Explorer les théorèmes d'une TBox

Julien Corman

Explorer les théorèmes d'une TBox

Julien Corman ⋆

IRIT, Université de Toulouse
Julien.Corman@irit.fr

Résumé : Nous présentons deux tests appliqués à plusieurs ontologies de domaine, afin d'identifier de possibles erreurs de modélisation, comprises ici comme des décalages entre ce que l'ontologie exprime et ce que son auteur souhaite exprimer. Ces tests viennent en complément de méthodologies et outils existants, et s'appliquent en particulier à des ontologies dont la consistance logique a déjà été vérifiée. Ils sont basés sur l'exploration d'une classe de théorèmes de l'ontologie évaluée. Le premier test fait également usage d'un sous-ensemble de catégories de l'ontologie fondationnelle DOLCE.

Mots-clés : ontologie, ontologie fondationnelle, logiques de description, représentation des connaissances

1 Introduction

Les ontologies, ou bases de connaissances, sont souvent présentées comme un élément essentiel au déploiement des technologies sémantiques. Elles peuvent être vues comme des bases de données dont le modèle de données et les données sont représentées par des formules logiques, le tout formant une théorie. Cette axiomatisation facilite l'interopérabilité entre sources de données hétérogènes, ou l'interrogation de données incomplètes.

La conception d'une ontologie est cependant une tâche réputée complexe. Il est notamment difficile, même pour un expert du domaine modélisé :

– de maintenir la cohérence de l'ontologie au-delà d'une certaine taille,
– de produire une ontologie suffisamment axiomatisée pour prévenir des usages non souhaités.

⋆. Merci à Nathalie Aussenac-Gilles et Laure Vieu pour leurs conseils et relectures

De nombreux outils d'aide à la modélisation ont été développés afin de faciliter ce processus. En complément de ces outils, nous proposons ici deux tests originaux appliqués à la partie terminologique (*TBox*) de l'ontologie, c'est à dire son modèle de données.

Ces tests visent à identifier de possibles erreurs de modélisation, comprises comme des décalages entre ce que l'ontologie implique ou autorise, et ce que son auteur souhaite exprimer. Ils sont basés sur l'exploration d'une partie des théorèmes de l'ontologie. Le premier test fait également usage d'une ontologie fondationnelle externe, en l'occurrence un sous-ensemble des catégories de DOLCE (Masolo *et al.*, 2002).

L'article est découpé comme suit : la première section donne un bref aperçu des outils et méthodes existants pour détecter ce type d'erreurs de modélisation. La seconde section détaille les objectifs et la méthode suivie. Les troisième et quatrième sections décrivent les tests eux-mêmes. Les deux dernières sections présentent les résultats de ces tests appliqués à plusieurs ontologies de domaines, et quelques prolongements envisagés.

2 État de l'art

Une première étape pour la détection de potentielles erreurs de modélisation consiste à vérifier la consistance d'une ontologie (et la satisfiabilité de chacun de ses concepts), vérification qui peut être effectué à l'aide un raisonneur. Nous avons par exemple utilisé Hermit (Shearer *et al.*, 2008) dans ce qui suit. Malheureusement, cela est loin d'être suffisant : une ontologie consistante peut néanmoins contredire les intentions de son auteur, ou autoriser des interprétations non souhaitées par celui-ci.

De nombreuses méthodologies ont donc été mises au point afin de prévenir ces cas de figure, basées sur des considérations philosophiques, comme la méthode OntoClean (Guarino & Welty, 2000), ou plus empiriques (Rector, 2003; Gangemi & Presutti, 2009). Les *ontologies fondationelles*, comme BFO ou DOLCE, toutes deux décrites dans (Masolo *et al.*, 2002), offrent quant à elles des représentations cohérentes et fondées philosophiquement de catégories indépendantes du domaine représenté.

L'automatisation de ces méthodes reste cependant une voie sous-explorée. Parmi les outils existants peuvent être mentionnés l'éditeur OntoUML (Benevides & Guizzardi, 2009; Guizzardi, 2005), basé sur OntoClean et DOLCE, qui contraint l'auteur à expliciter certains de ces choix conceptuels. OOPS ! (Poveda Villalon & Suárez-Figueroa, 2012) au contraire effectue une série de vérifications syntaxiques et sémantiques *a posteriori*. Corcho *et al.*

(2009) et Sales *et al.* (2012) ont également développé la recherche d' *antipatterns* susceptibles d'indiquer des erreurs de modélisation, tandis que Verdezoto & Vieu (2011) ont effectué une analyse ontologique des *synsets* supérieurs de Wordnet, afin d'identifier automatiquement de probables erreurs dans l'usage de la relation de méronymie. Le questionnaire mis au point par Pammer (2010) permet également à l'utilisateur de valider une série de théorèmes, ce qui se rapproche de ce que nous proposons ici.

Enfin d'autre outils privilégient l'exploration des interprétations possibles d'une ontologie, comme celui développé par Ferré & Rudolph (2012), qui s'intéresse aux concepts satisfiables, ou encore celui proposé par Benevides *et al.* (2010), qui présente à l'utilisateur des structures de Kripke satisfaisant un modèle de données en cours de construction.

3 Objectifs et méthode générale

Les deux tests présentés ici peuvent être vus comme un prolongement de ceux effectués grâce à un outils comme OOPS !. Le premier test requiert une rapide analyse ontologique manuelle d'une sélection de rôles, à l'aide d'un jeu de catégories du type de celles présentées en 3.2. Une partie des théorèmes de l'ontologie est ensuite parcourue à la recherche de possibles incohérences entre cette analyse et la TBox. Le second test en revanche vise à suggérer de possibles axiomes manquants. Il exploite le même ensemble de théorèmes, couplé à quelques tests de satisfiabilité. Nous n'abordons en revanche pas ici la question de la mise à jour de l'ontologie après détection d'une erreur.

Les ontologies auxquelles ces tests ont été appliqués sont CIDOC CRM (Doerr, 2003), utilisée dans le domaine de l'héritage culturel, l'ontologie du cinéma décrite dans (Pradel *et al.*, 2011), l'ontologie des plantes ornementales MOANO (Aussenac-Gilles *et al.*, 2013), et la célèbre ontologie des pizzas [1] en tant qu'exemple jouet. Leur consistance a été vérifiée, puis elles ont été soumises à OOPS !, et mises à jour en conséquence. [2]

Aucun jeu de données (*ABox*) n'a été utilisé, quoique le premier des deux tests puisse être étendu par intégration de données fiables (c'est un des prolongements immédiats du travail présenté ici). Enfin, pour des raisons pratiques, nous nous sommes substitués aux auteurs afin d'effectuer

1. http ://www.co-ode.org/ontologies/pizza/pizza.owl

2. Hormis les concepts insatisfiables de l'ontologie des pizzas, un emploi erronné de la transitivité pour CIDOC CRM, et deux axiomes manquants (symétrie et transitivité) pour un rôle de MOANO, les erreurs détectées étaient de nature syntaxique.

l'analyse ontologique requise par le premier test, en essayant de rester fidèles à leurs intentions manifestes.

Dans ce qui suit, conformément à la terminologie utilisée en logiques de description, les prédicats unaires (*owl :Class*) sont appelés des *concepts*, et les prédicats binaires (*owl :ObjectProperty*) des *rôles*. *Modèle* est utilisé avec son acception logique, tandis que *TBox* lui est préféré pour faire référence à un modèle de données. Enfin une *incompatibilité* (traduction de *disjointness*) désigne un axiome du type $A \sqsubseteq \neg B$, c'est-à-dire l'impossibilité pour un même individu d'être instance à la fois de A et de B.

3.1 Théorèmes

L'exploitation des théorèmes d'une ontologie (c'est-à-dire des connaissances qui y sont implicitement exprimées) en tant qu'aide à la modélisation est une possibilité encore peu exploitée. Pourtant, comme le montre Pammer (2010), il est difficile pour un individu d'appréhender les implications d'une ontologie à partir de ses seuls axiomes. La génération de théorèmes peut également améliorer la recherche d'*antipatterns*, comme l'ont récemment expérimenté Roussey & Zamazal (2013).

En pratique, l'exploitation de théorèmes pose au moins deux difficultés : calculer un sous-ensemble de théorèmes pertinents, et le parcourir. Pammer choisit ainsi de calculer un ensemble restreint, à savoir les subsomptions entre concepts (atomiques ou complexes) déjà présents dans la TBox. Nous pensons qu'il peut être utile d'exploiter un ensemble de théorèmes plus vaste, à condition de mettre au point des stratégies efficaces pour les parcourir.

Pour ce qui est du calcul, une procédure déductive ad-hoc peut souvent être appliquée afin de dériver une part importante d'une classe syntaxique finie de théorèmes (non triviaux). Dans ce qui suit cependant, afin de garantir la complétude des résultats, nous avons choisi d'employer une méthode beaucoup plus coûteuse : générer à partir de la signature de la TBox toutes les formules possibles de la classe syntaxique visée, et tester la satisfiabilité de leurs négations, à l'aide d'un raisonneur. [3]

Nous nous concentrerons ici sur le parcours de ces théorèmes, et plus exactement d'une certaine classe syntaxique, de la forme $A \sqsubseteq \exists R.B$, où A et B sont des concepts atomiques (non-niés), et R un rôle, par exemple :

Pizza$\sqsubseteq \exists$*hasbase.Pizzabase*

3. Moyennant quelques optimisations, non décrites ici.

A titre d'exemple, CIDOC CRM compte 2430 formules de ce type parmi ses théorèmes, dont seulement 49 sont des axiomes.

3.2 Catégories de DOLCE

Le premier test s'appuie également sur quelques catégories issues d'une ontologie fondationnelle, en l'occurrence DOLCE, choisies pour leur relative facilité d'utilisation (mais tout autre ensemble de catégories incompatibles peut être utilisé) :

- `Perdurant`, c'est-à-dire un événement, état ou processus,
- `Endurant`, partitionné en `PhysicalEndurant` (entité physique), et `NonPhysicalEndurant` (sans réalité physique, mais dont l'existence est limitée dans le temps, comme une œuvre, une entreprise ou une monnaie),
- `Quality`, qui désigne une propriété d'un individu, comme sa taille, sa durée ou son prix,
- `Region`, qui est la valeur d'une instance de `Quality` (partitionné en `TemporalRegion`, `PhysicalRegion` et `AbstractRegion`).

Il s'agit d'une description très informelle, les détails et implications philosophiques pouvant être trouvés dans (Masolo *et al.*, 2002). Nous réserverons le terme *catégorie* à ces catégories, par opposition à *concept*, qui désignera un concept de l'ontologie testée.

4 Test 1 : concepts hétérogènes vis-à-vis du jeu de catégories utilisé

Ce test peut être rapproché de certains de ceux appliqués par (Verdezoto & Vieu, 2011) à Wordnet, qui exploitent eux-aussi un sous-ensemble de catégories de DOLCE. Il s'en distingue cependant en ce qu'il requiert une analyse manuelle de rôles de l'ontologie testée, et non de concepts. Une catégorie doit être attribuée au domaine (*rdfs :domain*) et au codomaine (*rdfs :range*) d'un petit nombre de rôles, sélectionnés grâce aux théorèmes précédemment calculées. Ces théorèmes sont ensuite parcourus, afin d'identifier des concepts qui, dans tout modèle où chaque concept de l'ontologie est instancié, sont hétérogènes vis-à-vis du jeu de catégories utilisé [4] (ce qui est improbable, quoique pas impossible).

4. On pourrait choisir d'intégrer directement à l'ontologie les catégories et restrictions de domaine et codomaine. Mais un concept hétérogène n'en devient pas nécessairement insatisfiable (il est par exemple satisfait dans un modèle ou deux de ses instances instancient chacune une catégorie d'une paire de catégories incompatibles).

L'ontologie du cinéma sera utilisée ici en guise d'illustration. 699 formules valides (de la forme mentionnée en 3.1) ont été calculées pour cette ontologie, et 36 rôles distincts y apparaissent. Nous avons associé à chacun de ces rôles R deux catégories de DOLCE $dom(R)$ et $ran(R)$ en guise de domaine et codomaine, en nous basant sur les annotations de l'ontologie, et sur l'usage apparent de ces rôles. Par exemple, compte tenu de la distinction claire faite par l'ontologie du cinéma entre compétitions (considérées comme des événements) et récompenses, et d'après l'usage apparent des rôles *concourtPourCompétition* et *décerne*, nous avons assigné :

ran(***concourtPourCompétition***) $\leftarrow$ `Perdurant`,

ran(***décerne***) $\leftarrow$ `NonPhysicalEndurant`.

Ces assignations peuvent être effectuées rapidement, même sans maîtrise des notions philosophiques sous-jacentes. [5] [6] D'après les incompatibilités entre catégories de DOLCE, 294 paires de rôles $< R_1, R_2 >$ sont telles que $dom(R_1) \sqsubseteq \neg dom(R_2)$, 332 telles que $ran(R_1) \sqsubseteq \neg ran(R_2)$, et 687 telles que $ran(R_1) \sqsubseteq \neg dom(R_2)$.

L'ensemble des théorèmes calculés a ensuite été parcouru, à la recherche d'un des trois motifs suivants (où A, A', B, C, D et E sont des concepts atomiques) :

1.
$$\begin{array}{ll} B \sqsubseteq \exists R_1.C & B \sqcup D \sqsubseteq A \\ D \sqsubseteq \exists R_2.E & \forall A' : (((B \sqcup D) \sqsubseteq A') \Rightarrow (A \sqsubseteq A')) \\ dom(R_1) \sqsubseteq \neg dom(R_2) & \end{array}$$ [7]

2.
$$\begin{array}{ll} B \sqsubseteq \exists R_1.A & ran(R_1) \sqsubseteq \neg ran(R_2) \\ C \sqsubseteq \exists R_2.A & \forall A' : (((B \sqsubseteq \exists R_1.A') \sqcap (C \sqsubseteq \exists R_2.A')) \Rightarrow \\ & \quad (A \sqsubseteq A')) \end{array}$$

3.
$$\begin{array}{ll} B \sqsubseteq \exists R_1.A & ran(R_1) \sqsubseteq \neg dom(R_2) \\ A \sqsubseteq \exists R_2.C & \end{array}$$

5. Moins de 20 minutes en moyenne, vraisemblablement moins si effectuées par l'auteur de l'ontologie.

6. Nous avons constaté que les erreurs d'assignation ne sont généralement pas un réel problème, car elles deviennent évidentes lorsque les tests sont effectués.

7. La dernière contrainte permet d'exclure les cas redondants où un concept subsumant A est signalé comme hétérogène.

Dans tous les cas, A est considéré comme hétérogène, et signalé à l'utilisateur avec la paire de rôles $< R_1, R_2 >$. Par exemple, plusieurs théorèmes de l'ontologie du cinéma ont pour forme :

$B \sqsubseteq \exists$*concourtPourCompétition.PrixDécerné*,

tandis que plusieurs autres ont pour forme :

$C \sqsubseteq \exists$*décerne.PrixDécerné*,

ce qui correspond au second motif. Par conséquent, dans tout modèle satisfaisant cette ontologie, et où B et C sont instanciés, *PrixDécerné* est hétérogène, au sens où il compte parmi ses instances au moins un `Perdurant` et un `NonPhysicalEndurant`, deux catégories incompatibles. Il s'agissait d'une erreur dans l'axiome suivant :

CourtMétragePrimé $\equiv$ *(CourtMétrage* $\sqcap$
$\exists$*concourtPourCompétition.PrixOeuvreCinématographique)*

où *concourtPourCompétition* avait été utilisé au lieu de *concourtPourPrix*.

4.1 Suggestion d'axiomes

Ce test vise à suggérer de possibles axiomes manquants, en particulier des incompatibilités. Il est basé sur l'intuition suivante : lorsque l'utilisateur ajoute un concept atomique C à une ontologie $\mathcal{O}$, il sous-entend probablement que, dans tout modèle satisfaisant $\mathcal{O}$, s'il existe une instance de C, il existe au moins un autre individu [8] qui n'est pas un C, et instancie un autre concept de $\mathcal{O}$. [9] Nous pouvons donc vérifier si (en FOL) :

$\mathcal{O} \models \exists x C(x) \rightarrow \exists y \neg C(y)$

Pour effectuer cette vérification, il suffit d'ajouter (temporairement) à $\mathcal{O}$ l'axiome factice $\top \sqsubseteq C$. Si $\mathcal{O}$ reste consistante, alors il existe un modèle de $\mathcal{O}$ où tous les individus sont des C. Par exemple, ajouter à l'ontologie des pizzas l'axiome $\top \sqsubseteq$ *PizzaTopping* n'entraîne pas d'inconsistance. Il existe donc un modèle de cette ontologie où tous les individus sont des *PizzaToppings* (sans *Pizza* ou *PizzaBase*, ce qui est assez contre-intuitif).

Un motif simple permettant d'exclure cette possibilité est la conjonction d'une formule du type *PizzaTopping* $\sqsubseteq \exists R.A$, et d'une incompatibilité *PizzaTopping* $\sqsubseteq \neg A$. [10] Il est possible sur cette base de suggérer de nouveaux axiomes :

8. Cet autre individu peut être anonyme.

9. Il s'agit d'une simple heuristique : on peut notamment discuter sa pertinence pour des concepts "périphériques", comme *IceCream* ou *Spicyness* dans l'ontologie des pizzas.

10. Nous avons essayé des motifs plus complexes, sans grand succès.

- si $C \sqsubseteq \exists R.A$ est valide, pour tout B tel que $A \sqsubseteq B$, l'axiome $C \sqsubseteq \neg B$ est suggéré
- si $C \sqsubseteq \neg A$ est valide [11], pour tout B tel que $B \sqsubseteq A$, l'ajout d'un axiome $C \sqsubseteq \exists R.B$ est suggéré (le choix de R étant laissé à l'utilisateur).

Dans le cas de *PizzaTopping*, il est par exemple possible d'ajouter l'axiome *PizzaTopping* $\sqsubseteq \exists$*isToppingOf.Pizza*, étant donné que l'axiome *PizzaTopping* $\sqsubseteq \neg$*Pizza* est déjà présent.

5 Résultats

5.1 Test 1

Pour l'ontologie du cinéma, seules 6 paires de rôles ont été signalées, toutes ces paires contenaient le rôle *concourtPourCompétition*, renvoyant à l'axiome erroné déjà mentionné.

Pour CIDOC CRM, 6 concepts hétérogènes (et 286 paires de rôles) ont été signalés. Pour 3 d'entre eux, cette hétérogénéité est en adéquation avec leurs définitions dans les annotations de l'ontologie. Les 3 autres sont *Type*, *ConceptualObject* et *Material*, et les 13 paires de rôles correspondant contiennent toutes *consistsOf*. Il peut s'agir soit d'une réelle hétérogénéïté de ces trois concepts (instanciés à la fois par des `PhysicalEndurants` et des `NonPhysicalEndurants`), soit d'un mauvais étiquetage de notre part (*ran (consistsOf)* $\leftarrow$ `PhysicalEndurant`).

Pour MOANO, 13 paires de rôles et 6 concepts ont été signalés. Un axiome vraisemblablement erroné a pu être identifié, à savoir :

AlceaRosea $\sqsubseteq \exists$ *nécessiteElementInorganique.Humide*

Après discussion avec les auteurs, ce test a également permis de suggérer une réorganisation partielle de la taxonomie, basée sur le constat d'hétérogénéïté des concepts *FacteurEnvironnemental* (`Perdurant` et `PhysicalRegion`) d'une part, et *ComposantSol*, *Calcaire*, *Argile* et *QualitéDrainage* (`PhysicalRegion` et `PhysicalEndurant`) d'autre part.

5.2 Test 2

Le test a été appliqué en priorité aux concepts supérieurs de chaque taxonomie. Si $\top \sqsubseteq C$ entraîne l'inconsistance de $\mathcal{O}$, il est inutile d'ap-

11. A pourrait en fait être tout autre concept devenu insatisfiable lors de l'ajout de l'axiome $\top \sqsubseteq C$, mais le bruit obtenu nous a dissuadé de poursuivre dans cette voie.

pliquer le test aux concepts subsumés par C. Dans le cas contraire, et si aucun axiome n'était proposé pour C, le test a été appliqué aux concepts immédiatement subsumés par C, et ce récursivement. [12]

Pour 9 concepts de CIDOC CRM, le test ne provoquait pas l'inconsistance de $\mathcal{O}$, contre 99 pour le cinéma, et 1272 pour MOANO (ce nombre est cependant biaisé, dû à de volumineuses taxonomies importées). L'essentiel des propositions étaient des incompatibilités (246 sur 258), et la plupart étaient correctes (252 sur 258), quoique cela soit encore une fois biaisé par les taxonomies importées de MOANO. De nombreuses suggestions étaient triviales (comme *Acteur* $\sqsubseteq \neg$ *Film*). Dans plusieurs cas cependant, les propositions faites permettaient de prévenir de probables confusions (par métonymie). On peut par exemple mentionner *TemporalEntity* $\sqsubseteq \neg$ *TimeSpan* pour CIDOC CRM, *PrixDécerné* $\sqsubseteq \neg$ *Evénement* pour le cinéma, ou *Plante* $\sqsubseteq \neg$ *Taxon* pour MOANO.

6 Conclusions et perspectives

Nous avons montré dans cet article qu'il est possible, grâce à une exploration ciblée de théorèmes et à une analyse ontologique rapide, d'identifier de probables erreurs de modélisation dans une TBox, au-delà des tests de satisfiabilité. Parmi les prolongements (en cours ou à venir) de ce travail figurent :

- des tests basés sur un jeu plus important de théorèmes, et intégrant des données issues de l'ABox,
- une procédure efficace pour la dérivation de théorèmes,
- l'utilisation de catégories plus fines, en particulier parmi les `Non-PhysicalEndurants`.

Références

AUSSENAC-GILLES N., BUSCALDI D., COMPAROT C. & KAMEL M. (2013). Enrichissement d'ontologies grâce à l'annotation sémantique de pages web. *Revue des Nouvelles Technologies de l'Information*, **E-24**, 229–234.

BENEVIDES A. B. & GUIZZARDI G. (2009). A model-based tool for conceptual modeling and domain ontology engineering in OntoUML. *Enterprise Information Systems*, p. 528–538.

12. Avec une profondeur maximale de 4 pour MOANO, due au nombre important de concepts.

BENEVIDES A. B., GUIZZARDI G., BRAGA B. F. B. & ALMEIDA J. P. A. (2010). Validating modal aspects of OntoUML conceptual models using automatically generated visual world structures. *Journal of Universal Computer Science*, **16**(20), 2904–2933.

CORCHO O., ROUSSEY C., VILCHES-BLÁZQUEZ L. & PEREZ DOMINGUEZ I. (2009). Pattern-based OWL ontology debugging guidelines.

DOERR M. (2003). The CIDOC conceptual reference module : an ontological approach to semantic interoperability of metadata. *AI magazine*, **24**(3), 75.

FERRÉ S. & RUDOLPH S. (2012). Advocatus Diaboli–Exploratory enrichment of ontologies with negative constraints. *Knowledge Engineering and Knowledge Management*, p. 42–56.

GANGEMI A. & PRESUTTI V. (2009). Ontology design patterns. *Handbook on Ontologies*, p. 221–243.

GUARINO N. & WELTY C. (2000). A formal ontology of properties. *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, p. 191–230.

GUIZZARDI G. (2005). *Ontological foundations for structural conceptual models*. Centre for Telematics and Information Technology.

MASOLO C., BORGO S., GANGEMI A., GUARINO N., OLTRAMARI A., OLTRAMARI R., SCHNEIDER L., ISTC-CNR L. P. & HORROCKS I. (2002). Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology.

PAMMER V. (2010). *Automatic Support for Ontology Evaluation*. PhD thesis, Graz University of Technology.

POVEDA VILLALON M. & SUÁREZ-FIGUEROA M. (2012). OOPS ! OntOlogy pitfalls scanner !

PRADEL C., HERNANDEZ N., KAMEL M. & ROTHENBURGER B. (2011). Une ontologie du cinéma pour évaluer les applications du web sémantique.

RECTOR A. (2003). Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In *2nd international conference on Knowledge capture*.

ROUSSEY C. & ZAMAZAL O. (2013). Antipattern detection : How to debug an ontology without a reasoner. In *Second International Workshop on Debugging Ontologies and Ontology Mappings*.

SALES T. P., BARCELOS P. P. F. & GUIZZARDI G. (2012). Identification of semantic anti-patterns in ontology-driven conceptual modeling via visual simulation.

SHEARER R., MOTIK B. & HORROCKS I. (2008). HermiT : a highly-efficient OWL reasoner. In *5th International Workshop on OWL : Experiences and Directions*.

VERDEZOTO N. & VIEU L. (2011). Towards semi-automatic methods for improving WordNet. In *9th International Conference on Computational Semantics*.