av. Pierre Mendès-France
69676 Bron cedex - France
asayed@eric.univ-lyon2.fr, hhacid@eric.univ-lyon2.fr, zighed@univ-lyon2.fr,
http://eric.univ-lyon2.fr/

Abstract. A major lack in the existing semantic similarity methods is that no one takes into account the context or the considered domain. However, two concepts similar in one context may appear completely unrelated in another context. In this paper, our first-level approach is context-dependent. We present a new method that computes semantic similarity in taxonomies by considering the context pattern of the text corpus. In addition, since taxonomies and corpora are interesting resources and each one has its strengths and weaknesses, we propose to combine similarity methods in our second-level multi-source approach. The performed experiments showed that our approach outperforms all the existing approaches.

1 Introduction

Comparing two objects relevantly is still one of the biggest challenges and it now concerns a wide variety of areas in computer science, artificial intelligence and cognitive science. The end-goal is that our computational models achieve a certain degree of "intelligence" that makes them comparable to human's intentions over objects. That's obviously a hard task especially that two objects sharing any attribute(s) in common may be related by some abstract 'human-made' relation.

Beyond managing synonymy and polysemy, many applications need to measure the degree of semantic similarity between two words/concepts[1]; let's mention: Information retrieval, question answering, automatic text summarization and translation, etc. A major lack in existing semantic similarity methods is that no one takes into account the context or the considered domain. However, two concepts similar in one context may appear completely unrelated in another context. A simple example for that: While *blood* and *heart* seem to be very similar in a general context, they represent two widely separated concepts in a domain-specific context like medicine.

Thus, our first-level approach is context-dependent. We present a new method that computes semantic similarity in taxonomies by considering the context pattern of the text corpus.

[1] In the following, 'words' is used when dealing with text corpora and 'concepts' is used when dealing with taxonomies where each concept contains a list of words holding certain sense

In fact, taxonomies and corpora are interesting resources to exploit. We believe that each one has its strength and weakness, but using them both simultaneously can provide semantic similarities with multiple views on words from different angles. We propose to combine both methods in our second-level multisource approach to improve the expected performances.

The rest of this paper is organized as follows: Section 2 introduces quickly some semantic similarity measures. Our contribution dealing with a context-dependent similarity measure is described in Section 3. Section 4 presents the experiments made to evaluate and validate the proposed approach. We conclude and give some future works in Section 5.

2 Semantic Similarity in Text

2.1 Knowledge-based Measures

In this work, we focus on taxonomies since we don't need a higher level of complexity. A number of successful projects in computational linguistic have led to the development of some widely used taxonomies like Wordnet[13] (generic taxonomy) and Mesh[2] (for the medical domain). Many taxonomy-based measures for semantic similarity have made their appearance; they can be grouped into edge-based measures and node-based measures.

- **Edge-based Measures**. First, calculating similarities simply relied on counting the number of edges separating two nodes by an *'is-a'* relation [15]. Since specific concepts may appear more similar than abstract ones, the depth was taken into account by calculating either the maximum depth in the taxonomy [10] or the depth of the most specific concept subsuming the two compared concepts [19]. Hirst [8] considers that two concepts are semantically similar if they are connected by a path that is not too long and that does not change direction too often.

- **Node-based Measures** This approach came to overcome the unreliability of edge distances and based its similarities on the information associated with each node. This information can be either a node description (Feature based measures) or a numerical value augmented from a text corpus (Information content measures).

2.1.1 Feature based measures

In this category, we can cite the measure of Tversky [18] which assumes that the more common characteristics two objects have and the less non common characteristics they have, the more similar the objects are:

$$sim(c1, c2) = \frac{|C1 \bigcap C2|}{|C1 \bigcap C2| + k\,|C1/C2| + (k-1)\,|C2/C1|} \quad (1)$$

[2] http://www.nlm.nih.gov/mesh/

2.1.2 Information Content measures

The Information content (IC) approach was first proposed by Resnik[16]. It considers that the similarity between two concepts is "the extent to which they share information in common". Therefore, an IC value, based on a concept frequency in a text corpus, is assigned to each node in the taxonomy. IC represents the amount of information that a concept holds. After assigning IC values for each concept, Resnik defines the similarity between two concepts as the IC value of their Most Informative Subsumer (MIS).

Jiang[9] proposed next a model derived from the edge-based notion by adding the information content as a decision factor. Jiang assigns a link strength (LS) for each *"is-a"* link in the taxonomy which is simply the difference between the IC values of two nodes. Jiang has reached a success rate of 84.4% which led it to outperform all the other taxonomy-based measures. A similar measure to Jiang was proposed by Lin[11] which doesn't only consider the IC of the most informative subsumer but the IC of the compared concepts too.

2.2 Corpus-based Measures

Semantic similarity measures can also be derived by applying statistical analysis on large corpora and by using Natural language Processing (NLP) techniques. The advantage is that corpus driven measures are self-independent; they don't need any external knowledge resources, which can overcome the coverage problem in taxonomies. Three main directions have been pursued in this category of approaches:

- **Co-occurrence-Based Similarity:** This approach study the words co-occurrence or closeness in texts with the assumption that frequent words pairs reveal the existence of some dependence between these words. The first measure was introduced to computational linguistics by Church [4] as the Mutual Information (MI). Among the works we can quote here are those of Turney [17] who showed that Pointwise Mutual Information (PMI) computed on a very large corpus (the web) and using a medium-sized co-occurrence window can be efficiently used to find synonyms. Turney applied the PMI method to TOEFL synonym match problem and obtained an impressive success rate of 72.5% which exceeds by 8% the average foreign student making the test.

- **Context-based Similarity:** This approach is based on the intuition that similar words will tend to occur in similar contexts [3]. Vector-space model is used here as a semantic measuring device. Hindle's approach [7] considers lexical relationship between a verb and the head nouns of its subject and object. Nouns are then grouped according to the extent to which they appear in similar environments. Dagan [5] propose the L_1 norm measure to overcome the zero-frequency problems of bigrams. Turney [17] also proposed an extension of PMI which is an application of PMI on multiple words contexts.

- **Latent Semantic Analysis (LSA)**. LSA introduced by [6] came to overcome the high-dimensionality problem of the standard vector space model especially for the context-based methods. First text is represented as a matrix rows stand for words and columns stands for contexts and each cell contains some specified weight (frequency for instance).

Next, Singular Value Decomposition (SVD) is applied to the matrix in order to analyze the statistical relationships among words in a collection of text. In SVD, a rectangular matrix is decomposed into the product of three other matrices and then recomposed to a single compressed bidimensional matrix. Finally, LSA similarity is computed in a lower dimensional space, in which second-order relations among terms and texts are exploited. The similarity of two words is measured by the cosine of the angle between their corresponding compressed row vectors.

3 A Context-Dependent Similarity Measure

Context definition varies from one research area to another. Considering context is motivated by the fact that it can bring additional information to the reasoning process. Similarity judgments are made with respect to representations of entities, not with respect to entities themselves [12]. Thus, having a changeable representation, one can make any two items similar according to some criteria. To prevent this, a context may be used in order to focus the similarity assessment on certain features of the representation excluding irrelevant information. Barsalou[2] presents a nice example supporting the context-dependency and explaining the instability of similarity judgments.

In text, comparing two concepts doesn't make any sense if we ignore the actual context. Let's take the example of *heart* and *blood*. In a general context, these two concepts can be judged to be very similar. However, if we put ourselves in a medical context, *heart* and *blood* define two largely separated concepts. They will be even more distant if the context is more specifically related to *cardiology*. Our context-dependent approach suggest to adapt semantic similarities to the target corpus since it's the entity representing the context or the domain of interest in most text-based applications. This method is inspired by the information content theory [16] and by the Jiang[9] measure described above.

3.1 Problems with Information Content Measures

As we stated before, IC measures are mainly based on the concept frequency in a text corpus. According to the measure's purpose, we can show two main limitations for that approach:

- *On concept informativeness*. We believe that it's inaccurate to consider infrequent concepts as more informative than frequent ones. We argue that concept frequency is not a good decisive factor for concept informativeness. We follow Nuno's point of view [1] assuming that the taxonomic structure in WordNet is organized in an enough meaningful way to measure IC. We can simply say that the more hypernyms a concept has the more information it expresses. Nuno have shown that at least similar results can be obtained without using a text corpus.

- *On context-dependency*. If the motivation behind measuring the IC from a text corpus is to consider the actual context, we argue that the probability of encountering a concept in a corpus is not a sufficiently adaptive measure to determine whether it's representative for a given context. Thus, IC cannot meaningfully reflect the target context.

3.2 A New Context-Dependency Based Measure

Our approach tends to compute semantic similarities by taking into account the target context from a given text corpus. In order to represent the context, we assign weights for taxonomy's concepts according to their Context-Dependency CD to a corpus C. The goal is to obtain a weighted taxonomy, where 'heavier' subtrees are more context representative than 'lighter' subtrees. This will allow us to calculate semantic similarities by considering the actual context. Therefore, lower similarity values will be obtained in 'heavy' subtrees than 'light' subtrees. Thus, in our *heart/blood* example, we tend to give a high similarity for the concept couple in a general context, and a low similarity in a specific context like medicine.

As we said earlier, it's clear that a concept's frequency alone is not enough to determine its context-dependency. A concept very frequent in some few documents and absent in many others cannot be considered to be "well" representative for the corpus. Thus, the number of documents where the concept occurs is another important factor that must be considered. In addition to that, it's most likely that a concept c_1 -with a heterogeneous distribution among documents - is more discriminative than a concept c_2 with a monotone repartition which can reveal less power of discrimination over the target domain (Experimentations made assess our thesis).

Consequently, we introduce our CD measure which is an adapted version of the standard $tf - idf$. Given a concept c , $CD(c)$ is a function of its total frequency $freq(c)$, the number of documents containing it $d(c)$, and the variance of its frequency distribution $var(c)$ over a corpus C :

$$CD(c) = \frac{log(1 + freq(c))}{log(N)} * \frac{log(1 + d(c))}{log(D)} * (1 + log(1 + var(c))) \quad (2)$$

Where N denotes the total number of concepts in C and D is the total number of documents in C. The log likelihood seems adaptive to such purpose since it helps to reduce the big margins between values. This formula ensures that if a concept frequency is 0, its CD will equals 0 too. It ensures also that if c have an instance in C, its CD will never be 0 even if $var(c) = 0$.

Note that the CD of a concept c is the sum of its individual CD value with the CD of all its subconcepts in the taxonomy. The weights propagation from the bottom to the top of the hierarchy is a natural way to ensure that a parent even with a low individual CD will be considered as highly context-dependent if its children are well represented in the corpus(see Figure 1).

To compare two concepts using the CD values, we assign a Link Strength (LS) to each 'is-a' link in the taxonomy. Assume that c_1 subsumes c_2, the LS between c_1 and c_2 is then calculated as follows:

$$LS(c_1, c_2) = CD(c_1) - CD(c_2)$$

Then the semantic distance[3] is calculated by summing the log likelihood of LS along the shortest path separating the two concepts in the taxonomy.

$$Dist(c_1, c_2) = \sum_{c \in SPath(c_1, c_2)} log(1 + LS(c, parent(c)))$$

[3]Our measure deals with distance which is the inverse of similarity.

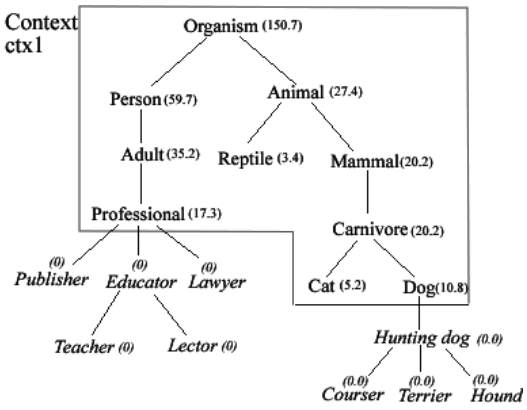


FIG. 1 – *A taxonomy extract showing CD values assigned in the context $cont1$*

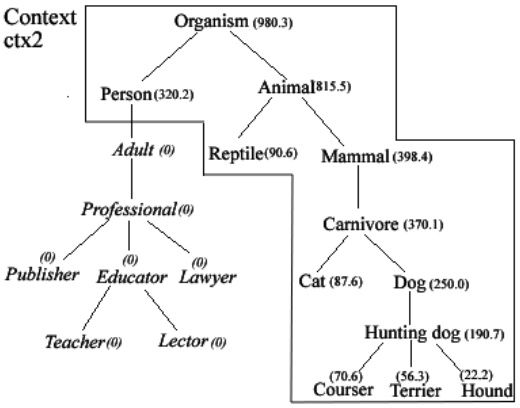


FIG. 2 – *A taxonomy extract showing CD values assigned in the context $cont2$*

Where $SPath$ denotes the shortest path between c_1 and c_2.

Let us illustrate the method with an example. Consider the taxonomy extract shown in figure 1. The related context $ctx1$ taken from a corpus C_1 is represented by the subtree where CD values are greater than 0. In $ctx1$, we notice that the corpus is likely general (talking about persons, professionals, carnivores,etc.). The obtained semantic distance between Cat and Dog in $ctx1$ is 2,2 while it's 4,5 in $ctx2$ illustrated in figure 2 where it seems to be more specialized in the animal domain. This states that Cat and Dog are closer in $ctx1$ than in $ctx2$ which respect human intuitions given the two different contexts.

3.3 A Corpus-based Combination Measure

The promising rates attained by the corpus-based word similarities techniques and especially for the co-occurence-based ones has pushed us to combine them with our context-dependent measure in order to reach the best possible rates. However, two similar words can appear in the same document, paragraph, sentence, or a fixed-size window. It's true that smaller window size can help identifying relations that hold over short ranges with good precisions, larger window size, yet too coarse-grained, allows to detect large-scale relations that could not been detected with smaller windows.

Consequently, we've choose to combine both techniques in order to view relations at different-scales. At the low scale, we use the PMI measure described above with a window size of 10 words (Table 1 Cooc). At the large scale, we calculate the Euclidian distance between words vectors where each word is represented by its tf.idf values over the documents (Table 1 Vecto).

4 Evaluation and Results

4.1 The Benchmark

In this study, Wordnet is used along with a corpus of 30,000 web pages in order to evaluate the proposed approach. The web pages are crawled from a set of news web sites (reuters.com, cnn.com, nytimes.com...).

The most intuitive way to evaluate a semantic similarity/distance is to compare machine ratings and human ratings on a same data set. A very common set of 30 word pairs is given by Miller and Charles [14]. M&C asked 38 undergraduate students to rate each pair on a scale from 0 (no similarity) to 4 (perfect synonymy). The average rating of each pair represents a good estimate on how similar the two words are. The correlation between individual ratings of human replication was 0.90 which led many researchers to take 0.90 as the upper bound ratio. For our evaluations, we've chosen the M&C subset of 28 words pairs which is the most commonly used subset for that purpose. Note that since our measure calculates distance, the M&C distance will be: $dist = 4 - sim$ where 4 represent the maximum degree of similarity.

4.2 Results

When comparing our distance results with the M&C human ratings, the context-dependency CD method gave a correlation of 0.83 which seems to be a very promising rate (See Table 1).In view of further improvements, we evaluated multiple combination strategies.

First, at the taxonomy level, we combine our CD measure with the feature-based measure (Feat) proposed by Tversky (equation 1:

- T1: $Dist = CD.Feat \Rightarrow$ Correlation = 0.83
- T2: $Dist = Feat.\sqrt{CD} \Rightarrow$ Correlation = 0.867

Second, at the corpus level, we combine the vectorial (Vecto) with the PMI measure (Cooc):

- C1: $Dist = Vecto.Cooc \Rightarrow$ Correlation = 0.649
- C2: $Dist = \alpha.vecto + \beta.coocc(\alpha = 0.3, \beta = 0.7) \Rightarrow$ Correlation = 0.564

Finally, at the overall level, we combine the taxonomy-based measures with the corpus-based measures:

- A1: $Dist = T1.Cooc \Rightarrow$ Correlation = 0.810
- A2: $Dist = T2.Cooc \Rightarrow$ Correlation = 0.833
- A3: $Dist = (1 + log(1 + CD)).(1 + log(1 + Cooc)) \Rightarrow$ Correlation = 0.884
- A4: $Dist = (1 + log(1 + T1)).(1 + log(1 + Cooc)) \Rightarrow$ Correlation = 0.890

Our method shows an interesting result whether on an individual level (CD) or on a combination level (T1-A4). We can notice that by using multiple resources (taxonomy and corpus) we could reached a correlation rate of 0.89 (Table 4.2 - A4) which is not too far from human

Word Pair	M&C	CD	Feat	Vecto	Cooc	T1	T2	C1	C2	A1	A2	A3	A4
car-automobile	0,08	1	0,52	2,801	0,332	0,52	0,52	0,93	1,073	0,173	0,173	2,178	1,826
gem-jewel	0,16	1	0,768	2,762	0,398	0,768	0,768	1,099	1,107	0,306	0,306	2,26	2,096
journey-voyage	0,16	3,783	0,847	2,765	0,439	3,203	1,647	1,214	1,137	1,407	0,723	3,499	3,323
boy-lad	0,24	1,635	0,81	2,77	0,389	1,325	1,036	1,078	1,103	0,515	0,403	2,616	2,449
coast-shore	0,3	1,426	0,862	2,762	0,416	1,229	1,03	1,149	1,12	0,512	0,429	2,543	2,429
magician-wizard	0,5	1	0,768	2,779	0,571	0,768	0,768	1,587	1,233	0,438	0,438	2,458	2,279
midday-noon	0,58	1	0,554	2,748	0,559	0,554	0,554	1,536	1,216	0,309	0,309	2,445	2,08
furnace-stove	0,89	14,182	0,886	2,79	0,355	14,182	3,766	0,99	1,086	5,028	1,335	4,849	4,849
food-fruit	0,92	8,489	1	2,793	0,324	8,489	2,914	0,905	1,065	2,751	0,944	4,163	4,163
bird-cock	0,95	3,606	0,858	2,79	0,513	3,094	1,629	1,431	1,196	1,587	0,836	3,574	3,407
bird-crane	1,03	4,286	0,86	2,79	0,499	3,687	1,781	1,392	1,186	1,839	0,889	3,744	3,575
tool-implement	1,05	2,01	0,85	2,768	0,392	1,708	1,205	1,085	1,105	0,67	0,472	2,797	2,657
brother-monk	1,18	1,473	0,905	2,765	0,506	1,333	1,098	1,399	1,184	0,674	0,555	2,685	2,603
crane-implement	2,32	8,982	1	2,754	1	8,982	2,997	2,754	1,526	8,982	2,997	5,589	5,589
lad-brother	2,34	12,745	0,842	2,762	0,398	12,643	3,262	1,099	1,107	5,028	1,297	4,833	4,823
journey-car	2,84	25,653	1	2,804	0,41	25,653	5,065	1,15	1,128	10,508	2,075	5,753	5,753
monk-oracle	2,9	13,64	1	2,776	0,49	13,64	3,693	1,36	1,176	6,683	1,81	5,153	5,153
food-rooster	3,11	13,53	1	2,798	0,597	13,53	3,678	1,67	1,257	8,077	2,196	5,397	5,397
coast-hill	3,13	7,24	1	2,762	0,422	7,24	2,691	1,166	1,124	3,054	1,135	4,203	4,203
forest-graveyard	3,16	21,004	0,902	2,77	1	18,939	4,133	2,77	1,531	18,939	4,133	6,927	6,76
shore-woodland	3,37	15,095	0,903	2,762	0,464	13,635	3,509	1,282	1,153	6,328	1,629	5,219	5,088
monk-slave	3,45	11,302	1	2,773	1	11,302	3,362	2,773	1,532	11,302	3,362	5,942	5,942
coast-forest	3,58	14,736	0,898	2,765	0,408	13,235	3,448	1,128	1,115	5,404	1,408	5,042	4,907
lad-wizard	3,58	11,853	1	2,765	1	11,853	3,443	2,765	1,53	11,853	3,443	6,017	6,017
chord-smile	3,87	15,701	1	2,762	1	15,701	3,963	2,762	1,529	15,701	3,963	6,46	6,46
glass-magician	3,89	17,276	1	2,784	1	17,276	4,156	2,784	1,535	17,276	4,156	6,613	6,613
noon-string	3,92	16,53	1	2,759	0,573	16,53	4,066	1,581	1,229	9,47	2,329	5,614	5,614
rooster-voyage	3,92	24,853	1	2,762	1	24,853	4,985	2,762	1,529	24,853	4,985	7,2	7,2
Correlation	**0,905**	**0.830**	**0.619**	**0.256**	**0.649**	**0.830**	**0,867**	**0.649**	**0.564**	**0.81**	**0.833**	**0.884**	**0.890**

TAB. 1 – *Similarity Results from the different strategies and their correlation to M&C Means.*

correlations of 0.905. The obtained rate led our approach to outperform the existing approaches for semantic similarity (see Table 2).

Our method shows an interesting result whether on an individual or on a combination scale. A part of its interesting correlation coefficient of 0.83, our CD method has the advantage to be context-dependent, which means that our results vary from one context to another. We argue that our measure could perform better if we "place" human subjects in our corpus context. In other terms, our actual semantic distance values reflect a specific context that doesn't necessarily match with the context of the human subjects during the R&C experiments.

Similarity method	**Type**	**Correlation with M&C**
Human replication	Human	0,901
Rada	Edge-based	0,59
Hirst and St-Onge	Edge-based	0,744
Leacock and Chodorow	Edge-based	0,816
Resnik	Information Content	0,774
Jiang	Information Content	0,848
Lin	Information Content	0,821
CD	Context-Dependent	0,830
our multisource measure	Hybrid	0,890

TAB. 2 – *Comparison between the principal measures and our two-level measure*

5 Conclusion and Future Work

We have shown the importance of considering the context when calculating semantic similarities between words/concepts. We've proposed a Context-Dependent method that takes the taxonomy as a principal knowledge resource, and a text corpus as a similarity adaptation resource for the target context. We've proposed also to combine it with other taxonomy-based and corpus-based methods. The results obtained from the experiments show the effectiveness of our approach which led it to outperform the other approaches. A much better way to evaluate the method and compare it with others is to perform context-driven human ratings, where human subjects will be asked to rank a same set of words pairs in different contexts. The machine correlation computed next according to each context will be able to show more significantly the added-value of our approach.

References

[1] N. S. And. An intrinsic information content metric for semantic similarity in wordnet.

[2] L. Barsalou. *Intraconcept similarity and its application for interconcept similarity*. Cambridge University Press, 1989.

[3] H. S. Christopher MANNING. *Foundations of statistical natural language processing*. 1999".

[4] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, B.C., 1989. Association for Computational Linguistics.

[5] I. Dagan, L. Lee, and F. C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.

[6] G. W. Furnas, S. C. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In Y. Chiaramella, editor, *SIGIR*, pages 465–480. ACM, 1988.

[7] D. Hindle. Noun classification from predicate-argument structures. In *Meeting of the Association for Computational Linguistics*, pages 268–275, 1990.

[8] G. Hirst and D. St-Onge. Lexical chains as representation of context for the detection and correction malapropisms, 1997.

[9] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy, 1997.

[10] C. Leacock, M. Chodorow, and G. A. Miller. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.

[11] D. Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[12] D. Medin. *Psychological essentialism*. Cambridge University Press, 1989.

[13] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[14] G. A. Miller and W. Charles. Contextual correlated of semantic similarity. *Language and Cognitive Processes*, 6:1–28, 1991.

[15] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.

[16] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, 11:95–130, 1999.

[17] P. D. Turney. Mining the Web for synonyms: PMI–IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–??, 2001.

[18] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

[19] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

Résumé

Un problème majeur dans les mesures existantes de similarité sémantique c'est qu'ils ne prennent pas en compte le contexte ou le domaine en question. Pourtant, deux concepts similaires dans un contexte donné, pourrait apparaître complètement distinct dans un autre contexte. Nous présentons une approche pour le calcul des distances sémantiques dans une taxonomie en prenant en compte le contexte d'un corpus de texte. Etant donné que les taxonomies et les corpus sont tous les deux des ressources "riches", nous proposons de combiner plusieurs mesures de similarité dans une approche multi-source. Les expérimentations établies démontrent que notre approche amène vers des résultats prometteurs.

Quelques pistes pour une distance entre ontologies

Jérôme Euzenat

INRIA Rhône-Alpes & LIG, Montbonnot, France
jerome.euzenat@inrialpes.fr

Résumé. Il y a plusieurs raisons pour lesquelles il est utile de mesurer une distance entre ontologies. En particulier, il est important de savoir rapidement si deux ontologies sont proches où éloignées afin de déterminer s'il est utile de les aligner ou non. Dans cette perspective, une distance entre ontologies doit pouvoir se calculer rapidement. Nous présentons les contraintes qui pèsent sur de telles mesures et nous explorons diverses manières d'établir de telles distances. Des mesures peuvent être fondées sur les ontologies elles-même, en particulier sur leurs caractéristiques terminologiques, structurelles, extensionnelles ou sémantiques; elles peuvent aussi être fondées sur des alignements préalables, en particulier sur l'existence ou la qualité de tels alignments. Comme on peut s'y attendre, il n'existe pas de distance possédant toutes les qualités désirées, mais une batterie de techniques qui méritent d'être expérimentées.

1 Motivations

Le web sémantique a pour but d'exploiter la connaissance formalisée à l'échelle du web. Il est, en particulier fondé sur les ontologies : des structures définissant les concepts et relations utilisés pour représenter la connaissance. Ces concepts sont utilisés pour décrire les services web sémantiques, annoter les ressources du web (images, textes, musiques, etc.) ou pour décrire des flux de données.

Le web sémantique est donc fondé sur un ensemble d'ontologies. Il est cependant prévisible que des sources de connaissance différentes utiliseront des ontologies différentes.

Dans de nombreux contextes, il est donc utile de déterminer si deux ontologies sont proches ou non (ou de savoir qu'elle est l'ontologie la plus proche d'une ontologie donnée). En particulier,

- si l'on veut déterminer les personnes avec lesquelles on est le plus susceptible de communiquer facilement, trouver celles qui utilisent des ontologies semblables peut être utile (Jung et Euzenat, 2007) ; ceci peut être exploité pour identifier les communautés dans les réseaux sociaux (Jung et al., 2007) ;
- dans les réseaux pair-à-pair sémantiques, il est plus facile de trouver une information si les requêtes peuvent être envoyées rapidement aux nœuds susceptibles d'y répondre. Utiliser des nœuds exploitant des ontologies similaires est utile car les requêtes pourront être transformées avec un minimum de perte d'information (Ehrig et al., 2005) ;
- en ingénierie de la connaissance, il est utile de trouver des ontologies similaires qui pourront être utilisées en conjonction avec une ontologie en cours de développement.

- lorsque l'on veut modulariser une ontologie importante en plus petites parties (Stuckenschmidt et Klein, 2004), on peut considérer les ontologies comme des ensembles d'objets à partitionner et les ensembles les plus distants sont susceptibles d'être séparés ;
- dans les moteurs de recherche sémantique qui retournent des ontologies correspondant à une requête (d'Aquin et al., 2007), il serait utile d'introduire le bouton "Find similar ontologies". Cela peut aussi être utilisé dans l'ordre des réponses à une requête (ontology ranking, Alani et Brewster (2005)) en fonction de la proximité des ontologies ;
- dans certains algorithmes de mise en correspondance d'ontologies (Gracia et al., 2007), lorsque l'on veut trouver des ontologies intermédiaires entre deux ontologies, il est naturel d'utiliser une ontologie proche des deux ontologies à aligner.

Bien entendu, chaque application a besoin d'une mesure de similarité avec des propriétés différentes. Cette "proximité" entre ontologies doit refléter différentes réalités : des ontologies peuvent être similaires parce qu'elles peuvent être facilement traduites l'une dans l'autre où parce qu'elles ont beaucoup de concepts en commun. À son tour, un opérateur de traduction entre ontologies peut être considéré comme facile s'il peut être obtenu rapidement (ou s'il est déjà disponible) ou s'il s'exécute en préservant le maximum d'information.

Nous cherchons à définir une ou des mesures de distance entre ontologies. Nous ne considérons que des distances ou des dissimilarités afin de pouvoir les comparer facilement. Au besoin, nous transformerons des similarités en dissimilarités. Nous ne nous attendons pas à ce qu'une distance particulière satisfasse toutes sortes d'applications mais qu'en fonction de la situation, certaines mesures soient plus appropriées. C'est pourquoi nous nous attacherons à poser les critères permettant de définir de telles mesures. À cette fin, nous examinerons certaines distances déjà définies et nous en introduirons de nouvelles.

Après le rappel de définitions générales sur les mesures de distance (§2), nous introduirons des contraintes s'appliquant spécifiquement aux distances entre ontologies (§3). Nous examinerons ensuite deux types de distances suivant que l'on peut se baser sur l'existence d'alignements (§5) entre ontologies ou non (§4). Après une présentation de travaux connexes (§6), nous terminons par une discussion de l'expérimentation de telles mesures (§7).

2 Propriétés algébriques des distances

Une dissimilarité est une fonction réelle positive δ de deux ontologies qui doit être d'autant plus élevée que les ontologies diffèrent.

Definition 1 (Dissimilarité) *Soit un ensemble O d'ontologies, une dissimilarité $\delta : O \times O \rightarrow \mathbb{R}$ est une fonction qui associe une valeur réelle à un couple d'ontologies telle que :*

$$\forall o, o' \in O, \delta(o, o') \geq 0 \qquad \textit{(positivité)}$$
$$\forall o \in O, \delta(o, o) = 0 \qquad \textit{(minimalité)}$$
$$\forall o, o' \in O, \delta(o, o') = \delta(o', o) \qquad \textit{(symétrie)}$$

Certains auteurs considèrent des dissimilarités et similarités "non symétriques" (Tverski, 1977) ; on préférera le terme de mesure non symétrique ou pré-dissimilarité. Il y a des notions plus contraignantes de dissimilarité comme les distances ou les ultramétriques.

Definition 2 (Distance) *Une distance (ou métrique)* $\delta : O \times O \rightarrow \mathbb{R}$ *est une fonction de dissimilarité définie et satisfaisant l'inégalité triangulaire :*

$$\forall o, o' \in O, \delta(o, o') = 0 \text{ si et seulement si } o = o' \qquad \text{(définition)}$$
$$\forall o, o', o'' \in O, \delta(o, o') + \delta(o', o'') \geq \delta(o, o'') \qquad \text{(inégalité triangulaire)}$$

Il y a de nombreux cas où il est approprié d'utiliser des mesures qui ne sont ni des distances, ni même des dissimilarités. En particulier, si l'on veut considérer la sémantique des ontologies : une mesure purement sémantique devrait retourner 0 lorsque les deux ontologies sont sémantiquement équivalentes, même si elles ne sont pas la même ontologie.

On verra ci-après qu'il peut aussi y avoir de bonnes raisons d'éviter la symétrie.

Très souvent on utilise des mesures normalisées, en particulier si la dissimilarité entre des types d'objets différents doit être comparée. Réduire toutes les valeurs à une même échelle, traditionnellement $[0\ 1]$, en proportion de la taille de l'image de la fonction est une manière commune pour normaliser les mesures utilisées.

Definition 3 (Mesure normalisée) *Une mesure est dite* normalisée *si elle prend ses valeurs dans l'intervalle réel unitaire* $[0\ 1]$. *Une version normalisée d'une mesure* δ *sera notée par* $\overline{\delta}$.

Dans la suite on considèrera principalement des mesures normalisées et l'on supposera qu'une fonction de dissimilarité retourne un nombre réel entre $0.$ et 1.

3 Propriétés spécifiques aux applications

On peut imaginer quelques propriétés des mesures liées à l'utilisation que l'on désire en faire et non à la notion de distance en général. En plus de propriétés algébriques, on voudrait exprimer des propriétés sur la mesure telle que plus une distance est petite :

- plus vite on pourra obtenir un alignement ;
- plus d'entités ont une entité proche dans l'autre ontologie ;
- plus les entités alignées des deux ontologies sont proches ;
- plus facile il sera de répondre à une requête.

Ainsi, on peut considérer la propriété qui veut que l'ajout d'information non comprise dans une ontologie ne puisse que la rendre plus distante :

$$\forall o, o', o'' \in O, o'' \cap o = \emptyset \Rightarrow \delta(o, o') \leq \delta(o, o' \cup o'')$$

On peut, au contraire, vouloir que l'ajout d'information comprise dans une ontologie ne puisse que la rapprocher :

$$\forall o, o', o'' \in O, o'' \subseteq o - o', \Rightarrow \delta(o, o' \cup o'') \leq \delta(o, o')$$

Ces premières propriétés reflètent l'idée que deux ontologies doivent être proches si elles ont beaucoup de concepts en commun et moins c'est le cas, plus elles sont éloignées. Cependant, elles ne sont utiles que si l'on considère des ontologies dont les entités doivent coïncider exactement. Dans la plupart des applications considérées ici, les ontologies sont suffisamment hétérogènes pour ne pas satisfaire cette propriété. Il faut alors prendre en compte l'existence d'une correspondance ou alignement entre ontologies exprimant leurs relations.

On ne considérera que des alignements dans lesquels les relations sont équivalence ($=$) ou subsomption ($\sqsubseteq, \sqsupseteq$) entre entités nommées de chaque ontologies (alignements simples). On considérera qu'ils ne relient que des termes nommés des ontologies, c'est-à-dire identifiés par des URIs.

Definition 4 (Alignement simple) *Soient deux ontologies o et o', un* alignement simple *est un ensemble de correspondances $\langle e, e', r\rangle$, telles que :*
- $e \in N(o)$ *et* $e' \in N(o')$ *sont des entités nommées des ontologies ;*
- $r \in \{=, \sqsubseteq, \sqsupseteq\}$.

On notera par $\Lambda(o, o')$ l'ensemble des alignements entre o et o'. L'existence d'un alignement ayant des propriétés désirées par une application particulière devrait conduire à considérer une ontologie comme proche d'une autre.

Les propriétés que l'on souhaiterait obtenir sont :

$$\forall e \in o, \exists \langle e, e', = \rangle \in A \qquad \text{(couverture)}$$
$$\forall e', e'' \in o, e' \neq e'' \Rightarrow \nexists \langle e', e, = \rangle \in A \vee \nexists \langle e'', e, = \rangle \in A \qquad \text{(injectivité)}$$

La couverture garanti qu'il est possible de traduire toute la connaissance exprimée dans la première ontologie dans la seconde ; l'injectivité garanti que les distinctions présentes dans la première ontologie sont préservées dans la seconde (ceci est vrai tant que la seconde ne dispose pas d'axiomes permettant d'égaliser les images des entités de la première ontologie).

Ces deux propriétés sont exprimées du point de vue d'une ontologie et n'induisent donc pas un comportement symétrique de la mesure. Pour cela, il faudra au minimum exiger que les propriétés soient satisfaites depuis les deux ontologies. Il est possible de transcrire cette exigence de couverture ou d'injectivité sous la forme d'une contrainte pour les mesures de distance. Cette contrainte peut être bâtie sur l'inclusion (une ontologie qui a des alignements plus couvrant, ou "plus injectifs", que ceux d'une autre ontologie doit être plus proche) ou sur la cardinalité (une ontologie qui a un alignement dont la partie couvrante ou injective est plus large qu'une autre doit être plus proche).

On peut définir les propriétés souhaitées à l'aide de $Dom_o(A) = \{e \in o; \exists \langle e, e', r\rangle \in A\}$.

Les formules suivantes traduisent l'idée qu'une bonne mesure doive refléter la couverture à l'aide de l'inclusion :

$$\forall o, o', o'' \in O, \forall A' \in \Lambda(o, o''), \exists A \in \Lambda(o, o'); Dom_o(A') \subseteq Dom_o(A) \Rightarrow \delta(o, o') \leq \delta(o, o'')$$

ou de la cardinalité :

$$\forall o, o', o'' \in O, \exists A \in \Lambda(o, o'); \forall A' \in \Lambda(o, o''), |Dom_o(A')| \leq |Dom_o(A)| \Rightarrow \delta(o, o') \leq \delta(o, o'')$$

La définition correspondante pour l'injectivité est un peu plus complexe.

Les deux définitions ci-dessus ne prennent en compte que la relation d'équivalence ($=$), elles devraient pouvoir être étendues aux relations de subsomption trouvées dans les alignements. Par exemple, si l'on dispose dans o d'un concept automobile et que o' ne dispose que du concept cabriolet reliés entre eux par la relation $\sqsupseteq$, il ne sera pas possible d'exporter les instances de o vers o', mais il sera possible de répondre aux requêtes sur l'extension d'automobile à l'aide des instances de cabriolet.

De la même manière si une entité e d'une ontologie n'a pas de correspondant mais que ses subsumants ou subsumés en ont, il est possible d'en tirer parti dans des notions affaiblies de couvertures. Par exemple, si l'on dispose dans o des concepts vehicule et automobile et que o' ne dispose que du concept car relié à automobile par la relation $=$, il ne sera pas possible d'exporter les instances de vehicule de o vers o', mais il sera possible de répondre aux requêtes sur l'extension de vehicule à l'aide des instances de car.

On voit bien que suivant l'application requise, une notion de couverture sera acceptable ou non.

On peut, en particulier dans une ontologie en logique de description, demander à ce que tout concept soit traductible, soit directement, soit en le ramenant à une définition traductible. Si cela est possible pour tous les concepts, on a alors interopérabilité.

4 Distances dans l'espace des ontologies

J'appelle "espace des ontologies" un ensemble d'ontologies. Dans l'espace des ontologies, aucun alignement entre ontologies n'est disponible a priori. Les distances entre ontologies doivent donc être celles à calculer avant de mettre, si besoin, les ontologies en correspondance. Sur la base de telles mesures, il est possible de décider entre quelles ontologies utiliser un algorithme d'alignement. De telles distances peuvent mesurer la facilité avec laquelle un alignement sera produit (sa rapidité mais aussi sa qualité). Une contrainte naturelle est que la distance soit calculable plus rapidement qu'un éventuel alignement.

La principale manière de mesurer une distance entre ontologies dans l'espace d'alignements est de comparer les ontologies. Ainsi, toute sorte de distance conçue pour mettre les ontologies en correspondance peut être étendue en une distance entre ontologies. Nous en considérons quelques exemples.

4.1 Distances lexicales

Par exemple, une distance entre ontologies peut être calculée à partir des étiquettes apparaissant dans les deux ontologies en utilisant une mesure telle que la distance de Hamming, c'est-à-dire le complément à 1 de la proportion de termes communs aux deux ontologies parmi tous les termes qu'elles utilisent. C'est une dissimilarité et elle s'exécutera assurément plus vite que n'importe quel algorithme sérieux de mise en correspondance, mais elle n'est pas très indicative des résultats d'un éventuel processus de mise en correspondance.

Definition 5 (Distance de Hamming sur les noms de classe) *Soient o et o' deux ontologies et $L(\cdot)$ une fonction retournant les noms des entités dans une ontologie, la distance de Hamming sur les noms de classe est caractérisée par :*

$$\delta_{hdcn}(o, o') = 1 - \frac{|L(o) \cap L(o')|}{|L(o) \cup L(o')|}$$

C'est une dissimilarité normalisée. Elle n'est pas une distance car non définie. Cette mesure est relativement facile à calculer et nos premières expériences montrent qu'elle est plutôt correcte.

Une proposition plus avancée consiste à utiliser des techniques de recherche d'information, c'est-à-dire de considérer tous les noms intervenant dans une ontologie comme une dimension, chaque ontologie comme un point dans un espace métrique de grande dimension et de calculer une distance entre ces points (distance Euclidienne ou cosine). Il est aussi possible d'utiliser des mesures telles que TFIDF (Robertson et Spärck Jones, 1976) pour mesurer combien une ontologie est pertinente vis-à-vis d'une autre. Cette approche est symétrique. Elle a cependant le défaut d'être basée sur un calcul global de fréquence des termes. Ainsi, à chaque fois qu'une nouvelle ontologie est à prendre en compte, les mesures changent.

Les mesures lexicales sont utilisables mais restent très dépendantes des langages utilisés : si les noms d'ontologies sont exprimées dans différents langages naturels, ces mesures ne seront pas les plus utiles. Cependant, on peut considérer que si l'utilisateur n'est pas capable de comprendre les termes utilisés dans une ontologie, alors on ne peut les considérer comme proches.

4.2 Mesures structurelles

Il y a beaucoup de propositions de distances entre concepts. En fait, la plupart des mesures entre ontologies sont fondées sur des distances entre concepts (Mädche et Staab, 2002; Euzenat et Valtchev, 2004; Hu et al., 2006; Vrandečić et Sure, 2007). À partir d'une telle distance δ_K, on peut facilement définir une distance entre ontologies. Parmi les mesures disponibles pour passer d'une distance entre concepts à une distance entre ontologies on trouve les mesures de lien, la distance de Hausdorff ou des mesures reposant sur un couplage.

Definition 6 (Lien moyen) *Soit un ensemble d'entités K et une mesure de dissimilarité $\delta_K : K \times K \rightarrow [0\ 1]$, la mesure de lien moyen entre deux ontologies o et o' est une fonction de dissimilarité $\delta_{alo} : 2^K \times 2^K \rightarrow [0\ 1]$ telle que $\forall o, o' \subseteq K$:*

$$\delta_{alo}(o, o') = \frac{\sum_{(e,e') \in o \times o'} \delta_K(e, e')}{|o| \times |o'|}$$

.

Definition 7 (Distance de Hausdorff) *Soit un ensemble d'entités K et une mesure de dissimilarité $\delta_K : K \times K \rightarrow [0\ 1]$, la distance de Hausdorff entre deux ontologies o et o' est une fonction de dissimilarité : $\delta_{Hausdorff} : 2^K \times 2^K \rightarrow [0\ 1]$ telle que $\forall o, o' \subseteq K$,*

$$\delta_{Hausdorff}(o, o') = \max(\max_{e \in o} \min_{e' \in o'} \delta_K(e, e'), \max_{e' \in o'} \min_{e \in o} \delta_K(e, e'))$$

Le problème de la distance de Hausdorff et des mesures de lien autres que le lien moyen est que sa valeur est une fonction de la distance entre une seule paire d'entités de l'ontologie. Le lien moyen, au contraire, prend en compte les dissimilarités avec toutes les entités. Aucune de ces deux approches n'est satisfaisante.

Les dissimilarités fondées sur un couplage (Valtchev, 1999) mesurent la dissimilarité entre deux ontologies en prenant en compte un couplage entre ces deux ontologies. Un couplage est un alignement. Il peut être défini indépendamment de tout alignement en utilisant des notions comme les couplages maximaux, c'est-à-dire impliquant le maximum d'entités, de poids minimaux, c'est-à-dire telle que la distance entre les entités appariées soit minimale. La qualité

d'une telle mesure est que la dissimilarité dépendra d'une mise en correspondance effective entre les deux ontologies et non d'une distance moyenne. Il sera ainsi possible de transcrire la connaissance d'une ontologie dans une autre. Cependant, de telles mesures sont plus difficiles à calculer.

Definition 8 (Distance de couplage maximal de poids minimal) *Soit un ensemble d'entités K et une mesure de dissimilarité $\delta_K : K \times K \to [0\ 1]$, un couplage maximal de poids minimal entre deux ontologies o et o' est un couplage maximal $M \subseteq o \times o'$, tel que pour tout autre couplage maximal $M' \subseteq o \times o'$,*

$$\sum_{\langle p,q\rangle \in M} \delta_K(p,q) \leq \sum_{\langle p,q\rangle \in M'} \delta_K(p,q)$$

On peut alors définir la distance entre ces deux ontologies par :

$$\delta_{mwmgm}(o,o') = \frac{\sum_{\langle p,q\rangle \in M} \delta_K(p,q) + \max(|o|,|o'|) - |M|}{\max(|o|,|o'|)}$$

On ne détaille pas ici les dissimilarités δ_K possibles. Cette mesure est symétrique, normalisée et définie si δ_K l'est. Beaucoup d'entre elles sont mentionnées dans (Euzenat et Shvaiko, 2007) car elles sont le moyen le plus commun de mettre en correspondance des ontologies. Un bon candidat est la distance, ou plutôt la similarité, définie pour OLA (Euzenat et Valtchev, 2004) parce qu'elle prend en compte tous les attributs des ontologies (étiquettes, structure, instances, etc.) d'une manière équilibrée et surtout parce qu'elle est déjà calculée de manière itérative de façon à obtenir une distance minimale. L'alignement obtenu est déjà le reflet de la structure de l'ontologie, et ceci sera donc pris en compte dans la distance de couplage.

Ce type de mesure peut être utilisé dans tous les types d'applications motivant ce travail.

4.3 Mesures sémantiques

Les mesures proposées jusqu'à présent n'offrent aucune garantie de satisfaction des contraintes sémantiques. Que pourrait être une distance sémantique ? Certainement une distance fondée sur l'interprétation des ontologies. On définit ce qui peut caractériser de telles mesures en se fondant sur la notion de conséquence ($\models$).

Definition 9 (Distance sémantique) *Soient un ensemble d'ontologies O et une relation de conséquence $\models$ pour la logique dans laquelle ces ontologies sont exprimées, une distance δ est sémantique si et seulement si :*

$$\begin{aligned} &\forall o,o',o'' \in O, o \models o' \text{ et } o' \models o'' \\ &\qquad \Rightarrow \delta(o,o') \leq \delta(o,o'') \text{ et } \delta(o',o'') \leq \delta(o,o'') && (\models\text{-compatibilité}) \\ &\forall o,o' \in O, o \models o' \text{ et } o' \models o, \text{ si et seulement si } \delta(o,o') = 0 && (\models\text{-définissabilité}) \end{aligned}$$

Ces contraintes peuvent être réécrites à l'aide de la notion de modèle au lieu de celle de conséquence. Toute sorte de relation entre ensembles peut être utilisée pour les comparer. Par exemple, on peut utiliser la distance de Hamming sur les conséquences :

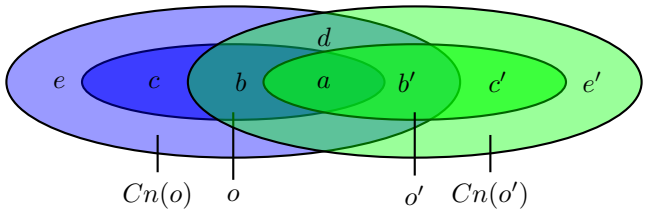


FIG. 1 – *Deux ontologies et leurs relations avec leurs ensembles de conséquences.*

set	definition	unique	invariant	finite
	o	√		√
	$Cn(o)$	√	√	
a	$o \cap o'$	√		√
b	$o \cap Cn(o') - (o \cap o')$	√		√
c	$o - Cn(o')$	√		√
d	$(Cn(o) \cap Cn(o')) - (o \cup o')$	√		
e	$Cn(o) - (o \cup Cn(o')$	√		

TAB. 1 – *Caractéristiques des différents ensembles de la figure 1.*

Definition 10 (Distance sémantique idéale) *Soient deux ontologies o et o' et une fonction Cn retournant leurs ensembes de conséquences, la distance sémantique idéale est définie par :*

$$\delta_{is}(o, o') = 1 - \frac{|Cn(o) \cap Cn(o')|}{|Cn(o) \cup Cn(o')|}$$

La distance sémantique idéale est une distance sémantique. Malheureusement, les ensembles de conséquences sont habituellement infinis. Une alternative classique serait d'utiliser la réduction des ontologies à la place de la clôture. Mais dans le cas général, ces réductions ne sont pas uniques et leur taille peut être variable. Il est donc difficile d'utiliser des mesures fondées sur la cardinalité.

La figure 1 illustre ceci, qui est mis en évidence par la table 1 : la seule chose qui ne dépend pas de la syntaxe des ontologies est $Cn(o)$ qui est infinie. Elle pourrait être comparée avec un ensemble fini, mais tous les ensembles finis sont dépendants de la syntaxe utilisée pour o (c'est-à-dire non invariant). Il est donc difficile de proposer une mesure qui dépend purement de la sémantique même s'il est possible de tester l'équivalence ou la conséquence.

Bien entendu, si les deux langages d'ontologies considérés ne sont pas très expressifs, par exemple, s'ils acceptent des clôtures finies ou des réductions uniques, il est alors possible de calculer une distance sémantique sur la clôture ou la réduction. C'est, en particulier, vrai pour les langages qui n'expriment que des taxonomies.

Cependant, nous avons considéré que les deux ontologies sont comparables. En fait, les ontologies du web sémantique, utilisant en général les URI comme étiquettes ne seront pas comparables : un alignement est requis pour comparer ces ontologies. Ainsi, ce type de mesure n'est utile que si un alignement est disponible.

On considère ci-dessous des distances prenant des alignements en compte.

5 Distances dans l'espace des alignements

On appelle "espace des alignements" un ensemble d'ontologies muni d'un ensemble d'alignements entre ces ontologies. L'ensemble d'alignements est censé ne pas changer. Ainsi, les distances ne mesureront pas la qualité espérée d'un alignement à produire mais celle des alignements existant. Ces alignements et les ontologies qu'ils alignement forment un espace d'alignements :

Definition 11 (Espace d'alignements) *Un* espace d'alignements $\langle \Omega, \Lambda \rangle$ *est composé d'un ensemble d'ontologies* Ω *et d'un ensemble d'alignements simples* Λ *entre ces ontologies. On notera* $\Lambda(o, o')$ *l'ensemble des alignements de* Λ *entre* o *et* o'*.*

Ici encore on ne considérera que des alignements simples. Un espace d'alignements peut être représenté par un multigraphe $G_{\Omega,\Lambda}$ dans lequel les nœuds sont les ontologies et les arcs sont les alignements.

5.1 Distances fondées sur les chemins

La première sorte de distance entre deux ontologies peut être fondée sur l'existence d'un chemin entre ces ontologies dans le graphe $G_{\Omega,\Lambda}$. En fait, l'existence d'un chemin permettra de transformer les requêtes ou les données d'une ontologie vers une autre.

Definition 12 (Distance de chemin d'alignement) *Soit un espace d'alignements* $\langle \Omega, \Lambda \rangle$*, la distance de chemin* δ_{apd} *entre deux ontologies* $o, o' \in \Omega$ *est :*

$$\delta_{apd}(o, o') = \begin{cases} 0 & \textit{si } o = o' \\ 1/3 & \textit{si } o \neq o' \textit{ et } \Lambda(o, o') \neq \emptyset \\ 2/3 & \textit{si } o \neq o' \textit{ et } \Lambda(o, o') = \emptyset \textit{ et } \exists o_0, \ldots o_n \in \Omega \textit{; } o_0 = o, \\ & \quad o_n = o' \textit{ et } \forall i \in [1\ n], \Lambda(o_{i-1}, o_i) \neq \emptyset \\ 1 & \textit{sinon} \end{cases}$$

Une telle distance est maximale entre deux ontologies non connectées et elle est normalisée. Elle est symétrique tant que les alignements le sont. Elle est relativement facile à calculer et est informative en ce qui concerne la possibilité de propager de l'information d'une ontologie à une autre. Cependant, elle n'est pas très précise à propos du nombre de transformations qui devront être réalisées pour propager cette information.

Une mesure naturelle est celle du plus court chemin dans le graphe $G_{\Omega,\Lambda}$. En effet, plus on applique de transformations à la connaissance, plus le processus est long et a de chances d'être dégradé (on peut supposer que chaque transformation perd un peu plus d'information). La mesure suivante est clairement une distance.

Definition 13 (Distance du plus court chemin d'alignement) *Soit un espace d'alignements* $\langle \Omega, \Lambda \rangle$*, la distance du plus court chemin d'alignement* δ_{sapd} *entre deux ontologies* $o, o' \in \Omega$

est la longueur du plus court chemin entre o et o' dans $G_{\Omega,\Lambda}$:

$$\delta_{sapd}(o,o') = \min_{\exists o_0,\ldots o_n \in \Omega; o_0=o, o_n=o' \text{ et } \forall i \in [1\ n], \Lambda(o_{i-1},o_i)\neq\emptyset} n$$

Cette distance simple peut être complétée pour être plus utilisable : elle peut être normalisée par la longueur du plus long chemin plus 1 et, si aucun chemin n'est disponible, le résultat doit être 1 et lorsque les arguments sont la même ontologie, 0.

Le calcul de cette distance n'est pas particulièrement plus long que celui de la précédente. Elle est plus précise car elle va refléter le nombre minimal de transformations nécessaires pour propager la connaissance.

Cependant, tout n'est pas si clair : un alignement entre deux ontologies peut parfaitement être vide. Cela n'indique pas que les ontologies sont très proches mais plutôt qu'elles sont très différentes. Même si les alignements ne sont pas vides, cette mesure n'indique pas la difficulté des transformations et surtout si elles peuvent perdre de l'information et combien. D'autres mesures doivent donc être proposées.

5.2 Distances fondées sur la préservation

Une autre mesure naturelle est de considérer la distance entre deux ontologies données par un alignement entre deux ontologies comme la proportion d'éléments de l'ontologie qui sont pris en compte par l'alignement. Cette mesure est assez naturelle puisque, plus l'ontologie est couverte, plus il y a de chances que l'information puisse transiter. De telle mesures sont destinées à satisfaire la propriété de couverture mentionnée au §3.

On va considérer ici des mesures respectant la couverture du point de vue de la cardinalité. Une telle mesure peut être exprimée comme la proportion d'éléments de l'ontologie de départ qui sont couverts (ou plutôt non couverts) par un alignement.

Definition 14 (Dissimilarité de couverture) *Soit un espace d'alignements $\langle\Omega,\Lambda\rangle$, la dissimilarité de couverture δ_{lcd} entre deux ontologies $o, o' \in \Omega$ est*

$$\delta_{lcd}(o,o') = 1 - \max_{A\in\Lambda(o,o')} \frac{|\{e \in N(o); \exists \langle e,e',r\rangle \in A\}|}{|N(o)|}$$

Cette mesure peut être complétée de la même manière que précédemment. Elle n'est plus symétrique : même si l'alignement n'est fait que d'égalités la proportion dépend de la taille de l'ontologie de départ et non de celle d'arrivée.

Mais nous avons appliqué cette mesure aux alignements et non aux chemins d'alignements. En effet, il y a deux manières de prendre en compte les chemins :

- composer les alignements dans le chemin et calculer la mesure résultante sur ce chemin. Cela peut être très lourd puisqu'il s'agit de calculer tous les chemins et toutes les compositions d'alignements.
- composer les mesures le long des chemins. Malheureusement ceci se révèle très difficile : imaginons que nous ayons à comparer un chemin fait d'un alignement qui couvre 64% de l'ontologie de départ à un chemin fait de deux alignements à 80% chacun. Le résultat devra être compris entre 0% et 80% de préservation ! Même si l'on peut faire du calcul d'intervalle, l'incertitude risque d'être souvent trop grande et exiger le retour à la solution précédente.

Résoudre les problèmes ci-dessus demandera sans doute d'expérimenter une combinaison de calcul d'intervalle, d'exploration heuristique et de composition.

On peut cependant chercher à offrir une première amélioration répondant aux problèmes ci-dessus.

La dissimilarité de couverture n'est déjà plus une pure mesure d'espace d'alignement puisqu'elle requiert de déterminer les entités couvertes en fonction de l'ontologie. Cependant, son calcul ne peut être basé uniquement sur la cardinalité. Les problèmes peuvent être résolus soit en considérant la couverture en fonction non plus de l'ontologie de départ mais de son image par le dernier alignement. Cela requiert de connaître, pour chaque élément A entre o' et o'' une mesure $m(A, A')$ dépendant de l'alignement A' incident à o' (en supplément de $m(A)$ la mesure de couverture définie ci-dessus). Ainsi, la dissimilarité associé avec la composition $A \cdot A' \cdot A''$ commençant en o sera $m(A) \times m(A', A) \times m(A'', A')$. C'est la dissimilarité de la plus grande couverture possible.

Definition 15 (Dissimilarité de la plus grande couverture possible) *Soit un espace d'alignement $\langle \Omega, \Lambda \rangle$, la proportion d'entiés préservées par un chemin $A_0 \cdot \ldots A_n$ est donnée par :*

$$pres(A_0 \cdot \ldots A_n) = \prod_{i=1}^{n} \frac{|\{e; \exists \langle e'', e, r' \rangle \in A_{i-1} \wedge \exists \langle e, e', r \rangle \in A_i\}|}{|\{e; \exists \langle e'', e, r' \rangle \in A_{i-1}\}|}$$

et la dissimilarité de la plus grande couverture possible δ_{lcpd} entre deux ontologies $o, o' \in \Omega$ est :

$$\delta_{lcpd}(o, o') = 1 - \max_{\substack{A_0 \cdot \ldots A_n \in \Lambda^*; \\ \forall i \in [1\ n], A_i \in \Lambda(o_{i-1}, o_i), \\ o_0 = o, \text{ et } o_n = o'}} \left(\frac{|\{e \in N(o); \exists \langle e, e', r \rangle \in A_0|}{|N(o)|} \times pres(A_0 \cdot \ldots A_n) \right)$$

Cette mesure n'est pas parfaite car elle fonctionne seulement étape par étape (il est possible que l'image de l'ontologie initiale ne soit pas dans les objets préservés par un alignement) mais elle devrait fournir une approximation statistiquement correcte.

On peut imaginer d'emblée deux variations :

- La première considérera que ce n'est pas suffisant car un alignement peut projeter beaucoup de concepts dans le même concept. Ceci peut conduire à des ontologies très proches mais de faible qualité en termes de précision. De plus l'utilisation de relations différentes de l'équivalence devrait être prise en compte. Il est nécessaire de trouver un moyen de le faire.
- La seconde est d'aller encore plus loin dans la direction proposée et d'évaluer la distance non plus par rapport aux ontologies mais par rapport à une requête particulière. Cela requièrerait de nouveau de propager la requête le long des chemins et ne sera pas très efficace.

Bien entendu, ces deux types de distances sont complémentaires. Dans la réalité, on n'est jamais dans un contexte où aucun alignement n'existe ou aucun alignement ne peut être créé. Il devrait donc être utile de concevoir des mesures qui peuvent tirer parti simultanément des deux types de situations, par exemple, en utilisant les alignements existants mais sans négliger la possibilité de calculer directement la similarité entre ontologies.

Une dernière proposition qui combine les alignements existants et l'évaluation fondée sur les ontologies consiste à adapter la distance de couplage maximal de poids minimal avec l'existence d'alignements :

Definition 16 (Distance de couplage) *Soit un ensemble d'entités K et une fonction de dissimilarité $\delta_K : K \times K \to [0\ 1]$, pour tout couple d'ontologies $o, o' \subseteq K$ et tout alignement $A \in \Lambda(o, o')$ la distance de couplage entre o et o' est*

$$\delta_{gm}(o, o') = \frac{2 \times \sum_{\langle p,q \rangle \in A} \delta_K(p, q) + (|o| - |A|) + (|o'| - |A|)}{|o| + |o'|}$$

C'est une mesure symétrique défine si δ_K l'est. Elle respecte l'inégalité triangulaire si Λ est clos par composition. Cette distance pondère l'existence d'un alignement par la force de celui-ci, c'est-à-dire qu'elle est fonction de sa couverture (dans les deux sens) et de la distance présumée entre les entités mises en correspondance. Cette mesure peut-être complétée en utilisant toujours le minimum de la distance de couplage pour tous les alignements et d'une distance entre ontologies dans le cas contraire (elle devrait aussi être combinée avec les chemins).

6 Travaux connexes

Les travaux sur le sujet (Mädche et Staab, 2002; Hu et al., 2006; Vrandečić et Sure, 2007) concernent la mesure d'une distance entre concepts dans l'espace des ontologies. Ils sont souvent rapidement étendus aux ontologies sans considérer tous les choix qu'il est nécessaire de faire. De telles mesures sont largement utilisées dans les systèmes d'alignement (Euzenat et Shvaiko, 2007) et peuvent être étendues de la même manière.

Mädche et Staab (2002) ont introduit une similarité entre concepts fondée sur une partie lexicale et une partie structurelle. Cette proposition très détaillée est une combinaison de distance d'édition sur les chaînes et de distance syntaxique sur les hiérarchies (distance de cotopie). La similarité entre ontologies est dépendante d'un couplage fortement fondé sur la similarité lexicale. L'expérimentation relatée dans cet article n'évalue pas réellement la mesure mais plutôt les processus de construction d'ontologies.

Euzenat et Valtchev (2004) ont proposé une mesure de similarité entre concepts de deux ontologies a des fins d'alignement. L'intérêt de la mesure proposée est qu'elle tire parti de tous les aspects des ontologies et retient la similarité maximale (qui peut être transformée en une distance minimale). Elle offre donc d'emblée une base sûre pour une mesure de distance.

Le cadre présenté dans (Ehrig et al., 2005) a pour but de comparer des concepts entre ontologies et non les ontologies elles-même. Il propose une similarité qui combine des similarités entre chaînes, entre concepts – vus comme des ensembles – et entre des traces de l'usage que les utilisateurs font de l'ontologie (ce qui n'est pas forcément toujours disponible).

Un cadre assez élaboré est défini dans (Hu et al., 2006). Il est principalement consacré à la comparaison de concepts mais peut aussi être étendu aux ontologies. Tout d'abord, les concepts sont expansés de sorte qu'ils soient exprimés en fonction de concepts primitifs. Chaque concept est exprimé sous la forme d'une disjonction de concepts composés mais dépourvus de disjonctions. Cela fonctionne si aucun cycle terminologique n'est toléré. Ensuite les concepts primitifs constituent les dimensions d'un espace vectoriel et chaque concept est placé dans cet espace. On utilise TFIDF pour normaliser les axes en fonction de leur pouvoir discriminant. La distance entre deux concepts est la plus petite cosine distance entre les vecteurs associés à deux de leurs concepts disjoints. Comme ce cadre ne permet que de comparer des concepts réductibles

au même ensemble de concepts primitifs, pour comparer des ontologies on suppose qu'une simple distance sur les chaînes de caractères est suffisante pour les concepts primitifs. La manière de passer ensuite aux ontologies n'est pas très clairement expliquée mais les méthodes évoquées au §4.2 fonctionneront.

Vrandečić et Sure (2007) ont considéré plus directement des métriques évaluant la qualité des ontologies. Cependant, c'est un pas vers des mesures sémantiques car ils introduisent des formes normales pour les ontologies qui pourraient permettre de développer des mesures syntaxiquement neutres.

Ces travaux, ainsi que ceux présentés ici, se caractérisent par un manque criant d'évaluation.

7 Vers l'expérimentation

L'ensemble de mesures présentées pour calculer des distances entre ontologies n'ont pas été évaluées à ce jour, que ce soit par nous ou par d'autres auteurs. Nous avons émis des avis sur leur pertinence fondés sur leur seule forme mathématique. Dans le cas présent, ces avis doivent être étayés par l'expérimentation.

Il est nécessaire d'évaluer à la fois la vitesse de calcul des mesures et leur acuité et ceci dans des situations diverses. En ce qui concerne l'acuité, il faudra prendre en compte, sinon des valeurs des mesures, au moins l'ordre qu'elles doivent induire sur la proximité des ontologies. En ce qui concerne la vitesse, il faudra la mesurer sur l'ensemble des expérimentations.

Une telle expérimentation doit disposer d'un corpus d'ontologies, alignées ou non. Le corpus devrait proposer à la fois des ontologies très proches et des ontologies très éloignées afin de connaître leur pouvoir discriminant. On se propose d'utiliser le corpus d'ontologies proposées pour l'évaluation des algorithmes d'alignements (OAEI[1]). Ce corpus se compose des ensembles d'ontologies suivants :

benchmark est un ensemble d'ontologies très proches puisqu'elles sont le résultat de l'altération d'une ontologie initiale. Par ailleurs, on connaît l'ordre de proximités entre ces ontologies car on connaît la force des altérations effectuées ;

conference est aussi un ensemble d'ontologies très proches entre elles (et relativement proches du domaine bibliographique). Par contre, cet ensemble ne comprend pas d'alignements de référence mais on peut disposer de nombreux alignements produits automatiquement.

anatomy deux ontologies sur l'anatomie devraient être proches entre elles et n'avoir rien à faire avec les autres : les alignements sont connus ;

directory deux taxonomies sur divers sujets qui pourraient être proches : il est possible que l'on connaisse les alignements ;

food deux thesauri sur l'agriculture et la nourriture qui sont proches entre eux et devraient être éloignés des autres. Des alignements partiels sont connus.

Les mesures proposées ici devront être évaluées sur chaque paire d'ontologies et leur temps de calcul enregistré. Leurs résultats pourront aussi être comparés à d'autres mesures objectives

[1] `http://oaei.ontologymatching.org`

(lorsque l'on dispose de leurs valeurs, c'est-à-dire d'alignement de référence) comme la préservation, la couverture et l'accessibilité dans l'espace d'alignement.

Une question qui devrait au moins être tranchée par une telle expérience est celle de savoir si ces mesures tendent à converger vers les mêmes valeurs ou si au contraire elles divergent.

8 Conclusion

Mesurer une distance entre ontologies a de nombreuses applications (trouver une ontologies pour en remplacer une autre, trouver une ontologie dans laquelle une requête peut être traduite, trouver des personnes utilisant des ontologies similaires). Il n'y a donc pas de critère universel pour décider si une ontologie est proche ou éloignée d'une autre.

Nous avons passé en revue diverses mesures destinées à proposer une distance entre ontologies. Ces mesures sont résumées dans la table 2 où nous avons indiqué leurs propriétés telles que nous les connaissons ainsi que les arguments en leur faveur. La diversité des mesures présentées en termes de propriétés est déjà frappante. On y distingue les deux types de mesures, fondées sur les ontologies ou sur les alignements, et dans les deux cas on part de mesures simples mais rapides pour aller vers des mesures plus utiles mais difficiles à calculer. Une question importante est donc : les premières peuvent-elles approcher les secondes ? Cette question pourrait être tranchée expérimentalement.

	définie	symétrique	inégalité triangulaire	couverture	injectivité	pour	contre
hdcn		√	√			rapide	langage dep., peu sem.
tfidf		√				assez rapide	langage dep., peu sem.
alo		√	√			rapide	moyenne
Haussdorf	√	√				rapide	maxima
mwmgm	√	√				alignement	lent
is		√	√			sémantique	lent voire impossible
apd	√	√	√			rapide, alignement	peu sem.
sapd	√	√	√			rapide, alignement, chemin	peu sem.
lcd				√		assez rapide, couverture	pas chemin
lcpd				√		meilleure couverture	assez lent
gm	√	√	√			alignement+ontologie	lent

TAB. 2 – *Liste des mesures présentées et leurs propriétés.*

Nous travaillons actuellement à l'organisation d'une évaluation de l'ensemble de ces mesures les unes par rapport aux autres. Ceci requiert la sélection d'un corpus d'ontologies adapté et surtout de préciser a priori ce que l'on attend des distances sur des critères objectifs

Il semble clair qu'il n'existe pas une mesure qui résout tous les problèmes. Les perspectives de recherche pour améliorer chacune des propositions faites ici sont donc plutôt importantes. Leur principal point commun est qu'elles devront arbitrer entre facilité de calcul et pertinence. Parmi les points que nous avons laissé en suspend on peut citer, outre l'évaluation :

- la conception de mesures donnant un bon indice d'injectivité ;
- l'impact de l'utilisation de relations de subsomption ($\sqsubseteq$ et $\sqsupseteq$) dans les alignements ;
- l'intérêt de mesures couvrantes du point de vue de l'inclusion ;
- la conception de mesures extensionnelles si elles restent compatibles avec l'exigence de rapidité.

Remerciements

L'auteur remercie Jérôme David pour ses nombreux commentaires sur différentes versions de ce texte. Ce travail est partiellement financé par le projet intégré européen NeOn (IST-2004-507482).

Références

Alani, H. et C. Brewster (2005). Ontology ranking based on the analysis of concept structures. In *Proc. 3rd International conference on Knowledge Capture (K-Cap), Banff (CA)*, pp. 51–58.

d'Aquin, M., C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, et E. Motta (2007). Watson : a gateway for next generation semantic web applications. In *Proc. Poster session of the International Semantic Web Conference (ISWC), Busan (KR)*.

Ehrig, M., P. Haase, M. Hefke, et N. Stojanovic (2005). Similarity for ontologies – a comprehensive framework. In *Proc. 13th European Conference on Information Systems, Information Systems in a Rapidly Changing Economy (ECIS), Regensburg (DE)*.

Euzenat, J. et P. Shvaiko (2007). *Ontology matching*. Heidelberg (DE) : Springer.

Euzenat, J. et P. Valtchev (2004). Similarity-based ontology alignment in OWL-lite. In *Proc. 16th European Conference on Artificial Intelligence (ECAI)*, Valencia (ES), pp. 333–337.

Gracia, J., V. Lopez, M. d'Aquin, M. Sabou, E. Motta, et E. Mena (2007). Solving semantic ambiguity to improve semantic web based ontology matching. In *Proc. 2nd ISWC Ontology matching workshop (OM), Busan (KR)*, pp. 1–12.

Hu, B., Y. Kalfoglou, H. Alani, D. Dupplaw, P. Lewis, et N. Shadbolt (2006). Semantic metrics. In *Proc. 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, Volume 4248 of *Lecture notes in computer science*, Praha (CZ), pp. 166–181.

Jung, J. et J. Euzenat (2007). Towards semantic social networks. In *Proc. 4th European Semantic Web Conference, Innsbruck (AT)*, Volume 4519 of *Lecture Notes in Computer Science*, pp. 267–280.

Jung, J., A. Zimmermann, et J. Euzenat (2007). Concept-based query transformation based on semantic centrality in semantic peer-to-peer environment. In *Proc. Advances in Data*

and Web Management, Joint 9th Asia-Pacific Web Conference (APWeb) and 8th International Conference, on Web-Age Information Management (WAIM), Huang Shan(CN), Volume 4505 of *Lecture Notes in Computer Science*, pp. 622–629.

Mädche, A. et S. Staab (2002). Measuring similarity between ontologies. In *Proc. 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, Volume 2473 of *Lecture notes in computer science*, Siguenza (ES), pp. 251–263.

Robertson, S. et K. Spärck Jones (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science 27*(3), 129–146.

Stuckenschmidt, H. et M. Klein (2004). Structure-based partitioning of large concept hierarchies. In *Proc. 3rd International Semantic Web Conference (ISWC), Hiroshima (JP)*, Volume 3298 of *Lecture Notes in Computer Science*, pp. 289–303. Springer.

Tverski, A. (1977). Features of similarity. *Psychological Review 84*(2), 327–352.

Valtchev, P. (1999). *Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets*. Thèse d'informatique, Université Grenoble 1, Grenoble (FR).

Vrandečić, D. et Y. Sure (2007). How to design better ontology metrics. In *Proc. 4th European Semantic Web Conference, Innsbruck (AT)*, Volume 4519 of *Lecture Notes in Computer Science*, pp. 311–325.

Summary

There are many reasons for measuring a distance between ontologies. In particular, it is useful to know quickly if two ontologies are close or remote, before deciding to match them. To that extent, a distance between ontologies must be quickly computable. We present constraints applying to such measures and investigate several ways to compute ontology distances. Measures can be based on ontology themselves, in particular on their terminological, structural, extensional and semantic characteristics; they can also be based on available alignments. As can be expected, there is not a unique distance that can satisfy all needs, but various techiques that deserve to be evaluated.

Semantic Similarities and General-Specific Noun Relations from the Web

Gaël Dias*, Raycho Mukelov*, Guillaume Cleuziou** and Veska Noncheva***

*University of Beira Interior Covilhã - Portugal
ddg@di.ubi.pt , raicho@penhas.di.ubi.pt
**University of Orléans, Orléans - France
guillaume.cleuziou@univ-orleans.fr
***University of Plovdiv, Plodiv - Bulgaria
wesnon@pu.acad.bg

Résumé. Dans cet article nous proposons une nouvelle méthodologie utilisant les graphes orientés pondérés et l'algorithme TextRank proposé par Mihalcea et Tarau (2004) dans le but d'extraire automatiquement des relations de généralités entre noms à partir de collocations observées sur un corpus Web. Plusieurs mesures d'association (non-symétriques) ont été implémentées pour construire les graphes sur lesquels l'algorithme TextRank a été appliqué afin de produire une liste de noms ordonnés du plus général au plus spécifique. Les résultats ont été évalués quantitativement en utilisant la hiérarchie des noms de WordNet comme base de référence.

1 Introduction

Taxonomies are crucial for any knowledge-based system. They are in fact important because they allow to structure information, thus fostering their search and reuse. However, it is well known that any knowledge-based system suffers from the so-called knowledge acquisition bottleneck, *i.e.* the difficulty to actually model the domain in question. As stated by Caraballo (1999), WordNet has been an important lexical knowledge base, but it is insufficient for domain specific texts. So, many attempts have been made to automatically produce taxonomies (Grefenstette (1994)), but Caraballo (1999) is certainly the first work which proposes a complete overview of the problem by (1) automatically building a hierarchical structure of nouns based on bottom-up clustering methods and (2) labeling the internal nodes of the resulting tree with hypernyms from the nouns clustered underneath by using patterns like "B is a kind of A".

In this paper, we are interested in dealing with the second problem of the construction of an organized lexical resource *i.e.* discovering general-specific noun relations, so that correct nouns are chosen to label internal nodes of any hierarchical knowledge base, such as the one proposed by Dias et al. (2006). Most of the works proposed so far have (1) used predefined patterns or (2) automatically learned these patterns to identify hypernym/hyponym relations. From the first paradigm, Hearst (1992) first identifies a set of lexico-syntactic patterns that are easily recognizable *i.e.* occur frequently and across text genre boundaries. These can be called seed patterns. Based on these seeds, he proposes a bootstrapping algorithm to semi-automatically

acquire new more specific patterns. Similarly, Caraballo (1999) uses predefined patterns such as “X is a kind of Y” or “X, Y, and other Zs” to identify hypernym/hyponym relations. This approach to information extraction is based on a technique called selective concept extraction as defined by Riloff (1993).

A more challenging task is to automatically learn the relevant patterns for the hypernym/hyponym relations. In the context of pattern extraction, there exist many approaches as summarized by Stevenson et Greenwood (2006). The most well-known work in this area is certainly the one proposed by Snow et al. (2006) who use machine learning techniques to automatically replace hand-built knowledge.

Links between words that result from manual or semi-automatic acquisition of relevant predicative or discursive patterns (Hearst (1992); Caraballo (1999)) are fine and accurate, but such an acquisition is a tedious task that requires substantial manual work. On the other side, works done by Snow et al. (2006) have proposed methodologies to automatically acquire these patterns mostly based on supervised learning to leverage manual work. However, training sets still need to be built. Unlike other approaches, we propose an unsupervised methodology which aims at discovering general-specific noun relations which can be assimilated to hypernym/ hyponym relations detection The advantages of this approach are clear as it can be applied to any language or any domain without any previous knowledge, based on a simple assumption : specific words tend to attract general words with more strength than the opposite. As Michelbacher et al. (2007) state : *“there is a tendency for a strong forward association from a specific term like adenocarcinoma to the more general term cancer, whereas the association from cancer to adenocarcinoma is weak”*.

Based on this assumption, we propose a methodology based on directed weighted graphs and the TextRank algorithm (Mihalcea et Tarau (2004)) to automatically induce general-specific noun relations from web corpora frequency counts. Indeed, asymmetry in Natural Language Processing can be seen as a possible reason for the degree of generality of terms (Michelbacher et al. (2007)). So, different asymmetric association measures are implemented to build the graphs upon which the TextRank algorithm is applied and produces an ordered list of nouns from the most general to the most specific. Experiments have been conducted based on the WordNet noun hierarchy and a quantitative evaluation proposed using the statistical language identification model (Beesley (1998)).

2 Asymmetric Association Measures

Michelbacher et al. (2007) clearly point at the importance of asymmetry in Natural Language Processing. In particular, we deeply believe that asymmetry is a key factor for discovering the degree of generality of terms. It is cognitively sensible to state that when someone hears about "mango", he may induce the properties of a *"fruit"*. But, when hearing *"fruit"*, more common fruits will be likely to come into mind such as *"apple"* or *"banana"*. In this case, there exists an oriented association between *"fruit"* and *"mango"* (mango $\rightarrow$ fruit) which indicates that *"mango"* attracts more *"fruit"* than *"fruit"* attracts *"mango"*. As a consequence, *"fruit"* is more likely to be a more general term than *"mango"*.

Based on this assumption, asymmetric association measures are necessary to induce these associations. Pecina et Schlesinger (2006) and Tan et al. (2004) propose exhaustive lists of association measures from which we present the asymmetric ones that will be used to measure

the degree of attractiveness between two nouns, x and y, where $f(.,.)$, $P(.)$ and $P(.,.)$ are respectively the frequency function, the marginal probability function and the joint probability function, and N the total of digrams.

$$Braun - Blanquet = \frac{f(x,y)}{\max\left(f(x,y) + f(x,\bar{y}), f(x,y) + f(\bar{x},y)\right)} \tag{1}$$

$$J\ measure = \max \left[\begin{array}{l} P(x,y)\log\frac{P(y|x)}{P(x)} + P(x,\bar{y})\log\frac{P(\bar{y}|x)}{P(\bar{y})}, \\ P(x,y)\log\frac{P(x|y)}{P(x)} + P(\bar{x},y)\log\frac{P(\bar{x}|y)}{P(\bar{x})} \end{array} \right]. \tag{2}$$

$$Confidence = \max[P(x|y), P(y|x)] \tag{3}$$

$$Laplace = \max\left[\frac{N.P(x,y)+1}{N.P(x)+2}, \frac{N.P(x,y)+1}{N.P(y)+2}\right]] \tag{4}$$

$$Conviction = \max\left[\frac{P(x).P(\bar{y})}{P(x,\bar{y})}, \frac{P(\bar{x}).P(y)}{P(\bar{x},y)}\right] \tag{5}$$

$$Certainty\ Factor = \max\left[\frac{P(y|x)-P(y)}{1-P(y)}, \frac{P(x|y)-P(x)}{1-P(x)}\right] \tag{6}$$

$$Added\ Value = \max[P(y|x) - P(y), P(x|y) - P(x)] \tag{7}$$

All seven equations show their asymmetry by evaluating the maximum value between two hypotheses *i.e.* by evaluating the attraction of x upon y but also the attraction of y upon x. As a consequence, the maximum value will decide the direction of the general-specific association *i.e.* $(x \rightarrow y)$ or $(y \rightarrow x)$.

3 TextRank Algorithm

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

Our intuition of using graph-based ranking algorithms is that more general words will be more likely to have incoming associations as they will be associated to many specific words. On the opposite, they will have few outgoing associations as they will not attract specific words. As a consequence, the voting paradigm of graph-based ranking algorithms should give more strength to general words than specific ones, thus resulting in an ordered list of words from general to specific.

For that purpose, we first need to build a directed graph. Informally, if x attracts more y than y attracts x, we will draw an edge between x and y as follows ($x \rightarrow y$) as we want to give more credits to general words. Formally, we can define a directed graph $G = (V, E)$ with the set of vertices V (in our case, a set of words) and a set of edges E where E is a subset of $V \times V$ (in our case, defined by the asymmetric association measure value between two words). In Figure 1, we show the directed graph obtained by using the set of words $V = \{$*isometry, rate of growth, growth rate, rate*$\}$ randomly extracted from WordNet where *"rate of growth"* and *"growth rate"* are synonyms, *"isometry"* an hyponynym of the previous set and *"rate"* an hypernym of the same set. The weights associated to the edges have been evaluated by the confidence association measure (Equation 3) based on web search engine counts[1].

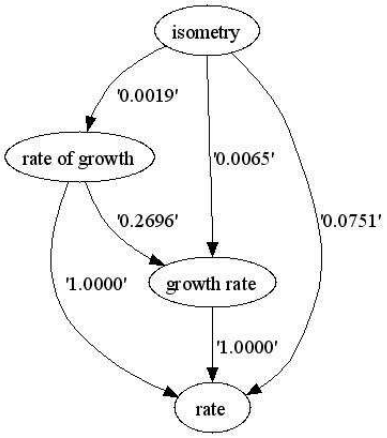


FIG. 1 – *Directed Graph Construction.*

Figure 1 clearly shows our assumption of generality of terms as the hypernym *"rate"* only has incoming edges whereas the hyponym *"isometry"* only has outgoing edges. As a consequence, by applying a graph-based ranking algorithm, we aim at producing an ordered list of words from the most general (with the highest value) to the most specific (with the lowest value). For that purpose, we present the TextRank algorithm proposed by Mihalcea et Tarau (2004) both for unweighted and weighted directed graphs.

Unweighted Directed Graph

For a given vertex V_i let $In(V_i)$ be the set of vertices that point to it (predecessors), and let $Out(V_i)$ be the set of vertices that vertex V_i points to (successors). The score of a vertex V_i is defined in Equation 8 where d (usually set to 0.85) is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph.

$$S(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} \times S(V_j) \quad (8)$$

[1]We used counts returned by http ://www.yahoo.com.

Weighted Directed Graph

In order to take into account the weights of the edges, a new formula is introduced in Equation 9.

$$WS(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{k \in Out(V_j)} w_{j,k}} \times WS(V_j) \quad (9)$$

After running the algorithm in both cases, a score is associated to each vertex, which represents the "importance" of the vertex within the graph. Notice that the final values obtained after TextRank runs to completion are not affected by the choice of the initial values randomly assigned to the vertices, only the number of iterations needed for convergence may be different. As a consequence, after running the TextRank algorithm, in both its configurations, the output is an ordered list of words from the most general one to the most specific one. In table 1, we show both the lists with the weighted and unweighted versions of the TextRank based on the directed graph shown in Figure 1.

Unweighted		Weighted		WordNet	
$S(V_i)$	Word	$WS(V_i)$	Word	Category	Word
0.50	*rate*	0.81	*rate*	Hypernym	*rate*
0.27	*growth rate*	0.44	*growth rate*	Synset	*growth rate*
0.19	*rate of growth*	0.26	*rate of growth*	Synset	*rate of growth*
0.15	*isometry*	0.15	*isometry*	Hyponym	*isometry*

TAB. 1 – *TextRank ordered lists.*

The results show that asymmetric measures combined with directed graphs and graph-based ranking algorithms such as the TextRank are likely to give a positive answer to our hypothesis about the degree of generality of terms. Moreover, we propose an unsupervised methodology for acquiring general-specific noun relations. However, it is clear that deep evaluation is needed.

4 Experiments and Results

Evaluation is classically a difficult task in Natural Language Processing. Human judgment or evaluation metrics are two possibilities. However, human evaluation is time-consuming and generally subjective even when strict guidelines are provided. As a consequence, in order to validate our assumptions, we propose an automatic evaluation scheme based on statistical language identification techniques (Beesley (1998)).

Evaluation Metric

To identify the language of a text, a distance between its frequency-ordered list of N-grams and language baseline frequency ordered-lists can be computed. For each N-gram in the test document, there can be a corresponding one in the current language profile it is compared to.

N-grams having the same rank in both profiles receive a zero distance. If the respective ranks for an N-gram vary, they are assigned the number of ranks between the two as shown in Figure 2. Finally all individual N-gram rank distances are added up and evaluate the distance between the sample document and the current language profile.

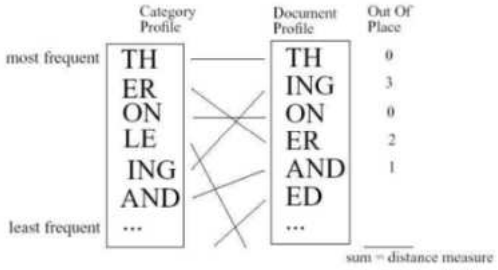


FIG. 2 – *Statistical Language Identification.*

For our purpose, we aim at calculating the distance between the lists of general-specific relations encountered by the TextRank algorithm and the original list given by WordNet. However, we face one problem. WordNet does not give an order of generality inside a synset. Then, we decided to order the words in each synset by their estimated frequency given by WordNet[2] and their frequency calculated in the web space, as our work is based on document hits. An example of both ordered lists is given in Table 2 showing different results.

WordNet Estimated Frequency		Web Estimated Frequency	
Category	Word	Category	Word
Hypernym	*statement*	Hypernym	*statement*
Synset	*answer*	Synset	*reply*
Synset	*reply*	Synset	*response*
Synset	*response*	Synset	*answer*
Hyponym	*rescript*	Hyponym	*feedback*
Hyponym	*feedback*	Hyponym	*rescript*

TAB. 2 – *Estimated Frequencies ordered lists.*

So, calculating the distance $d(.,.)$ on the lists proposed in Table 3 results in : *d(A,B)=5+1+0+2+1+1=10* and *d(A,C)=4+1+1+0+2+0=8*.

Weighted list (A)	WordNet Esti. List (B)	Web Esti. List (C)
feedback	*statement*	*statement*
statement	*answer*	*reply*
reply	*reply*	*response*
answer	*response*	*answer*
response	*rescript*	*feedback*
rescript	*feedback*	*rescript*

TAB. 3 – *Ordered lists to calculate* $d(.,.)$.

It is clear that this distance is a penalty factor which must be averaged by the length of the list. For that purpose, we propose the $matching - score(.,.)$ in Equation 10 (where $length(.)$

[2]We use WordNet 2.1.

is the number of words in a list and $n \in \mathbb{N}^+$) which aims at weighting positively the fact that two lists A and B are similar.

$$matching - score(A,B) = \begin{cases} 1 - \frac{d(A,B)}{2n^2} \; if \; length(A) = length(B) = 2n, \\ 1 - \frac{d(A,B)}{2n^2+2n} \; otherwise \end{cases} \quad (10)$$

Evaluation Scheme

In order to evaluate our methodology, we randomly extracted 115 seed synsets from which we retrieved their hypernym and hyponym synsets. For each seed synset, we then built the associated directed weighted and unweighted graphs based on the asymmetric association measures referred to in section 2[3] and ran the TextRank algorithm to produce a general-specific ordered lists of terms. For each produced list, we finally calculated their $matching - score(.,.)$ with both WordNet and Web Estimated Lists. In Table 4, we present the average results of the $matching - score(.,.)$ for the 115 synsets.

Equation	Type of Graph	Average *Matching-score* with Wordnet Estimated List	Average *Matching-score* with Web Estimated List
Braun-Blanquet	Unweighted	**51.94**	**52.83**
	Weighted	**51.94**	**52.83**
J Measure	Unweighted	47.41	48.74
	Weighted	46.76	48.93
Confidence	Unweighted	**51.94**	**52.83**
	Weighted	**51.94**	**52.83**
Laplace	Unweighted	**51.94**	**52.83**
	Weighted	**51.94**	**52.83**
Conviction	Unweighted	47.42	48.74
	Weighted	46.74	48.94
Certainly Factor	Unweighted	51.63	52.85
	Weighted	51.75	52.58
Added Value	Unweighted	51.63	52.85
	Weighted	51.77	52.58

TAB. 4 – *Average score in % for entire list comparison.*

In order to be more precise, we proposed another evaluation scheme by looking at the lists such as a sequence of three sub-lists.

In fact, we calculated the average $matching - score(.,.)$ for the three sub-lists that are contained in any general-specific list. Indeed, we can look at a list as the combination of the hypernym list, the synset list and the hyponym list. The idea is to identify differences of results in different parts of the lists (*e.g.* if hypernyms are more easily captured than hyponyms). In Table 5, we illustrate the results by representing a list of words as three sub-lists just in the case of weighted graphs as results between weighted and unweighted are negligible.

[3]The probability functions are estimated by the Maximum Likelihood Estimation (MLE).

Equation	Sub-List	Average *Matching-score* with Wordnet Estimated List	Average *Matching-score* with Web Estimated List
Braun-Blanquet	Hypernym	**68.34**	65.84
	Synset	**55.95**	54.17
	Hyponym	**56.19**	54.54
J Measure	Hypernym	61.98	60.83
	Synset	52.47	51.12
	Hyponym	52.91	54.62
Confidence	Hypernym	**68.34**	65.84
	Synset	**55.95**	54.17
	Hyponym	**56.19**	54.54
Laplace	Hypernym	**68.34**	65.84
	Synset	**55.95**	54.17
	Hyponym	**56.19**	54.54
Conviction	Hypernym	62.14	60.89
	Synset	51.75	50.62
	Hyponym	53.87	55.68
Certainly Factor	Hypernym	67.96	65.34
	Synset	56.03	54.32
	Hyponym	56.07	54.25
Added Value	Hypernym	67.32	64.70
	Synset	55.29	53.70
	Hyponym	56.55	54.52

TAB. 5 – *Average score in % for sub-list comparison.*

Discussion

Based on Table 4, the first conclusion to be drawn from our experiments is that unweighted graphs and weighted graphs perform the same way *i.e.* the importance of the graph is its topology and not its weights. In fact, the number of incoming compared to the number of outgoing edges makes the difference in the results.

The second conclusion is the fact that using any of the asymmetric measures does not drastically influence the results. This is a clear consequence of our first conclusion, as the topology is more important than the values given to the edges and most of the asymmetric association measures are able to catch the correct directions of the edges. In fact, the simplest measure, the Confidence, performs best with a $matching - score(.,.)$ of 52.83% which means that the list obtained with our methodology overlaps more than a half the Web Estimated List.

An important remark needs to be made at this point of our discussion. There is a large ambiguity introduced in the methodology by just looking at web counts. Indeed, when counting the occurrences of a word like *"answer"*, we count all its occurrences for all its meanings and forms. For example, based on WordNet, the word *"answer"* can be a verb with ten meanings and a noun with five meanings. Moreover, words are more frequent than others although they are not so general, unconfirming our original hypothesis. Looking at Table 3, *"feedback"* is a clear example of this statement. As we are not dealing with a single domain within which

one can expect to see the "one sense per discourse" paradigm, it is clear that the $matching - score(.,.)$ would not be as good as expected as it is clearly biased by "incorrect" counts. For that reason, we proposed to use Web Estimated Lists to evaluate the $matching - score(.,.)$. As expected, the results show improvements although negligible for most measures. Lately, with (Kilgarriff (2007)), there has been great discussion whether one should use web counts instead of corpus counts to estimate word frequencies. In our study, we clearly see that web counts show evident problems, like the ones mentioned by Kilgarriff (2007). However, they cannot be discarded so easily. In particular, we aim at looking at web counts in web directories that would act as specific domains and would reduce the space for ambiguity. Of course, experiments with well-known corpora will also have to be made to understand better this phenomenon.

Finally, Table 5 shows very interesting results. On average, the $matching - score(.,.)$ works better to discover hypernyms (68.34%) and hyponyms (56.19%). The worst results are shown for the words in the seed synsets (55.95%). These results are encouraging as defining an order in the seed synset is a difficult task or even impossible. Indeed, it would mean that one is capable of giving a fine-grained level of generalization-specification between synonyms. For example, is it possible to clearly define a level of generalization between the *"answer"* and *"response"* ? It does not seem so. However, with our algorithm, each word has a specific order, even within the seed synset. Based on these results, we clearly believe that future research will lead to improved results.

5 Conclusions and Future Work

In this paper, we proposed a new methodology based on directed weighted/unweighted graphs and the TextRank algorithm to automatically induce general-specific noun relations from web corpora frequency counts. To our knowledge, such an unsupervised experiment has never been attempted so far. In order to evaluate our results, we proposed a new evaluation metric, the $matching - score(.,.)$, based on an adaptation of the statistical language identification model. The results obtained by using seven asymmetric association measures based on web frequency counts showed promising results reaching levels of $matching - score(.,.)$ of 68.34% for hypernyms detection.

Nevertheless, future work is needed. First, based on the statements of Kilgarriff (2007), we aim at reproducing our experiments based on web directories and reference corpora such as the Reuters to avoid large scale ambiguity from web counts. Second, the $matching - score(.,.)$ generally penalizes the overall results as hypernyms and hyponyms are not so much represented in terms of words than the seed synset. As a consequence, we aim at gathering more hypernyms and hyponyms of the seed synset to provide a more representative test set. Third, we want to propose another way of evaluating the results. Instead of applying the $matching - score(.,.)$ function, we could run clustering algorithms to reproduce the three original sub-lists of words. So far, our experiments with the K-means and the PAM algorithm have not been fruitful but we aim at using more sophisticated algorithms such as the PoBOC or the QT-Clustering to perform this task. Finally, we want to study the topologies of the built graphs to understand if simplifications can be made based on their topologies as it is done in (Patil et Brazdil (2007)).

Références

Beesley, K. (1998). Language identifier : A computer program for automatic natural-language identification on on-line text.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Conference of the Association for Computational Linguistics (ACL 1999)*.

Dias, G., C. Santos, et G. Cleuziou (2006). Automatic knowledge representation using a graph-based algorithm for language-independent lexical chaining. In *Proceedings of the Workshop on Information Extraction Beyond The Document (COLING/ACL 2006)*, Sydney, Australia, pp. 36–47.

Grefenstette, G. (1994). *Explorations in automatic thesaurus discovery*. Kluwer Academic Pub.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, Morristown, NJ, USA, pp. 539–545. Association for Computational Linguistics.

Kilgarriff, A. (2007). Googleology is bad science. *Comput. Linguist. 33*(1), 147–151.

Michelbacher, L., S. Evert, et H. Schütze (2007). Asymmetric association measures. In *Recent Advances in Natural Language Processing (RANLP 2007)*.

Mihalcea, R. et P. Tarau (2004). TextRank : Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*.

Patil, K. et P. Brazdil (2007). Sumgraph : Text summarization using centrality in the pathfinder network. *International Journal on Computer Science and Information Systems 2*(1), 18–32.

Pecina, P. et P. Schlesinger (2006). Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, Morristown, NJ, USA, pp. 651–658. Association for Computational Linguistics.

Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. In *National Conference on Artificial Intelligence*, pp. 811–816.

Snow, R., D. Jurafsky, et A. Y. Ng (2006). Semantic taxonomy induction from heterogenous evidence. In *ACL '06 : Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, Morristown, NJ, USA, pp. 801–808. Association for Computational Linguistics.

Stevenson, M. et M. A. Greenwood (2006). Comparing information extraction pattern models. In *Proceedings of the Workshop on Information Extraction Beyond The Document (COLING/ACL 2006)*, Sydney, Australia, pp. 29–35.

Tan, P.-N., V. Kumar, et J. Srivastava (2004). Selecting the right objective measure for association analysis. *Information Systems 29*(4), 293–313.

Summary

In this paper, we propose a new methodology based on directed weighted graphs and the TextRank algorithm (Mihalcea et Tarau (2004)) to automatically induce general-specific noun relations from web corpora frequency counts. Different asymmetric association measures are implemented to build the graphs upon which the TextRank algorithm is applied and produces an ordered list of nouns from the most general to the most specific. Experiments are conducted based on the WordNet noun hierarchy with a quantitative evaluation.

Protocole d'évaluation d'une mesure de degré de relation sémantique

Laurent Mazuel*, Nicolas Sabouret*

*Laboratoire Informatique de Paris 6 - LIP6
104 av du Président Kennedy, 75016 Paris
{laurent.mazuel, nicolas.sabouret}@lip6.fr,
http://www-poleia.lip6.fr/ mazuel/

Résumé. Le traitement de l'approximation sémantique est un sujet encore très ouvert, notamment pour la désambiguïsation de texte ou les moteurs de recherche documentaire. En particulier, l'une des voies les plus récentes concerne la définition de mesures de degré de relation sémantique, plus complètes que les mesures de similarité hiérarchique. Si quelques travaux sur les mesures de degré de relation sémantique ont été proposé, peu ont été évalué sur les protocoles classiques (e.g. Miller et Charles). Cet article présente une évaluation comparative de notre proposition de mesure de degré de relation sémantique et montre qu'elle peut obtenir une corrélation jusqu'à deux fois supérieure aux autres mesures classique de la littérature en fonction des couples de mots pris en compte.

1 Introduction

La grande majorité des systèmes d'interaction homme-machine actuels en langue naturelle pour la commande, les systèmes de questions/réponses, le dialogue... utilise une ontologie pour l'interprétation sémantique (Corby et al. (2004); Milward et Beveridge (2003)). Cette approche permet de concevoir des systèmes robustes et capables de comprendre des commandes ou des requêtes imprécises ou au contraires sur-spécifiées et de proposer des alternatives à l'utilisateur. Néanmoins, pour exploiter toute la puissance d'une ontologie, il faut disposer d'une mesure adaptée au modèle.

Les mesures sémantiques actuelles reposent essentiellement sur l'analyse d'une taxonomie (*e.g.* Budanitsky et Hirst (2006)). Ainsi, il existe peu de formules qui peuvent utiliser les relations sémantiques non-hiérarchique, tel que la méronymie ou l'antonymie. Pourtant, ces types de formules permettent d'améliorer l'interprétation ou la désambiguïsation sémantique en considérant le contexte fonctionnel d'un concept en plus de sa description (*e.g.* Strube et Ponzetto (2006)). Ainsi, nous avons proposé récemment dans Mazuel et Sabouret (2007a) une mesure de degré de relation sémantique pour une taxonomie augmentée de relations non-hiérarchique.

Cet article présente l'évaluation de cette mesure de degré de relation sémantique, évaluation qui n'avait pas encore été réalisée à ce jour. La section 2 définit notre formule, avec les notions de point d'articulation, de composante relationnelle et de composante hiérarchique. La

section 3 décrit brièvement les protocoles d'évaluation classiques, présentent notre approche basée sur le test de Finkelstein et al. (2001) et discute les résultats obtenus. Enfin, la section 4 présente un état de l'art sur les mesures de degré de relation sémantique existantes et les évaluations effectuées pour les valider.

2 Une mesure de degré de relation sémantique

Nous présentons brièvement dans cette section notre mesure de degré de relation sémantique. Nous définirons d'abord une mesure de *non-relation* sémantique (*i.e.* plus le score de notre mesure est faible, plus les concepts sont reliés), que nous noterons $dist_{ONT}$. Cette mesure étant bornée, nous pourrons alors la convertir en une mesure de degré de relation sim_{ONT} en utilisant la méthode proposée par Resnik (1995).

2.1 Le point d'articulation

Une taxonomie augmentée peut être vue comme un graphe orienté avec un certain nombre d'arêtes de types différents réparti sur ce graphe. Il existe donc une multitude de chemins entre deux concepts, passant par des arêtes aux types très variables. En effet, alors que le plus court chemin est unique dans le cas d'une hiérarchie, plusieurs chemins candidats sont envisageables dans le cas des graphes. Basé sur les hypothèses d'Aleksovski et al. (2006) et de Hirst et St-Onge (1998), nous considérerons que pour qu'un chemin soit correct en tenant compte des différents types de relations, ce chemin doit respecter :[1]

1. D'abord une suite de relation X quelconque non-hiérarchique, sans changer de type ;
2. Ensuite une suite de relation hiérarchique $is\text{-}a$.

Ainsi, pour un chemin donné entre deux concepts c_1 et c_2, il existe un concept central, ou point d'articulation. Ce point d'articulation est relié vers le concept c_1 par un chemin composé uniquement de relations X quelconques non-hiérarchiques et vers le concept c_2 par un chemin hiérarchique. Nous posons ainsi notre mesure de *degré de non-relation sémantique* :

$$dist_{ONT}(c_1, c_2) = \min_{t \in C, X \in \mathcal{R}} \{dist_{REL}(c_1, t) + dist_{HIE}(t, c_2)\}$$

où $\mathcal{C}$ est l'ensemble des concepts, $\mathcal{R}$ l'ensemble des relations dans l'ontologie, $dist_{REL}$ la *composante relationnelle* et $dist_{HIE}$ la *composante hiérarchique*. Autrement dit, nous cherchons parmi tous les points d'articulation possibles, le point t qui minimise la distance sémantique et tel que t soit atteignable par la relation X en partance de c_1 et que c_2 soit atteignable à partir de t par un chemin hiérarchique.

2.2 Composante relationnelle & composante hiérarchique

La composante hiérarchique mesure la distance entre deux concepts selon la taxonomie. Nous avons choisi d'utiliser comme base hiérarchique la formule de Jiang et Conrath (1997) pour deux raisons. D'abord, car les évaluations montrent que c'est une formule efficace (*e.g.*

[1] Une discussion détaillée sur la justification de nos hypothèses peut être trouvée dans Mazuel et Sabouret (2007a).

Budanitsky et Hirst (2006)), ensuite car sa justification dans l'article initial est basée sur une somme des pondérations des arêtes traversées, ce qui est en adéquation avec l'idée que nous cherchons à développer pour les chemins non-hiérarchiques. Cette formule est définie tel que :

$$dist_{JC\text{-}HIE}(c_1, c_2) = (IC(c_1) + IC(c_2)) - 2 \times IC(ccp(c_1, c_2))$$

où $IC(c)$ représente une fonction de pondération des noeuds de la hiérarchie.[2] Nous poserons ainsi notre composante hiérarchique comme égale à la distance hiérarchique simple de Jiang & Conrath :

$$dist_{HIE}(c_1, c_2) = dist_{JC\text{-}HIE}(c_1, c_2)$$

Le calcul de la composante relationnelle repose sur une pondération des arêtes et sur la longueur de chemin à parcourir sur cette relation. Ce calcul est fait directement à partir du plus court chemin suivant une relation X donnée dans le sous-graphe correspondant à l'ontologie limitée à la relation X. Pour c_1 et c_2 dans l'ontologie, nous noterons $sp_X(c_1, c_2)$ le plus court chemin suivant X entre c_1 et c_2 et $|sp_X(c_1, c_2)|$ la longueur de ce chemin (∞ lorsque c_1 et c_2 sont dans deux sous-graphes disjoints). Nous voulons que notre calcul soit borné (pour simplifier une conversion linéaire) ainsi que pseudo-logarithmique (pour respecter l'hypothèse proposée par Resnik (1995) de théorie de l'information appliquée aux mesures de similarité sémantique hiérarchique). La composante relationnelle est ainsi définie par :

$$dist_{REL}(c_1, c_2) = TC_X \times \frac{|sp_X(c_1, c_2)|}{(|sp_X(c_1, c_2)| + 1)}$$

où $|sp_X(c_1, c_2)|$ est le nombre d'arêtes du plus court chemin entre c_1 et c_2 suivant X et TC_X est la pondération associée au type d'arête X.[3]

2.3 La formule de degré de relation sémantique

Avec la forme définie en section 2.1 et les définitions précédentes de composantes, nous obtenons finalement notre mesure de *degré de non-relation sémantique*. Nous pouvons remarquer que cette mesure respecte et est compatible avec la distance de Jiang & Conrath hiérarchique : si l'ontologie ne contient que des relations hiérarchiques, nous obtenons la relation $dist_{ONT}(c_1, c_2) = dist_{JC\text{-}HIE}(c_1, c_2)$.

Notre mesure $dist_{ONT}$ est bornée sur $[0, 2 + \max_{X \in \mathcal{R}} TC_X]$. Ainsi nous pouvons définir une transformation linéaire de $dist_{ONT}$ pour obtenir notre *mesure de degré de relation sémantique* (méthode de Resnik (1995)). Nous devons faire attention à cette transformation dans le cas où le chemin ne suit que des relations hiérarchiques (*i.e.* si $dist_{ONT}(c_1, c_2) \equiv dist_{JC\text{-}HIE}(c_1, c_2)$). Pour garder une valeur conforme à la transformation linéaire de Jiang & Conrath, nous ne devons pas utiliser dans ce cas le poids TC_X dans la normalisation d'un chemin entièrement hiérarchique. Notre mesure finale de degré de relation sémantique est donc :

$$sim_{ONT}(c_1, c_2) = 1 - \min \left\{ \frac{dist_{ONT}(c_1, c_2)}{2 + \max_{X \in \mathcal{R}} TC_X}, \frac{dist_{JC\text{-}HIE}(c_1, c_2)}{2} \right\}$$

[2] La fonction est maintenant le plus souvent calculé hiérarchiquement selon la méthode top-down de Resnik (1995) ou la méthode bottom-up de Seco et al. (2004).

[3] Dans notre modèle, les pondérations des arêtes dépendent uniquement du type de relation de l'arête – et non de l'arête elle-même, comme envisagé initialement par Jiang et Conrath (1997).

3 Évaluation

3.1 Description du protocole utilisé

Il existe trois types d'évaluation pour une mesure sémantique (Budanitsky et Hirst (2006); Zargayouna et Salotti (2004)) :

1. L'approche théorique consiste à chercher une démonstration mathématique des principes utilisés dans la formule (*e.g.* Lin (1998))
2. L'approche humaine consiste à faire évaluer sur un ensemble de couples de mots la corrélation entre un jugement humain et le résultat du calcul (*e.g.* Resnik (1995))
3. L'approche applicative consiste à intégrer la formule dans un système (moteur de recherche, système de dialogue, désambiguïsation de texte, etc.) et à évaluer le gain de performance obtenu (*e.g.* Budanitsky et Hirst (2006)).

Notre objectif final est d'évaluer notre formule par l'approche applicative avec nos agents conversationnels (Mazuel et Sabouret (2007b)). Néanmoins, nous n'avons pas défini actuellement d'agent sémantiquement suffisamment ouvert pour obtenir une évaluation efficace. Nous proposerons donc ici une évaluation « approche humaine », afin d'étudier les premières grandes lignes de résultats.

Pour une évaluation « approche humaine », le test de Miller et Charles (1991) est considéré par beaucoup de chercheurs comme une référence à atteindre par leur mesure de similarité (*e.g.* Budanitsky et Hirst (2006); Jiang et Conrath (1997); Lin (1998); Resnik (1995)). Un calcul de corrélation entre le résultat produit par une mesure informatique et un jugement humain sur un ensemble de couple de mots permet d'étudier la « validité » de la mesure informatique. La recherche de la meilleure corrélation sur les 30 mots de Miller et Charles fut (et est encore) utilisé comme une « preuve » pour démontrer l'efficacité d'une mesure de similarité. Néanmoins, la question se pose de la représentativité de ce test pour les mesures de degré de relation sémantique. En effet, la plupart des couples proposés n'ont pas de relations fonctionnelles entre eux et ceux qui en possèdent sont ceux qui donnent les plus mauvais résultats du test (*e.g.* le couple « journey-car »). Il devient ainsi nécessaire de chercher un test plus représentatif du degré de relation sémantique.

Le test WordSimilarity-353[4] de Finkelstein et al. (2001) est un test récent contenant 353 couples de mots reliés essentiellement par des liens *fonctionnels* (*e.g.* « computer-keyboard », « telephone-communication », etc.) et non plus uniquement *attributionels* (*e.g.* « bird-cock », « gem-jewel », etc.) . Les résultats (tableau 1) montrent ainsi que les mesures de similarité actuelles ne passent pas à l'échelle sur ce type de test et ne suffisent pas pour modéliser le degré de relation sémantique. En effet, alors que les mesures classiques atteignent une corrélation de $0,8$ sur le test de Miller & Charles, cette corrélation tombe à $0,3$ sur le test de Finkelstein. Ainsi, pour obtenir des couples plus représentatifs du degré de relation sémantique, nous utiliserons le test WordSimilarity-353 pour évaluer notre mesure.

Pour calculer nos similarités, nous prendrons comme ontologie la partie *nom* de WordNet 3.0. Nous avons retiré 9 couples de mots aux 353 initiaux de Finkelstein, quand au moins un des termes n'était pas défini dans la partie *nom* de WordNet. La quantité d'information IC

[4] `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html`

	Corrélation	
Mesures	Miller & Charles	WordSimilarity-353
Resnik	$0,774$	$0,381$
Jiang & Conrath	$0,850$	$0,370$
Lin	$0,829$	$0,383$

TAB. 1 – *Mesure de corrélation de trois mesures classiques sur le test de Miller et Charles (1991) et 344 couples du test WordSimilarity-353 de Finkelstein et al. (2001) (WordNet 3.0 et formule IC de Seco et al. (2004)).*

hiérarchique d'un concept est modélisée par la formule de Seco et al. (2004). Comme relations, nous utilisons les 3 relations de méronymie/holonymie définies dans WordNet (*part-of, member-of* et *substance-of*), dans les deux sens. Nous avons choisi dans un premier temps de n'utiliser qu'un seul facteur TC pour ces relations (que nous noterons TC_*). Cette évaluation nous permettra par ailleurs de vérifier la validité de cette dernière hypothèse.

Notre objectif final est ainsi double :

1. Évaluer la corrélation de notre mesure avec la référence humaine du test WordSimilarity-353 ;
2. Évaluer empiriquement (si possible) une valeur pour TC_*, celle qui correspond à la meilleure corrélation.

Pour ce dernier point, nous ferons varier le coefficient TC_* de 0 à $2,0$ par pas de $0,1$ et nous étudierons la corrélation des couples en fonction de ce poids.

3.2 Nos résultats & interprétations

Le calcul direct de la corrélation sur l'ensemble des 344 termes donnent un résultat compris dans l'intervalle $[0,377; 0,393]$, en fonction du poids TC_*. La valeur $0,393$ correspond au poids $TC_* = 1,1$. Ce résultat montre que notre meilleure corrélation est plus forte que celle des trois autres mesures, mais avec une différence très faible (et difficilement significative, voir tableau 2). Cette faible différence est due en partie au manque de relations non-hiérarchiques présentes dans WordNet (environ 20% seulement des relations sont non-hiérarchiques), certains chemins fonctionnels ne pouvant pas être pris en compte. Par exemple, il n'existe aucun lien entre le mot *téléphone* et le mot *communication*, ce qui se traduit par une impossibilité de mesurer un degré de relation entre ces deux termes, alors que le test de Finkelstein donne comme évaluation humaine $0,75$. Ce test ne nous permet pas ainsi directement d'évaluer les avantages/inconvénient de notre formule, du à l'utilisation de WordNet.

Afin d'évaluer notre formule en condition optimale, nous testerons donc de plus un sous-ensemble de l'ensemble des 344 couples du test. Pour définir ce sous-ensemble, nous avons étudié le sous-ensemble de couples tel qu'il existe au moins un chemin correspondant à nos hypothèses de construction de notre mesure (*i.e.* tel que notre formule donne un résultat diffèrent du résultat de Jiang & Conrath). A noter que la construction de cet ensemble étant dépendant de nos hypothèses, nous n'avons pas forcément tous les couples reliés par un chemin relationnel considéré correct par un humain (par exemple les couples *war-troops* et *OPEC-oil* ne sont pas dans l'ensemble, ce qui nous amène à revoir une partie de nos hypothèses). A l'inverse,

	Corrélation		
Mesures	WordSimilarity-353	32 couples pour $TC_* = 1,1$	Meilleur Δ
Mazuel	$[0,377; 0,393]$	$0,583$	22
Resnik	$0,381$	$0,241$	3
Jiang & Conrath	$0,370$	$0,291$	8
Lin	$0,383$	$0,236$	0

TAB. 2 – *Mesure de corrélation de trois mesures classiques sur 344 couples du test WordSimilarity-353 et un sous-ensemble de 32 couples (WordNet 3.0 et formule IC de Seco et al. (2004)). La troisième colonne présente, pour les 32 couples, le nombre de fois où la mesure citée était la plus proche du jugement humain.*

nous n'avons pas non plus la certitude que tous les couples de l'ensemble sont reliés par un chemin sémantiquement correct (nous sommes juste certains qu'un chemin existe).

Pour le poids $TC_* = 1,1$ qui représente la meilleure corrélation sur l'ensemble complet, ce sous-ensemble contient 32 couples de mots (ce qui est plus que le test de Miller & Charles et devrait constituer un ensemble suffisamment grand pour être représentatif). Les résultats complets et détaillés pour tous les couples sont donnés dans le tableau 3 et les corrélations de ces 32 couples dans le tableau 2.

On observe, sur cet ensemble de couples, une corrélation avec notre mesure en moyenne 2 fois supérieure à la corrélation obtenue avec les mesures comparées. De plus, statistiquement, notre mesure est la plus proche du jugement humain pour 69% des couples (22 pour 32).

Néanmoins, certaines valeurs obtenues sont étonnement forte et amènent à réviser nos hypothèses. Par exemple, dans le couple « train-car », le résultat très élevé (*i.e.* $0,82$) s'explique par le fait que « car » est pris dans le sens « voiture : véhicule ferroviaire réservé au transport de voyageurs ». Ce problème a déjà été soulevé par Resnik (1995) avec le couple « horse-heroin » étonnamment fort dans ses résultats. En effet, les formules de similarité sont définies pour des couples de synset, et non pas des couples de mots. Le consensus actuel est donc de considérer toutes les permutations entre chaque synset des deux mots et de prendre le maximum obtenu, ce qui conduit parfois au problème précédent. Resnik proposait ainsi comme solution de pondérer l'importance d'un synset dans la formule par sa fréquence d'apparition dans un corpus, afin de mettre en avant le sens « le plus courant ». Cette solution n'a jamais été utilisée en pratique, mais il est possible que le fait d'utiliser beaucoup de relations relance le problème de manière plus conséquente.

Autre exemple, le résultat fort (*i.e.* $0,58$) de « Mars-water » est du au fait que la Terre (synset *Earth*) est composé d'eau (*water* $partOf$ *hydrosphere* $partOf$ *Earth*). De plus, la « Terre » et « Mars » sont très proche hiérarchiquement car toutes les deux sont des planètes (synset *terrestrial planet*). Ce genre d'erreur amène à penser que nos hypothèses de chemin sont incomplètes. En effet, suivre la relation d'hyponymie entre *terrestrial planet* et *Mars* est clairement une erreur. Cela revient à dire qu'un attribut ou propriété d'un concept donné peut être considéré comme proche des frères de ce concept, ce qui est faux.

De plus, il semble que l'hypothèse d'uniformité pour la pondération des arêtes est à améliorer. En effet, par exemple, les couples « century-year » et « computer-keyboard » possèdent la même évaluation humaine $(0,76)$. Or, « century » et « year » sont reliés au bout de 2 relations *part-of* (en passant par *decenny*), alors que « computer » et « keyboard » sont directement

#	Couples		Humain	Mazuel	J&C	Lin	Resnik
1	computer	keyboard	$0,762$	$0,823$	$0,535$	$0,404$	$0,315$
2	plane	car	$0,577$	$0,728$	$0,562$	$0,363$	$0,249$
3	train	car	$0,631$	$0,823$	$0,517$	$0,340$	$0,249$
4	stock	egg	$0,181$	$0,570$	$0,436$	$0,394$	$0,366$
5	fertility	egg	$0,669$	$0,490$	$0,083$	$0,0$	$0,0$
6	bank	money	$0,831$	$0,707$	$0,524$	$0,471$	$0,425$
7	Jerusalem	Israel	$0,846$	$0,822$	$0,366$	$0,366$	$0,366$
8	forest	graveyard	$0,185$	$0,443$	$0,124$	$0,0$	$0,0$
9	coast	forest	$0,315$	$0,488$	$0,223$	$0,095$	$0,082$
10	planet	galaxy	$0,811$	$0,764$	$0,242$	$0,114$	$0,097$
11	planet	space	$0,792$	$0,562$	$0,434$	$0,130$	$0,097$
12	cup	coffee	$0,658$	$0,481$	$0,471$	$0,455$	$0,441$
13	Mars	water	$0,294$	$0,576$	$0,308$	$0,098$	$0,098$
14	sign	recess	$0,238$	$0,542$	$0,533$	$0,438$	$0,364$
15	skin	eye	$0,622$	$0,595$	$0,433$	$0,200$	$0,190$
16	theater	history	$0,391$	$0,511$	$0,231$	$0,184$	$0,179$
17	century	year	$0,759$	$0,763$	$0,601$	$0,523$	$0,437$
18	hospital	infrastructure	$0,463$	$0,570$	$0,094$	$0,082$	$0,082$
19	life	death	$0,788$	$0,822$	$0,793$	$0,764$	$0,669$
20	OPEC	country	$0,563$	$0,672$	$0,505$	$0,437$	$0,384$
21	car	flight	$0,494$	$0,425$	$0,351$	$0,227$	$0,190$
22	street	place	$0,644$	$0,614$	$0,433$	$0,296$	$0,287$
23	street	block	$0,688$	$0,601$	$0,496$	$0,274$	$0,231$
24	network	hardware	$0,831$	$0,581$	$0,356$	$0,279$	$0,249$
25	day	summer	$0,394$	$0,680$	$0,658$	$0,561$	$0,437$
26	nature	man	$0,625$	$0,659$	$0,447$	$0,239$	$0,180$
27	man	governor	$0,525$	$0,573$	$0,543$	$0,300$	$0,249$
28	focus	life	$0,406$	$0,407$	$0,356$	$0,308$	$0,287$
29	country	citizen	$0,731$	$0,807$	$0,405$	$0,066$	$0,042$
30	planet	people	$0,575$	$0,561$	$0,357$	$0,0$	$0,0$
31	space	world	$0,653$	$0,709$	$0,349$	$0,259$	$0,231$
32	preservation	world	$0,619$	$0,479$	$0,324$	$0,324$	$0,324$

TAB. 3 – *Tableau des 32 couples de mots pour $TC_* = 1,1$. Toutes les valeurs ont été normalisé entre 0 et 1 pour faciliter la lecture.*

reliés. Une solution possible à envisager serait de rajouter une « pondération locale » à l'arête, par exemple en considérant la densité relationnelle d'un concept. En pratique, un objet serait moins bien relié sémantiquement à ses composants s'il en possède 10 que s'il en possède 2. Autrement dit, le nombre de relations d'un type X donné partant d'un concept pourrait jouer sur la pondération à appliquer pour ses arêtes sortantes.

4 Autres mesures de degré de relation sémantique

Dans leur article, Thieu et al. (2004) proposent une extension de la mesure de Jiang et Conrath pour le degré de relation sémantique. Le poids d'une relation non-hiérarchique est dépendant des poids IC des deux concepts extrémités, ce qui est surprenant dans la mesure où le calcul de la fonction IC ne dépend que de la hiérarchie. Malheureusement, la formule n'a été évaluée que sur une application et une ontologie spécifique avec peu de relations.

La formule de Cho et al. (2003) reprend une partie des idées exposées dans les mesures de similarité classiques (tel que l'utilisation du plus proche parent commun (le ccp)) avec quelques hypothèses nouvelles (la distinction des différents types de lien (hiérarchie, méronymie, etc.) par une pondération). Ils proposent ainsi la formule suivante :

$$sim_{CHO}(c_i, c_j) = IC(ccp(c_i, c_j)) \times \left[D_{i \to j} \times \sum_{\{c_k, c_l\} \in sp(c_i, c_j)} W_{k \to l} \times d_{k \to l} \times depth(c_k) \right]$$

avec $D_{i \to j}$ « le facteur de distance » entre c_i et c_j (*i.e.* dépendant de la distance en nombre d'arêtes sur le chemin), $W_{k \to l}$ la fonction poids associée au type de lien entre c_k et c_l et $d_{k \to l}$ la « fonction densité » entre c_k et c_l (*i.e.* dépendant du nombre de descendant d'un concept).

Comme on l'observe, le défaut majeur de cette formule est qu'elle utilise plusieurs poids différents et qu'aucune piste n'est donnée pour déterminer leurs valeurs respectives. L'évaluation de l'article utilise des poids non justifiés et le protocole de Miller et Charles (1991), protocole n'est pas le plus efficace dans le cadre du degré de relation sémantique (section 3.1).

La formule de Yang et Powers (2005) se base sur le principe d'une formule paramétrée à base d'*apprentissage supervisée*. La structure générale de la formule est pondérée et basée sur le décompte du nombre d'arêtes sur le chemin le plus court. Ils introduisent ainsi une formule paramétrée, et proposent d'utiliser les données humaines de Miller et Charles pour évaluer les différents paramètres. La formule est donc une simple somme de coefficients en fonction de la longueur du plus court chemin :

$$sim_{YANG}(c_1, c_2) = \alpha_t \prod_{i=1}^{len(c_1, c_2)} \beta_{t_i} \text{ Si } len(c_1, c_2) < \gamma \text{ (0 sinon)}$$

où $len(c_1, c_2)$ est la longueur du plus court chemin entre c_1 et c_2, α_t un facteur lié au type d'arête t, β_t un facteur de profondeur lui aussi paramétré en fonction du type de lien et γ un seuil au delà duquel la similarité est suffisamment négligeable pour être nulle. Pour les auteurs, ce dernier facteur représente « les limitations du système cognitif humain ».

L'apprentissage des bons coefficients se fait ainsi sur le test de Miller et Charles, qui, encore une fois, n'est pas le plus adapté aux mesures de degré de relation sémantique. De plus, l'évaluation ne propose pas de résultat sur des données hors-apprentissage, il est ainsi difficile de connaître la qualité de la formule sur des mots autres que les 30 couples du protocole initial.

Les formules basées sur Google (*e.g.* Cilibrasi et Vitanyi (2006)) ou Yahoo (*e.g.* E. Iosif and A. Potamianos (2007)) sont potentiellement des mesures de degré de relation sémantique car elles n'utilisent pas de hiérarchie de concepts pour leur calcul, mais uniquement les co-occurences de termes. Malheureusement, elles n'ont pas encore été évaluées sur les protocoles classiques. La formule de co-occurrence classique de Jaccard donne par contre de mauvais résultats sur le test de Finkelstein (évaluation dans Strube et Ponzetto (2006)).

5 Conclusion

Nous avons présenté dans cet article une mesure de degré de relation sémantique basée sur une extension d'interprétation de la formule de Jiang et Conrath (1997) et dont les hypothèses sur les chemins valides sont issus des travaux d'Aleksovski et al. (2006) et Hirst et St-Onge (1998). De plus, nous avons proposé un protocole d'évaluation basé sur WordNet pour les mesures de degré de relation sémantique.

L'évaluation faite sur WordNet 3.0 montre que notre formule fonctionne (corrélation deux fois supérieure aux mesures classiques sur un sous-ensemble de couples), mais que certaines de nos hypothèses doivent être revues. En particulier, l'uniformité des poids sur tous les arcs *part-of*, la gestion des hyponymes et de la spécialisation lors du calcul du chemin et la notion de chemin « sémantiquement correct » qui est trop stricte.

Lors de notre évaluation, nous avons du réduire notre ensemble de couples suite au manque de relation dans WordNet. Nos résultats tendent à montrer qu'il est difficile (actuellement) d'utiliser WordNet comme base de connaissance pour calculer un degré de relation sémantique. Certains travaux (*e.g.* Strube et Ponzetto (2006)) montrent que l'utilisation de base de connaissances contenant plus de relations (tel que Wikipedia[5]) permet d'obtenir de meilleurs résultats sur les mesures relationnelles. Une prochaine étape dans notre travail consistera donc à évaluer notre mesure sur ce type de base. Néanmoins, notre objectif final est de l'évaluer *in-situ* pour l'interprétation sémantique d'un agent conversationnel.

Références

Aleksovski, Z., W. ten Kate, et F. van Harmelen (2006). Exploiting the structure of background knowledge used in ontology matching. In *Proc. Workshop on Ontology Matching in ISWC2006*. CEUR Workshop Proceedings.

Budanitsky, A. et G. Hirst (2006). Evaluating wordnet-based measures of semantic distance. *Computational Linguistics 32*(1), 13–47.

Cho, M., J. Choi, et P. Kim (2003). *An Efficient Computational Method for Measuring Similarity between Two Conceptual Entities*, Volume 2762/2003 of *LNCS - Time Series, Similarity, and Ontologies*, pp. 381–388. Springer Berlin / Heidelberg.

Cilibrasi, R. et P. Vitanyi (2006). Automatic Extraction of Meaning from the Web. In *Proc. IEEE International Symposium on Information Theory*, pp. 2309–2313.

Corby, O., R. Dieng-Kuntz, et C. Faron-Zucker (2004). Querying the Semantic Web with the CORESE search engine. In I. Press (Ed.), *Proc. of the ECAI'2004*, Valencia, pp. 705–709.

E. Iosif and A. Potamianos (2007). Unsupervised Semantic Similarity Computation usingWeb Search Engines. *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*.

Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, et E. Ruppin (2001). Placing search in context : the concept revisited. In *WWW '01 : Proceedings of the 10th international conference on World Wide Web*, New York, pp. 406–414. ACM Press.

[5] `http://fr.wikipedia.org/wiki/Accueil`

Hirst, G. et D. St-Onge (1998). Lexical chains as representation of context for the detection and correction malapropisms. In C. Fellbaum (Ed.), *WordNet : An Electronic Lexical Database*, Chapter 13, pp. 305–332. MIT Press.

Jiang, J. et D. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. on International Conference on Research in Computational Linguistics*, Taiwan, pp. 19–33.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pp. 296–304. Morgan Kaufmann, San Francisco, CA.

Mazuel, L. et N. Sabouret (2007a). Degré de relation sémantique dans une ontologie pour la commande en langue naturelle. In *Plate-forme AFIA, ingénierie des connaissances 2007 (IC 2007)*, pp. 73–83.

Mazuel, L. et N. Sabouret (2007b). Vers une approche générique pour l'interprétation de commandes en langage naturel. In *Rencontre des Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2007)*, pp. 133–149.

Miller, G. et W. Charles (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes 6*(1), 1–28.

Milward, D. et M. Beveridge (2003). Ontology-based dialogue systems. In *Proc. 3rd Workshop on Knowledge and reasoning in practical dialogue systems (IJCAI03)*, pp. 9–18.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pp. 448–453.

Seco, N., T. Veale, et J. Hayes (2004). An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *Proc. ECAI'2004, the 16th European Conference on Artificial Intelligence*, pp. 1089–1090.

Strube, M. et S. Ponzetto (2006). WikiRelate ! Computing semantic relatedness using Wikipedia. *Proc. of AAAI 6*, 1419–1424.

Thieu, M., O. Steichen, E. Zapletal, M. Jaulent, et C. L. Bozec (2004). Mesures de similarité pour l'aide au consensus en anatomie pathologique. *Ingénierie des Connaissances (IC)*.

Yang, D. et D. M. W. Powers (2005). Measuring semantic similarity in the taxonomy of wordnet. In *ACSC '05 : Proceedings of the Twenty-eighth Australasian conference on Computer Science*, Darlinghurst, Australia, Australia, pp. 315–322. Australian Computer Society, Inc.

Zargayouna, H. et S. Salotti (2004). Mesure de similarité dans une ontologie pour l'indexation sémantique de documents XML. *Ingénierie des Connaissances (IC)*.

Summary

The problematic of semantic approximation still needs a lot of work, especially for text disambiguation or search engine for documents. Even if some work on semantic relatedness has been proposed, few have evaluated using classical protocols such as the Miller and Charles test. This paper evaluates our own semantic relatedness measure presented in an earlier paper and shows that it obtains a correlation factor twice as much than the others classical measures.

Distances sémantiques dans des applications de gestion d'information utilisant le web sémantique.

Fabien Gandon*, Olivier Corby*, Ibrahima Diop*, Moussa Lo**

*INRIA Edelweiss, 2004 rt des Lucioles BP93, 06902 Sophia Antipolis
Fabien.Gandon@sophia.inria.fr
http://www-sop.inria.fr/edelweiss/
**LANI, UFR SAT, Université Gaston Berger, Saint-Louis, Sénégal
lom@ugb.sn
http://www.ugb.sn/

Résumé. Nous résumons ici divers utilisations que nous avons faites des distances sémantiques et discutons notre position sur ce domaine. Nous présentons en suite un travail préliminaire sur l'extension de leur utilisation dans des applications reposant au minimum sur des modèles RDF/S.

1 Introduction : intuitivement proches

Intuitivement nous sommes tous portés à dire que le concept de *berline* est plus proche du concept de *monospace* que de celui d'*avion* ; cependant nous pensons aussi que le concept de *berline* est plus proche du concept d'*avion* que du concept de *livre*. Ces distances intuitives peuvent être simulées par exemple pour améliorer les moteurs de recherche du Web dans leurs algorithmes de filtrage et de tri des réponses.

En informatique, une ontologie est une théorie logique partielle rendant explicite une conception de la réalité (Gruber, 93) (Guarino et Giaretta, 1995). Les définitions en intension d'une ontologie sont donc naturellement traduites en des représentations logiques exploitées notamment dans des inférences de dérivation, par exemple pour améliorer le rappel en recherche d'information. Cependant, ces mêmes définitions et leurs relations peuvent être vues comme des espaces, notamment des graphes ou réseaux sémantiques, qui peuvent être dotés de métriques servant de base à toute une autre gamme d'inférences.

L'idée d'évaluer la proximité conceptuelle sur des réseaux sémantiques remonte aux travaux de (Quillian, 1968) et (Collins & Loftus, 1975) sur la mémoire sémantique humaine. La proximité de deux concepts peut venir d'une complémentarité fonctionnelle (ex: un clou et un marteau), d'une similarité fonctionnelle (ex: un marteau et un tournevis), etc. Ce dernier exemple appartient à la famille des similarités sémantiques dans laquelle la proximité est basée sur une caractéristique définitionnelle partagée (ex: être un outil).

Une structure supportant naturellement le raisonnement sur les similarités sémantiques est la hiérarchie des types telle que l'on peut la trouver dans un support en graphes conceptuels, dans la TBox des logiques de description, dans un schéma RDFS, etc. En effet, dans cette structure, les liens de subsomption groupent les types suivant les caractéristiques définitionnelles qu'ils partagent. Lorsqu'elle est appliquée au graphe d'une hiérarchie, une proximi-

té calculée par propagation donne une distance sémantique, la première et la plus simple étant celle qui compte les arcs séparant deux sommets (Rada *et al.*, 1989).

Deux grandes familles d'approches peuvent être identifiées pour le calcul de telles distances: (1) celles qui incluent des informations externes à la hiérarchie, par exemple, des statistiques sur l'utilisation des types de concepts (Resnik, 1995) (Jiang & Conrath, 1997), et (2) les approches reposant uniquement sur les modèles en intensions comme par exemple une hiérarchie de types (Rada *et al.*, 1989)(Wu et Palmer, 1994). Dans le domaine des graphes conceptuels la deuxième approche est utilisée, en particulier pour proposer une projection ne donnant plus uniquement des valeurs booléennes *i.e.* une similarité $S:C^2 \rightarrow [0,1]$ où 1 correspond à la valeur vraie de la projection classique et toute autre valeur donne une idée de la similarité entre le graphe projeté et le graphe source. L'utilisation initiale faite par Sowa visait à permettre des déplacements latéraux dans le treillis des types. (Ralescu et Fadlalla, 1990) l'ont utilisée pour relaxer les contraintes de l'opérateur de jointure. Plus récemment, (Zhong *et al.*, 2002) ont utilisé une distance atténuée par la profondeur des types dans l'ontologie pour construire une mesure de similarité entre graphes conceptuels.

Dans la suite, nous donnons un aperçu des applications et des travaux que nous menons autour de cette notion de proximité ou distance sémantique à travers une opérationnalisation du web sémantique basée sur les graphes conceptuels.

2 Expériences passées pour des distances sur web sémantique

Nous résumons ici les scénarios d'utilisation et les distances utilisées dans un échantillon représentatif de projets que nous avons menés. Pour nous, cette première série d'expériences a permis de démontrer l'intérêt et le potentiel des distances, et aussi de souligner l'importance du travail restant à faire pour identifier et caractériser les familles de distances existantes et leur adéquation respective aux tâches pour lesquelles nos utilisateurs souhaitent être assistés. Trois expériences sont présentées chronologiquement et discutées dans la quatrième section.

2.1 Distance et bases de connaissances distribuées

Dans un web sémantique d'entreprise les scénarios amènent souvent la contrainte de bases d'annotations distribuées (assertions RDF[1] à propos de ressources intranet). Pour gérer cette distribution nous avons proposé une architecture et des protocoles permettant en particulier de maintenir la spécialisation des bases d'annotations quant aux sujets abordés dans leurs assertions (Gandon, 2002).

Chaque archive de notre architecture maintient une structure appelée ABIS (Annotation Base Instances Statistics) décrivant des statistiques sur les types de triplets (relations binaires imposées par le modèle RDF) présents dans leur base d'annotations. Par exemple si, dans l'ontologie, il existe une propriété Auteur avec la signature:

`[Document] → (Auteur) → [Personne]`

L'ABIS pourra contenir des statistiques sur l'existence des instances de triplets suivantes:

`[Article] → (Auteur) → [Etudiant]`
`[Livre] → (Auteur) → [Philosophe]`
…

[1] http://www.w3.org/RDF/

L'ABIS est construit lors de la transformation des annotations RDF en graphes conceptuels et capture la contribution d'une archive à la mémoire globale en terme de types de connaissances contenues dans cette archive. L'ABIS fournit un moyen de comparer le contenu de deux bases et nous l'utilisons pour maintenir la spécialisation des bases d'annotations grâce à une distance sémantique définit entre un ABIS et une nouvelle annotation.

Pour comparer deux types primitifs, nous utilisons la distance de (Rada *et al.*, 1989) comptant le nombre d'arcs sur le chemin le plus court qui relie ces deux types à travers la hiérarchie; voir formule (1). En utilisant cette distance on peut définir une distance entre deux triplets RDF (ou deux instances d'une relation binaire), comme étant la somme des distances entre: les types des deux relations, les types des deux concepts en premier argument (domain) et les types des deux concepts en deuxième argument (range); voir formule (2). La distance entre un triplet et un ABIS est alors définie comme la distance minimale entre ce triplet et les triplets recensés par l'ABIS; voir formule (3). Et finalement, la distance entre une annotation et un ABIS est la somme des distances entre chaque triplet de l'annotation figurant dans l'ABIS; voir formule (4).

$$dist(t_1,t_2) = length(t_1,lcst(t_1,t_2)) + length(t_2, lcst(t_1,t_2)) \quad (1)$$

où , *lcst(t1,t2))* est le plus proche supertype commun de t_1 et t_2.

$$\begin{aligned} dist(triple1,triple2) = \\ & dist(domain(triple1), domain(triple2)) + \\ & dist(predicate(triple1), predicate(triple2)) + \\ & dist(range(triple1), range(triple2)) \end{aligned} \quad (2)$$

$$dist(triple, ABIS) = \min_{triple_i \in ABIS} (dist(triple, triple_i)) \quad (3)$$

$$dist(An, ABIS) = \sum_{triple_j \in An} dist(triple_j, ABIS) \quad (4)$$

Cette distance donne une fonction d'évaluation / fonction de coût utilisée comme critère dans un protocole de mise aux enchères des nouvelles annotations à archiver : chaque nouvelle annotation est mise aux enchères entre les archives existantes; chaque archive fait une offre qui correspond à la distance entre son ABIS et l'annotation; l'archive avec l'offre la plus petite gagne l'annotation. Ce protocole permet de maintenir la spécialisation des bases et ainsi de faciliter l'optimisation de la résolution de requêtes distribuées en utilisant les ABIS pour la décomposition et le routage des projections.

Dans ce premier exemple, la définition d'une distance conceptuelle sur la hiérarchie des types permet de construire un consensus calculatoire (distance) au dessus du consensus ontologique (hiérarchie), et de l'utiliser dans un consensus protocolaire (enchères). Initialement utilisée pour un protocole de mémoire distribuée, la section suivante explique comment cette distance a ensuite été intégrée au moteur de recherche de chaque base pour proposer une nouvelle fonctionnalité: la recherche approchée de connaissances par relaxation des contraintes de typage.

2.2 Distance et projection de graphes approchée

La plateforme CORESE (Corby et al, 2006) intègre une fonctionnalité de recherche approchée qui démontre une autre application des inférences simulant la proximité conceptuelle. CORESE utilise une extension de la distance atténuée par la profondeur (Zhong *et al.*, 2002) des types dans le treillis de l'ontologie; voir formules (5) et (6).

$$\forall (t_1,t_2) \in {H_c}^2; t_1 \le t_2 \text{ on a } l_{H_c}(t_1,t_2) = \sum_{\{t \in \langle t_1,t_2 \rangle, t \neq t_1\}} \left[\frac{1}{2^{depth(t)}} \right] \quad (5)$$

avec H_c la hiérarchie des types de concepts, le chemin le plus court entre t_1 et t_2, et *depth(t)* la profondeur de *t* dans l'ontologie *i.e.* le nombre d'arcs sur le chemin le plus court entre *t* et la racine ⊤

$$\forall (t_1,t_2) \in {H_c}^2 \text{ on a } dist(t_1,t_2) = \min_{\{t \ge t_1, t \ge t_2\}} \left(l_{H_c}(t_1,t) + l_{H_c}(t_2,t) \right) \quad (6)$$

En utilisant cette distance, on peut relaxer la contrainte d'égalité ou de spécialisation des types lors de la projection en la remplaçant par une contrainte de proximité utilisant une distance conceptuelle comme celle définie en (6). On obtient alors une projection approchée pour l'appariement de graphes requêtes (ex: un motif recherché par un utilisateur) et de graphes faits (ex: une annotation RDF) ; une telle projection préserve l'adjacence et l'ordre des arcs mais permet de relaxer les contraintes de typage. Dans ce deuxième exemple, la distance est utilisée pour remplacer une contrainte logique ($t_1(x) \Rightarrow t_2(x)$) par une contrainte numérique ($d(t_1,t_2) < seuil$). Cette transformation permet de relaxer une requête donnée par un utilisateur lorsqu'elle ne donne pas (suffisamment) de résultats. Pour accéder aux connaissances d'une base, une alternative aux requêtes est la navigation dans les annotations. La section suivante montre comment, là encore, les distances peuvent être utilisées.

2.3 Distance et ultra-métrique de clustering

Dans le cadre du projet KmP nous nous sommes intéressés à la construction d'un algorithme de regroupement (clustering) des compétences présentes sur la Télécom Valley de Sophia Antipolis et annotées en RDF (Gandon *et al.*, 2006). Le regroupement normalement effectué manuellement par les experts en gestion s'est révélé être un algorithme de regroupement monothétique (monothetic clustering). La représentation recherchée demandait de pouvoir fournir des moyens de contrôler simplement le niveau de détail et de granularité choisi pour générer le regroupement. En analyse de données (Jain *et al.*, 1999), une structure classique supportant le choix des niveaux de détail est le dendrogramme, un arbre qui, à chaque niveau de coupure, donne une solution de regroupement plus ou moins fin.

Un dendrogramme repose sur une ultra-métrique c'est-à-dire une distance avec une inégalité triangulaire sur-contrainte : $\forall t', t_1, t_2 \in {H_c}^3 \quad dist(t_1,t_2) \le max(dist(t_1,t'), dist(t_2,t'))$

Nous avons donc cherché à construire cette ultramétrique à partir de la distance sémantique de CORESE. Nous n'avons considéré pour cela que des hiérarchies en arbres simples, ce qui nous donne une distance exacte ayant pour formule (7).

$$dist(t_1,t_2) = \frac{1}{2^{depth(lcst(t_1,t_2))-2}} - \frac{1}{2^{depth(t_1)-1}} - \frac{1}{2^{depth(t_2)-1}} \quad (7)$$

Nous avons ensuite proposé une transformation produisant une ultramétrique et améliorant le nombre de niveaux de détail disponibles dans le dendrogramme obtenu en favorisant le regroupement des classes ayant une descendance peu profonde, formule (8) et figure 1.

$$dist_{CH}(t_1,t_2) = \max_{\forall st \le lcst(t_1,t_2)} \left(dist\left(st, lcst(t_1,t_2)\right)\right) \text{ quand } t_1 \neq t_2 \quad (8)$$

$$dist_{CH}(t_1,t_2) = 0 \text{ quand } t_1 = t_2$$

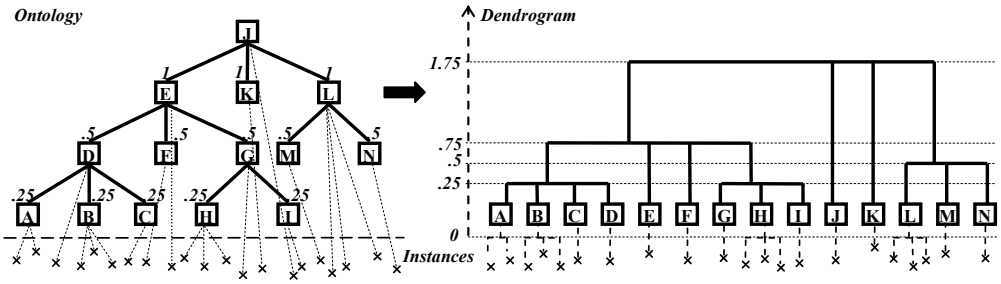


Figure 1. Transformation de la distance en ultramétrique.

Le regroupement suivant la hiérarchie des types, on peut nommer chaque regroupement. Un exemple de regroupement des compétences sur la Télécom Valley est donné en Figure 2.

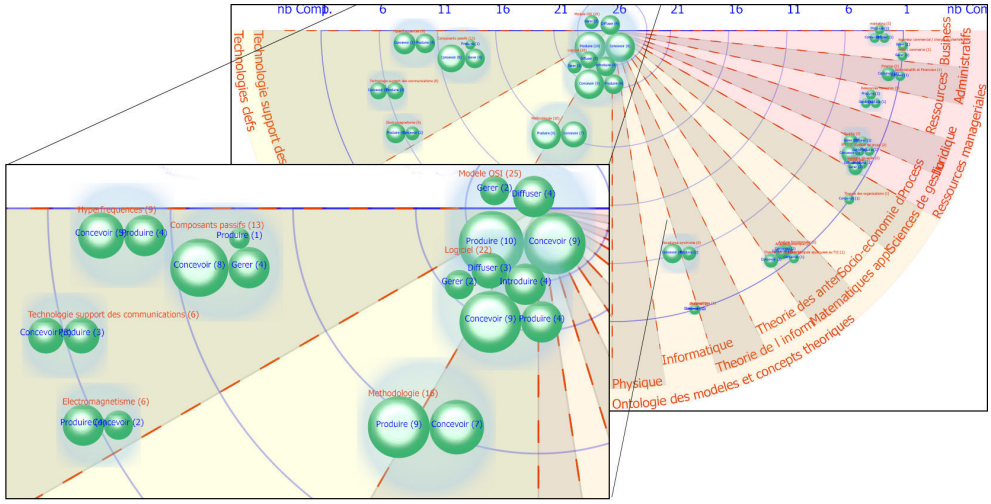


Figure 2. Vue en Radar sur 180° des regroupements de compétences.

2.4 Une première discussion sur les distances et leur utilisation

En parallèle avec ces trois premières explorations des caractéristiques, interprétations et applications des distances conceptuelles, nous avons commencé à questionner la valeur de ces distances et leur fidélité par rapport aux proximités naturellement ressenties par les humains. Pour cela nous avons commencé une étude empirique et statistique. La première hypothèse testée fut *"Est-il juste de considérer que les frères sont à égale distance du père et à égale distance les uns des autres ou est-ce un effet secondaire du fait que l'on repose sur la structure des chemins de subsomption?"*.

Afin d'étudier ces distances dans leur milieu naturel et de les comparer avec leurs simulations informatiques, nous avons conçu une plateforme permettant de réaliser, gérer, et d'ana-

lyser des expériences où les participants organisent et regroupent spatialement des concepts selon leur proximité intuitive (Boutet *et al.*, 2005); voir figure 3.

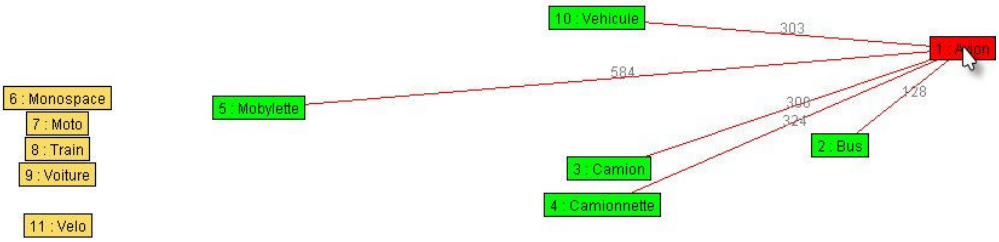


Figure 3. Applet de l'exercice de placement

A partir de ces exercices, des analyses statistiques sont faites. Prenons l'exercice de la figure 3 effectué par 30 participants de 13 à 50 ans. Les distances capturées ont été normalisées avant d'en calculer la moyenne, l'écart type et la variance. La figure 4 montre les distances entre le concept *camion* et la liste des autres concepts à placer.

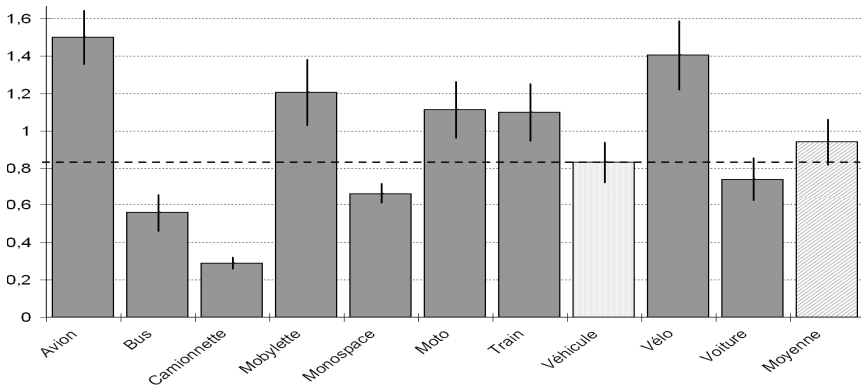


Figure 4. Distances entre Camion et d'autres véhicules

On peut lire sur ce graphique que le concept *camionnette* est en moyenne très proche de *camion* et, étant donné que la variance est très faible, qu'il s'agit d'un consensus. Le concept *véhicule* est particulier dans cette liste puisque dans une ontologie il serait naturellement placé comme père (ou ancêtre) des autres concepts et que ces autres concepts seraient entre eux des frères (ou des cousins). On voit notamment que la distance entre *camion* et ses frères est parfois plus petite (4 cas) et parfois plus grande (5 cas) que la distance à son père ou ancêtre *véhicule* ce qui serait impossible à faire dans un hiérarchie de types classique comme celles utilisées dans la majorité des distances puisque l'on passe forcément par le père pour aller à un frère. Il reste beaucoup à faire dans cette étude mais si ces résultats se confirmaient, ils montreraient qu'une structure de subsomption seule ne permet pas de simuler de tels comportements.

3 Expériences en cours d'extension des distances

Nous sommes convaincus que la tâche devant être assistée par un système conditionne la ou les distances employées et leur combinaison éventuelle. Notre démarche actuelle est donc

d'essayer les différents espaces métriques disponibles. Dans le cadre de plusieurs projets en cours, nous explorons actuellement de nouvelles distances ou extensions de distances. Les sections suivantes donnent des descriptions préliminaires de ces travaux en cours. Nous l'avons vu en introduction, il existe deux grandes approches pour calculer les distances: utiliser des informations extérieures aux modèles des connaissances (ex: statistiques sur des corpus de textes) ou reposer uniquement sur la structure des modèles (ex: la hiérarchie des types). Nous observons actuellement comment ces approches se déclinent sur l'utilisation d'informations extérieures (extraites d'une base d'annotations RDF) et sur des extensions de la structure habituellement exploitée dans les modèles (ici la hiérarchie de types en RDFS).

3.1 Distance de cooccurrence et contexte en extension

Notre premier essai consiste à considérer la proximité d'usage de deux types *i.e.* la fréquence avec laquelle ces deux types sont employés ensemble dans des descriptions. Soient deux types de concepts $t_x,t_y \in H_c^2$ on définit le comptage de cooccurrences comme le nombre de triplets impliquant ces deux types:

$$count(t_x,t_y) = || \{t \text{ triplets RDF} \mid t = (x,T_p,y) \wedge (x,rdf{:}type,T_x) \wedge (y,rdf{:}type,T_y)\} ||$$

On définit alors une distance inverse:

$$dist_{count}(T_x,T_y) = \frac{1}{1+count(T_x,T_y)} \in [0,1]$$

Cette distance capture une proximité d'usage des types. En effet, comme nous le disions en section 2.1, une même signature de relation peut engendrer bien des familles de triplets par spécialisation des types spécifiés. De plus, détail important, la signature en RDFS est utilisée pour de l'inférence de type (ajout supplémentaire de types) et non de la validation de type ; par conséquent la variété effective des types sur les instances desquels une propriété est utilisée peut être bien plus grande que la liste des types obtenue par la fermeture transitive de la subsumption de ses domaine et co-domaine.

Une utilisation immédiate de cette distance est d'assister les utilisateurs dans des interfaces de navigation (ex: tag cloud figure 5), de requête ou d'annotation en suggérant des types co-occurrents au dernier type sélectionné (figure 6).

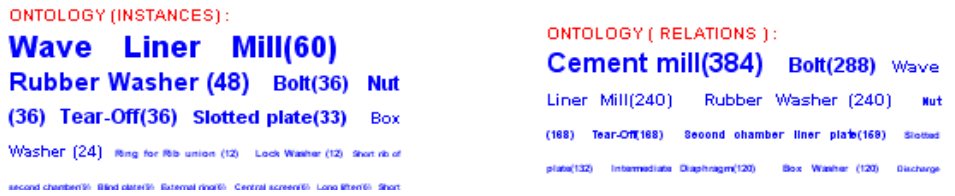


Figure 5. Nuages de termes obtenus avec la distance

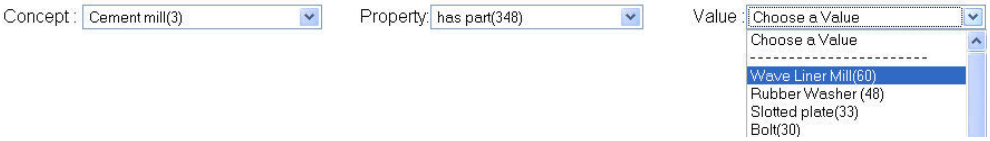


Figure 6. Suggestion dans l'interface en utilisant la distance pour ordonner les options

3.2 Distance de signatures et contexte en intension

L'ontologie capture des caractérisations en intension des concepts et des relations du domaine, et en particulier des relations en intension entre ces concepts. Nous l'avons vu, l'espace le plus utilisé par les distances à base d'ontologies est le graphe de la hiérarchie des types de concepts. Nous nous intéressons ici à augmenter ce graphe avec les chemins de la hiérarchie des types de relations et les chemins des signatures des relations.

Formellement, on repose sur une première extension du méta-modèle RDFS introduisant une propriété symétrique parente directe du domaine et co-domaine et appelée signature (définition 1).

Définition 1: la propriété `cos:signature` est telle que

`rdfs:domain`(T_p, T_x) ⇒ `cos:signature`(T_p, T_x)

`rdfs:range`(T_p, T_x) ⇒ `cos:signature`(T_p, T_x)

`cos:signature`(T_x,T_y) ⇔ `cos:signature`(T_x,T_y)

Définition 2: un chemin de signatures $C_s(T_x,T_y)$ entre les types $T_x,T_y \in H_C^2$ est un chemin de T_x à T_y composé exclusivement d'arcs de type `cos:signature` inférés à partir des signatures de H_R : $C_S(T_x,T_y):=< T_x$, *signature*, T_1, *signature*, T_2, *signature*,...*signature*, T_n, *signature* $T_y>$

Définition 3: une distance de signature $d_S(T_x,T_y)$ entre les types $T_x,T_y \in H_C^2$ est définie par $d_S(T_x,T_x):=0$ et $d_S(T_x,T_y):= min_{\{Ci \in \{ Cs(Tx,Ty) \}\}}\ long(C_i)$ avec
$long(<T_x$, *signature*, T_1, *signature*, T_2, *signature*, ... *signature*, T_n, *signature* $T_y >) := n$

Intuitivement, avec cette distance, deux types sont proches s'il y a (en intension) une possibilité de les impliquer dans un graphe d'annotation concis. Par exemple *document* et *pays* peuvent être proches dans une ontologie s'il existe les déclarations suivantes (et les implications par subsomption et symétrie):

rdfs:domain(auteur, document) ⇒ *cos:signature(document, auteur)*
rdfs:range(auteur, personne) ⇒ *cos:signature(auteur, personne)*
rdfs:domain(nationalité, personne) ⇒ *cos:signature(personne, nationalité)*
rdfs:range(nationalité, pays) ⇒ *cos:signature(nationalité, pays)*
soit $d_S(document, pays)=3$

Cette première version permet de comprendre le principe mais ne rend pas compte de la subsomption des types de relations et des types de leur signature; en d'autres termes si une relation (ex: titre) spécifie un type (ex: document) dans sa signature, elle inclut les sous-types de ce type (ex: livre, roman, etc.) et cette spécification se propage au sous-type de la relation (ex: sous-titre, titre court, etc). Pour rendre compte de ce point nous avons défini et nous expérimentons la distance suivante (définition 4).

Définition 4: la propriété `cos:sous-type-et-signature` est telle que

`cos:signature`(T_p, T_x) ⇒ `cos:sous-type-et-signature`(T_p, T_x)

`rdfs:subClassOf`(T_p, T_x) ⇒ `cos:sous-type-et-signature`(T_p, T_x)

`rdfs:subPropertyOf`(T_p, T_x) ⇒ `cos:sous-type-et-signature`(T_p, T_x)

`cos:sous-type-et-signature`(T_x,T_y) ⇔ `cos:sous-type-et-signature`(T_x,T_y)

Les définitions du chemin et de la distance sont des adaptations des définitions 2 et 3 *mutatis mutandis*.

Une utilisation testée actuellement pour cette distance est le pilotage de l'analyse de la langue naturelle utilisée pour lever l'ambigüité dans la génération automatique des annotations à partir de textes. Il s'agit de prédire les patrons d'annotation les plus probables pour l'extraction de connaissances d'un texte donné en fonction d'une ontologie donnée. La figure 7 montre une expérience dans laquelle on passe en neuf étapes par pondération d'une distance n'utilisant que les chemins de subsomption à une distance n'utilisant que des chemins de signatures. Pour chaque jeu de poids la courbe donne la précision de la désambiguïsation pour 10 termes apparaissant chacun dans 100 documents. Le point intéressant ici est que le meilleur résultat est obtenu pour un espace métrique mixte.

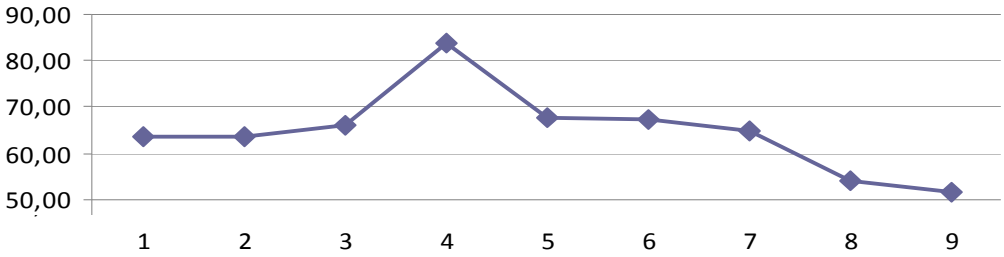


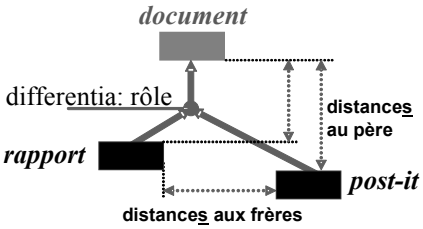
Figure 7. Précision de la désambiguïsation entre subsomption et signature

4 Discussion : intuitivement loin

Les distances conceptuelles laissent de nombreuses questions de recherche nourries par un ensemble grandissant d'applications, en particulier:

Proximité naturelle vs. distance mathématique: la distance utilisée par CORESE dans sa recherche approchée est une semi-distance *i.e.* elle ne vérifie pas l'inégalité triangulaire. Quelle est la valeur des conditions nécessaires de la définition mathématique des distances? Devons nous à tout prix essayer de les respecter ou est-ce simplement une limite de la métaphore des distances? Quelles seraient sinon les caractéristiques définitionnelles d'une distance conceptuelle?

Distance conceptuelle vs. espaces métriques: les structures de la hiérarchie de types sont le terrain favori pour la définition des distances conceptuelles. Faut-il considérer des représentations plus riches (Bachimont, 2000) incluant, par exemple, des liens frère-frère, qui permettraient de mieux simuler de telles distances? Comment mieux recenser et définir ces espaces métriques *i.e.* les espaces pertinents et les métriques adaptées? Comment étudier les différentes familles de distances qui semblent cohabiter dans nos inférences au quotidien? Comment les capturer, les apprendre, pour les utiliser dans des inférences de recherche d'information?



S'il est clair que depuis près de trente ans ces similarités ne cessent de resurgir dans l'exploitation de modèles et d'espaces de représentation, intuitivement, il nous semble qu'il reste encore beaucoup à faire pour identifier, étudier, caractériser et simuler ces similarités, dans la continuité des travaux de Blanchard *et al*, 2005.

Distances sémantiques dans des applications utilisant le web sémantique.

Références

Bachimont B., (2000) *Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances,* In "Ingénierie des connaissances Evolutions récentes et nouveaux défis", Jean Charlet, Manuel Zacklad, Gilles Kassel, Didier Bourigault; Eyrolles, ISBN 2-212-09110-9

Blanchard E., Harzallah M., Briand H., Kuntz, P., (2005), *A Typology Of Ontology-Based Semantic Measures*, Workshop EMOI-INTEROP at CAISE'05.

Boutet, M., Canto, A., Roux, E., (2005) *Plateforme d'étude et de comparaison de distances conceptuelles*, Rapport de Master, Ecole Supérieure En Sciences Informatiques,

Collins, A., Loftus, E., (1975) *A Spreading Activation Theory of Semantic Processing*. Psychological Review, vol. 82, pp. 407-428,

Corby O., Dieng-Kuntz R., Faron-Zucker C., Gandon F., (2006) *Searching the Semantic Web: Approximate Query Processing Based on Ontologies*, IEEE Intelligent Systems, January/February (Vol. 21, No. 1), pp. 20-27, ISSN: 1541-1672.

Gandon F., (2002) *Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web*, PhD Thesis., INRIA

Gandon F., Corby O., Giboin A., Gronnier N., Guigard C., (2005) *Graph-based inferences in a Semantic Web Server for the Cartography of Competencies in a Telecom Valley*, ISWC, Lecture Notes in Computer Science LNCS 3729, Galway,

Gruber, T.R. (1993). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, In Formal Ontology in Conceptual Analysis and Knowledge Representation, edited by Nicola Guarino and Roberto Poli, Kluwer Academic Publishers

Guarino N., Giaretta P., (1995) *Ontologies and Knowledge Bases: Towards a Terminological Clarification.* In N. J. I. Mars (ed.), Towards Very Large Knowledge Bases, IOS Press.

Jain, A.K., Murty, M.N., and Flynn, P.J. (1999) *Data Clustering: A Review*, ACM Computing Surveys, 31(3) 264-323.

Jiang, J., Conrath, D., (1997) *Semantic Similarity based on Corpus Statistics and Lexical Taxonomy*. In Proc. of International Conference on Research in Computational Lin-guistics, Taiwan,

Quillian, M.R., (1968) *Semantic Memory*, in: M. Minsky (Ed.), Semantic Information Processing, M.I.T. Press, Cambridge.

Rada, R., Mili, H., Bicknell, E., Blettner, M., (1989) *Development and Application of a Metric on Semantic Nets*, IEEE Transaction on Systems, Man, and Cybernetics, vol. 19(1), pp. 17-30.

A.L. Ralescu, A. Fadlalla, (1990) *The Issue of Semantic Distance in Knowledge Representation with Conceptual Graphs*, In Proc. Of AWOCS90, pp. 141-142,

Resnik, P., (1995) *Semantic Similarity in a Taxonomy: An Information-Based Measure and its Applications to Problems of Ambiguity in Natural Language*. In Journal of Artificial In-telligence Research, vol 11, pp. 95-130,

Sowa., J.F., (1984) *Conceptual structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, Massachusetts,

Wu Z, Palmer, M. (1994) Verb Semantics and Lexical Selection. In Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico.

J. Zhong, H. Zhu, J. Li, Y. Yu. (2002) *Conceptual Graph Matching for Semantic Search*, In Proc. of 10th International Conference on Conceptual Structures, ICCS2002, LNCS 2393, Springer Verlag, pp. 92-106, Borovets, Bulgaria,

Mesures sémantiques pour la comparaison des « constructs » des langages de modélisation d'entreprise

Mounira Harzallah*, Emmanuel Blanchard* Giuseppe Berio**, Pascale Kuntz*

*LINA, Université de Nantes
mounira.harzallah@univ-nantes.fr,
http://www.polytech.univ-nantes.fr/COD/
**Dépt Informatique de Turin, Italie
Giuseppe.berio@di.unito.it
http://www.di.unito.it/~berio/

Résumé. La comparaison « sémantique » des objets est devenue une opération nécessaire dans de nombreux domaines. Dans cet article, nous nous intéressons à la comparaison des éléments de base constitutifs des langages de modélisation d'entreprise (les « constructs ») en utilisant l'approche UEML et son exploitation des mesures sémantiques. Après une présentation de l'approche UEML, nous présentons l'adaptation des mesures pour comparer deux concepts d'une ontologie à la comparaison de deux constructs. Et nous montrons l'utilisation des ces mesures pour évaluer l'application UEML : l'ontologie résultante, l'alignement d'un construct avec cette ontologie et l'espace d'information des mesures utilisées.

1 Introduction

La comparaison « sémantique » des objets est devenue une opération nécessaire dans de nombreux domaines (e.g. la gestion des documents, la bioinformatique, la recherche d'information, la gestion des compétences (Baziz et al., 2007), (Zargayouna et Salotti 2004) (Lord et al., 2003), (Corbey et al., 2004)). On identifie si ces objets sont similaires ou non, si un objet compose un autre objet, s'il peut le remplacer, etc. Pour cela, ces objets sont définis (ou caractérisés) à l'aide de l'ontologie de leur domaine. Ensuite, leurs définitions sont comparées. Il s'agit de la comparaison de l'annotation de ces objets avec une ontologie.

Dans le domaine de la modélisation d'entreprise sur lequel nous nous focalisons dans cet article, une problématique actuelle majeure concerne l'interopérabilité des modèles de représentation définis avec des langages différents. Il est alors nécessaire de définir des correspondances –i.e. des relations sémantiques de comparaison- entre les « constructs » de ces langages. Le terme « construct » qui est couramment employé dans ce domaine définit « un élément de base constituant un langage et utilisé pour définir des modèles dans ce langage». L'objectif d'identification des correspondances entre constructs est d'effectuer des opérations, par exemple de translation ou de fusion, sur des modèles définis avec les langages définis autour de ces constructs (Berio et al., 2004a). Dans les approches dirigées par les modèles qui visent le développement, le déploiement et la gestion du logiciel d'entreprise, des mécanismes sont disponibles pour représenter des correspondances entre ces constructs mais il n'y a pas de support spécifique pour les découvrir (Berio et al., 2004b).

Dans cet article, nous nous intéressons à l'identification des correspondances entre constructs en utilisant l'approche UEML (Unified Enterprise Modelling Language) (Interop,

2005), et plus particulièrement son exploitation des mesures de comparaison des concepts dans une ontologie.

La suite de cet article est organisée comme suit. D'abord, nous exposons brièvement l'approche UEML et son application expérimentale. Ensuite, nous présentons des mesures de comparaison existantes et leur adaptation à la comparaison des constructs en utilisant l'ontologie UEML. Nous présentons ensuite l'outil de comparaison de constructs que nous avons développé et nous montrons, à l'aide d'exemples, l'utilisation des mesures de comparaison pour analyser l'application de l'approche UEML.

2 L'approche UEML

L'objectif de UEML est de supporter l'utilisation intégrée des modèles d'entreprises définis dans des langages différents. UEML est conçue comme un mécanisme pour interconnecter des langages différents et leurs modèles. UEML comprend (Opdahl et Berio, 2006) :

- un méta-méta modèle pour organiser la description des différents aspects d'un construct,
- une ontologie permettant de décrire la sémantique des constructs,
- un cadre pour définir et évaluer la qualité des langages de modélisation d'entreprise pour aider à sélectionner les langages à décrire,
- une approche d'analyse des correspondances pour déterminer les correspondances sémantiques entre les constructs,
- un ensemble d'outils pour aider à l'utilisation des descriptions des langages considérés.

UEML propose une nouvelle approche pour décrire les langages de modélisation, leurs types de diagrammes et leurs constructs. L'objectif de cette approche est d'incorporer les langages existants et leurs constructs dans ce qui est nommé « web de langages », qui se définit comme un ensemble de langages sélectionnés par rapport à leur qualité d'une façon standardisée, intégrante et évolutive. Lors de la description d'un langage en informatique, on distingue souvent trois concepts : sa syntaxe concrète, sa syntaxe abstraite et sa sémantique. Pour les langages de modélisation, ces trois concepts sont renommés respectivement : présentation (visualisation d'un construct), représentation (le méta-modèle de sa structure et ses relations) et alignement de la représentation (qui décrit comment les éléments de la représentation sont alignés avec les concepts de l'ontologie).

Un construct d'un langage est donc intégré dans ce web de langages par la définition de sa présentation, sa représentation et l'alignement de cette dernière avec l'ontologie UEML. En comparant les alignements des représentations des constructs, on peut identifier les relations sémantiques entre ces constructs. Tous les constructs incorporés dans ce web de langages sont par conséquent inter-reliés au niveau le plus détaillé via l'ontologie UEML.

2.1 L'ontologie UEML

L'ontologie UEML a été développée à partir de l'ontologie de Bunge et le modèle BWW (Bunge, 1979 ; Wand et Weber, 1993) et enrichie avec d'autres concepts pour mieux couvrir la sémantique des constructs incorporés dans le web de langages de UEML. Elle a été conçue et structurée selon quatre concepts centraux : Classe, Propriété, Etat et Transformation. Chaque concept est spécialisé en d'autres concepts pour former une hiérarchie. Ces quatre hiérarchies sont liées par des relations entre leurs concepts. Les types de ces relations sont

bien définis : une classe peut avoir une ou plusieurs propriétés, un état est défini par des propriétés et une transformation est définie par des états avant et après. Actuellement, les hiérarchies des classes (35 concepts) et propriétés (56) sont assez développées pour les exploiter, ce qui n'est pas le cas des deux autres hiérarchies (moins que 9 concepts par hiérarchie). La figure 1 représente la hiérarchie des classes de l'ontologie UEML après son enrichissement avec l'incorporation des langages suivants utilisés en modélisation d'entreprise : RdPC, GRL, KAOS, ISO/DIS 19440, UML 2.0, BPMN et IDEF3.

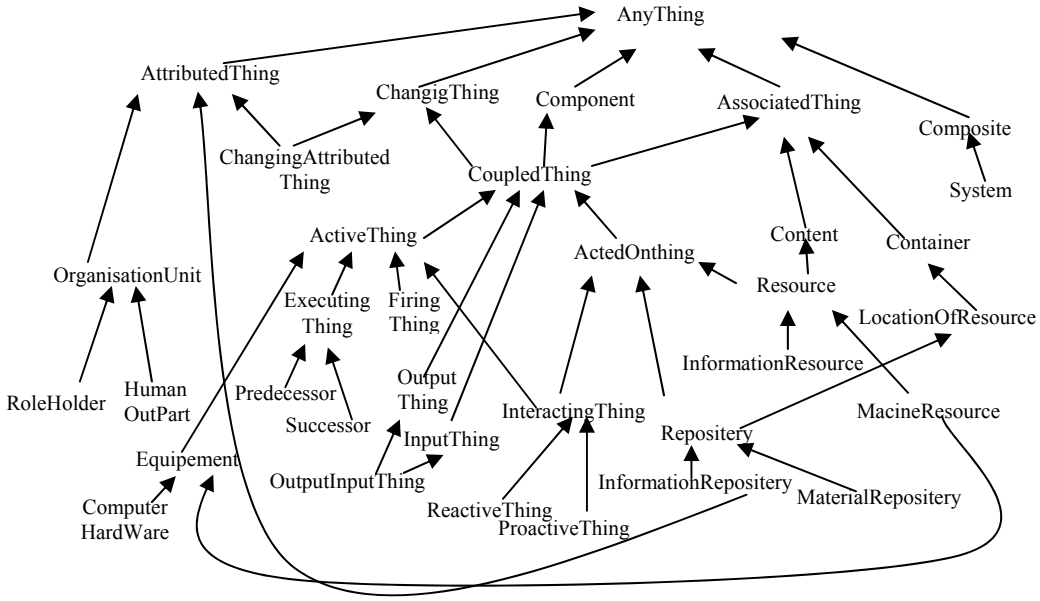


Figure 1 : Hiérarchie des classes de UEML

2.2 Représentation du construct UOB et son alignement

Nous illustrons dans cette section à l'aide de l'exemple du construct UOB (Unit of Behaviours) du langage IDEF3 (Knowledge Based Systems, 2006), le résultat de l'incorporation d'un construct dans le web de langage UEML. D'abord, l'incorporation de ce construct n'avait pas enrichi l'ontologie UEML déjà construite : l'analyse de sa sémantique ne nous a pas suggéré d'étendre l'ontologie UEML. La partie en haut de la figure 2 montre la représentation de ce construct, la partie en bas, les concepts et les relations de l'ontologie UEML qui ont été nécessaires pour décrire l'alignement de sa représentation. Les concepts de l'ontologie utilisés pour son alignement sont trois classes (ExecutingThing, System et ActedOnThing) liées aux propriétés (NamedProperty, ValueProperty, SystemPartwholeRelation, Precondition, PostCondition, ActingOnRelation, IsActive), la propriété IsActive définissant les états Active et NonActive (Harzallah et al., 2007).

Le résultat de l'alignement de la représentation de l'UOB, et d'un construct en général, est un graphe orienté dont les sommets représentent les classes, les propriétés, les états et les transformations, et les arcs représentent les relations entre ces différentes composantes dans l'ontologie UEML. Nous pouvons considérer que le graphe obtenu constitue une annotation

du construct avec l'ontologie UEML. D'habitude, on annote un objet avec un concept, un groupe de concepts ou un template de l'ontologie.

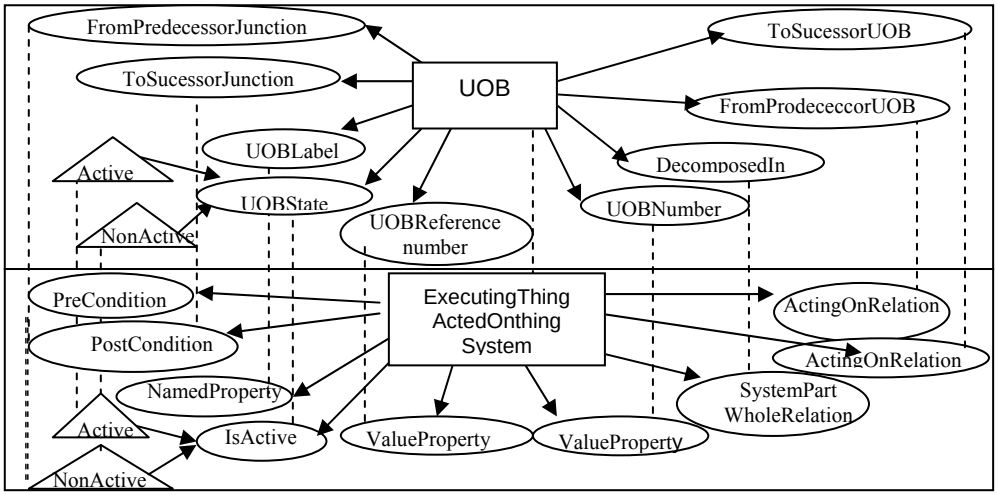


Figure 2 : La représentation du construct UOB et son alignement

3 Mesures sémantiques de comparaison et constructs

Pour évaluer les alternatives de modélisation qui se présentent lors de l'utilisation de langages différents, il est nécessaire de pouvoir comparer leurs paires de constructs. Comme nous l'avons souligné dans le paragraphe précédent, cette comparaison peut se poser comme un problème de comparaison des graphes associés à leurs représentations. Dans cet article, nous restreignons à la comparaison des ensembles de concepts (les nœuds) qui composent les graphes – et non à la comparaison des graphes-. Pour cet objectif, nous proposons de recourir à des mesures que nous développons ci-dessous.

3.1 Mesures sémantiques de comparaison de deux concepts

De nombreuses mesures sémantiques ont été définies dans la littérature. Blanchard et al. (2007) les ont classées en trois classes :

- Mesures utilisant seulement la structure hiérarchique de l'ontologie (Rada et al., 89), (Wu et Plamer, 94), (Sussna, 93);
- Mesures utilisant la structure hiérarchique de l'ontologie et une extension (e.g. un corpus du domaine, des instances des concepts de l'ontologie (Resnik, 93), (Jiang Conrath, 97) et (Lin, 98) ;
- Mesures utilisant la structure hiérarchique de l'ontologie et une intension (les caractéristiques intrinsèques ou les attributs des concepts de l'ontologie).

La majorité de ces mesures est basée sur le principe qui considère que la fonction de comparaison de deux objets dépend de leur information en commun et de l'information qui caractérise chacun d'entre eux. Ce principe est bien connu en taxonomie numérique (Sokal et Sneath, 63). Il a été également utilisé en psychologie dans le modèle du contraste de Tversky

(77). Considérons deux ensembles A et B, et une fonction f qui permet d'associer une valeur numérique à un ensemble (par exemple sa cardinalité). Une première mesure M1(A,B) a été définie par

$$M1(A, B)= \theta\, f(A\cap B) - \alpha f(A-B) - \beta f(B-A) \qquad (1)$$

où β, α et θ sont trois réels positifs : α et β permettent de pondérer l'importance distinctive de chaque ensemble par rapport à l'autre et θ l'importance leur partie commune. Une version normalisée de cette mesure M2(A,B) peut se définir par

$$M2(A,B) = \frac{f(A \cap B)}{\alpha f(A - B) + \beta f(B - A) + f(A \cap B)} \qquad (2)$$

Ces deux mesures peuvent être adaptées au problème de la comparaison sémantique de deux concepts : A et B deviennent deux concepts C_1 et C_2 d'une ontologie et f devient une fonction qui quantifie l'information que porte chaque concept. Dans ce cas, $f(C_1 \cap C_2)$ mesure l'information commune portée par ces deux concepts et, $f(C_1-C_2)$ et $f(C_2-C_1)$ mesurent l'information spécifique que porte chaque concept par rapport à l'autre.

Pour quantifier l'information d'un concept C d'une ontologie, Resnik (95) a proposé d'évaluer le contenu informationnel (IC(C)) par l'entropie de Shannon : la probabilité d'existence P(C) d'une instance de ce concept C est $IC(C) = -\log P(C)$. Cette probabilité peut être estimée en recourant à un corpus : on considère alors la fréquence de l'occurrence de C dans le corpus ainsi que celle des concepts qui le subsument. Blanchard et al. (2008) ont récemment proposé d'autres estimations de P(C) qui sont fonction uniquement de l'information structurelle associée à la taxonomie et ne nécessitent pas de corpus additionnel. Ces estimations varient selon les hypothèses faites sur la distribution des instances de l'ontologie sur ses différents concepts. Nous discutons de trois hypothèses dans la section 4.

Notons que la définition M1 (équation 1) est suffisamment générique pour s'adapter à différentes problématiques de la comparaison : les valeurs de β, α et θ permettent de définir le type de la comparaison requise (e.g. une similarité, une inclusion) et la fonction f peut s'adapter au type des données disponibles reliées à l'ontologie (e.g. structure hiérarchique, le corpus, les instances). Par exemple, pour la définition M2 (équation 2) quand α = β, les mesures obtenues sont des mesures symétriques permettant l'évaluation de la similarité de deux concepts. Quand $\alpha \neq \beta$, l'ensemble de ces mesures sont des mesures asymétriques permettant l'évaluation de l'inclusion ou de l'intersection de deux concepts.

3.2 Mesures sémantiques de comparaison de deux groupes de concepts

Certaines mesures sémantiques de comparaison de deux concepts sont étendues à la comparaison de deux groupes de concepts, en effectuant l'agrégation de ces mesures à l'aide d'une fonction d'agrégation (e.g. une somme, une moyenne). Par exemple, l'extension de la mesure sémantique entre deux concepts de Rada et al (89) est une mesure de comparaison de deux groupes de concepts qui comprennent k concepts C_i et m concepts C'_j. Elle est définie par

$$\text{Distance}(C_1 \wedge ..... \wedge C_k, C'_1 \wedge ..... \wedge C'_m) = \frac{1}{km} \sum_{i=1}^{k} \sum_{j=m}^{m} \text{Distance}(C_i, C'_j) \qquad (3)$$

Un autre point de vue considère seulement la meilleure valeur de la mesure sémantique entre une paire de concepts de deux groupes. Dans ce cas, une mesure asymétrique exprime

la contribution sémantique des concepts du groupe 1 en relation aux concepts du groupe 2. Elle peut être définie comme suit (Azuaje et al., 05)

$$\text{Distance}(C_1 \wedge .. C_k, C'_1 \wedge .. C'_m) = \frac{1}{k+m}\left(\sum_{i=1}^{k} \min_j (\text{Distance}(C_i, C'_j) + \sum_{j=1}^{m} \min_i (\text{Distance}(C_i, C'_j))\right)$$

où la distance utilisée est celle de Rada et al.(89) pour deux concepts.

Blanchard et al. (2008) ont étendu les mesures pour deux concepts à des mesures pour deux groupes de concepts. Leur approche reprend la définition M1 (équation 1) en définissant le contenu informationnel d'un groupe de concepts. Ce dernier est basé sur la détermination de l'ensemble de tous les subsumants du groupe (y compris les concepts du groupe) et l'addition de ce qu'un concept de cet ensemble apporte en plus comme quantité d'information par rapport à ce qu'a apporté son père. Il s'agit de la quantité d'information apportée par une spécialisation Père-fils, notée dans la suite: ICc/p. Soient C et P deux concepts de l'ontologie où P est le père de C :

$$IC_{c/p}(C) = IC(C) - IC(P) \quad (4)$$

Etant donné deux ensembles quelconques de concepts C_i et C_j, la définition générique Mg de la mesure proposée par Blanchard et al. (2008) s'exprime comme suit

$$Mg(C_i, C_j) = \frac{IC^{\cap}(C_i, C_j)}{\alpha IC^{-}(C_i, C_j) + \beta IC^{-}(C_j, C_i) + IC^{\cap}(C_i, C_j)} \quad (5)$$

$IC^{\cap}(C_i, C_j)$ est le contenu informationnel de la partie commune de deux groupes de concepts et $IC^{-}(C_i, C_j)$ ($IC^{-}(C_i, C_j)$) est le contenu informationnel de la partie distinctive de C_i par rapport à C_j (respectivement de C_j par rapport à C_i). Quand $\alpha = 0$ and $\beta = 1$, Mg1 mesure l'inclusion de C_i dans C_j, où $IC^{U}(C_j)$ est le contenu informationnel du groupe de concepts C_j:

$$Mg1(C_i, C_j) = \frac{IC^{\cap}(C_i, C_j)}{IC^{\cup}(C_j)} \quad (6)$$

3.3 Formalisation des mesures de comparaison de constructs

Pour comparer deux constructs, nous comparons les alignements de leur représentation avec l'ontologie UEML via les graphes définis dans la section 2.2. La démarche choisie procède en deux étapes : 1) une comparaison des ensembles de concepts, puis 2) une agrégation des valeurs obtenues lors de la comparaisons pour les quatre types de groupes..

Soient G_{Cc} le groupe de concepts appartenant à la hiérarchie des classes du construct C, G_{Cp} celui appartenant à la hiérarchie des propriétés de ce construct, G_{Ct} celui appartenant à la hiérarchie des transformations de ce construct et G_{Ce} celui appartenant à la hiérarchie des états du construct C. Nous définissions le degré de correspondance Dc de deux constructs C_1 et C_2 comme suit :

$$Dc(C_1, C_2) = AGG(M(G_{C1c}, G_{C2c}), M(G_{C1p}, G_{C2p}), M(G_{C1e}, G_{C2e}), M(G_{C1t}, G_{C2t})) \quad (7)$$

où M est une mesure de comparaison de deux groupes de concepts (cf 3.2), AGG est une fonction d'agrégation des mesures de comparaison des quatre types des groupes de concepts.

Le choix d'une fonction d'agrégation est une question délicate. Dans un premier temps, nous pouvons considérer une fonction additive qui pondère avec des coefficients a, b, c et d la contribution au degré de correspondance des groupes de concepts de chaque hiérarchie :

$$Dc1(C_1,C_2)= a(M(G_{C1c}, G_{C2c})+bM(G_{C1p}, G_{C2p}) + cM(G_{C1e}, G_{C2e}) + dM(G_{C1t}, G_{C2t})) \quad (8)$$

Afin de normaliser la mesure pour permettre des comparaisons sur une même échelle, nous avons privilégié la définition ci dessous

$$Dc2(C_1,C_2) = \frac{aIC^{\frown}(G_{C1c}, G_{C2c}) + bIC^{\frown}(G_{C1p}, G_{C2p}) + cIC^{\frown}(G_{C1e}, G_{C2e}) + dIC^{\frown}(G_{C1t}, G_{C2t})}{aIC^{\smile}(G_{C1c}, G_{C2c}) + bIC^{\smile}(G_{C1p}, G_{C2p}) + cIC^{\smile}(G_{C1e}, G_{C2e}) + dIC^{\smile}(G_{C1t}, G_{C2t})} \quad (9)$$

4 Outil des mesures de comparaison des constructs

Pour faciliter la mise en œuvre opérationnelle de l'approche UEML, l'outil UEML a été développé sous Protégé2000 (Interop, 2005). Le méta-méta-modèle de UEML, son ontologie et plusieurs langages de modélisation ont été implémentés. Pour évaluer le degré de correspondance de deux constructs, nous avons développé un plug-in appelé « UEMLBase Correspondence » avec Java sous Protégé2000. Ce plug-in offre une interface à l'utilisateur pour choisir et définir des mesures sémantiques de comparaison à appliquer sur les constructs de la base UEML. Pour cela, quatre paramètres sont à fixer: 1) l'état de la racine, 2) l'hypothèse de distribution des instances permettant d'estimer le contenu informationnel, 3) la forme de la mesure sémantique et 4) les coefficients d'importance des quatre hiérarchies (à savoir a, b, c, d indiqués dans la section ci-dessus) (Figure 3).

L'état de la racine permet de choisir si la racine de l'ontologie (ici la racine de chaque hiérarchie) est informative ou non. La racine informative participe dans l'augmentation de la similarité de deux concepts de même hiérarchie. Si la racine est non informative la similarité de deux concepts l'ayant comme seul subsumant commun est nulle.

Les hypothèses de distribution portent sur la répartition des instances de l'ontologie sur ses concepts. Quatre hypothèses sont considérées dans l'outil, nous en étudions seulement trois qui nous paraissent les plus pertinentes pour notre domaine applicatif.

Hypothèse 1 : la distribution est uniforme sur les concepts de même profondeur. Le nombre d'instances d'un concept est divisé par le même scalaire à chaque spécialisation. Une spécialisation apporte toujours la même quantité d'information. Le contenu informationnel d'un concept dépend uniquement de sa profondeur.

Hypothèse 2 : la distribution des instances est uniforme sur l'ensemble des fils de chaque concept. Le nombre d'instances d'un concept est divisé par le nombre de ses fils à chaque spécialisation. Une spécialisation d'un concept apporte une quantité d'information qui dépend du nombre des fils de ce concept. Plus un concept a de frères plus la quantité d'information qu'il apporte en plus par rapport à ce qu'apporte son père est importante. Le contenu informationnel d'un concept dépend du nombre des frères de ses subsumants (y compris ses frères à lui).

Hypothèse 3 : la distribution est uniforme sur l'ensemble des feuilles de la hiérarchie. Le nombre d'instances d'un concept est réparti sur ses fils en fonction de leur nombre de feuilles. Plus un concept a de feuilles, moins son contenu informationnel est important. Cette mesure associe un contenu informationnel maximum aux feuilles, indépendamment de leur profondeur et le nombre de fils de leur subsumants.

Finalement, il est possible de considérer trois formes de mesure : Jaccard, Précision et Rappel. Elles sont définies comme suit (C_i et C_j deux groupes de concepts)

Jaccard : $$M(C_i,C_j)=\frac{IC^{\cap}(C_i,C_j)}{IC^{\cup}(C_i,C_j)} \quad (11)$$

Précision : $$M(C_i,C_j)=\frac{IC^{\cap}(C_i,C_j)}{IC^{-}(C_i,C_j)+IC^{\cap}(C_i,C_j)} \quad (12)$$

Rappel : $$M(C_i,C_j)=\frac{IC^{\cap}(C_i,C_j)}{IC^{-}(C_j,C_i)+IC^{\cap}(C_i,C_j)} \quad (13)$$

L'interprétation d'une mesure dépend de la forme de la mesure choisie. La définition de Jaccard définit une similarité stricte ente deux groupes de concepts si le résultat est égal à 1. Les deux définitions Précision et Rappel évaluent l'inclusion (or un recouvrement) d'un groupe de concepts dans l'autre.

Un coefficient « d'importance » pour chaque hiérarchie permet de quantifier l'importance que l'on souhaite donner à chacune dans la phase d'agrégation. Ceci est d'autant plus important qu'actuellement, les quatre hiérarchies ne soient pas aussi développées les unes que les autres. Le poids utilisé dans chaque mesure est le produit du cœfficient d'importance et d'un facteur d'échelle.

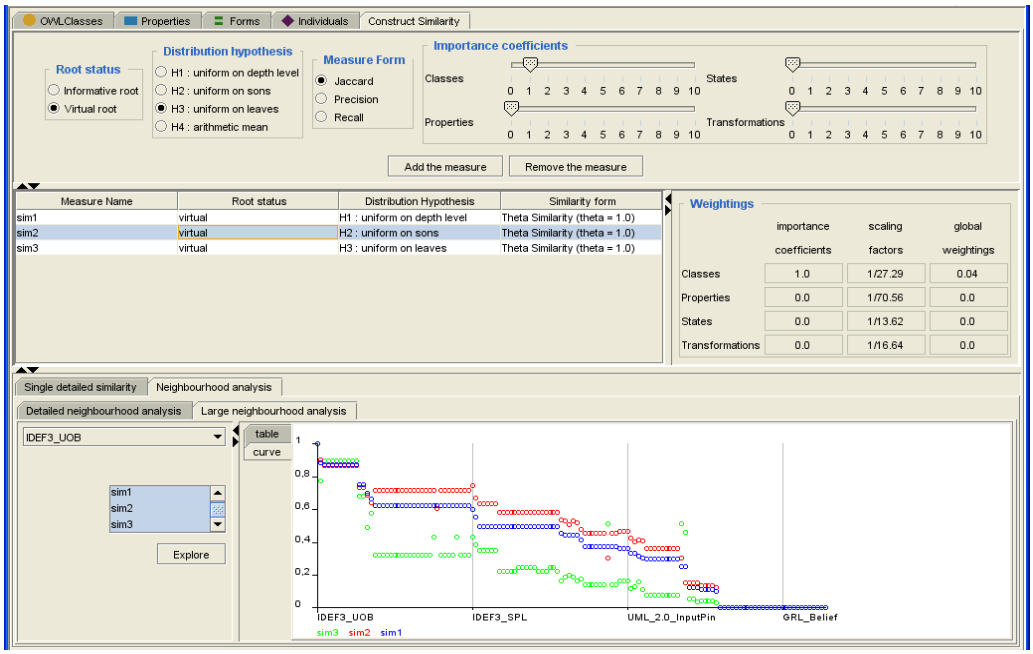


Figure 3 : Les paramètres de l'outil d'analyse des correspondances de UEML

Le facteur d'échelle est propre à une hiérarchie : il dépend de sa taille et il participe à la définition du poids d'une hiérarchie suivant l'idée que « plus une ontologie est de taille importante moins un concept de cette ontologie apporte de l'information ».

Les résultats de ces mesures sont des valeurs entre 0 et 1 affichés de deux façons :

- Les degrés de correspondance entre un construct et tous les autres constructs de la base UEML, sous une forme graphique ou tabulaire.
- Les degrés de correspondance de tous les couples de constructs de la base UEML.

Si les variations des mesures de comparaison en fonction des paramètres Racine, Forme et Coefficient sont bien identifiées, il n'en est pas de même pour le paramètre de Distribution. Dans la suite, nous privilégions la forme de Jaccard (des similarités) avec une racine Informative et des coefficients égaux à 0 pour les trois hiérarchies des propriétés, états, transformations et nous varions les distributions.

5 Analyse des distributions dans l'application UEML

Nous avons analysé les degrés de correspondance des constructs estimés par l'outil en variant les trois distributions afin de comprendre leur influence. Cette analyse nous a aidé à étudier pendant la phase d'incorporation des langages :

- l'ontologie résultante de l'incorporation des langages de modélisation dans le web de langages UEML (en identifiant des éventuelles lacunes);
- la consistance et la précision de l'alignement des représentations des constructs ;
- la complétude des espaces d'information des mesures utilisées.

Nous montrons ces trois points sur des exemples d'analyse de l'application UEML. Dans cette analyse, nous avons considéré la comparaison du construct UOB avec les différents constructs intégrés dans la base UEML.

D'abord, nous avons remarqué qu'il n'y a pas de relation d'ordre entre ces les mesures des trois distributions : Sim1 (Hypothèse 1), Sim2 (Hypothèse2) et Sim 3 (Hypothèse3)). Sim 3 semble plus discriminante que les deux autres mesures. Les valeurs de Sim1 et Sim 2 sont souvent proches. Pour mieux analyser leurs résultats, nous nous sommes focalisés sur certaines valeurs que nous avons comparées au jugement humain (JgH). Ce dernier est défini à l'aide de trois experts du domaine en utilisant trois degrés de similarité : FO (une similarité forte entre 1et 0.65), MO (une similarité moyenne entre 0.65 et 0.40), FA (une similarité faible entre 0.40 et 0).

En analysant les résultats des mesures au JgH, nous avons identifié trois cas possibles pour un construct : 1) la valeur du JgH correspond aux valeurs des trois mesures, 2) la valeur du JgH ne correspond à aucune des valeurs des trois mesures, 3) il y a au moins une mesure qui correspond au JgH. Le tableau 1 comprend des résultats des trois mesures et le JgH que nous utilisons dans la suite. Pour bien identifier les origines des différences entre ces trois mesures et l'écart avec le JgH, nous avons détaillé le calcul du contenu informationnel de chaque mesure pour chaque construct (Tableau 2) : nous utilisons les mesures IC et $IC_{c/p}$ d'un concept (équation 4).

UML-Activity est concerné par le premier cas. Les valeurs des trois mesures sont en adéquation avec le JgH. En comparant le contenu informationnel de ce construct avec celui de l'UOB, nous remarquons que le contenu informationnel de l'UOB a une quantité d'information en plus par rapport à celui du UML-Activity ; elle est apportée par ActedOn-

thing qui s'ajoute à celle apportée par son père. Elle est évaluée par des valeurs proches par les trois mesures.

Construct	Sim 1	Sim2	Sim3	JgH	Concepts de l'alignement du construct
IDEF3-UOB	**1**	**1**	**1**	**FO**	ActedOnThing, System, ExecutingThing
UML-Activiy	0.88	0.87	0.90	FO	System, ExecutingThing
Cpnet-Token	0.38	0.45	0.14	FA	CoupledThing
BPMN-Activity	0.50	0.58	0.22	FO	ActiveThing
IDEF3-SPL	0.60	0.75	0.43	FA	ActedOnthing, PredecessorThing, Suces-sorThing
UML-Aggrgation	0.25	0.30	0.51	FA	Composite, Component

Table 1 : Comparaison des valeurs des trois mesures avec le jugement humain

Lacunes dans l'espace d'information des mesures. IDEF3-SPL et BPMN-Activity sont concernés par le deuxième cas. Pour IDEF3-SPL, nous attendions une faible similarité avec l'UOB. En effet, l'UOB représente principalement une classe qui a des propriétés et des états alors que les deux autres constructs représentent une propriété d'une classe. Cependant les résultats des mesures donnent une similarité importante. En considérant l'alignement de ce construct nous remarquons qu'il comprend un groupe de concepts similaires à celui de l'UOB, sans préciser quel type de concept représente principalement ce construct. L'écart entre les valeurs des mesures et du JgH est dû donc ici au manque d'information prise en compte dans la mesure de comparaison (i.e. le type de concept que le construct représente principalement).

Construct	IC1	IC2	IC3
UOB	IC(ActiveThing)+$IC_{c/p}$(ExecutingThing)+ $IC_{c/p}$(ActedOnThing)+IC(System)		
	0.57+0.14+0.14+ 0,28	0.75+0.17+0.17+0.19	0.26+0.18+0.10+0.45
UML-Activity	IC(ActiveThing)+$IC_{c/p}$(ExecutingThing)+ IC(System)		
	0.57+0.14+ 0,28	0.75+0.17+0.19	0.26+0.18+0.45
IDEF3_SPL	IC(ExecutingThing)+ 2$IC_{c/p}$(PredecessorThing) +$IC_{c/p}$(ActedOnThing)		
	0,71+ 0,28+ 0,14	0,92+0,16+0,17	0,44+0,24+0,10
Cpnet-Token	IC(CoupledThing)		
	0.43	0.58	0.16
BPMN-Activity	IC(ActiveThing)		
	0.57	0.75	0.26
UML-Aggregation	IC(Component)+IC(Composite)		
	0,14+0.14	0.19+0.19	0.05+0.46

Tableau 2 : Détail du contenu informationnel de chaque construct

Eventuelles lacunes dans l'alignement. Pour BPMN-Activity, nous attendions une forte similarité avec l'UOB alors que les résultats des trois mesures sont moyens ou faibles. En comparant les concepts de l'alignement de ces constructs à ceux de l'UOB, il nous semble que cet écart peut être dû, au fait que l'alignement de BPMN-Activity n'est pas complet par rapport à ce lui de l'UOB (BPMN-Activity est il un système, agit-il sur une autre activité ?)

Eventuelles lacunes dans l'ontologie. UML-Aggregation et CPNET-Token, sont concernés par le troisième cas. Pour UML-Aggregation, Sim 3 donne une valeur importante pour le concept Composite (concept avec une seule feuille) qui appartient à l'alignement de

l'UOB, alors que les deux autres mesures donnent une valeur faible (concept de profondeur 1 et avec 4 frères). Pour CPNET-Token, son seul concept de l'alignement est commun avec ceux de l'UOB. Cependant, Sim3 évalue ce concept d'une façon faible (concept avec plusieurs feuilles) par rapport à son évaluation par Sim 1 et Sim2. Cette différence entre Sim3 et Sim1 et Sim2 est due à l'existence d'une quantité d'information qui est commune aux deux constructs ou qui les différencie et qui est apportée par un concept ayant peu de feuilles, un nombre de frères de ses subsumant faible et une profondeur faible. Cette quantité est évaluée d'une façon très importante par Sim3 et moindre par Sim1 et Sim2.

Un grand écart d'évaluation de la quantité d'information apportée par un concept entre ces trois mesures peut déceler des lacunes dans l'ontologie. Ici Sim 2 et Sim1 considèrent que le concept système est peu spécifique (peu profond et ses subsumants ont peu de frères) alors que Sim3 considère qu'il est très spécifique (il a une seule feuille). Cette opposition montre qu'il y a un problème dans l'ontologie. En effet, seule l'une des deux considérations est correcte. Si Système est un concept très spécifique, alors Sim2 nous informe que ses susbumants n'ont pas assez de frères et elle nous propose d'en rajouter dans l'ontologie. Sim1 nous informe qu'il n'est pas assez profond dans l'ontologie et qu'il faut rajouter des concepts intermédiaires comme des subsumants à lui. Si Système est un concept qui n'est pas très spécifique, Sim 3, nous informe qu'on n'a pas définit ses subsumés et surtout les frères de ses subsumés.

Ce manque de concepts dans l'ontologie peut être dû à l'oubli ou à l'ignorance de concepts. En complétant l'ontologie avec des concepts en fonction des résultats de ces trois mesures, on pourrait peut-être faire converger les valeurs des trois mesures.

6 Conclusion

Nous avons présenté, dans le cadre de l'approche UEML, l'adaptation de mesures de comparaison de paires de concepts d'une ontologie à la comparaison des constructs utilisés en modélisation d'entreprise. Ces travaux ont été implémentés dans un outil qui permet de paramétrer et de définir des mesures sémantiques, et d'automatiser le calcul des degrés de correspondance. Nous avons montré sur des exemples comment les mesures de comparaison peuvent nous aider à analyser l'ontologie utilisée, ainsi que l'alignement des constructs avec l'ontologie et l'espace d'information associé aux mesures.

Les retours d'expérience montrent que ces travaux permettent de détecter des éventuels manques de concepts et des spécialisations dans l'ontologie. Ils permettent ainsi de fournir une assistance à l'utilisateur-expert dans la construction de l'ontologie. Dans ce cadre, nous nous intéressons maintenant, une fois la détection des lacunes effectuée, à l'intégration des nouveaux concepts à ajouter dans l'ontologie (quels concepts ? et à quelle place ?).

Par ailleurs, dans l'approche UEML, l'enrichissement de l'ontologie manipulée repose sur la capacité d'expression sémantique des constructs de modélisation intégrés dans le web de langages UEML. Nous étudions actuellement une procédure d'enrichissement ciblée visant à aligner les valeurs calculées avec le jugement humain.

Références

Azuaje, F., H. Wang, et O.Bodenreider (2005). Ontology-driven similarity approaches to supporting gene functional assessment. In *Proc. of the ISMB'2005,* 9-10.

Baziz, M., M. Boughanem, G. Pasi, et H. Prade (2007). An Information Retrieval Driven by Ontology from Query to Document Expansion. *Proc. In the 8th Conference Large Scale Semantic Access to Content*.

Berio, G., A. Opdahl, V. Anaya, et M. Dassisti (2004a). UEML 2.0. Deliverable 5.1 – *INTEROP Network of Excellence*, IST (Confidential).

Berio, G., V. Anaya, et A. Ortiz (2004b). Supporting Enterprise Integration through a Unified Enterprise Modeling Language. CAiSE04 Workshop Proceedings, *Enterprise Modelling and Ontologies for Interoperability (EMOI 2004)*, 3, 165-176.

Blanchard, E., M. Harzallah, P. Kuntz, et H. Briand (2007). Vers une classification des similarités basées sur le contenu informationnel des concepts d'une hiérarchie de subsomption. *Actes des journées francophones d'Ingénierie des Connaissances*, 145-156.

Blanchard, E., M. Harzallah, P. Kuntz, et H. Briand (2008). Sur l'évaluation de la quantité d'information d'un concept dans une taxonomie et la proposition de nouvelles mesures. *Revue des Nouvelles Technologies de l'Information*. A paraître.

Bunge, M. (1979). Treatise on Basic Philosophy. *Ontology II: A World of Systems*, Boston:Reidel,4.

Corby, O., R. Dieng-Kuntz, et C. Faron-Zucker (2004). Querying the Semantic Web with the CORESE search engine. In *Proc. of the 16th European Conference on Artificial Intelligence* : IOS Press, 705-709.

Harzallah M., G. Berio, et A.L. Opdahl (2007). Incorporating IDEF3 into the Unified Enterprise Modelling Language (UEML). In *Proc. of EDOC07 workshop, VORTE 2007*,

INTEROP (2005). Interop Network of Excellence. *www.interop-noe.org*.

Jiang, J. J., et D. W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Conf. on Research in Computational Linguistics*, 19–33.

Knowledge Based Systems, Inc. (2006). IDEF: Integrated DEfinition Methods, *http://www.idef.com/*.

Lin D. (1998). An information-theoretic definition of similarity. In *Proc. of the 15th Int. Conf. on Machine Learning*: Morgan Kaufmann, 296–304.

Lord, P.W., R.D. Stevens, A. Brass, et C.A. Goble C.A. (2003) Semantic similarity measures s tools for exploring the Gene Ontology. *Pac Symp Biocomput*, 601-612.

Opdahl, A.L., et G. Berio (2006). Interoperable Language and Model Management Using the UEML Approach. *Proc. of Workshop on Global Integrated Model Management* .

Rada, R. H. Mili, E. Bicknell, et M. Blettner (1989). Development and application of a metric on semantic nets, *IEEE Transactions on Systems, Man, and Cybernetics*, 1, 17–30.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14th int. Joint conf. on Artificial Intelligence*, 1, 448–453

Sokal, R., et P.H. Sneath (1963). *Principles of numerical taxonomy*. W. H. Freeman and Co.

Tversky, A. (1977). Features of similarity, *Psychol. Rev*, 84, 327-352.

Wand, Y., et R.Weber (1993). On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems*, 217-237.

Wu, Z. et M. Palmer (1994). Verb semantics and lexical selection. In *Proc. of the 32nd annual meeting of the associations for Comp. Linguistics*, 133–138.

Zargayouna, H., et S. Salotti (2004). Mesure de similarité sémantique pour l'indexation de documents semi-structurés. *Actes des journées francophones d'Ingénierie des Connaissances.*

Impact du choix de la distance sur la classification d'un ensemble de molécules

Gilles Bisson[1], Samuel Wieczorek[3]*, Samia Aci[2], Sylvaine Roy[3]

[2]Centre de Criblage pour Molécules Bioactives
[3]Laboratoire Biologie, Informatique, Mathématiques,
CEA-DSV-iRTSV
17, avenue des martyrs, 38054 Grenoble Cedex 9, France
{samia.aci, samuel.wieczorek, sylvaine.roy}@cea.fr
[1] Laboratoire TIMC-IMAG, CNRS / UJF 5525
Université de Grenoble
Domaine de la Merci, 38710 La Tronche, France
{gilles.bisson}@imag.fr

Résumé : Nous comparons ici différentes distances structurelles sur une tâche de catégorisation de molécules chimiques. Nous proposons une nouvelle mesure de similarité asymétriques capable de capturer directement la topologie des molécules. Les expérimentations sont effectuées sur des bases de molécules qui sont utilisées pour réaliser du criblage « haut-débit ».

1 Contexte

Un sujet important pour la recherche et ses impacts potentiels sur la santé est celui de la découverte ou de la synthèse de molécules ayant une activité biologique. Le criblage « haut débit » (ou HTS : High Throughput Screening) de collections de molécules est une approche systématique de ce problème qui est couramment utilisée dans l'industrie pharmaceutique et plus récemment dans la recherche académique.

L'objectif est de tester rapidement, à l'aide d'appareils robotisés, l'activité d'un large ensemble de molécules issues de la chimie (une chimiothèque) sur une cible qui peut-être une enzyme ou une cellule. En pratique, un test de criblage permet de mettre en évidence quelques dizaines de molécules actives, les « hits », ne représentant qu'un faible pourcentage de la chimiothèque. En effet, sa taille est typiquement comprise entre 10^3 à 10^6 molécules. Toutefois, ces tests ne constituent que le début du travail car le plus souvent les molécules identifiées n'ont pas les caractéristiques souhaitées notamment en terme de *sensibilité* et de *spécificité*. De surcroît, les résultats d'un criblage comportent des taux relativement élevés de faux positifs (molécules retenues à tort) et faux négatifs (molécules rejetées à tort).

Dans ce contexte, il est important de fournir aux chimistes des outils d'exploration du contenu des chimiothèques et notamment de faciliter la recherche de molécules structurellement similaires à celles jugées actives lors du criblage. Une approche possible, si l'on dispose d'une distance appropriée, est de faire la recherche des plus proches voisins des « hits ». Mais plus généralement les chimistes ont aussi besoin de méthodes pour organiser *automatiquement* les chimiothèques afin de mieux localiser l'emplacement des molécules actives dans l'espace chimique et surtout de pouvoir évaluer globalement la *diversité* des structures chimiques qui s'y trouvent.

---

* Ce travail fait l'objet d'une thèse en collaboration avec le laboratoire TIMC-IMAG.

La catégorisation automatique (Berkhin 2002), ou « clustering », permet d'effectuer ce type de travail. Cependant, sur des objets aussi structurellement complexes que des molécules, il est clair que la qualité des résultats dépend de la capacité qu'a la distance utilisée de capturer les ressemblances/dissemblances structurelles. Il est cependant difficile de définir « a priori » une distance universelle car la proximité de deux molécules est fonction des propriétés que le chimiste juge importantes vis à vis de ses objectifs. Il n'en reste pas moins qu'il est le plus souvent possible d'injecter une telle connaissance en ajoutant, ou en pondérant, des descripteurs dans la représentation informatique des molécules.

L'objectif de cet article est donc d'étudier expérimentalement, sur un problème de catégorisation de molécules, le comportement de quelques distances structurelles qui sont soit classique en chimie ou qui soit ont été conçues pour comparer spécifiquement des graphes. Le plan est le suivant : dans la section 2 nous présenterons les distances que nous allons utiliser, et en particulier Ipi qui est la mesure que nous proposons ; dans la section 3 nous détaillerons le matériel expérimental ainsi que la méthodologie employée pour évaluer les distances ; enfin, les résultats obtenus seront décrits et discutés dans la section 4.

2 Etat de l'art

2.1 Distances entre molécules

L'évaluation d'une distance entre deux molécules (plus généralement de graphes) est complexe car elle repose directement ou indirectement sur la recherche de sous-graphes isomorphes partiels. Toutefois, cette difficulté peut être contournée en utilisant des représentations basées sur une linéarisation « a priori » de la molécule (langage SMILE de Weininger 1988) ou sur une propositionalisation de la représentation. De la sorte, on peut représenter les molécules à l'aide d'un vecteur de descripteurs, chaque descripteur correspondant à une séquence d'atomes (ou fragment) présent dans la molécule. Ces descripteurs peuvent être soit construits automatiquement (Cf. Chemaxon) soit générés à partir d'un ensemble de contraintes (Helma 2003).

Plus récemment, des *fonctions noyaux*, assimilables à des distances, entre graphes (voir Gärtner 2003) ont été proposées dans le contexte des machines à vecteur support (SVM). Ces approches obtiennent de bonnes performances dans le cadre de l'apprentissage supervisé pour prédire la bio-activité des molécules (Mahé, 2005) ou des problèmes en bio-informatique (Menchetti *et al*. 05). Dans toutes ces approches, la représentation des molécules s'effectue de manière *globale* en construisant de manière explicite ou implicite un ensemble de *chemins* (c-à-d des fragments de la molécule) choisis ou tirés au hasard.

Cependant, il est également possible d'évaluer la distance entre les structures en calculant les chemins de manière plus dynamique en fonction des appariements qui sont effectivement réalisables entre deux molécules (ou de graphes). L'idée dans ce cas est d'explorer *localement* la correspondance entre les paires d'atomes qui les composent puis de trouver le meilleur appariement global des atomes. C'est ce que propose (Froëlich et al. 2005) dans son noyau dit « d'appariement optimal » afin de prédire l'activité de molécules sur des données de criblage. De même, l'indice de similarité que nous proposons ici, appelé indice Ipi, (Wieczorek 2006) suit une stratégie voisine avec des motivations différentes.

Dans la suite de cette section, nous présentons les fonctions noyaux et la façon d'en dériver une distance. Puis nous décrivons deux noyaux classiques basées sur une représentation linéaire des molécules : le noyaux de Tanimoto et le noyaux de décomposition pondéré. Enfin, nous décrivons le noyau d'appariement optimal et l'indice Ipi.

2.2 Fonctions noyaux et distances euclidienne

Les fonctions noyaux constituent la base des méthodes d'apprentissage de type Machine à Vecteur Support (SVM). Ces fonctions permettent de travailler sur les données initiales comme si elles étaient représentées dans un espace de très grande dimension F, appelé l'espace des caractéristiques, mais sans avoir à faire explicitement cette transformation. Pour ce faire, considérons l'ensemble de données X = {x_1, …, x_n} et la fonction ϕ définie telle que $\phi : x \in X \mapsto \phi(x) \in F$. Une fonction noyau est une fonction k (x, y) telle que pour tout $(x,y) \in X$ on satisfasse $k(x,y) = \langle \phi(x), \phi(y) \rangle$ ou $\langle .,. \rangle$ représente le produit cartésien. On peut dériver une distance euclidienne entre les images $\phi(x)$ et $\phi(y)$ de la manière suivante :

$$\begin{aligned} \left\| \phi(x) - \phi(y) \right\|^2 &= \phi(x).\phi(x) - 2\phi(x).\phi(y) + \phi(y).\phi(y) \\ &= k(x,x) - 2k(x,y) + k(y,y) \end{aligned} \quad (1)$$

2.2.1 Noyau de Tanimoto

Le noyau de Tanimoto (Ralaivola *et al.* 2005) est la mise sous forme d'un noyau de la distance de même nom classiquement utilisée en chimie. On y représente les molécules comme un vecteur dont chaque cordonnée indique la présence ou l'absence d'un chemins (fragments) donné. Etant donnée deux molécules x et y, la fonction k_d(x, y) dénombre le nombre de chemins communs entre x et y. Le noyau de Tanimoto k_d^t est définit ainsi :

$$k_d^t(x,y) = \frac{k_d(x,y)}{k_d(x,x) + k_d(y,y) - k_d(x,y)} \quad (2)$$

Dans cette approche, il est nécessaire de fournir la longueur maximale des chemins que l'on va considérer pour représenter les molécules. Dans nos expérimentations nous avons utilisé la valeur de 8 qui est classique en chemoinformatique.

2.1.2 Noyaux de décomposition pondéré

Dans le *noyau de décomposition pondéré* (Menchetti *et al*. 05) (nommé ici 2D-WDK), les molécules sont représentées par un sous-ensemble de tous les sous-graphes possibles à une profondeur bornée. Etant donné un sommet v et un entier $l \geq 0$, on note x(v, l) le sous-graphe de x induit par l'ensemble des sommets atteignables depuis v par un chemin de longueur au plus égale à l. Le noyau 2D-WDK est alors défini par :

$$k(x,y) = \sum_{v \in V(x), v' \in V(y)} \delta\left(x(v), y(v')\right).\kappa\left(x(v,l), y(v',l)\right) \quad (3)$$

La fonction *K*(x(v,l), y(v',l)) est définie comme le noyau de la probabilité de distribution des deux sous-graphes x(v,l) et y(v',l) et la fonction δ est un second noyau entre les sommets x(v) et y(v') dont la valeur est égale à 1 s'ils sont égaux, 0 sinon.

L'intérêt de ce noyau de décomposition est qu'il ne tient pas compte de tous les sous-graphes possibles, ce qui induirait une trop grande complexité en temps de calcul, mais seulement des plus fréquents dans les deux molécules x et y. Dans nos expérimentations nous avons utilisé la longueur de chemin l=3 qui est la valeur par défaut du système.

2.1.3 Optimal Assignment Kernel

Le noyau *d'Appariement Optimal* (OAkernel), proposé par (Frölich et *al*., 2005), est basé sur une exploration dynamique et locale des graphes moléculaires. Contrairement au noyau de Tanimoto la représentation adoptée prend explictement en compte la structure des molécules. Le calcul du noyau se déroule en deux étapes qui sont conceptuellement proches de celles proposées dans (Bisson 95). La première consiste à évaluer une distance entre chaque paire d'atomes a_i et b_j (représentée par la matrice de Gram) des deux molécules par le biais de la fonction noyau k_{nei}. Une fois que la matrice k_{nei} a été calculée, la seconde étape consiste à apparier les atomes a_i de la molécule A avec ceux b_j de la molécule B qui leurs sont les plus similaires, en cherchant à maximiser la somme des $k_{nei}(a_i, b_j)$. Cette phase se ramène à la recherche du couplage de poids maximal dans un graphe biparti.

2.2 Indice de similarité structurelle Ipi

La distance Ipi est une adaptation au domaine de la chimie de l'indice de similarité entre formules logiques proposé dans (Bisson 92, 95 ; Wieczorek et al. 2006). Chaque molécule M est décrite comme un graphe non-orienté définie par un couple (A, L) dans lequel :

- A correspond aux atomes composants la molécule $\{a_1, \ldots, a_n\}$
- L correspond aux liaisons covalentes entre ces atomes $\{l_1, \ldots, l_p\}$

La similarité entre deux molécules M et M' est évaluée à partir de 2 composantes. D'une part, le calcul de la *similarité relative* de M vis à vis de M', ce qui correspond a évaluer le degré d'inclusion de M dans M' noté INC(M, M') $\in [0, 1]$; d'autre part, la *similarité relative* de M' vis à vis M (noté INC(M', M)). De la sorte, on peut comparer des molécules ayant des tailles différentes en terme de nombre d'atomes. Par exemple, si M est plus petite que M' et que M est quasiment incluse dans M' on aura INC(M, M') $\approx$ 1, alors qu'avec une similarité symétrique classique cette valeur reflèterait la différence de taille entre M et M'. Pour calculer la *similarité relative* INC(M, M') entre deux molécules (ou respectivement INC(M', M)), on utilise une méthode qui se décompose en deux étapes :

Evaluation des similarités relatives locales. On calcule ici la similarité relative entre tous les atomes a_i de M et tous ceux a'_m de M' (cette similarité sera noté SUB[a_i, a'_m] $\in [0, 1]$)[1]. Deux atomes sont d'autant plus similaires qu'ils partagent des propriétés communes, mais également, et c'est l'intérêt de la méthode, que les atomes « voisins » auxquelles ils sont reliées par les liaisons de covalence sont eux-mêmes similaires (et réciproquement). Cette définition récursive conduit à exprimer le calcul de similarité entre couples d'atomes sous la forme d'un système d'équations non linéaires. De cette manière, la similarité locale entre une paire d'atome a_i et a'_m caractérise simultanément les ressemblance entre les propriétés des atomes et celles des voisinages dans lesquels ils apparaissent.

Evaluation de la similarité relative globale. Une fois évaluées les valeurs de similarité locales entre toutes les paires d'atomes de M et M', il faut décider de l'appariement à effectuer entre ces atomes de manière à identifier les parties de M présentes dans M' et ainsi évaluer la ressemblance globale INC(M, M'). Cette recherche, qui s'apparente à celle du plus grand sous-graphe commun (MSC), est dirigée par les valeurs contenues dans SUB[].

[1] On fera ensuite de même entre M' et M, les résultats étant stockés dans la matrice SUB'.

Finalement, une fois que les deux valeurs d'inclusion INC(M, M') et INC(M', M) ont été calculées, la similarité totale entre les 2 molécules est la moyenne de ces deux valeurs (ce qui a pour effet de redonner une similarité symétrique). C'est cette moyenne qui est ensuite utilisée par l'algorithme de catégorisation. Nous allons maintenant détailler les procédures de calcul des similarités locales et globales qui sont au cœur de notre méthode.

2.2.1 Similarité locale

Le problème est donc ici de calculer les valeurs des éléments de SUB[] qui correspondent à la similarité locale entre les atomes a_i de M=(A, L) et a'_m de M'=(A', L'). Avant de détailler l'algorithme nous allons introduire quelques fonctions nécessaires.

- Dans la représentation utilisée ici, chaque atome a_i est caractérisé par son seul type (C, N, O, …). Cependant comme dans notre méthode a_i peut correspondre à un objet « composite » contenant différentes propriétés et valeurs, nous considérerons par soucis de généralité qu'il existe une fonction générique S_A : AxA'→[0,1] qui évalue la similarité entre deux atomes. Elle est ici implémentée de la manière suivante :

```
function SA (ai, a'm)
  if (type(ai) = type(a'm) Then return 1 Else return 0
```

- Nous définissons la fonction *Link-of* : A → L qui renvoie pour chaque atome a_i d'une molécule la liste $\{l_1, \ldots, l_p\}$ des liaisons covalentes dans lesquelles a_i apparaît.
- Nous définissons la fonction *Neighbor* : AxL → A qui renvoie, pour un atome donné a_i la référence sur l'atome voisin qui est lié avec a_i par une liaison l_j donnée.
- Dans la représentation, les liaisons l_i ne sont caractérisées que par le type de liaison de covalence. Comme pour les atomes, nous considérons qu'il existe une fonction générique S_L : SUB[]xLxL'xAxA'→[0,1] évaluant la similarité de deux liaisons du point de vue de deux atomes donnés. Cette fonction est implémentée ainsi :

```
function SL (SUB, lj, l'n, ai, a'm)
  if (valence(lj) = valence(l'n) Then simlink = 1 Else simlink = 0
  return ((SUB [Neighbor(lj, ai), Neighbor(l'n, a'm)]) + simlink)/2
```

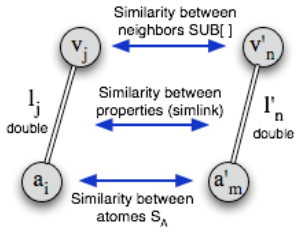
D'une part, S_L prend en compte la ressemblance entre les propriétés des deux liaisons (variable *Simlink*). D'autre part, le terme « (SUB [Neighbor(l_j, a_i), Neighbor(l'_n, a'_m)]) » est celui qui permet d'avoir une description récursive de la similarité locale entre atomes et ainsi qui permet de prendre en compte la topologie. Avant d'expliquer le mécanisme exact, il est nécessaire de décrire l'algorithme de calcul de la matrice SUB[] qui contient l'ensemble des valeurs de ressemblance locale entre deux atomes quelconques de la molécule M et M'

```
function local-similarity (M, M', Sub[], Iter)
(IT1)for k=1 to Iter
(IT2)  for each pair of atoms (ai,a'm) of respectively M and M'
         let L  = Link-of (ai)        //neighbors of ai
         let L' = Link-of (a'm)       //neighbors of a'm
         Nsim = Find-Max-Mapping (SUB[], L, L', ai,a'm)
         SUB[ai,a'm] = (SA(ai,a'm)+(Nsim/(length(L))))/2 // normalize by M
       return Sub[]
```

Le calcul est donc constitué de 2 boucles imbriquées IT1 et IT2. Commençons par décrire la seconde qui effectue le calcul de similarité locale entre 2 atomes a_i et a'_m. Le problème central est de trouver comment apparier de manière optimale les voisins de a_i et a'_m et donc les liaisons de covalence correspondantes. Cela se ramène à un problème d'appariement classique qui est géré à l'aide de la procédure *Find-Max-Mapping*. En effet,

considérons les liaisons contenues dans les listes L et L' comme des éléments d'un graphe biparti dont on connaît, pour chaque couple (l_j, l'_n), la valeur de ressemblance grâce à la fonction S_L. Trouver le meilleur appariement à effectuer entre ces éléments, correspond à maximiser la somme des S_L et donc résoudre un problème *de couplage de poids maximum dans un graphe biparti*. Pour cela, nous utilisons l'algorithme hongrois (Kuhn 57) dont la complexité est en $O(n^3)$. Cela ne pose pas de problème car les listes L et L' sont ici petites : la taille correspond à la valence de l'atome concerné, typiquement 4 pour le carbone.

L'originalité de l'approche réside évidemment dans le terme « SUB [Neighbor(l_j, a_i), Neighbor(l'_m, a'_m)] » qui apparait dans S_L. Il exprime le fait que pour 2 atomes a_i ($\in$ M) et a'_m ($\in$ M') la similarité entre 2 liaisons covalente quelconques l_j et l'_n qui « partent » de ces atomes est fonction de la ressemblance entre les atomes voisins v_j et v'_n qui sont reliés par les liaisons l_j et l'_n. De la sorte, la similarité est définie sous la forme d'un système d'équations, puisque la similarité de deux atomes a_i et a'_m est fonction de celle de ses voisins SUB[v_j, v'_n] (et réciproquement). Finalement, le calcul de la similarité locale SUB[a_i, a'_m] entre deux atomes correspond à la moyenne entre leur ressemblance (S_A) et la ressemblance moyenne normalisée de leur voisinage (variable *Nsim*). La figure 1 ci-contre résume l'ensemble des informations qui sont prises en compte lorsque l'on compare deux atomes quelconques.



La résolution du système d'équations s'effectue à l'aide d'une méthode itérative (Jacobi) l'itération IT1 permettant de calculer les valeurs successives de la matrice SUB []. Il faut noter que ces itérations ont également une seconde interprétation : lors de la première itération, chaque couple a_i et a'_m prend en compte ses voisins immédiats[2] ; à la deuxième itération également mais cette fois les voisins ont eux-mêmes pris en compte leurs propres voisins, de la sorte le calcul de la similarité entre a_i et a'_m intègre des informations provenant de voisins d'ordre 2 et ainsi de suite pour les itérations suivantes. En d'autres termes, le nombre d'itération caractérise la profondeur de diffusion des informations, c'est à dire la taille du voisinage pris en compte pour comparer deux atomes. Cette propagation des informations suit ici une loi de décroissance géométrique en $1/(n+1)^2$ ou n est la distance des voisins. C'est pourquoi on peut fixer le paramètre *Iter* à une valeur faible (*Iter* =5) car après cette profondeur la similarité ne change plus guère d'une itération à l'autre.

2.2.2 Similarité globale

Une fois que les valeurs de similarité locales ont été calculées dans la matrice SUB[], il reste à évaluer la similarité globale INC (M, M'). Cela revient à rechercher la liste des meilleurs appariements possibles entre tout ou partie des atomes de M et M' en se basant sur les similarités locales, le but étant de trouver le plus grand sous-graphe partiel connexe. La valeur de INC (M, M') correspond alors à la somme des similarités locales entre les atomes appariés, somme qui est normalisée par la taille de la molécule M (puisque INC (M, M') correspond à la mesure d'inclusion de M dans M'). Pour rechercher le meilleur appariement possible, on pourrait à nouveau utiliser l'algorithme hongrois, comme le fait d'ailleurs (Frölich et *al*., 2005), en considérant que l'on a un graphe biparti constitué par les atomes de M et de M'. Toutefois cette solution n'est pas satisfaisante pour deux raisons :

- La similarité locale entre deux atomes a_i et a'_m correspond à une surestimation de la similarité réelle. En effet, lorsque l'on apparie les voisins de a_i et a'_m rien ne garantie

[2] C'est le cas si l'on initialise la matrice SUB [] en mettant SUB[a_i, a'_m] = S_A(a_i, a'_m).

que les appariements décidés, à leur tour, par ces voisins sont compatibles. Aussi, le couplage maximum est trop « optimiste » quant à la similarité réelle des molécules.

- Comme les similarités locales peuvent correspondre à des appariements différents, il n'y a aucune garantie que le sous-graphe commun (MCS) trouvé par la procédure de recherche du coupage maximal soit connexe. Or cela est crucial en chimie.

De ce fait, la recherche de l'appariement est effectuée à l'aide de l'heuristique suivante. On prend le meilleur score B de similarité entre atomes que l'on peut trouver dans SUB[], comme une « graine » puis on propage la mise en appariement en se guidant sur la structure de M et sur les valeurs contenues dans SUB[]. Toutefois, comme B n'est qu'un point de départ possible, cette procédure est itérée plusieurs fois (10 essais[3]), en prenant à chaque fois un couple d'atomes différents, non déjà appariés lors des explorations précédentes. La similarité INC(M, M') finale correspond au meilleur de ces essais d'appariements.

3 Matériel expérimental et méthodologie

3.1 Les bases de tests

Les 4 chimiothèques qui ont été utilisées pour effectuer les tests sont issues du travail de (Sutherland et *al*., 2003) et elles sont représentées au format SDF. Cela permet de connaître la structure 2D des molécules. Voici une brève description du contenu de ces bases :

- **Cox2 :** 467 inhibiteurs de la cyclooxygenase-2, divisés en 13 familles.
- **Dhfr :** 756 inhibiteurs de la dihydrofolate reductase, divisés en 18 familles.
- **Bzr :** 405 ligands pour le récepteur de la benzodiazepine, divisé en 14 familles
- **Er :** 393 ligands récepteurs d'œstrogène, divisé en 3 familles.

3.2 Représentation et méthodes

L'objectif de cette étude est de tester la capacité des différentes mesures de similarité à retrouver, par classification, les familles chimiques de molécules définies par les experts en se basant sur la simple structure 2D. La représentation adoptée est homogène pour toutes les approches : on ne prend en compte que le nom des atomes (C, N, …) et le type de liaison covalente (simple, double, …) entre ceux-ci. Cette représentation minimale présente l'avantage d'être exploitable par l'ensemble des approches testée ici, ce qui élimine le biais représentationnel. Dans le cas des distances de TanimotoK et de 2D-WDK, la description des molécules sous forme de chemins est automatiquement réalisée par ces logiciels.

La distance de Tanimoto que nous avons utilisée est celle implémentée dans l'outil de Chemaxon (www.chemaxon.com). Pour les autres approches (2D-WDK, OAkernel et Ipi) nous avons utilisé les implémentations fournies par les auteurs respectifs.

Une fois calculées les matrices de distances avec chacune des approches, la catégorisation des molécules a été réalisée à l'aide de la méthode de classification ascendante hiérarchique *hcluster* qui est implémentée dans le logiciel R (www.r-project.org/) en utilisant comme, distance d'aggrégation interclasse, l'indice de Ward.

---

[3] Ce nombre d'itération a été établi de manière expérimentale, en effet dans toutes les bases le meilleur appariement apparaît au cours des 10 premières itérations dans plus de 99% des cas.

3.3 Evaluation des classifications

L'évaluation des résultats d'une catégorisation est délicat en l'absence de critère de validation (voir parmi d'autres : Candillier et *al*. 2006). Ce n'est heureusement pas notre cas ici puisque nous connaissons pour chacune des bases le nombre de familles (classes) qui doivent être retrouvées par le système ainsi que la description extensionnelle de ces classes. Nous pouvons donc évaluer les résultats produits par les différentes mesures et méthodes en mesurant l'écart qu'il y a entre les classifications attendues et obtenues. Le résultat d'une catégorisation peut être représenté quantitativement sous la forme d'une matrice de confusion (C, L). Dans cette matrice les C_i (lignes) représentent les classes originales et L_j (colonnes) les classes construites par le système de classification, chacune des valeurs $n_{i,j}$ représente le nombre de molécules qui sont simultanément présentes dans les classes C_i et L_j. Il y accord parfait entre les deux classifications lorsque cette matrice ne contient qu'une seule valeur non nulle pour chaque ligne et chaque colonne (autrement dit, la matrice est diagonale à un ensemble de permutations prêt). Une manière simple de quantifier la qualité de la classification est donc de mesurer les entropies moyennes qui sont associées aux lignes et aux colonnes de la matrice. On a ainsi deux indices :

- *Indice de confusion (IC)* = quantifie le nombre de classes mélangées.
- *Indice de segmentation (IS)* = quantifie le nombre de classes subdivisées.

$$IC = \sum_{i}^{p} \frac{\overline{C}_i}{N} \sum_{j}^{q} -\frac{n_{i,j}}{\overline{C}_i} \times \log_2 \frac{n_{i,j}}{\overline{C}_i}$$

$$IS = \sum_{j}^{q} \frac{\overline{L}_j}{N} \sum_{i}^{p} -\frac{n_{i,j}}{\overline{L}_j} \times \log_2 \frac{n_{i,j}}{\overline{L}_j}$$

$$Avec : \begin{cases} \overline{C}_i = \sum_{j}^{q} n_{i,j} \\ \overline{L}_j = \sum_{i}^{p} n_{i,j} \\ N = \sum_{i}^{p} \sum_{j}^{q} n_{i,j} \end{cases}$$

L'indice le plus important est IC (entropie moyenne sur les colonnes) puisqu'il indique si classification initiale a bien été retrouvée par l'algorithme de catégorisation. Plus la valeur de cet indice est faible meilleure est l'adéquation entre les 2 classifications.

4. Expérimentations et discussions

Les 4 graphiques (figure 2) montrent l'évolution de l'indice IC pour les quatre bases. Les résultats obtenus sont homogènes. Pour le nombre de familles attendues, le classement entre les distances ou similarités est, à une exception près (Cox2), toujours le même : Ipi, TanimotoK, puis OAkernel et 2D-WDK. Nous allons présenter les résultats obtenus base par base puis essayer d'expliquer les résultats observés.

La base Cox2 contient des molécules ayant des structures générales (*scaffold*) très voisines, mais où les familles sont facilement reconnaissables à partir de la position de quelques atomes discriminants dans les cycles aromatiques. L'indice de similarité Ipi qui est capable de percevoir l'environnement de chaque atome grâce au calcul de similarité locale, retrouve quasiment toutes les familles attendues. Les seules erreurs correspondent à des familles de petits effectifs qui ont été fusionnées. Le score de TanimotoK est également très bon. Dans le cas de 2D-WDK et de OA-kernel les résultats sont assez mauvais.

La base Dhfr est constituée de molécules possédant des sous-structures similaires, mais organisées (connectées) différemment d'une famille à l'autre. Les résultats obtenus sont les mêmes que pour Cox2 mais avec une plus grande dispersion entre les méthodes. L'indice de similarité Ipi montre sa capacité à reconnaître globalement la structure des molécules.

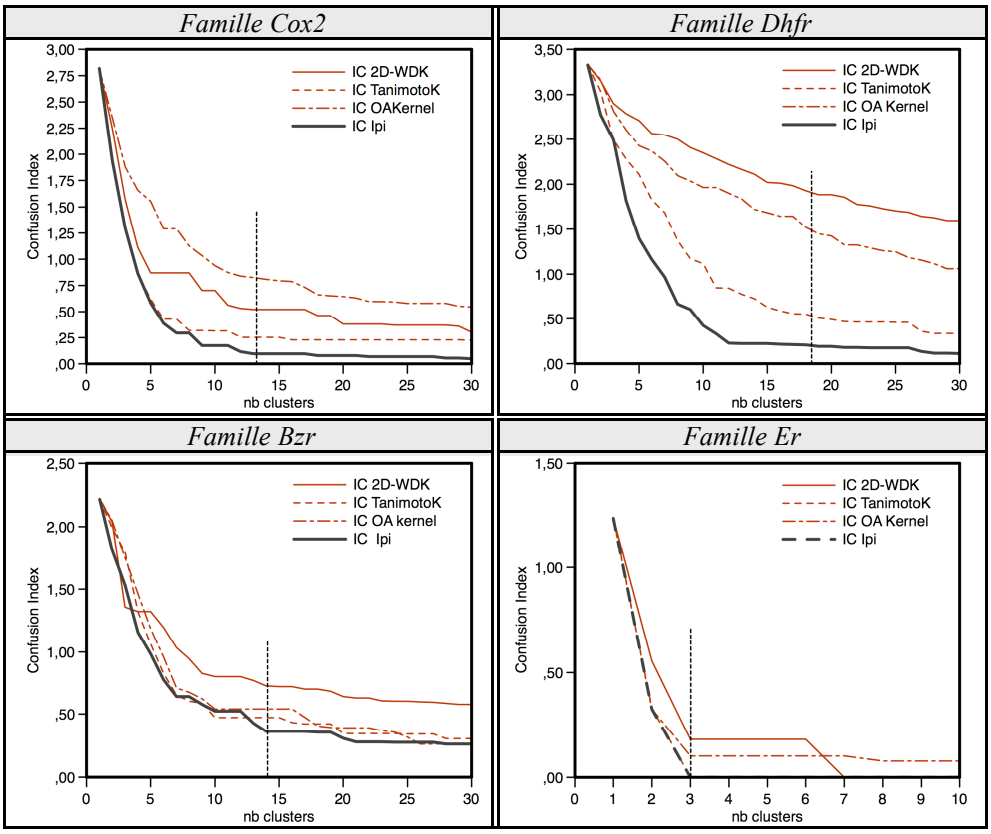


Figure 2 – Indices IC pour les 4 distances utilisées avec la CAH sur les quatre bases : Cox2, Dhfr, Bzr et ER. Le trait vertical représente le nombre de familles chimiques qui est attendu.

La base Bzr contient des molécules où l'on observe, au sein même d'une famille, une assez grande variabilité de la structure des molécules. De ce fait, toutes les méthodes ont du mal à retrouver la classification effectuée par les chimistes.

Enfin, la base Er, ne contient que 3 famille de molécules. Chaque famille est caractérisée par une structure bien spécifique qu'il est a priori assez simple à identifier. C'est effectivement le cas pour Ipi et Tanimotok (les 2 courbes se superposent). Par contre, une nouvelle fois, 2D-WDK et OA-Kernel obtiennnent de moins bons résultats.

Les approches testées correspondent à deux stratégies différentes : TanimotoK et 2D-WDK linéarisent les molécules pour les représenter alors que Ipi et OA-kernel traitent directement la structure 2D. Il est donc intéressant de se demander pourquoi dans chaque famille deux approches ont de bons résultats (Ipi et TanimotoK) et les deux autres non. Dans le cas de OA-kernel et Ipi, qui utilisent des algorithmes assez voisins, nous avons pu

déterminer plus précisément les principales modifications qui expliquaient les résultats divergents. C'est pour Ipi :

- l'utilisation d'un *indice asymétrique* pour comparer les atomes et les molécules.
- la recherche d'un *appariement global* entre les atomes cohérents vis à vis des structures des molécules et non pas avec la recherche du couplage de poids maximum (cf. 2.2.2).

Si l'on modifie Ipi de manière à retirer ces deux éléments, on observe que les comportements de OA-kernels et Ipi deviennent assez identiques.
Dans le cas de TanimotoK et 2D-WDK, c'est la sélection des chemins qui permettent de passer à une représentation vectorielle des molécules qui est déterminant.

5. Conclusion

L'analyse de résultats expérimentaux issus de criblage de molécules est une opération complexe et les chimistes sont demandeurs de mécanismes de catégorisation leur permettant de mettre en évidence des analogues structuraux des molécules actives et surtout d'évaluer la diversité chimique de leurs chimiothèques. Dans cet article, nous avons étudié expérimentalement quatre distances (ou indice de similarité) capables de travailler sur des données structurées afin d'évaluer leur capacité à retrouver des familles de molécules prédéfinies. Les résultats obtenus avec noyaux 2D-WDK et OA-kernel sont relativement décevants ce qui tend à montrer que les « distances » qui montrent de bonnes performances dans un cadre supervisé ne sont pas forcément applicables avec les algorithmes de catégorisation classiques. Il serait nécessaire d'approfondir cette étude, notamment en y intégrant les travaux qui ont été réalisés avec des SVM sur le clustering (entre autres : Finley et al. 05 ; Ben-Hur et al. 01) et en testant l'extension du MG kernel de (Mahé et *al.*, 2004).

Dans ce travail, malgré la simplicité de l'approche, on peut être surpris du bon résultat obtenu par le classique indice de Tanimoto qui est clairement moins complexe à calculer que l'indice Ipi. Toutefois, ce dernier présente un double avantage (indépendamment du fait que cette mesure arrive en tête de tous les tests). D'une part, celui de travailler directement à partir de la représentation 2D des molécules sans avoir à la linéariser et donc à choisir la taille des clefs structurelles à utiliser ; d'autre part, comme nous l'avons vu, en modifiant les fonction S_A et S_L il est possible de prendre en compte facilement dans la mesure toutes les informations physico-chimiques que le chimiste juge utiles.

Références

1. BEN-HUR A., BIOWULF, HORN D., SIEGELMANN H.T., VAPNIK V. (2001). Support Vector Clustering. Journal of Machine Learning Research, vol 2, pages 125-137. 2001.
2. BISSON G. (1992). Learning in FOL with a similarity measure. In Proceeding of 10th AAAI Conference, San-Jose. 82-87.
3. BISSON G. (1995. Why and how to define a similarity measure for object-based representation systems. Proceedings of 2nd international conference on building and sharing very large-scale knowledge bases (KBKS). IOS press. Enschede (NL). 10-13 avril 1995. pp 236-246.
4. BERKHIN, P. (2002). Survey of clustering data mining techniques. Tech. rep., Accrue Software, San Jose, CA. http://citeseer.nj.nec.com/berkhin02survey.html.
5. CANDILLIER L., TELLIER I., TORRE F., BOUSQUET O. (2006) Cascade Evaluation of Clustering Algorithms, in Proc. of ECML, Berlin.
6. DAYLIGHT : http://www.daylight.com/dayhtml/doc/theory/theory.finger.html
7. ESTER M., KRIEGEL H.-P., SANDER J., XU X. (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Proc 2nd Int. Conf. On Knownledge Discovery and Data Mining, Portland, 226-231.
8. FINLEY T. AND JOACHIMS T. (2005). Supervised Clustering with Support Vector Machines, Proceedings of the International Conference on Machine Learning (ICML). Bonn, Germany. Pages: 217 - 224
9. FRÖHLICH H., WEGNER J., SIEKER F., ZELL (2005) A Optimal Assignment Kernels for Attributed Molecular Graphs, In Proc. of Int. Conf. on Machine Learning (ICML), pp. 225 – 232.
10. GARTNER T., FLACH P., WROBEL S. (2003) On graph kernels : Hardness results and efficient alternatives. Proc. of the 16th Annual Conference on Computational Learning Theory and the 7th Annual Workshop on Kernel Machines. Heidelberg : Springer-Verlag
11. KASHIMA, HISAHI K., KOJI T., AKIHIRO I. (2003), Marginalized Kernels Between Labeled Graphs, in Proc. the International Conference on Machine Learning (ICML). Washington DC
12. MAHE P., UEDA N., AKUTSU T., VERT J.-P. (2004), Extensions of Marginalized Graph Kernels, Proc. the International Conference on Machine Learning (ICML). Banff, Alberta.
13. MAHE P., UEDA N., AKUTSU T., PERRET J.-L., VERT J.-P. (2005) Graph kernels for molecular structure-activity relationship with support vector machines. J. Chem. Inf. Model. 45(4) :939-951.
14. MENCHETTI S, COSTA F., FRASCONI P. (2005). Weighted decomposition kernels, Proceedings of the International Conference on Machine Learning (ICML). Bonn, Germany. Pages: 585 – 592
15. SUTHERLAND J.J., O'BRIEN L. A., WEAVER D. F. (2003) Spline-Fitting with a Genetic Algorithm: A Method for Developing Classification Structure-Activity Relationships. J. Chem. Inf. Comput. Sci. 43, 1906-1915
16. WEININGER D. (1988), "SMILES 1. Introduction and Encoding Rules", Weininger, D., J. Chem. Inf. Comput. Sci., 1988, 28, 31. Voir aussi : http://www.daylight.com/
17. WIECZOREK S., BISSON G. AND GORDON MB. (2006). Guiding the Search in the NO Region of the Phase Transition Problem with a Partial Subsumption Test. In proceeding of ECML 2006. LNCS 4212/2006. 18-22 september, Berlin. p 817-824.

Détection de la similarité sémantique entre pages visitées durant une session d'apprentissage

Mouna Khatraoui*, Nabila Bousbia*,**
Amar Balla*

*I.N.I. (Institut National d'Informatique), BP 68M, 16309 Oued-Smar, Alger, Algérie
m_khatraoui@ini.dz, n_bousbia@ini.dz, a_balla@ini.dz
**LIP6, Université Pierre et Marie, 104 Avenue du Président Kennedy, F-75016 Paris, France
nabila.bousbia@etu.upmc.fr

Résumé. Les mesures de similarité sémantique jouent un rôle important dans la recherche d'information, elles sont devenues une voie très explorée. Dans le présent article, nous appliquons les mesures de similarité entre concepts dans une ontologie, afin de détecter les liens sémantiques entre les pages Web visitées par un apprenant, et le cours qui lui est proposé dans le cadre d'une Formation Ouverte et A Distance (FOAD). La similarité entre le cours et ces pages visitées en parallèle à la formation nous permet d'apprendre le degré d'intérêt que porte un apprenant aux contenus pédagogiques.

1 Introduction

L'évaluation de la similarité sémantique entre concepts dans une ontologie est un problème connu dans le domaine de la Recherche d'Information (R.I). Plusieurs méthodes ont été proposées dans ce sens. Deux grandes approches se sont détachées : les approches basées sur la distance, c'est-à-dire sur la structure de l'ontologie (Rada et al., 1989), et les approches utilisant le contenu informatif des concepts (Sanderson et Croft, 1999), (Seco et al., 2004). Ces mesures sont intégrées dans différentes applications, telles que le calcul de similarité entre documents, le clustering de documents, la désambigüisation sémantique, l'indexation automatique, etc.

D'autre part, la mesure de la similarité entre documents est également une question qui a été traitée dans plusieurs travaux. Le modèle vectoriel, par exemple, consiste à représenter les documents par des vecteurs de termes pondérés par des poids. La similarité entre documents est calculée par des méthodes statistiques telles que le cosinus, la distance euclidienne, etc.

Cependant, les méthodes syntaxiques prennent en compte l'appariement exact des mots, sans pour autant tenir compte des relations sémantiques entre ces derniers, telle que la synonymie. Avec l'expansion du Web sémantique, les ontologies qui se voient intégrées de plus en plus dans les processus de recherche d'information, viennent pallier au problème de la sémantique dans la mesure de similarité.

Dans cet article, nous nous intéressons à l'application des mesures de similarité entre concepts d'une ontologie, pour l'évaluation de la similarité entre documents. Notre travail s'inscrit dans le cadre d'une formation ouverte et à distance. L'objectif étant de détecter le

degré de similitude entre les pages Web visitées par un apprenant, avec le cours proposé. Ceci permettra au tuteur de déterminer à quel point l'apprenant visite-t-il des documents (en particulier des pages Web) en relation avec le cours consulté, et donc de l'informer sur le degré d'intérêt que porte un apprenant au cours. Ainsi, nous proposons dans cet article un indicateur informant sur la proximité sémantique entre un cours dispensé dans un dispositif de FOAD et les pages Web visitées en dehors du cours, en tenant compte de la similarité sémantique entre les concepts les représentant respectivement, mais aussi en tenant compte des durées de consultation de chaque page. Cet indicateur, -en plus d'autres indicateurs tels que : le taux de participations de l'apprenant dans des forums en relation avec le cours, le taux d'enregistrements des pages de cours, etc.- intervient dans le calcul du degré d'intérêt que porte l'apprenant au cours proposé et sa manière d'apprendre. L'obtention des autres indicateurs que nous venons de citer n'est pas l'objectif du présent article ; elle sera présentée dans d'autres communications.

Cet article s'articule comme suit : nous commençons par explorer les différentes méthodes pour le calcul de similarité entre concepts dans une ontologie. Après l'évaluation de ces méthodes, nous présentons notre démarche permettant de détecter la similarité sémantique entre le cours et les pages visitées par un apprenant dans une formation ouverte et à distance. Enfin, nous terminons par présenter nos perspectives quant aux futurs travaux.

2 Evaluation de similarité sémantique entre concepts dans une ontologie

La similarité entre concepts d'une ontologie a été étudiée par de nombreux auteurs, et deux grandes approches se sont détachées : les approches basées sur la distance, c'est-à-dire sur la structure de l'ontologie (Rada et al., 1989), et les approches utilisant le contenu informatif des concepts (Sanderson et Croft, 1999), (Seco et al., 2004).

2.1 Approche basée sur la distance

L'ontologie est représentée par un graphe dont les nœuds sont des concepts, et les arcs sont les liens entre concepts. Les mesures reposant sur la distance considèrent que la similarité entre deux concepts peut être calculée à partir du nombre de liens qui séparent les deux concepts.

Plusieurs variantes existent en fonction du chemin pris en compte pour calculer la distance entre les concepts.

Ainsi, la mesure du *edge counting* proposée par Rada et al. (1989) utilise une métrique dist(c_1 ;c_2), qui indique le nombre d'arcs séparant les deux concepts c1 et c2 par le plus court chemin dans la hiérarchie. Plus deux concepts sont distants, moins ils sont similaires.

$$Sim(c_1, c_2) = \frac{1}{1 + dist(c_1, c_2)}$$

Wu et Palmer (1994) ont proposé une autre méthode basée sur le plus petit généralisant commun, c'est-à-dire le généralisant commun à c_1 et c_2 le plus éloigné de la racine.

$$Sim(c_1, c_2) = \frac{2 * depth(c)}{depth(c_1) + depth(c_2)}$$

Où *depth(c_i)* correspond au niveau de profondeur du concept c_i, et « c »représente le plus petit ancêtre commun à c_1 et c_2. Cependant, l'approche basée sur la distance présente l'inconvénient que la similarité dépend de l'organisation des concepts dans la hiérarchie. Les choix pris lors de la construction de la hiérarchie des concepts influencent donc la valeur de la similarité.

2.2 Approche basée sur le contenu informationnel

Cette approche prend en considération le contenu informatif des concepts de l'ontologie. La similarité est alors calculée à partir de l'information partagée par les concepts. Deux méthodes existent. La première méthode utilise un corpus de référence et mesure la probabilité de trouver un concept ou un de ses descendants dans ce corpus. Le contenu en information d'un concept « c »se calcule donc de la façon suivante (Resnik, 1999) :

$$ic_{res}(c) = -logp(c)$$

Par la suite, il faut trouver l'ensemble des concepts qui subsument les deux concepts soit S(c_1,c_2). Une des mesures de similarités est donnée par la formule suivante (Resnik, 1999) :

$$Sim(c_1, c_2) = Max[IC(c)], cdansS[c_1, c_2]$$

La seconde méthode calcule le contenu informatif des noeuds à partir de WordNet[1] au lieu d'un corpus. Seco et al. (2004) utilisent les hyponymes descendants des concepts pour calculer le contenu informatif de ceux-ci.

$$\frac{log(\frac{hypo(c)+1}{max_{wn}})}{log(\frac{1}{max_{wn}})} = 1 - \frac{log(hypo(c)+1)}{log(max_{wn})}$$

Où : *hypo(c)* est le nombre d'hyponymes du concept c ; et max_{wn} : une constante qui indique le nombre de concepts de la taxonomie.

2.3 Approche mixte

Il existe aussi une approche mixte, utilisant les résultats des deux approches définies précédemment. Le principe des mesures mixtes est de considérer le plus court chemin reliant deux concepts dans l'ontologie et de pondérer ces liens à partir de leur poids sémantique. Le poids sémantique des liens prend notamment en compte le contenu en information des concepts. La mesure de Jiang et Conrath (1997) est donnée par la formule suivante :

$$dist_{jcn}(c_1, c_2) = (ic_{res}(c_1) + ic_{res}(c_2)) - 2 \times sim_{res}(c_1, c_2)$$

Aussi, la mesure de Lin (1998) est donnée par la formule suivante :

$$sim_{lin}(c_1, c_2) = \frac{2 \times sim_{res}(c_1, c_2)}{(ic_{res}(c_1) + ic_{res}(c_2))}$$

[1] http ://www.cogsci.princeton.edu/ wn/

Notons que les approches citées précédemment pour calculer la similarité entre concepts d'une ontologie, sont toutes basées sur les liens taxonomiques (c'est-à-dire la relation *est un*). D'autres travaux calculent la similarité en se basant sur d'autres types de relations telle que *partie de* (Lord et al., 2003), (Thieu et al., 2004), (Bernstein et al., 2005). Seulement, ces méthodes ont l'inconvénient soit de ne pas différencier le poids des liens en fonction de leur type, soit d'ignorer une partie de la sémantique qu'ils représentent.

2.4 Évaluation

Les différentes mesures présentées ont été comparées dans plusieurs travaux (Resnik, 1995), (Jiang et Conrath, 1997), (Lin, 1998), (Budanitsky et Hirst, 2001). L'évaluation consiste à comparer la valeur donnée par les mesures pour différentes paires de termes avec des valeurs de similarités affectées par des humains. En considérant la hiérarchie de concepts issue de Wordnet, la mesure qui permet d'obtenir les résultats les plus proches des jugements humains est la méthode proposée par Jiang, suivie de très près la mesure de Lin. Cette conclusion est vraie pour les évaluations utilisant WordNet car le corpus est désambiguïsé à partir de ses synsets[2].

C'est pour cette raison que nous utilisons la mesure de Jiang, mais en appliquant la méthode de Seco pour le calcul du contenu informatif des concepts (c'est-à-dire à partir de WordNet). Car selon les travaux de Seco et al. (2004), elle apporte de meilleurs résultats.

3 Notre démarche

Nous partons des trois phases d'observation des usages dans le cadre du Web Usage Mining (Srivastava et al., 2003) : la collecte, le prétraitement et l'analyse comme illustré dans la figure1, tout en proposant une méthode pour la détection du degré de similitude entre les pages parcourues par un apprenant lors de sa session d'apprentissage et les pages de navigation Web.

3.1 Première phase : collecte des traces

La collecte consiste à capturer les données de parcours qui constituent la matière première pour l'analyse et l'interprétation. La collecte du côté client fournit les données relatives au parcours de l'apprenant, que ce soit à l'intérieur de la plate-forme d'apprentissage, ou bien en dehors de celle-ci, c'est-à-dire sa navigation sur le Web et les différentes applications exécutées sur son poste en local : documents consultés, les notes prises, etc. Nous nous intéressons particulièrement aux données de navigation Web.

Les données capturées constituent tous les événements et actions effectuées sur le poste de l'apprenant : URL, entrées clavier, clics de souris, sélection de texte, gestion de fichiers, etc. Ces données collectées du coté client s'appellent données brutes. Volumineuses et très minutieuses qu'elles soient, il est difficile, voire impossible de les interpréter telles quelles par les humains. Il est donc indispensable d'effectuer quelques traitements dessus afin de les rendre interprétables.

[2]Ensembles de synonymes

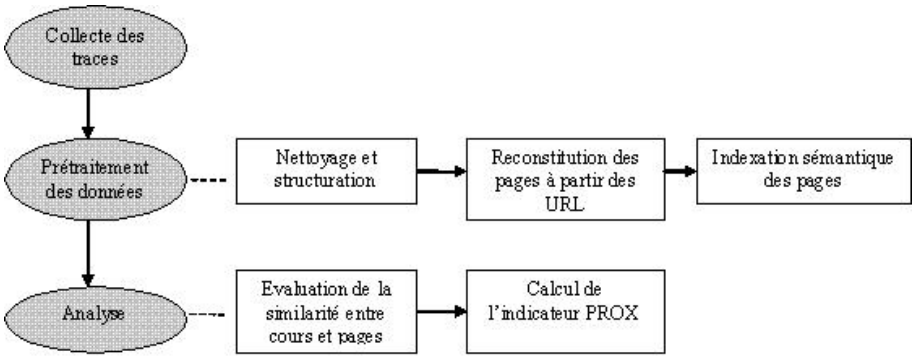


FIG. 1 – *Schéma global de la démarche suivie.*

3.2 Deuxième phase : Prétraitement des données

Le prétraitement se scinde en deux étapes : le nettoyage et la structuration. Vu que nous nous intéressons à la sémantique des pages visitées, nous ajoutons une étape à la phase de prétraitement : *l'extraction des métadonnées des pages*.

Nettoyage et structuration des données. Le nettoyage consiste à écarter les données insignifiantes et superflues. Dans notre contexte, il s'agit des pages consultées une seule fois durant la session, et pendant un laps de temps très réduit. De plus, dans le cadre de la présente étude, nous ne considérons que les URL consultées par l'apprenant. Les données sont ensuite structurées pour faciliter leur interprétation. Le tableau 1 illustre la structure adoptée pour une session d'un apprenant.

URL	Titre de la page	Heure d'accès	Durée
http ://www.efad.ufc.dz/mescours/ index.php ?mu=vi060001	Cours de Grammaire anglaise (GRAMMAR)	12 :50 :08	60 mn
http ://www.doctissimo.fr/html/ sante/sante.htm	Santé- votre santé avec Doctissimo	13 :00 :55	15 mn
http ://www.ucl.ac.uk/internet-grammar/	The internet grammar of English Google -	13 :05 :10	30 mn
http ://www.developpez.com/	Accueil - Club d'entraide des développeurs francophones	13 :20 :11	10 mn

TAB. 1 – *Structure du fichier de l'historique de navigation d'un apprenant durant une session d'apprentissage.*

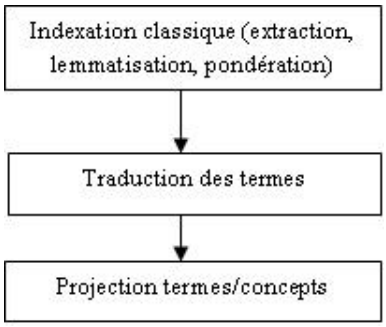


FIG. 2 – *Etapes du processus d'indexation.*

Afin d'évaluer la similarité entre le cours et les pages Web consultées, qui est l'objectif de notre système, nous représentons chaque page par ses mots clés. Il est à noter que les cours sont indexés par le concepteur, en revanche les pages consultées par l'apprenant en dehors du contenu ne sont pas forcément indexées. Nous distinguons deux cas de figure. Le premier est celui où les pages Web visitées sont accompagnées de métadonnées. Dans ce cas, les mots clés sont extraits directement des pages. Dans le deuxième cas, les pages Web ne contiennent pas de métadonnées. Ainsi, nous proposons d'extraire les métadonnées par des algorithmes d'indexation sémantique.

Indexation sémantique des pages. Pour effectuer l'indexation, nous devons disposer du contenu textuel des pages Web. Pour ce faire, nous intégrons un module de reconstitution des pages Web consultées lors d'une session de navigation à partir des URL. La reconstitution se fait en aspirant les pages, et en traitant les balises HTML afin d'extraire le contenu significatif. Une fois que nous disposons du contenu intégral de chaque page du fichier de trace, ces pages sont sauvegardées et nous procédons à leur indexation grâce au module d'indexation.
Le module d'indexation extrait les termes de chaque page et les pondère comme dans une indexation classique. Les termes sont ensuite convertis en concepts par une projection sur l'ontologie générale WordNet. Nous avons en effet opté pour une ontologie générale vu que les pages appartiennent à des domaines distincts.
La projection terme/concept se fait de la manière suivante : pour chaque mot clé, les concepts de l'ontologie sont extraits, bien entendu, il est possible de trouver plusieurs sens pour un terme. Un processus de désambigüisation examine donc les différents sens associés à un mot clé et localise les sens les plus proches du contexte du document Web. Nous envisageons d'intégrer un module de traduction des vecteurs de termes pour les pages non anglo-saxonnes en utilisant le service Web Babelfish[3] d'Altavista, vu que nous travaillons avec la version anglaise (originale) de WordNet.

L'évaluation de la similarité entre les pages de cours et les pages de navigation Web revient à évaluer la similarité entre les vecteurs de concepts les décrivant respectivement.

[3] http ://babelfish.altavista.com

3.3 Troisième phase : interprétation et analyse

Similarité sémantique entre deux documents. Comme l'objectif est d'évaluer la similarité sémantique entre documents (pages web visitées et cours étudié), et sachant que ces pages sont représentées par des vecteurs (concept, poids), il nous faut employer une technique qui permet de mesurer la similarité entre ensembles pondérés de concepts.

Varelas et al. (2005) proposent un nouveau modèle de recherche d'information (SSRM), basé sur la similarité sémantique. Nous adopetons leurs formule, mais au lieu de sommer toutes les valeurs de similarité deux à deux, nous ne prenons que les similarités maximales. A partir de là, nous calculons la similarité sémantique entre chaque page visitée et le cours étudié de la plate-forme de formation, deux à deux, comme le montre la formule suivante :

$$Sim(q,d) = \frac{\sum_i q_i d_j sim(i,j)}{\sum_i \sum_j q_i d_j} (1)$$

Où :
- i représente les concepts du document q ;
- j : le concept du document d, ayant la similarité maximale avec i ;
- qi est le poids du concept i dans le document q ;
- dj est celui du concept j dans le document d.

Ici, Sim (i, j) est la similarité sémantique entre les deux concepts i et j, calculée à partir de l'ontologie utilisée, dans notre cas Wordnet.

Proximité entre le cours et les pages Web visitées. Afin de déterminer le degré d'intérêt que porte un apprenant à un cours donné, plusieurs facteurs sont pris en considération : le taux de participation de l'apprenant dans des forums en relation avec le cours, le taux d'enregistrements des pages du cours, et la consultation de pages Web en relation avec le cours. Afin d'obtenir ce dernier facteur, nous proposons l'indicateur de proximité *Prox* donné par la formule suivante :

$$Prox(cours, pagesendehorsducours) = \frac{\sum D_i \times Sim(Cours, P_i)}{\sum D_i}$$

Où :
- Cours fait référence au cours étudié ;
- Pi est la page de navigation Web consultée en même temps que le cours ;
- Di : la durée de consultation de la page d'indice i ;
- Sim (Cours, Pi) : la similarité sémantique entre le cours et la page i. Cette similarité est calculée par la formule (1).

Nous calculons la similarité sémantique entre le cours et les pages de navigation Web apparaissant dans le même intervalle de temps (i.e. : consultées en même temps que le cours), deux à deux. L'indicateur de Proximité Sémantique entre un cours consulté et les pages de Navigation de l'apprenant est donc égal à la moyenne des valeurs de similarité entre le cours et chacune des pages Web.

Notons que l'indicateur que nous venons de proposer prend ses valeurs dans un intervalle [0, 1], vu que les mesures de similarité entre concepts/documents sont toujours comprises dans cet intervalle. L'indicateur prend la valeur maximale 1 lorsque les concepts visités par l'apprenant sont identiques à ceux qu'il consulte sur le Web, et 0 dans le cas contraire. Notons

également que le fait d'associer les durées de navigations aux pages attribue un poids aux pages selon leur durée de navigation. Ainsi, une page qui a été visitée pendant un temps court aura moins d'influence qu'une autre qui l'a été pendant un temps plus long.

Considérons l'exemple de l'historique de navigation illustré dans le tableau1. La première URL représente le cours étudié par l'apprenant. Il s'agit d'un cours de grammaire anglaise, décrit par les concepts suivants : grammar, english, language, linguistics. Selon le tableau1, l'apprenant en question a visité en même temps que le cours, une page d'un site de grammaire anglaise(P1), une page d'un site médical (P2) et une page d'un site d'informatique(P3). P1 étant représentée par les concepts suivants : (english, grammar, language, syntax, linguistics), P2 par (health, magazine, guide, drug, diseases) et P3 par (data-processing, computer, developpment).

Nous calculons la similarité sémantique entre le cours et chacune de ces pages en utilisant la formule (1). Nous considérons dans cet exemple que tous les poids sont égaux à 1. Nous obtenons :

Sim(Cours, P1) =0.8126 , Sim(Cours, P2) =0.0369 , Sim (Cours, P3)=0.0701

L'indicateur *Prox* sera égal à 0.46. Nous pouvons donc dire que les pages visitées par l'apprenant se rapprochent moyennement du cours étudié.

4 Conclusion

Dans cet article, nous avons exploité les mesures de similarité sémantique entre concepts d'une ontologie, pour la détection du degré d'intérêt que portent les apprenants aux cours qui leurs sont proposés. Ainsi, nous ajoutons un nouveau champ d'application de ces mesures de similarité.

Dans nos futurs travaux, nous partirons de l'indicateur de proximité entre cours et pages consultées, pour tirer d'autres indicateurs plus génériques. Par exemple, en rapprochant plusieurs sessions d'apprenants différents, nous pouvons déduire le degré d'intérêt porté à un cours donné (par la totalité des apprenants). De même, en rapprochant plusieurs sessions d'un même apprenant, nous déterminerons la motivation de cet apprenant par rapport à la formation en général, sa manière d'apprendre, ses points forts et ses points faibles, etc.

D'autre part, il serait intéressant si nous exploitons les différentes relations entre concepts dans une ontologie, et non uniquement la relation *est un*, telles que les relations : *fait-partie-de*, *se-compose- de*, etc. La prise en compte de ces relations permet de considérer des concepts qui, en prenant en compte la relation *est un* serait très dissimilaires aux concepts du cours, alors qu'ils ne le sont pas en réalité.

Références

Bernstein, E. Kaufmann, C. Buerki, et M. Klein (2005). Classification and regression trees. *In Proceedings of the 7 Internationale Tagung Wirtschaftsinformatik*, 1347–1366.

Budanitsky, A. et G. Hirst (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. *In Proceedings of the Workshop on WordNet and Other Lexical Resources, ACL.*

Jiang, J. et D. Conrath (1997). Semantic similarity based on corpus statistics and lexical terminology. *In Proceedings of the International Conference on Computational Linguistics, RoclingX*.

Lin, D. (1998). An information-theoric definition of similarity. *In Proceedings of the 15th international conference on Machine Learning*, 296–304.

Lord, P., R. Stevens, A. Brass, et C. Goble (2003). Semantic similarity measures as tools for exploring the gene ontology. *In Proceedings of the Pacific Symposium on Biocomputing*, 601–612.

Rada, R., H. Mili, E. Bicknell, et M. Blettner (1989). Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man and Cybernetics 19(1)*, 17–30.

Resnik, P. (1995). Using information content to evaluate similarity in a taxonomy. *In Proceedings of the 14th joint conference in Artificial Intelligence*.

Resnik, P. (1999). Semantic similarity in a taxonomy: an information based measure and its application to problems of ambiguity in natural langage. *Journal of Artificial Intelligence Research 11*, 95–130.

Sanderson, M. et W. Croft (1999). Deriving concept hierarchies from text. *Proceedings of the 22nd International ACM SIGIR Conference*, 206–213.

Seco, N., T. Veale, et J. Hayes (2004). An intrinsic information content metric for semantic similarity in wordnet. *Proceedings of ECAI'2004, the 16th European Conference on Artificial Intelligence, Valence, Espagne*.

Srivastava, J., P. Desikan, et et V. Kumar (2003). Web mining - concepts, applications and research directions. *Boston : Data Mining: Next Generation Challenges and FutureDirections, AAAI/MIT Press, MA*.

Thieu, O. Steichen, C. L. Bozec, E. Zapletal, et M.-C. Jaulent (2004). Mesures de similarité pour l'aide au consensus en anatomie pathologique. *In Proceedings of the 5th International Conference on Internet Computing*, 225–236.

Varelas, G., E. Voutsakis, P. Raftopoulou, E. Petrakis, et E.E.Milios (2005). Semantic similarity methods in wordnet and their application to information retrieval on the web. *In Proc. of WIDM 2005*, 10–16.

Wu, Z. et M. Palmer (1994). Verb semantics and lexical selection. *In: Proceedings of the 32nd annual meetings of the associations for computational linguistics*, 133–138.

Summary

Semantic similarity measures play an important role in information retrieval; they have become a very prospected way. In this paper, we apply the similarity measures to concepts in ontology, in order to identify the semantic links between Web pages, visited by a learner, and the course being consulted in the context of an Open and Distance Training. The similarity between the course and Web pages, visited in the same while as the training, allows us to find out how interested a learner is in the learning contents to discover the stage of interest a learner reaches in the learning contents.

Tables des matières