Short Paper: Non-Taxonomic Concept Addition to Ontologies

Artemis Parvizi, Chris Huyck, and Roman Belavkin

Middlesex University
The Borough
London NW4 4RL

Abstract. Concept addition, an ontology evolution's edit operation, includes adding taxonomic (hierarchical structure) and non-taxonomic (concept properties) relations. Generating concept properties requires information extraction from various sources, such as WordNet. Other than semantic similarities generated by WordNet, self-information generated from existing non-taxonomic relations has aided non-taxonomic relation addition to ontologies. Evaluation is based on using an ontology as gold standard and detaching and reattaching the nodes. Non-taxonomic relation generation without accessing an enormous amount of information has proven to be quite difficult; the results displayed in this work are an indication of this difficulty.

Keywords: Ontology Evolution, Ontology Learning, Non-Taxonomic Relations, Concept Addition

1 Introduction

Ontology is commonly defined as a formal, explicit specification of a shared conceptualisation [14], and often has been used for modelling concepts of the world. Due to the experts' limitations of producing a complete image of the world with flexible boundaries for a domain, change is inevitable. Change in ontologies has some common causes [29]: change in the domain, change in the shared conceptualisation, or change in the specification. Ontology update has been a subject of debate for many years, and many methods have been proposed to address it. Ontology evolution and ontology learning are among these proposed methods. *Ontology evolution* is "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts" [39], such as systems defined in [5, 30, 22, 40, 13, 4, 42, 19, 21]; *ontology learning* involves changing an ontology automatically or semi-automatically by consulting some structured data sources, such as databases; semi-structured data sources, such as WordNet, or Cyc; or some unstructured data sources, such as text documents and web pages [10]. A few examples of ontology learning systems can be found in [20, 9, 36, 27, 11, 41, 33].

Changing an ontology involves both changing the concepts and the relations. Ontology relations have been divided into two categories: taxonomic relations such as `subClassOf` and `disjointWith` in OWL [2], and non-taxonomic relations which covers most of the other OWL relations. On one hand, taxonomic relations provide a structure to ontologies and are crucial. On the other hand, non-taxonomic relations by presenting meaning add depth to the ontology. Regardless of using the term ontology evolution or ontology learning, commonly, ontology update involves changing both taxonomic and non-taxonomic relations.

A fundamental design operation for having a successful ontology evolution application includes concept addition [24, 15]. To address concept addition, two approaches (Approach I (see Section 4.1) and Approach II (see Section 4.2)) have been introduced in which ontology graphs (see Section 2) and semantic similarity (see Section 3) have been employed.

2 Ontology Graph

The definition of an ontology in this paper is a set C of concepts and a set of relations $R_1, \ldots, R_n$, $R_i \subset C \times C$. Since multiple relations with different labels are allowed to exist in ontologies, labelled graphs also known as multigraphs ($G = (V, E_1, \ldots, E_n)$) with the set of vertices $V \iff C$ and a set of edges $E_i \iff R_i$ are a logical choice of representing them. A graph with the stated characteristics is called an *ontology graph* and is able to cover all important structural OWL ontology features including individuals, classes, relations, object properties, datatype properties, and restrictions [23]. The notion of ontology graph in this work is an extended version represented in [26, 16, 34, 17, 3, 25]; vertices represent concepts, individuals, restrictions, and values, and edges, include taxonomic OWL relations, such as `subClassOf` and `disjointWith`, and non-taxonomic relations.

3 Semantic Similarity

A successful ontology change application must be able to detect what needs to be changed, gather sufficient information about the element that needs to be changed, and finally decide how to implement change. Extracting relevant and sufficient information is crucial. In this work, WordNet [38] and Wikipedia as general purpose semi-structured data sources are consulted; they both are capable of generating semantic similarity distances between concepts. Semantic similarity between two or more concepts refers to the level of closeness that their meanings possess, and it is very difficult to acquire. It is common practice to use ontologies for computing the distance between two concepts and normalising the final result. In RiTa WordNet [18], the minimum distance between any two senses for the two words in the WordNet tree is returned and the result is normalised; if there is a similarity a number is retuned, and 1 if no similarity is found.

This work has generated semantic similarities from Wikipedia as well. Although many have mentioned that Wikipedia is much richer and a far better source [35, 7, 32, 37], the result acquired from Wikipedia were not as promising as WordNet. Often semantic Wikipedia APIs only consult the infoboxes for generating semantic similarity; lack of word sense when extracting concepts is identified as another shortcoming [37].

4 Methodology

Ontology development is highly dependent on ontology experts, and domain experts. The perception of an expert about a correct or an incorrect relation may differ from another expert. This issue has contributed to the complexity of ontology development and update. Nonetheless, this work proposes that when adding a non-taxonomic relation, provided that the consistency of the ontology holds and the ontological statement is semantically correct, the new statement is as welcomed as any existing statement. For example when given the three concepts `Student`, `Library`, and `Group`, and the relation `memberOf`, an expert might generate `Student memberOf some Library`, `Student memberOf some Group`, or both. Absence of either of these two statements will not make the ontology incorrect but in certain circumstances it can be claimed that the ontology is less accurate. The same justification holds when a system is automatically generating non-taxonomic statements.

Commonly when generating non-taxonomic statements, a common approach is to provide a set of possible properties for each concept, rank them according to their frequencies, and finally according to some criteria select the highly probably one. However, this work does not intend to generate new properties for concept, but to assign an existing property to an input concept. Non-taxonomic relations can be classified into two general groups: object properties (intrinsic and extrinsic), and data-type properties [28]. The aim of this work is to generate intrinsic properties for a new input concept based on the existing intrinsic properties. The hypothesis is that siblings of a vertex in an ontology graph often have the same intrinsic properties assigned to different concepts.

In this work, the complete set of possible answers (Ans list) is generated, and the existing statements in the ontology (Cur list) are extracted. Ans list is a combination of an input concept I, the set of vertices $V = V_1, V_2, \ldots, V_n$, the set of edges $E = E_1, E_2, \ldots, E_n$, and the set of restrictions. Note that in this work only the two restrictions `some` and `only` are considered. Sample statements for the following approaches are as follows:

Existing Statement: $\mathtt{V_1}$ $\mathtt{E_1}$ `some` $\mathtt{V_2}$
Generated Statement: `I` $\mathtt{E_1}$ `some` $\mathtt{V_3}$

4.1 Approach I

The members of list Ans for an input concept I, m vertices, the two restrictions, and n edges will be $4 \times m \times n$ which comparative to list Cur are numerous. This approach consists of a number of filters to prune Ans list according to Cur list with the aid of various semantic similarities. To be able to apply semantic similarities, a random entropy or self-information approach has been selected.

Probability of the event of randomly connecting a to b by an R_i relation is defined by $P(e) = P((a, b) \in R_i)$. The prior probability therefore being $P(e) = \frac{1}{k}$, where k is the number of possible links $(a, b) \in R_i$. Given some semantic similarity distances (see Section 3) $s(a, b) \in [0, 1]$, the posterior probability of a connection assuming a dependency between e and $s(a, b)$ is:

$$P(e \mid s(a, b)) \neq P(e)$$

Since $s(a, b)$ is a similarity distance (taking values in $[0, 1]$ with 0 corresponding to the most similar), it can be assumed that the posterior probability of connection monotonically depends ($\propto$) on $1 - s(a, b)$:

$$P(e \mid s(a, b)) \propto 1 - s(a, b)$$

The monotonicity for two events $e_1 = (a, b)$ and $e_2 = (a, c)$ means the following:

$$\begin{aligned} s(a, b) \geq s(a, c) &\Longleftrightarrow 1 - s(a, b) \leq 1 - s(a, c) \\ &\Longrightarrow P(e_1 \mid s(a, b)) \leq P(e_2 \mid s(a, c)) \end{aligned}$$

The probability can be used to compute self-information as follows [6]:

$$\begin{aligned} h(a, b) &= -\log(P(e \mid s(a, b))) \\ &\approx -\log(1 - s(a, b))) \end{aligned} \tag{1}$$

The first filter is called hierarchy filtering; it is based on the patterns of the siblings of the input concept. A sibling is referred to a concept with a `disjointWith` relation. This work focuses on non-taxonomic patterns. For the input concept I, assuming that one of the statements in Ans is $IE_1 only V_1$, the patterns would be $IE_1 only$ and $E_1 only V_1$. This approach only makes use of the forward patterns which in this example is $E_1 only V_1$. Any member of the Ans list which does not contain the same pattern as one of the members of Cur list will be excluded from Ans. Also, if the input concept I and the first concept of a member of Cur list do not have the same parent, the statement will be excluded from Ans. Presuming both the pattern and the parent is matched, when the success rate of comparing the generated statement with all the members of Cur list is more than 50%, the statement will still remain in Ans, otherwise dropped. At this stage, only the statements with the patterns similar to the existing non-taxonomic statements remain.

From this point onwards, Equation 1 will aid the pruning process. For the second filter $Q_1 = h(I, E_1)$, $Q_2 = h(V_3, E_1)$, $Q_3 = h(V_2, E_1)$, and $Q_4 = h(V_1, E_1)$ are generated. The goal of this filter is to investigate $Q_1 + Q_2 \leq Q_3 + Q_4 \in [0.1]$; if in more than half the occurrences this function holds, then the generated statement will be accepted; otherwise rejected. The aim is for the self-information of the generated statement to be less than or equal to the self-information of the current statements.

For the third filter $Q_5 = h(I, V_1)$ and $Q_6 = h(V_2, V_3)$ are calculated. This filter will examine that in more than half the occurrences $Q_5 \leq Q_6 \in [0, 1]$ holds.

The forth filter will generate $Q_7 = h(I, V_2)$ and $Q_8 = h(V_1, V_3)$; the relation $Q_7 \leq Q_8 \in [0, 1]$ must hold in more than half the occurrences for the generated statement to be accepted.

The last filter will generate the self-information among all the members of the generated and the current statement:

$$Q_i = h(Statement\ from\ Ans\ list, Statement\ from\ Cur\ list)$$

The result generated by Q_i are sorted and the k most similar statements selected. Tables 1 and 2 display the results when $k = 2$.

4.2 Approach II

The members of the *Ans* list have to be pruned according to the members of *Cur* list. A comparison between all the members of both lists is made. Providing that a statement from one of lists has the same relation and restriction (for example E_K `Some` or E_K `Only`) as the other list, the occurring pattern and its frequency is recorded. The list containing the patterns *Pat* will be sorted ascending with regard to the frequencies, and the top half selected. Those statements in *Ans* which do not contain these patterns will be omitted from the final answer pool. The statement V_1 E_1 `some` V_2 contains two patterns; (1) E_1 some V_2 and (2) V_1 E_1 some.

The aim of this step is to prune *Ans* list according to the patterns in *Cur* list; there is a trade off to this filter, some semantically correct statements will not be validated due to the low or lack of frequencies.

Hierarchy filtering as discussed in approach (I) will filter the remaining members of the *Ans* list. When the siblings of the input concept contain a non-taxonomic relation which have occurred in more than 50% of the cases and this taxonomic relation is among the remaining members of the *Ans* list, this statement will be accepted, otherwise rejected from *Ans* list.

4.3 Transitive Reduction

Both of the introduced approaches have the potential of producing transitive relations, which from the consistency point of view have to be eliminated. Inheritance through the hierarchy has to be modelled in an ontology graph. Transitive reduction on directed graphs is the answer to this problem. Presuming there is the possibility of representing information in the directed graph G with fewer arcs than the current amount, then that is the solution [1]. Graph G' will be the transitive reduction of G if it satisfies the following conditions:

1. A direct path between v and u in G' exists, if a direct path between v and u in G exists.
2. There is no graph with fewer arcs than G' satisfying the above condition.

For approach (II), since all the remaining members of the *Ans* list are selected, transitive reduction is applied after the last step. However, approach (I) is more complicated due to selecting the top k generated relations. Transitive reduction can be applied before or after the top k selection, which this work has adopted the latter. Regardless of the approach, in situations in which a child inherits a property and the algorithm identifies this transitive property, the property is dropped.

4.4 Evaluation

This work has adopted an evaluation mechanism based on precision and recall measurements [8, 12]. The strategy is to select a well-structured ontology and after converting it into an ontology graph, detach the vertices one by one; the system will attempt to reattach the vertex to the graph optimally with the original relations and at the original location [31]. A comparison between the number of *removed* edges in the original ontology graph (O) and the modified graph (O') is made. Assuming concepts c_1 and c_2 and relation R_k are present in O', the hypothesis is to examine O and determine whether c_1 and c_2 are related by R_k or not. Accepting the hypothesis indicates that O contains an edge corresponding to $c_1 R_k c_2$; rejecting is when there is no such edge in O. The overall count of correct edges in O' relative to the numbers of all edges in O' or O respectively will produce precision and recall. F-measure is a more just measurement since precision and recall are distributed evenly.

$$P(E', E) = \frac{|E \cap E'|}{|E'|} \qquad R(E', E) = \frac{|E \cap E'|}{|E|}$$

$$F(E', E) = 2 \times \frac{P(E', E)R(E', E)}{P(E', E) + R(E', E)}$$

Other than studying the effect of a single concept addition, the effect of adding a sequence of concepts also has to be studied. The order in which concepts a and b are added to the system has an effect on the non-taxonomic relations generated; generally, the semantic richness of the ontology is affected by the existing concepts and relations. This work has studied the effect of adding two ($p = 2$) and ten ($p = 10$) concepts to the ontology graph. Due to all the input concepts being known, the average of all the possible orders have been displayed.

Approaches (I) clearly has better results than approaches (II) excluding one exception. The more frequent a pattern, the higher the probability of it being selected; also, the closer the pattern in the hierarchy, the greater the likelihood of it being the final answer. The major difference between the two approaches other than the F-measure is in the number of statements being selected as the final answer. In the approach (I), the number of statements selected has a limit; as a result, fewer unmatched statements are selected. However, approach (II) has no limit on the number of generated statement, but at the same time more unmatched statements are in the final answer pool. The reason this paper is using the expression unmatched instead of incorrect is that studying the final

Table 1. The experimental results of non-taxonomic learning for approach (I). The results are displayed in percentage.

	p=1			p=2			p=10		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Pizza	0	0	unknown	0	0	unknown	0	0	unknown
Travel	25.0	50.0	33.33	25.0	50.0	33.33	25.0	50.0	33.33
Amino Acid	31.11	11.20	16.47	31.11	11.20	16.47	33.33	12.00	17.64
Career	20.00	26.66	22.85	20.00	26.66	22.85	15.00	20.00	17.14
Human and Pets	16.66	17.39	17.02	16.66	17.39	17.02	14.28	16.66	15.38
Movie	23.52	11.32	15.28	20.00	9.43	12.82	17.5	7.27	10.29
OBOE	0	0	unknown	0	0	unknown	0	0	unknown
University	19.56	14.51	16.66	19.56	14.51	16.66	11.36	8.06	9.43
Vehicle	14.28	18.18	16.0	14.28	18.18	16.0	23.07	27.27	25.0

Table 2. The experimental results of non-taxonomic learning for approach (II). The results are displayed in percentage.

	p=1			p=2			p=10		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Pizza	18.76	71.71	29.69	18.76	71.71	29.69	15.84	52.25	24.31
Travel	46.15	42.85	44.44	46.15	42.85	44.44	56.25	32.14	40.90
Amino Acid	52.50	63.00	57. 27	52.50	63.00	57. 27	59.42	41.0	48.52
Career	50	50	50	50	50	50	37.5	25.0	30.00
Human and Pets	52.77	39.58	45.23	52.77	39.58	45.23	57.57	39.58	46.91
Movie	45.16	70.0	54.90	48.83	70	57.53	49.33	61.66	54.81
OBOE	0	0	unknown	0	0	unknown	0	0	unknown
University	20.40	28.57	23.80	20.40	28.57	23.80	25.00	28. 57	26.66
Vehicle	0	0	unknown	0	0	unknown	0	0	unknown

results has shown that more than 50% of the unmatched statements are actually semantically and logically accurate, although, not present in the original answer pool. Nevertheless, Table (1) and 2 only display the result of correctly matched edges to the original graph.

5 Conclusion and Future Work

One ontology evolution operation is concept addition, which implies adding a concept by taxonomic and non-taxonomic relations. Commonly for changing an ontology some external information is required. In this work WordNet as an external source for generating similarities between concepts and relations has been selected. The semantic similarities generated by WordNet, self-information produced from patterns within ontologies, and the hierarchical structure of ontologies are the basis of approaches introduced in this paper. The focus is on intrinsic properties; presuming that intrinsic properties already exist, the assumption is that an input concept is more likely to have the same intrinsic properties as its siblings. Evaluation is based on calculating the precision and recall of detaching a node from an ontology and attempting to reattach it. The results displayed in this paper are based on this evaluation technique. Due to the poor F-measures generated by the introduced approaches, an investigation into the cause of this poor performance revealed that more than 50% of the statements that were considered incorrect are actually semantically accurate. These results if generated by an ontology expert, could easily be regarded as correct.

The next step for this research is to generate more complex non-taxonomic relations, such as statements including conjunction and disjunction. Throughout the development of this work, the need for a ternary and a quaternary comparison has been visible. Such a comparison is essential for generating more meaningful ontology statements. Another future direction is to develop a source capable of ternary and quaternary comparison.

References

1. A V Aho, M R Garey, and J D Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.
2. H. Peter Alesso and Craig F. Smith. *Thinking on the Web: Berners-Lee, Godel and Turing*. Wiley-Interscience, New York, NY, USA, 2008.
3. Christoph Böhm, Philip Groth, and Ulf Leser. Graph-based ontology construction from heterogenous evidences. In *International Semantic Web Conference*, pages 81–96, 2009.
4. Silvana Castano, Irma Sofia Espinosa Peraldi, Alfio Ferrara, Vangelis Karkaletsis, Atila Kaya, Ralf Moeller, Stefano Montanelli, Georgios Petasis, and Michael Wessel. Multimedia interpretation for dynamic ontology evolution. *Journal of Logic and Computation*, 19(5):859–897, October 2009.
5. Fabio Ciravegna, Alexiei Dingli, David Guthrie, and Yorick Wilks. Integrating information to bootstrap information extraction from web sites. In *IJCAI'03 Workshop on Intelligent Information Integration*, pages 9–14, 2003.
6. Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.
7. Gerard de Melo and Gerhard Weikum. Menta: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1099–1108, New York, NY, USA, 2010. ACM.
8. K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of the 5th International Semantic Web Conference (ISWC)*, 2006.
9. Takahira Yamaguchi Dept and Takahira Yamaguchi. Acquiring conceptual relationships from domain-specific texts. In *Proceedings of the Second Workshop on Ontology Learning OL'2001*, pages 0–2, 2001.

10. Lucas Drumond and Rosario Girardi. A survey of ontology learning procedures. In Frederico Luiz Gonçalves de Freitas, Heiner Stuckenschmidt, Helena Sofia Pinto, Andreia Malucelli, and Óscar Corcho, editors, *WONTO*, volume 427 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
11. E. Drymonas, K. Zervanou, and E. Petrakis. Unsupervised ontology acquisition from plain texts: the ontogain system. In *Proceedings of the 15th International Conference on Applications of Natural Language to Information Systems (NLDB)*, Wales, UK, 2010.
12. Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 348–353, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
13. Ademir Roberto Freddo and Cesar Augusto Tacla. Integrating social web with semantic web - ontology learning and ontology evolution from folksonomies. In *KEOD*, pages 247–253, 2009.
14. T Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
15. Mark Hall. Ontology integration and evolution. *SE Data and Knowledge Engineering*, 10, May 2004.
16. J. Hartmann, P. Spyns, A. Giboin, D. Maynard, R. Cuel, M. C. Suarez-Figueroa, and Y. Sure. D1.2.3 methods for ontology evaluation. Technical report, Knowledge Web Consortium, 2005. Version 1.3.1, Available at: http://knowledgeweb.semanticweb.org/, Downloaded 2005-05-10.
17. Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. *A Practical Guide To Building OWL Ontologies Using Prot eg e 4 and CO-ODE Tools*. The University Of Manchester, 1.2 edition, March 2009.
18. Daniel C. Howe. Rita: creativity support for computational literature. In *Proceeding of the seventh ACM conference on Creativity and cognition*, pages 205–210, New York, NY, USA, 2009. ACM.
19. Pieter De Leenheer. *On Community-based Ontology Evolution*. PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium., 2009.
20. A. Maedche and S. Staab. Mining non-taxonomic conceptual relations from text. In *Proceedings of the 12th European Knowledge Acquisition Workshop (EKAW)*, Juan-les-Pins, France, 2000.
21. Yuxin Mao. A semantic-based genetic algorithm for sub-ontology evolution. *Information Technology Journal*, 9:609–620, 2010.
22. Diana Maynard, Diana Maynard, Wim Peters, and Marta" Sabou. Change management for metadata evolution. *International Workshop on Ontology Dynamics (IWOD) at European Semantic Web Conference*, 2007.
23. D.L. McGuinness and F. van Harmelen. *OWL web ontology language overview.* World Wide Web Consortium, Feb 2004.
24. Mohamed Mhiri and Faïez Gargouri. Using ontologies to resolve semantic conflicts in information systems design. In *Proceedings of The first International Conference on ICT and Accessibility*, Hammamet, Tunisia, April 2007. The first International Conference on ICT and Accessibility.
25. Victoria Nebot and Rafael Berlanga. Efficient retrieval of ontology fragments using an interval labeling scheme. *Information Sciences*, 179(24):4151 – 4173, 2009.
26. Chokri Ben Necib and Johann Christoph Freytag. Using ontologies for database query reformulation. In *ADBIS (Local Proceedings)*, 2004.
27. Mathias Niepert, Cameron Buckner, and Colin Allen. A dynamic ontology for a dynamic reference work. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '07, pages 288–297, New York, NY, USA, 2007. ACM.

28. Natalya Noy and Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, March 2001.
29. Natalya F. Noy and Mark A. Musen. Ontology versioning in an ontology management framework. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, 19(4):6–13, 2004.
30. Philip O'Brien and Syed Sibte Raza Abidi. Modeling intelligent ontology evolution using biological evolutionary processes. In *IEEE International Conference on Engineering of Intelligent Systems*, pages 1–6, 2006.
31. Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Anastasia Krithara, and Elias Zavitsanos. *Ontology Population and Enrichment: State of the Art*, volume 6050 of *Lecture Notes in Computer Science*, pages 134–166. Springer Berlin Heidelberg, 2011.
32. Simone Paolo Ponzetto and Roberto Navigli. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 2083–2088, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
33. Janardhana Punuru and Jianhua Chen. Learning non-taxonomical semantic relations from domain texts. *Journal of Intelligent Information Systems*, pages 1–17, 2011. 10.1007/s10844-011-0149-4.
34. Sang Keun Rhee, Jihye Lee, and Myon-Woong Park. Ontology-based semantic relevance measure. In *Proceedings of the The First International Workshop on Semantic Web and Web 2.0 in Architectural, Product and Engineering Design*, 2007.
35. Navigli Roberto, Velardi Paola, and Faralli Stefano. A graph-based algorithm for inducing lexical taxonomies from scratch. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Spain, July 2011. IJCAI/AAAI.
36. David Sánchez and Antonio Moreno. Discovering non-taxonomic relations from the web. In *7th International Conference on Intelligent Data Engineering and Automated Learning. LNCS 4224*, pages 629–636, 2006.
37. Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 801–808, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
38. Michael M. Stark and Richard F. Riesenfeld. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*. MIT Press, 1998.
39. Ljiljana Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, Germany, 2004.
40. Carlo Torniai, Jelena Jovanovic, Scott Bateman, Dragan Gasevic, and Marek Hatala. Leveraging folksonomies for ontology evolution in e-learning environments. In *ICSC*, pages 206–213, 2008.
41. C. Trabelsi, A. Ben Jrad, and S. Ben Yahia. Bridging folksonomies and domain ontologies: Getting out non-taxonomic relations. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 369 –379, dec. 2010.
42. P. Wongthongtham, N. Kasisopha, and S. Komchaliaw. Community-oriented software engineering ontology evolution. *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*, pages 1–4, November 2009.