Effects of Ontology Pitfalls on Ontology-based Information Retrieval Systems

Davide Buscaldi[1] and Mari Carmen Suárez-Figueroa[2]
[1]*Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS, (UMR7030), Villetaneuse, France*
[2]*Ontology Engineering Group (OEG), Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain*

Keywords: Ontology-based Information Retrieval Systems, Ontology Quality, Ontology Pitfalls.

Abstract: Nowadays, a growing number of information retrieval systems make use of ontologies to improve the access to textual information, especially in domain-specific scenarios, where the knowledge provided by ontologies represents a key factor. Such kinds of retrieval systems are often referred to as ontology-based or semantic information retrieval systems. The quality of ontologies plays an important role in such systems in the sense that modelling errors in the ontologies may deteriorate the quality of the results obtained by these systems. In this paper we provide a comprehensive analysis of how ontology pitfalls have an influence on these kinds of systems. This study allows us to have a more complete understanding of the role of ontology quality in the information retrieval field. Our survey shows that pitfalls may act as an indicator not only of possible problems in ontology design, but also of OWL features overseen by system developers.

1 INTRODUCTION

Since the introduction of the concept of Semantic Web by Tim Berners-Lee in 2001 (Berners-Lee et al., 2001), the Information Retrieval (IR) research community has shown a growing interest in Semantic Web technologies. Ontologies are one of the most important technologies proposed in the context of the Semantic Web. Many IR researchers saw the possibility to use ontologies as an external knowledge source to be integrated into IR systems in order to improve their performance when accessing to textual information. Therefore, in the last decade, a growing number of IR systems based on ontologies have appeared, such as KIM (Kyriakov et al., 2004), MELISA (Abasolo and Gomez, 2000), and TextViz (Reymonet et al., 2010).

Like any other resources used in software systems, ontologies need to be evaluated before (re)using them in other ontologies and/or applications. Results obtained by ontology-based applications can be affected by the quality of the ontologies used. Ontology quality improvement, by specifying equivalent and disjoint classes, adding instances and properties, can significantly enhance question answering (Poveda et al., 2010) or Web search results (Tomassen and Strasunkas, 2009). Independently from the way an IR system exploits an ontology, it is clear that problems, anomalies or pitfalls that occurred in the design of an ontology built for this specific purpose may affect the results obtained by the IR systems. The use of an analysis tool could help to implement better IR systems and/or correct the detected pitfalls. OOPS! [1] (OntOlogy Pitfall Scanner) is such a tool, independent of any ontology development environment, originally intended to help ontology developers during the ontology validation (Gómez-Pérez, 2004). Currently, OOPS! provides mechanisms to automatically detect 21 pitfalls out of 29 identified in the on-line catalogue[2].

The objective of this paper is to provide an overview of the potential effects of these pitfalls on ontology-based IR systems. In order to accomplish this objective, we selected 12 out of the state-of-the-art systems and studied how they work, identifying common features and understanding which pitfalls may affect them. Unfortunately, it is very difficult to evaluate directly these systems since in most cases they are not publicly available, they have been built to work under very specific conditions, and they do not comply with W3C standards. Therefore, our

[1] http://www.oeg-upm.net/oops
[2] http://www.oeg-upm.net/oops/catalogue.jsp

Buscaldi D. and Carmen Suarez Figueroa M..
Effects of Ontology Pitfalls on Ontology-based Information Retrieval Systems.
DOI: 10.5220/0004550203010307
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2013), pages 301-307
ISBN: 978-989-8565-81-5

analysis is based exclusively on the study of the systems as described by the authors in their papers.The remainder of this paper is structured as follows: Section 2 presents related work in ontology evaluation and the pitfall catalogue used in our study. Section 3 describes general characteristics of the state-of-the-art systems analyzed. In Section 4 the analysis of the possible effects of every pitfall on each system is included. Finally, Section 5 outlines some conclusions and future steps.

2 ONTOLOGY EVALUATION AND PITFALLS

In the last decade a huge amount of research on ontology evaluation has been performed. Some of these attempts have defined a generic quality evaluation framework (Ciorascu et al;, 2003), (Gangemi et al., 2006), (Gómez-Pérez, 2004), other authors proposed to evaluate ontologies depending on the final (re)use of them (Suárez-Figueroa, 2010), others have proposed quality models based on features, criteria and metrics (Burton-Jones et al., 2005), (Djedidi et al., 2010), and in recent times methods for pattern-based evaluation have also emerged (Presutti et al., 2008). A summary of guidelines and specific techniques for ontology evaluation can be found on (Sabou et al., 2012).

Despite vast amounts of frameworks, criteria, and methods, ontology evaluation is still largely neglected by developers and practitioners. The result is many applications using ontologies following only minimal evaluation with an ontology editor, involving, at most, a syntax checking or reasoning test. Also, ontology practitioners could feel overwhelmed looking for the information required by ontology evaluation methods, and then, to give up the activity. That problem could stem from the time-consuming and tedious nature of evaluating the quality of an ontology. To alleviate such a dull task technological support that automate as many steps involved in ontology evaluation as possible have emerged (ODEClean and ODEval (Corcho et al., 2004), XDTools plug-in for NeOn Toolkit and OntoCheck plug-in for Protégé, and MoKi (Pammer, 2010)).

One of the crucial issues in ontology evaluation is the identification of anomalies or bad practices in the ontologies. Different research works have been focused on establishing sets of common errors (Rector et al., 2004), (Poveda et al., 2010). The ontology pitfalls catalogue presented in (Poveda et al., 2010)is being maintained and improved and it is available on-line. Such a version consists on the 29 pitfalls. In addition, such pitfalls can be checked using OOPS! (Poveda et al., 2012), is a web-based tool, independent of any ontology development environment, for detecting potential pitfalls that could lead to modelling errors. This tool is intended to help ontology developers during the ontology validation activity , which can be divided into diagnosis and repair. Currently, OOPS! provides mechanisms to automatically detect as many pitfalls as possible, thus helps developers in the diagnosis activity. In the near future OOPS! will include also methodological guidelines for repairing the detected pitfalls. We refer the reader to the OOPS! Site for a complete explanation of each pitfall.

3 ONTOLOGY-BASED INFORMATION RETRIEVAL SYSTEMS

The classical Information Retrieval task consists in, given a user request (usually in natural language) q and a collection of text documents D, retrieving a subset R, $|R| << |D|$ of documents that are relevant with respect to the information need expressed by the user request. IR systems are usually composed of the following components:

an indexing module, which process the collection of documents to transform each document in a representation stored in a way that allows to search the collection efficiently

a search module, which transforms the natural language query in the same way and calculates the score for each document with respect to the query

An ontology-based IR system may use the knowledge included in the ontology in the indexing module, to expand the index with information that otherwise could remain implicit in the text (for instance, extending the information that “car is a vehicle”), in the search module, to expand the query in the same way, or in both. In the first two cases we talk, respectively, about *index* and *query expansion*. In order to carry out an expansion of this kind, it is necessary to map a concept to the terms that are supposed to *denote* the concept. In many cases, the concept name is also the term that denotes the concept; in other cases, terms are stored in the ontology or in different structures. The process to map a term in a text to the corresponding concept in the ontology is called *annotation*.

Since no ontology-based IR systems are

currently publicly distributed, to perform our analysis we selected the following 12 ontology-based information retrieval systems from the state-of-the-art. The choice was determined by the level of detail provided for the description of the system. We have studied how they work and identified common features.

A. Castells (Castells et al., 2007) use an ontology structure very similar in principle to the one used in TextViz. Document annotations are stored together with concepts, but terms are not modelled as concepts. Concept labels contain the terms that are used in the annotation phase. The query is translated into a RDQL query that is run on the ontology to retrieve the relevant documents. The user is allowed to specify weights on concepts of his choice at the query formulation time.

B. KIM (Kiryakov et al., 2004). This system focuses on Named Entities (NE), that is, people, organizations, places, etc. The ontology contains, for each entity, a link to its most specific class (for instance, "Arabian Sea" is an instance of the "Sea" class). The entities are identified thanks to pattern-matching grammars. Lucene[3] is used to store the entities IDs together with the document. Entities in the queries are also converted to their respective IDs, therefore allowing to resolve cases in which an entity may have different names.

C. knOWLer (Ciorascu et al., 2003). This system uses three different OWL ontologies: the first one corresponds to the WordNet[4] ontology, where synsets have been mapped to concepts and the WordNet relationships to OWL properties. A second OWL ontology contains the terms related to each concept (terms are represented using their stemmed form). The last ontology is used to represent the documents, extracted from the Wall Street Journal corpus. This last ontology actually serves as index since the document is represented using the concepts from the other two ontologies. Queries are transformed in logical forms which are used to filter the relevant documents.

D. K-Search (Bhagdev et al., 2008). This system was developed to search technical documentation about jet engines. It uses two indexes: a standard keyword-based index, and a triple store where triples are of the form <*subject*, *relation*, *object*>. The search module extracts concepts to build triples that are translated into SPARQL[5] queries. The words appearing in the original query that cannot be mapped into concepts are sent to a traditional information retrieval system. The final result is obtained ranking documents using the traditional approach and filtering the relevant ones by means of the SPARQL query results.

E. Liu (Liu et al. 2009). do not use an ontology to carry out query or index expansion; they instead use the ontology as an index, storing terms, documents as concepts and the occurrence relationship as a property connecting terms and documents. They rely on OWL to model the ontology.

F. MELISA (Abasolo and Gomez, 2000). This system uses a medical ontology where concepts correspond to MeSH[6] terms. They expand queries using the medical ontology and the results are presented to the user to receive an additional feedback. Finally, the expanded query is re-sent to the search engine to present the final search results.

G. OWLIR (Shah et al., 2002). This system is tailored to work on Web documents, especially news documents. The ontology contains an event taxonomy (sport event, movie show event, etc.) with spatio/temporal concepts that are connected to event concepts in order to establish the relationships between an event and where and when it took place. The extraction of events from free text is carried out using an annotation tool named AeroText. The document index is expanded with the annotated concepts and relationships (triples subject-relation-object). At search phase, the queries are converted in triples which are searched into the index.

H. TARGET (Pruski et al., 2011). This system is a web search engine that is based on OWL primitives, enriched with the meronymy and antonymy relations. An ontology is used to store concepts about a specific domain. The concepts contained in queries are expanded using the concepts that are directly connected to them in the ontology. The query and the results of the Web search are transformed in graphs and a score is assigned to the top 100 retrieved pages, as a result of a graph similarity calculation.

I. Terrier-SIR (Bannour and Zargayouna, 2012). This is a Terrier[7] extension that allows, given an ontology and a terminology associated to this ontology, to index and retrieve documents using concepts as index terms. Documents weights are calculated using a concept-based version of the well-known *tf.idf* weighting scheme. The ontology is used to compute similarity values between concepts, by

[3] http://lucene.apache.org

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

[5] http://www.w3.org/TR/rdf-sparql-query/

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

[7] http://terrier.org/

taking into account the hierarchical relationships between concepts.

J. Textpresso (Müller et al., 2004). This system uses an ontology of biological concepts (e.g., gene, allele, cell, etc.) and relations connecting them (association, regulation, etc.) to expand the index and the query. In order to identify concepts in text, regular expressions are used to find the terms associated to each concept. The concepts in the ontology are structured in "categories" and "sub-categories", thus a retaining a (shallow) hierarchical structure. Queries can be expanded with more generic or specific concepts, according to the user needs.

K. TextViz (Reymonet et al., 2010). In TextViz, terms denoting concepts are stored in the same OWL ontology containing the concept themselves (terms are modelled as concepts). The ontology is also used for indexing, to store the concept instances identified in documents. Document and queries are annotated using term labels, then a similarity is calculated between document and query instances, for each document, exploiting hierarchies, using a concept similarity formula named Proxigénéa. In a test scenario, the score was also modified depending on the presence or not in the document of a relation expressed in the query, but in general the weighting scheme proposed takes into account only concepts. An important factor seems to be how terms (keywords representing the ontology concepts) are processed. Some systems consider concept names as terms, others separate terms from concepts and in this second case, terms may be also stored in the ontology as concepts of a different class. The ontology itself may or may not be used as an index. In the affirmative case, queries may be transformed in a language such as SPARQL. Some systems may use or not the taxonomic information (is-a relationship) to enrich queries (query expansion), documents (index expansion) or both, or to calculate concept similarity. Other relations (not OWL primitives) may also be used by the system.
Here we present first our analysis of the potential effect of each pitfall on the results of an ontology-based IR system, on the basis of the description provided by authors. We remember that IR systems are usually evaluated using *precision* (number of relevant document retrieved divided by the number of retrieved documents) and *recall* (number of relevant documents retrieved divided by the number of relevant documents in the collection). Secondly, we show the qualitative analysis of the impact the pitfalls in the OOPS! catalogue could have in the 12 ontology-based IR systems described in Section 3.

P1. Creating Polysemous Elements: if the concept name is used to annotate the text, this pitfall would imply having ambiguous annotations, with a possible decrease in the precision.

P2. Creating Synonyms as Classes: if the system exploits hierarchical information, or calculates distances between concepts to determine a similarity value, this pitfall may affect precision.

P3. Creating the Relationship "is" instead of using `rdfs:subClassOf`, `rdf:type` or `owl:sameAs`: if a system exploits hierarchical information, the concepts that are connected using this re-implementation of an OWL primitive may actually never be taken into account, affecting both precision and recall.

P4. Creating Unconnected Ontology Elements: the appearance of this pitfall in the ontology would affect both precision and recall, meaning that some ontology elements could not be reached.

P5. Defining Wrong Inverse Relationships: this pitfall would affect precision if the system exploits property features, such as inverse.

P6. Including Cycles in the Hierarchy: having a cycle between classes in one of the ontology hierarchies would imply that a system that exploits hierarchies in a recursive way could not finish its process.

P7. Merging Different Concepts in the same class: if the merged concepts should have different parents, the appearance of this pitfall would affect the precision of the system.

P8. Missing Annotations: if the system uses labels and/or comments to carry out some tasks, the pitfall may affect the precision and recall of the system.

P9. Missing basic Information: this pitfall may indicate that the information included in the ontology is not complete, affecting recall and/or precision. However, the ontologies used by the analysed systems do not seem to use ORSD.

P10. Missing Disjointness: the analysed systems do not use disjoint axioms. The pitfall could affect precision if a system can take into account this information.

P11. Missing Domain or Range in Properties: if a system exploits relationships other than "is-a", the appearance of this pitfall in the ontology would affect its precision.

P12. Missing Equivalent Properties: this pitfall may cause same concepts to have different parents. Therefore, if a system exploits hierarchical information, it may affect its precision and recall.

P13. Missing Inverse Relationships: this pitfall would affect precision if the system is able to exploit property features, such as inverse.

P14. Misusing `owl:allValuesFrom`: currently, the appearance of this pitfall in the ontology does not affect in any sense. This pitfall may affect if the system exploits more language primitives.

P15. Misusing "not some" and "some not": currently, the appearance of this pitfall in the ontology does not affect in any sense. This pitfall may affect if the system exploits more language primitives.

P16. Misusing Primitive and Defined Classes: currently, the appearance of this pitfall in the ontology does not affect in any sense. This pitfall may affect if the system exploits more language primitives.

P17. Specializing too Much a Hierarchy: in most analysed systems, this is not perceived as a pitfall. Many systems model instances directly into the ontology. However, in some cases, when the individual is not really an instance of a concept but it is connected to the concept by means of a relation, this pitfall may indicate an error in the instance creation.

P18. Specifying too Much the Domain or the Range: if relationships other than "is-a" are used, some relations may be missed due to this pitfall. Therefore, precision could be affected.

P19. Swapping Intersection and Union: if relationships other than "is-a" are used, some relations may be missed due to this pitfall. Therefore, precision could be affected.

P20. Misusing Ontology Annotations: systems that exploits annotation properties to operate (for instance, TextViz) may be affected by this pitfall.

P21. Using a Miscellaneous Class: if a concept is not used, it should not appear. This pitfall may affect systems if the miscellaneous concept can be actually instantiated, leading to a decrease in precision.

P22. Using Different Naming Criteria in the Ontology: this pitfall may affect systems that use concept names in the annotation process. Using concepts with names that do not usually occur in the text may compromise their correct annotation, causing a deterioration in both precision and recall.

P23. Using Incorrectly Ontology Elements: the appearance of this pitfall would affect depending on the modelling decisions (classes or properties). In ISCO, for instance, relations are modelled as concepts.

P24. Using Recursive Definition: definitions should not affect the IR process in any way.

P25. Defining a Relationship Inverse to itself: currently, the appearance of this pitfall does not affect any of the analysed systems. This pitfall would affect if the system exploits property features, such as inverse and symmetric.

P26. Defining Inverse Relationships for a Symmetric one: currently, the appearance of this pitfall does not affect any of the analysed systems. These pitfalls would affect if the system exploits property features, such as inverse and symmetric.

P27. Defining Wrong Equivalent Relationships: if a system uses relationships and OWL primitives, the appearance of this pitfall in the ontology would affect to the precision.

P28. Defining Wrong Symmetric Relationships: currently, the appearance of this pitfall does not affect any of the analysed systems. This pitfall would affect if the system exploits property features, such as inverse and symmetric.

P29. Defining Wrong Transitive Relationships: if a system exploits the transitive property in relationships, the pitfall may affect its precision.

P30. Missing equivalent classes: this pitfall may cause same concepts to have different parents. Therefore, if a system exploits hierarchical information, it may affect its precision and recall.

Table 1 provides an overview of how every pitfall may or not affect each of the analysed systems.

4 CONCLUSIONS

We carried out a survey of existing state-of-the-art ontology-based information retrieval systems with respect to the pitfalls listed in the OOPS! catalogue. Our analysis shows that indeed OOPS! may prove useful to the developers of ontology-based IR systems in order to verify the quality of the ontology they use in their systems and prevent errors. Our analysis highlights also the fact that most of current available systems do not use some advanced features (especially with respect to relationships) that are provided by the OWL language. It is difficult to say whether this issue derives from the fact that developers ignore the existence of these features, or whether it is consequence of the state of the art of the available Natural Language Processing tools.We hope that this work will be viewed as an incentive for people working on ontology-based IR systems to: make their systems available for comparative testings; get used to adopt existing standards; evaluate their ontologies with an existing tool like OOPS!, in order to benefit of having some degree of quality in such ontologies. As a further work, we plan to carry out an evaluation of the speculated effects on a new version of the TextViz system which takes into account relations in a more advanced way than TextViz. This new version of TextViz is being completed and should be available soon. In order to carry out such evaluation, we will have to produce a test environment with different ontology benchmarks that include different combinations of pitfalls. Thanks to the results of this study, we are also planning to sketch some advices to help developers of ontology-based information retrieval systems to avoid pitfalls that may prevent their systems from working properly or deteriorate their performance.

ACKNOWLEDGEMENTS

This work has been partially supported by (a) the Spanish projects *BabelData* (TIN2010-17550) and *BuscaMedia* (CENIT 2009-1026), and (b) the Post-Doctoral Exchange Programme of the French-Spanish Laboratory for Advanced Studies in Information, Representation and Processing (LIRP Associated European Laboratory (LEA)), and (c) the EFL Labex project.

REFERENCES

Abasolo, J. M., Gómez, M.: *MELISA*. An ontology-based agent for information retrieval in medicine. Proceedings of the First International Workshop on the Semantic Web (SemWeb2000), pp. 73-82. (2000).

Bannour, I., Zargayouna, H.: Une plateforme open-source de recherche d'information sémantique. CORIA 2012: 167-178. (2012).

Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, vol. 284 no.5, pp. 34-43 (2001).

Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli D.: Hybrid Search : Effectively Combining Keywords and Semantic Searches. ESWC'08 Proceedings of the *5th European semantic web conference on The semantic web: research and applications Page*s 554-568. (2008).

Burton-Jones, A., Storey, V. C., and Sugumaran, V., and Ahluwalia, P.: A Semiotic Metrics Suite for Assessing the Quality of Ontologies. Data and Knowledge Engineering, (55:1), pp. 84-102. (2005).

Castells, P., Fernández, M., Vallet, D.: An adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering,* 19(2): 261-272 (2007).

Ciorascu, C., Ciorascu, I., Stoffel, K.: knOWLer – Ontological Support for Information Retrieval Systems. *Proceedings of 26th Annual International ACM SIGIR Conference, Workshop on Semantic Web* (2003).

Corcho, O., Gómez-Pérez, A., González-Cabero, R., Suárez-Figueroa, M. C.: ODEval: a Tool for Evaluating RDF(S), DAML+OIL, and OWL Concept Taxonomies. In: 1st IFIP Conference on Artificial Intelligence *Applications and Innovations (AIAI* 2004), August 22-27, 2004, Toulouse, France. (2004).

Djedidi, R., Aufaure, M. A.: Onto-Evoal an Ontology Evolution Approach Guided by Pattern Modelling and Quality Evaluation. Proceedings of the Sixth *International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2010), February 15-19 2010, Sofia, Bulgaria (2010).*

Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann J.: Modelling Ontology Evaluation *and* Validation. *Proceedings of the 3rd European Semantic Web Conference (ESWC2006),* number 4011 in LNCS, Budva. (2006).

Gómez-Pérez, A.: Ontology Evaluation. Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. (2004).

Gulla, J. A., Borch, H. O., Ingvaldsen, J. E.: Ontology learning for search applications. In Proceedings of the 2007 OTM Confederated *International Conference on On The Move to Meaningful Internet Systems: CoopIS, DOA, ODBASE, GADA, and IS* - Volume Part I (OTM'07), Robert Meersman and Zahir Tari (Eds.), Vol. Part I. *Springer-Verlag, Berlin, Heidelberg*, 1050-1062. (2007).

Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, indexing and Retrieval. Journal of Web Semantics 2, 49–79. (2004).

Liu, C. H., Hung, S. C., Jain, J. L., Chen, J. Y.: Semi-automatic Annotation System for OWL-based Semantic Search. CISIS, pp.475-480, *International Conference on Complex, Intelligent and Software Intensive Systems.* (2009).

Müller, H., Kenny E., Sternberg, P.: Textpresso: An Ontology-based Information Retrieval and Extraction System for biological Literature. PLoS Biology*, Vol.* 2, Nr. 11, p. 1984-1998. (2004).

Pammer, V.: PhD Thesis: Automatic Support for Ontology Evaluation Review of Entailed Statements and Assertional Effects for OWL Ontologies. *Engineering Sciences. Graz University of Technology.* (2010).

Poveda, M., Suárez-Figueroa, M. C., Gómez-Pérez, A.: Validating Ontologies with OOPS!. EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings. *Lecture Notes in Computer Science Volume* 7603, pp 267-281. ISBN: 978-3-642-33875-5. (2012).

Poveda, M., Suárez-Figueroa, M. C., Gómez-Pérez, A.: A Double Classification of Common Pitfalls in Ontologies. OntoQual 2010 - *Workshop on Ontology Quality at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010).* Lisbon, Portugal. (2010).

Presutti, V., Gangemi, A., David S., Aguado, G., Suárez-Figueroa, M. C., Montiel-Ponsoda, E., Poveda, M.: NeOn D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. *NeOn project. (FP6-27595)*. (2008).

Pruski, C., Guelfi, N., Reynaud, C.: Adaptive Ontology-Based Web Information Retrieval: The TARGET Framework. *IJWP 3(3):* 41-58 (2011).

Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns. *In Proc. of EKAW 2004, pp: 63–81. Springer. (2004).*

Reymonet, A., Thomas, J., Aussenac-Gilles, N.: Ontologies et Recherche d'Information: Une application au diagnostic automobile. *Journées Francophones d'Ingénierie des Connaissances (IC 2010), Nîmes* (France) (2010).

Sabou, M., Fernandez, M.: Ontology (Network) Evaluation. Ontology Engineering in a Networked World. Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., Gangemi, A. *Editors. Pp. 193-212, Springer.* ISBN 978-3-642-24793-4. (2012).

Shah, U., Finin, T., Joshi, A., Cost, R. S., Mayfield, J.: Information Retrieval on the Semantic Web. *CIKM* '02, November 4-9, McLean, Virginia, USA. (2002).

Suárez-Figueroa, M. C.: PhD Thesis: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. Spain. *Universidad Politécnica de Madri*d. (2010).

Tomassen, S. L., Strasunskas, D.: An ontology-driven approach to web search: analysis of its sensitivity to ontology quality and search tasks. In Proceedings of the *11th International Conference on Information Integration and Web-based Applications & Services (iiWAS '09).* ACM, New York, NY, USA, 130-138. (2009).

Table 1: How pitfalls may affect each of the analysed systems[8]. Black: pitfall may have a negative effect on the system. Gray: pitfall could affect the system if it was designed to take into account a specific feature. White: pitfall has no impact in the system.

Sys	**J**	**B**	**C**	**D**	**H**	**I**	**E**	**F**	**K**	**A**	**G**	
P1		■		■	■		■		■	■	■	(6)
P2		■	■			■			■	■		
P3	(1)	■	■		■	■		(1)	■	■	■	(1)
P4	■	■	■	■	■	■	■	■	■	■	■	■
P5			(5)	(1)(5)	(5)		(5)		(5)	(5)	(5)	
P6	■		■			■		■	■	■	■	■
P7	■		■	■	■	■	■	■	■	■	■	■
P8 (2)									■	■		
P9	(1)(3)	(3)	(3)	(1)(3)	(3)	(3)	(3)	(1)(3)	(3)	(3)	(3)	(3)
P10												
P11	■	■	■	■	■		■			■	■	■
P12	(1)	■	■	(1)	■	■	■	(1)	■	■	■	(1)
P13									(4)	(4)		
P14			(4)		■				(4)	(4)		
P15			(4)		■				(4)	(4)		
P16												
P17											■	
P18		■	■	(1)	■					■	■	■
P19		■	■	(1)	■					■		
P20(2)									■	■		
P21	■	■	■	(1)	■		■	■			■	■
P22												
P23	(1)	■	■	(1)	■	■	■		■	■	■	(1)
P24			(4)		(4)	(4)			(4)	(4)	(4)	(4)
P25			■	(1)(5)	■				(5)	(5)		
P26			■	(1)(5)	■				(5)	(5)		
P27			■	(1)(5)	■		■		■	■		
P28			■	(1)(5)	■							
P29			▨	(1)	▨		▨		▨	▨		
P30	(1)	■	■	(1)	■	■	■	■	■	■	■	(1)

[8] Notes: (1) may affect if OWL is used; (2) only TextViz and Castells use labels; (3) may affect if ORSD is used; (4) may affect if system exploits some language primitives that are not currently exploited; (5) may affect if the system exploits property features; and (6) the paper did not provide enough insights to determine whether the pitfall may affect or not.