International Journal of Scientific and Technical Research in Engineering (IJSTRE)
www.ijstre.com Volume 7 Issue 2 ǁ Mar-Apr 2022.
ISSN: 2581-9941

Alignment Finder: An Interactive Ontology Alignment Framework

Shehzad Khan[1], Saifur Rehman Khan[2], Asif Raza[3]
[1]*Department, of Computer Science & Engineering / Air University Multan, Pakistan*
shehzadmcs87@gmail.com
[2]*Department of Computer Science/ National University of Computer & Emerging Sciences, Pakistan*
mr.saifurrehman.khan@gmail.com
[3]*Department of Computer Science/ BahauddinZakariya University, Pakistan*
asifraza.raza14@gmail.com

Abstract: *Ontologies are being utilized as a core element in services of the semantic web and systems which require automatic and semantic interpretation of data. Ontologies work as a mediator between the system and heterogenous information retrieved from various data sources. This allows interoperability and semantic manipulation of data between the system and sources of data. But, ontologies across different data sources tend to be heterogenous as well. Ontology Alignment finds mappings between various ontologies and between various versions of an ontology to maintain consistency among similar data in different ontologies and systems using this data as well. Many frameworks have been proposed which can find mappings between various ontologies and their versions following a semi-automatic approach.The purpose of writing this paper is to propose a Framework which is Generic and supports Multiple Strategies for the alignment of ontologies following a semi-automatic approach. The proposed framework is interactive as it lets the user to choose an overall strategy for the alignment task by considering the complexity of input ontologies.*

Keywords: *Ontology, Ontology Matching, Ontology Mapping, Ontology Merging, Ontology Alignment*

I. INTRODUCTION

Semantic Interoperability is the most important requirement and a main characteristic of the Semantic Web.This ensures meaningful and automatic exchange and interpretation of data. Ontologies are distributed and heterogenous because of the decentralized nature of World Wide Web. Syntactical, structural and semanticalheterogeneitiesare needed to be removed for achieving semantic interoperability [1, 2, 3, 4].Heterogeneity in the syntax is because of the data represented in different formats. By bringing this heterogenous data retrieved from different sources into a common representation, syntactic heterogeneity can easily be handled. Heterogeneity in the structure is due to people having different perspective regarding a similar problem. Naturally, they come up with a solution different to one another and structure the data accordingly.Structural heterogeneity cannot be removed by just uniform representation of data. For resolving structural heterogeneity, data tagging and mediation is required. Using different abbreviations and terminologies for defining different taxonomies that model similar data causes semantic heterogeneity. Semantic heterogeneity can be resolved by performing Ontology Alignment [1, 2, 4, 5, 6, 7] and Ontology Matching [8, 9, 10,11, 12, 13] strategies. Thesestrategiesand the different types of similarity matchers that are used in them are discussed later in this paper. Moreover, an important significance of ontology alignment is to maintain consistency among similar data in different ontologies, different versions of similar ontologiesand systems using this data as well. Ontologies evolve when data is added or changed which can cause inconsistency in different sources of data and systems accessing this data.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 gives an overview of the Framework proposed in this paper. Section 4 illustrates different types of Similarity Matchers which are used in various combinations in the Alignment Task, elucidates Similarity Matrix and strategies for Aggregating the results obtained from individual matchers.Section 5 presents experiments and evaluation of the system. And, Section 6 concludes the paper.

II. Related Work

In this section, we are going to highlight some of the methods already proposed for Ontology Alignment and Matching.

[14] uses Machine Learning Techniques to determine correlations between ontological annotations. The process is carried out in two phases. In the Training Phase, ontologies are parsed and a Training Model is derived with the

support of various classifiers. In the Testing Phase, similarity matrix of the input ontologies is computed using the training model and Alignment is extracted.

RiMOM [15]automatically assesses which aspects of similarity should be compared for a given alignment task in its preprocessing phase. Using this information, it then calculates similarity by using Label based or Structure based similarity methods. Similarity Combination methods are used to aggregate measures of similarity.These measures are used to generate alignment result.

The Infrastructure of GOMMA [16] operates in three levels.Storage and Management of ontology versions, elements and mappings is carried out in the Repository level. Functional Components level has MATCH for finding mappings between attribute pairs, DIFF for finding changed regions between various ontology versions and EVOLUTION for providing evolution analysis by using statistics and change history. Tools level has Ontology Matcher which is used by MATCH, COntoDiff and OnEX which are used by DIFF, Region Analyzer and Stability Analyzer which are used by EVOLUTION.

MAFRA [17] generates a Semantic Bridge Ontology for presenting the computed alignment. The Infrastructure works in two dimensions.Components in Horizontal DimensionHandle Preprocessing Ontologies, estimating Similarity, creating and executing Semantic Bridges and refining results. Semantic Bridge is a structure which completely describes correspondence among one entity pair. Components in Vertical Dimension are GUI, Background Knowledge, Consensus Building and Evolution.

III. Alignment Finder Overview

This section outlines the modules of the framework proposed and elucidates the working of the system and its components. Figure 1 gives an overview of the infrastructure.

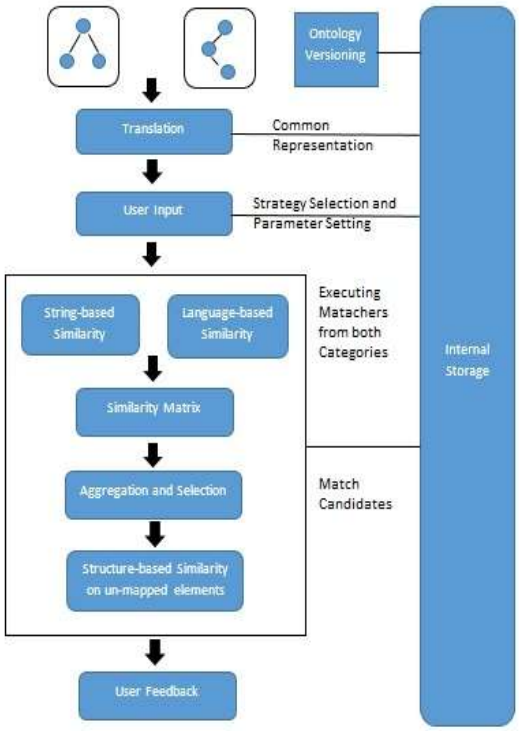


Figure 1 – Alignment Finder Infrastructure

3.1 Ontology Import and Translation

Source and Target Ontologies are loaded into the system and translated into a uniform representation by this module. This makes it easier to compare the two ontologies as syntactic heterogeneity is resolved is this step. Jena Ontology API [18] was used for this purpose.

Figure 2 – Translation Steps

Element names of input ontologies are preprocessed for this purpose. Tokenizing, eliminating stop words and eliminating special characters are the preprocessing techniques used.

3.2 User Input

This module involves user input in two ways. One is determining overall strategy for the alignment task. The user tells the system how entities of input ontologies should be semantically compared by the system by specifying the similarity matchers to be used. User also postulates a strategy for aggregating results generated by individual matchers and for the selection of candidates for mapping from the aggregated results. The other user involvement is parameter setting like threshold etc. for the matching process.

3.3 String-based Similarity

String-based matchers, picked by user during strategy selection, are executed on entities from input ontologies for determining their similarity. These matchers focus on the character sequence of entity names only. Measure of similarity will depend on identicality of the labels. This module does not require language support for matching as string-based matchers have no concern with the meanings of labels. A thorough discussion on such matchers which the system supports is presented in section 4.1.1.

3.4 Language-based Similarity

Language-based matchers, picked by user during strategy selection, are executed on entities from input ontologies for determining their similarity. In contrast with string-based similarity, this module requires language support because these matchers focus on meanings of entity names as well. A lexical database or any background knowledge of a language can be utilized for this purpose. A thorough discussion on such matchers which the system supports is presented in section 4.1.2.

3.5 Similarity Matrix

A matrix containing similarity measures between pairs of entities is obtained after executing string-based and language-based matchers. This matrix is called Similarity Matrix or Similarity Cubes. A floating-point number that ranges from 0 to 1 is called a similarity measure where 1 represents strong similarity and 0 represents strong dissimilarity. Section 4 gives a thorough discussion on similarity matrix and how it is used.

3.6 Aggregation and Selection

Similarity values calculated by each matcher are placed in the similarity matrix individually against each matcher for every pair of entities of the input ontologies. The results per every matcher are then combined using a user selected combination strategy for getting a final similarity score for each pair of entities. One aggregation strategy is selected by the user from Max, Weighted, Average or Min. Candidates for mappings are identified by using this combined similarity. One selection strategy is selected by the user from MaxN, MaxDelta or Threshold for determining mapping candidates. Section 4.2 and 4.3 present a detailed discussion of aggregation and selection.

3.7 Structure-based Similarity

Structure-based similarity is executed on the entities of input ontologies that were not recognized as correspondences by the previous phases of the system. Similarity matchers of this category focus on matching ontology entities based on their structure. The results of this match depend upon how classes and properties of

various ontologies correlate with each other with regards to interrelations of elements being matched and identicality of structures of entities. A thorough discussion on suchmatchers which the system supports is presented in section 4.1.3.

3.8 User Feedback

Results of the alignment task are presented to user by the system for taking user's feedback. In this phase, user can keep or discard correspondences determined automatically and can manually add other correspondences as well.

3.9 Internal Storage

Various stages of the system's infrastructure utilize internal storage right through the alignment task. Figure 3 depicts what data is accessed and stored in the internal storage.

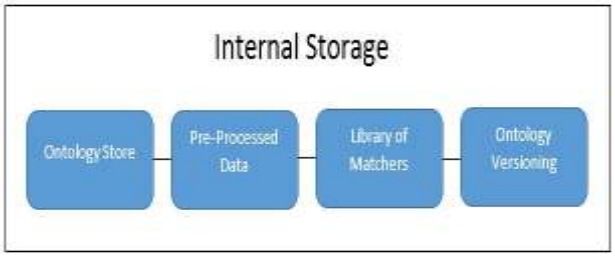


Figure 3 – Internal Storage

Ontology Store contains input ontologies loaded into the system which would be compared during the alignment task.

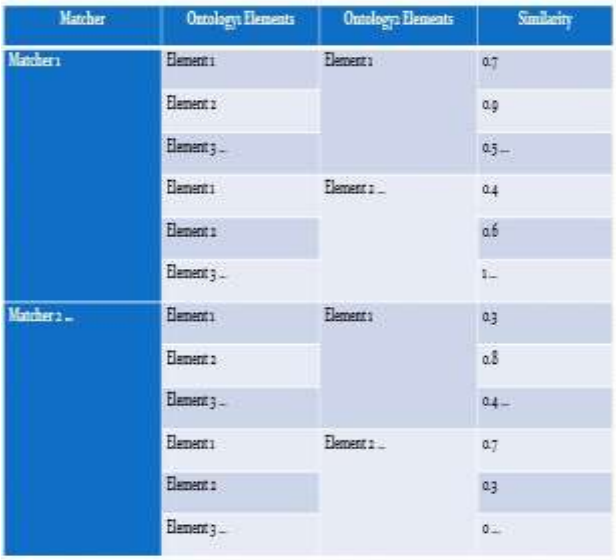
Pre-Processed Data contains entity names from target and source ontologies during the Translation module of the framework while passing them through the pre-processing stages.

Library of Matcherscontainscurrently supported similarity matchers' information. It is displayed to the user for determining match strategy in the User input phase. The information is stored in a database which contains all the similarity matchers' names classified in Structure-based, Language-based and String-based matchers categories.

Ontology Versioningcontains and maintains versions of ontology annotations. Versioning module is implemented by any system when it stores and manages ontology versions for handling changes in ontologies [19, 20, 21].

IV. Matching Process

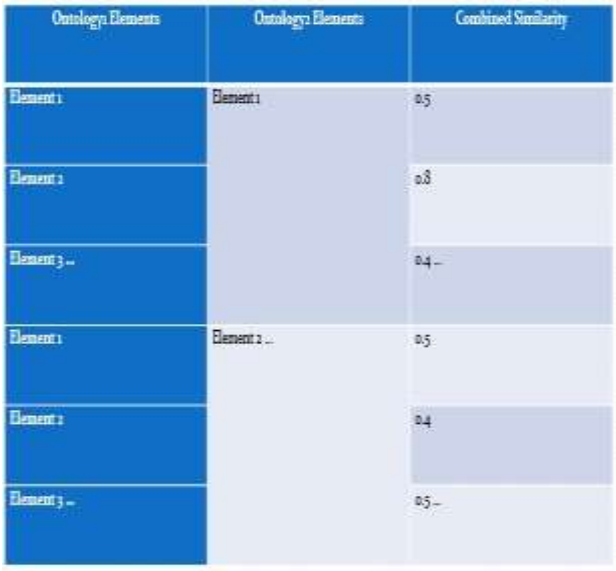
This operation identifies mappings between input ontologies by matching their element pairs [8, 9, 10, 11, 12, 13]. For this, the system executes different similarity matchers selected by the user from the matcher library. Classes from the target ontology are matched with the classes from the source ontology and properties from the target ontology are matched with the properties from the source ontology. Every match returns a value that ranges from 0 to 1 which represents the measure of similarity between the pair. If we have m matchers, n ontology1 elements and p ontology2 elements then there are m * n * p rows in the similarity matrix.



Matcher	Ontology1 Elements	Ontology2 Elements	Similarity
Matcher1	Element 1	Element 1	0.7
	Element 2		0.9
	Element 3 ..		0.5 ..
	Element 1	Element 2 ..	0.4
	Element 2		0.6
	Element 3 ..		1 ..
Matcher 2 ..	Element 1	Element 1	0.3
	Element 2		0.8
	Element 3 ..		0.4 ..
	Element 1	Element 2 ..	0.7
	Element 2		0.3
	Element 3 ..		0 ..

Figure 4 – Similarity Matrix

These per-matcher results are then aggregated for obtaining a combined similarity for every element pair.



Ontology1 Elements	Ontology2 Elements	Combined Similarity
Element 1	Element 1	0.5
Element 2		0.8
Element 3 ..		0.4 ..
Element 1	Element 2 ..	0.5
Element 2		0.4
Element 3 ..		0.5 ..

Figure 5 – Combined Similarity (by taking average)

Afterwards, a selection strategy picked by the user is used to select match candidates which formulate the mappings.

4.1 Similarity Matchers

Following are the Similarity Matchers supported by the system for matching ontology entities.

4.1.1 String-based Matchers

These matchers compare the labels of ontology entities (classes and properties) using a string-based match [1, 2, 22, 23]. SecondString API [24] was used for this purpose.

Edit Distance –This matcher computes the similarity based on the minimum number of edits necessary to transform first string to the second string. It is also calledLevenshtein Distance. The strings compute and computers have anedit distance of 2 between them.

Similarity = 1 – (edit-distance / length of the larger string)

Monge-Elkan Distance – This matcher computes similarity of the entities by the evaluation of each substring against anutmost similar substring.
UnsmoothedJS –is variation of Jensen-Shannon distance which computes similarity of language models in unsmoothed and unigram form.
Smith-Waterman Distance –This matcher determines similar regions in given sequences and computes local sequence alignments by matchingsegments of every possible length.
Jaccard Measure –This matcher estimates similarity by computing inner products and Euclidean norms of entity names represented as vectors.
Jaro Measure –This matcher uses number of transpositions between two strings for computing similarity between them.

4.1.2 Language-based Matchers
A Lexical Database or any Background Knowledge of a language are provided to Language-based matchers for calculating semantic similarity [12, 25].

Soundex –Soundex estimates phonetic similarity between strings by similar soundex codes. For example, it will first convert the input strings computing and computes into soundex codes and both will be the same C513 which depicts high similarity.

WordNet –A very popular lexical database of English is WordNet [25]. Words are grouped semantically in the form of synsets. Synsets are sets of synonyms and these are used to determine whether two words have the same sense or not, that is, if they belong to the same synset or the same set of synsets. This information is used to estimate similarity of the elements.

4.1.3 Structure-based Matchers
These matchers focus on matching ontology entities based on their structures and the results depend upon the identicality of the structures [26].

NamePath –A large string is built by this matcher by traversing the hierarchy of the element up to the root and concatenating names coming in this path. For estimating similarity between two elements, both such strings are tokenized and then string matching is performed.
Leaves –This structural matcher calculates similarity between elements based on the similarity found betweentheir leaf elements.

4.2 Combined Similarity
Following aggregation strategies[10, 13] are supported by the system for combining individual similarity matcher results.

Max – Final similarity score for any specific entity pair is determined by taking a maximum similarity score of that pair calculated by any matcher.

Weighted – Every matcher is assigned a weight based on its expected importance. These weights are used in combining similarity scores computed by each matcher.

Average – Every matcher is given equal importance and combined similarity score is computed by taking average of similarity scores determined by all matchers.

Min – Lowest similarity score computed by any matcher is taken.

4.3 Selection of Match Candidates
For determining match candidates of any entity in Ontology2, similarity scores of Ontology1 entities are ranked in descending order by using the combined similarity matrix. Following selection strategies [13] are supported by the system for selecting candidates of mappings.
MaxN – N Ontology1 entities having highest similarity scores are taken as match candidates. For one-to-one correspondence, the strategy will be Max1.

MaxDelta – Match candidates are selected from Ontology1 by picking all entities having mutual difference less than or equal to the specified tolerance value.
Threshold – Match candidates are selected by picking all Ontology1 entities that have a similarity score greater than the specified threshold.

V. Evaluation and Results

Evaluation and Results are illustrated in this section.

5.1 Evaluation Criteria

Following evaluation metrics are suggested by [27, 28] for evaluating Alignment Results.
Precision is calculated by dividing the number of correct alignments found automatically by total amount of alignments found automatically.

$$P = |m_a \cap m_m| / |m_a|$$

Recall is calculated by dividing the number of correct alignments found automatically by total amount of correct alignments.

$$R = |m_m \cap m_a| / |m_m|$$

F1-measure is a balanced value between P and R.

$$F1\text{-measure} = (2 * P * R) / (P + R)$$

5.2 Results

Benchmark Data Set, Anatomy Data Set and Conference Data Set from Ontology Alignment Evaluation Initiative 2016 were used as test sets for experimentations on the system. These Test Sets along with the Match Strategies depicted in Figure 6evaluate the system's performance in contrasting situations.

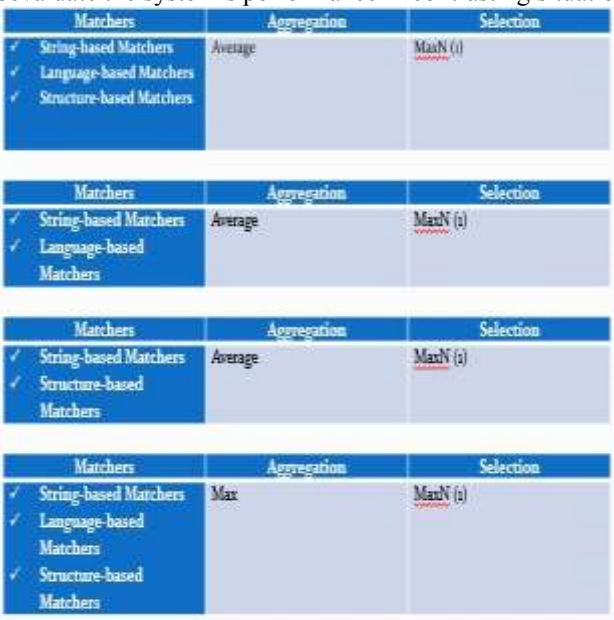


Figure 6 – Match Strategies A, B, C and D

Figures 7,8,9 and 10 graphically illustrate the results of this experimental evaluation of the system. It is notable how the results vary by changing strategies for the test sets.

Figure 7 – Precision, Recall and F1 for Strategy A

Figure 8 – Precision, Recall and F1 for Strategy B

Figure 9 – Precision, Recall and F1 for Strategy C

Figure 10 – Precision, Recall and F1 for Strategy D

VI. Conclusion

Interoperability is an issue in the World Wide Web because of its decentralized nature and distributed ontologies. Ontology Alignment hasimmense significance in this regard.This paper presented a multi-strategy and generic framework for aligning ontologies in a dynamic and interactive environment. For any ontology alignment task, ontology matching is a core process. Various algorithms supported by the system for similarity matching,

categorized based on their type of matching, are discussed. Calculating and storing match results in similarity matrix, strategies for aggregating similarity measures and strategies for candidate selection are also discussed in this paper along with evaluation and results of the system. This work can be extended by adding Natural Language Processing and Machine Learning techniques.

References

[1]. Euzenat, Jérôme, and Pavel Shvaiko. *Ontology matching*. Vol. 18. Heidelberg: Springer, 2007.

[2]. Euzenat, J., T. Le Bach, J. Barrasa, P. Bouquet, J. De Bo, and R. Dieng. *D2. 2.3: State of the art on ontology alignment–Knowledge Web project, realizing the semantic web*. IST-2004-507482 Programme of the Commission of the European Communities, 2004.

[3]. Visser, Pepijn RS, Dean M. Jones, Trevor JM Bench-Capon, and M. J. R. Shave. "An analysis of ontology mismatches; heterogeneity versus interoperability." In *AAAI 1997 Spring Symposium on Ontological Engineering, Stanford CA., USA*, pp. 164-72. 1997.

[4]. Predoiu, Livia, Cristina Feier, Francois Scharffe, Jos de Bruijn, Francisco Martín-Recuerda, DimitarManov, and Marc Ehrig. "D4. 2.2 State-of-the-art survey on Ontology Merging and Aligning V2." *EU-IST Integrated Project IST-2003-506826 SEKT* (2005): 79.

[5]. Do, Hong-Hai, and Erhard Rahm. "Matching large schemas: Approaches and evaluation." *Information Systems* 32, no. 6 (2007): 857-885.

[6]. Kalfoglou, Yannis, and Marco Schorlemmer. "Ontology mapping: the state of the art." *The knowledge engineering review* 18, no. 1 (2003): 1-31.

[7]. Choi, Namyoun, Il-Yeol Song, and Hyoil Han. "A survey on ontology mapping." *ACM Sigmod Record* 35, no. 3 (2006): 34-41.

[8]. Rahm, Erhard, and Philip A. Bernstein. "A survey of approaches to automatic schema matching." *the VLDB Journal* 10, no. 4 (2001): 334-350.

[9]. Maedche, Alexander, and Steffen Staab. "Measuring similarity between ontologies." In *International Conference on Knowledge Engineering and Knowledge Management*, pp. 251-263. Springer, Berlin, Heidelberg, 2002.

[10]. Doan, AnHai, Jayant Madhavan, Pedro Domingos, and Alon Halevy. "Learning to map between ontologies on the semantic web." In *Proceedings of the 11th international conference on World Wide Web*, pp. 662-673. ACM, 2002.

[11]. Bergamaschi, Sonia, SilvanaCastano, Maurizio Vincini, and Domenico Beneventano. "Semantic integration of heterogeneous information sources." *Data & Knowledge Engineering* 36, no. 3 (2001): 215-249.

[12]. Zhang, Rubo, Ying Wang, and Jing Wang. "Research on ontology matching approach in semantic web." In *Internet Computing in Science and Engineering, 2008. ICICSE'08. International Conference on*, pp. 254-257. IEEE, 2008.

[13]. Do, Hong-Hai, and Erhard Rahm. "COMA—a system for flexible combination of schema matching approaches." In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pp. 610-621. 2002.

[14]. Nezhadi, AzadehHaratian, BitaShadgar, and AlirezaOsareh. "Ontology alignment using machine learning techniques." *International Journal of Computer Science & Information Technology* 3, no. 2 (2011): 139.

[15]. Li, Juanzi, Jie Tang, Yi Li, and Qiong Luo. "RiMOM: A dynamic multistrategy ontology alignment framework." *IEEE Transactions on Knowledge and Data Engineering* 21, no. 8 (2009): 1218-1232.

[16]. Kirsten, Toralf, Anika Gross, Michael Hartung, and Erhard Rahm. "GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution." *Journal of biomedical semantics* 2, no. 1 (2011): 6.

[17]. Maedche, Alexander, Boris Motik, Nuno Silva, and Raphael Volz. "Mafra—a mapping framework for distributed ontologies." In *International Conference on Knowledge Engineering and Knowledge Management*, pp. 235-250. Springer, Berlin, Heidelberg, 2002.

[18]. Apache Jena", https://jena.apache.org/.

[19]. Noy, Natalya F., and Mark A. Musen. "Ontology versioning in an ontology management framework." *IEEE Intelligent Systems* 19, no. 4 (2004): 6-13.

[20]. Stojanovic, Ljiljana. "Methods and tools for ontology evolution." (2004).

[21]. Plessers, Peter, Olga De Troyer, and Sven Casteleyn. "Understanding ontology evolution: A change detection approach." *Web Semantics: Science, Services and Agents on the World Wide Web* 5, no. 1 (2007): 39-49.

[22]. Stoilos, Giorgos, GiorgosStamou, and StefanosKollias. "A string metric for ontology alignment." In *International Semantic Web Conference*, pp. 624-637. Springer, Berlin, Heidelberg, 2005.
[23]. Cohen, William, Pradeep Ravikumar, and Stephen Fienberg. "A comparison of string metrics for matching names and records." In *Kdd workshop on data cleaning and object consolidation*, vol. 3, pp. 73-78. 2003.
[24]. "SecondString Project Page", http://secondstring.sourceforge-.net.
[25]. Fellbaum, Christiane. *WordNet*. John Wiley & Sons, Inc., 1998.
[26]. Euzenat, Jèrôme. "Alignment API and server." *INRIA &LIG*(2008): 32.
[27]. Do, Hong-Hai, and Erhard Rahm. "Matching large schemas: Approaches and evaluation." *Information Systems* 32, no. 6 (2007): 857-885.
[28]. "Ontology Alignment Evaluation Initiative", http://oaei.ontologymatching.org. 2009.