Ontology Matching Algorithms for Data Model Alignment in Big Data

Ruth Achiaa Frimpong(✉)

University of South Australia, Adelaide, SA, Australia
ruth.frimpong@mymail.unisa.edu.au

Abstract. Big Data commonly refers to large data with different formats and sources. The problem of managing heterogeneity among varied information resources is increasing. For instance, how to handle variations in meaning or ambiguity in entity representation still remains a challenge. Ontologies can be used to overcome this heterogeneity. However, information cannot be processed across ontologies unless the correspondences among the elements are known. Ontology matching algorithms (systems) are thus needed to find the correspondences (alignments). Many ontology matching algorithms have been proposed in recent literature, but most of them do not consider data instances. The few that do consider data instances still face the big challenge of ensuring high accuracy when dealing with Big Data. This is because existing ontology matching algorithms only consider the problem of handling voluminous data, but do not incorporate techniques to deal with the problem of managing heterogeneity among varied information (i.e., different data formats and data sources). This research aims to develop robust and comprehensive ontology matching algorithms that can find high-quality correspondences between different ontologies while addressing the variety problem associated with Big Data.

Keywords: Big Data · Ontology matching · Data heterogeneity · Alignment

1 Introduction

Big Data is the "new oil", the substance that is expected to drive the information economy of tomorrow. It is commonly considered to have three main dimensions: Volume, Velocity and Variety. Volume refers to the problem of dealing with large data sets; Velocity refers to the problem of dealing with real-time streaming data where it may not be possible to store all data for later processing and; Variety refers to the need to deal with many different data sources and data formats [9]. Big Data applications and projects are everywhere and companies prepare for the future where they cannot survive without the information gleaned from a variety of data sources. The problem of managing heterogeneity among such varied information resources is increasing. For instance, most database research

E. Blomqvist et al. (Eds.): ESWC 2017, Part II, LNCS 10250, pp. 195–204, 2017.
DOI: 10.1007/978-3-319-58451-5_14

and self-assessment reports recognise that the question of semantic heterogeneity, that is, how to handle variations in meaning or ambiguity in entity interpretation, remains open [1]. Ontologies are often used as a model for knowledge representation and can help in overcoming these heterogeneities [16].

Ontologies play a prominent role for many applications, such as database integration, peer-to-peer systems, e-commerce, semantic web services and social networks [7]. They are a practical means to conceptualise what is expressed in a machine readable format. An ontology usually provides a vocabulary characterising a domain of interest and a specification of the meaning of terms in that vocabulary [5]. In open or evolving systems, such as dynamic Big Data analysis environments, different parties adopt different ontologies that typically need to be merged for analysis to be performed. Information cannot be processed across ontologies if the correspondences or semantic mappings between the elements are unknown. Manually finding such correspondences is time-consuming and prone to error. The success of the Semantic Web and other applications is dependent on the development of algorithms to assist in this process, also called *ontology matching*, which is the main focus of the research.

We seek to develop a robust and comprehensive ontology matching system that can find high-quality correspondences between different ontologies and also resolve the variety problem associated with Big Data.

The remainder of this paper is organised as follows. Section 2 reviews the related literature. The research problems and expected contributions are presented in Sect. 3. Section 4 outlines the research methodology and approach. Sections 5 and 6 discuss the preliminary test and evaluation plan respectively. The paper is finally concluded in Sect. 7.

2 State of the Art

This section discusses the recent developments in ontology matching (OM).

Given two or more ontologies, the OM problem is to find correspondences between them. OM has been extensively studied as an essential technology to achieve interoperability over ontologies [10]. Existing work on OM is mostly focused at the schema level, that is, finding correspondences between two schemas such as database schemas and ontologies. However, in recent literature, the ability to compare different ontologies with the objective of identifying similar instances which refer to the same real-world entity is drawing more research interest. To execute the matching process, OM systems (algorithms or tools) are used. Some of the challenges the OM systems face can be found in [17]. These matching systems are basically developed by selecting a matching strategy and combining two or more of the multitude of matching techniques (see [Chaps. 5–7 of [5]] for details). However, quality mappings cannot be obtained if these techniques are not selected and combined appropriately.

Ontology matching systems such as COMA [3] and Lily [18] find correspondences between ontologies at the schema level and do not consider instance data, hence, they are unable to identify important mappings which may be needed for future analysis. The few systems, such as Falcon-AO [8] (see survey in [14]), that do consider data instances during matching still cannot deal with voluminous and high-variety datasets properly. Existing systems such as GLUE [4] use iterative matching (IM) techniques to speed up the ontology matching process. IM finds the instance correspondences in multiple loops; only a fraction of instances are matched in each iteration, which are then used as sources for matching the remaining instances in the next iterations. Although these techniques are useful when matching large datasets, IM is likely to propagate errors of mismatched instances. To avoid this problem, other systems such as PRIOR+ [12] use the parallel workflow strategy. In this strategy, several matchers are executed independently on the ontologies. The results produced by the individual matchers are combined by some aggregation and extraction methods to obtain the final alignments. Several aggregation and extraction methods have been proposed [5], including: Max/Min method, which returns the maximum or minimum similarity value of individual matchers; Weighted method, which computes a weighted sum of similarity values of individual matchers; and Sïgmoid method, which combines the results of the individual matchers using a sïgmoid function. The weighting methods require that a threshold be set manually and hence, are unable to adapt to different matching tasks. For example, when the selected matcher changes or their number increases. Employing weighting methods as the only strategy to aggregate individual matchers and extract alignments has been identified in the literature [17] to be ineffective and may result in ambiguous, inconsistent and inaccurate (low-quality) mappings: especially for matching systems that adopt both syntactic and semantic techniques. To solve the above challenges, we propose a novel ontology matching system that uses a dynamic weighted method in addition to user knowledge (reasoning) to find high-quality correspondences in Big Data. The main idea behind the system is to maximize the utilization of available data instances of ontologies.

3 Problem Statement and Contributions

3.1 Problem Statement

The following examples illustrates variety problems associated with Big Data and some other ontology matching problems. The first example shows a problem that occurs when mappings are identified based solely on information in the ontologies, neglecting external background knowledge. The second scenario presents the variety problem associated with Big Data (i.e., different representation of the same attribute). The last example shows the importance of matching at both the schema and instance levels.

The diagrams in Figs. 1 and 2 represent ontologies for two companies A and B. These ontologies consist of concepts, instances, attributes and relations. Green rectangles are concepts, yellow diamonds represent relations between different concepts, attributes are the pale blue hexagons and instances are displayed as pink rounded rectangles.

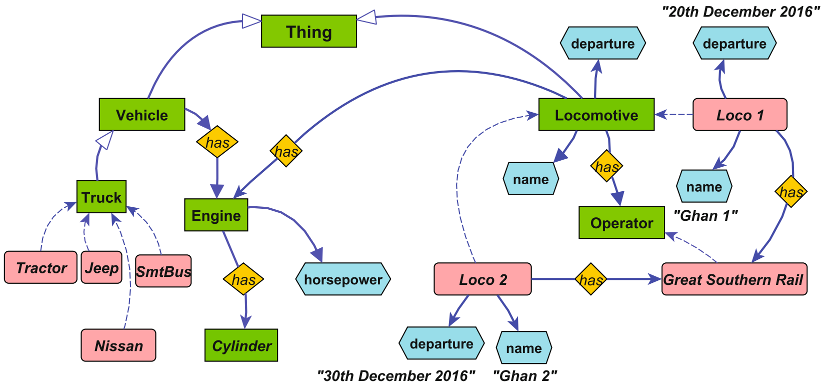

Fig. 1. Ontology A

Assume we would like to find correspondences between Ontology A (Fig. 1) and Ontology B (Fig. 2). Schema level matching would establish a match between Locomotive in ontology A and Train in Ontology B. However, in the real world, a locomotive is an engine of a train hence, matching Locomotive to Train is a false match. In such cases the use of external resources such as WordNet or reference ontologies can help reduce the mismatch.

Secondly, the attribute values associated with the *departure* attributes in ontologies A and B use different representations (i.e., full month name vs. short number form). Using syntactic (e.g. string-based) techniques, the attribute values would not match. However, these values represent dates in the real world hence techniques such as logical and numerical methods need to be combined appropriately with other syntactic techniques to resolve such varieties in data.

Lastly, using semantic and syntactic techniques (such as logical deduction, terminological structure, name-based, external resources), the Truck (Fig. 1) and Lorry (Fig. 2) will result in a match. This is because in the real world, Lorry and Truck are synonyms; Australia and United States of America call a motor vehicle designed to transport cargo as Truck while United Kingdom and Ghana refer to the same entity as Lorry. However, in this case, matching Truck to Lorry would be a mismatch. This is because the instances of Truck are similar to the instances of the Lorry, Autobus and Car hence, Truck would match Lorry, Autobus and Car in this scenario.

Additional problems that current matching systems cannot handle include:

- the inability to incorporate missing type information of instances. This is important since not all ontologies will be at the same level of granularity, which makes it difficult to find appropriate matches.
- the inability to effectively resolve mismatches, inaccuracies and inconsistencies between heterogeneous data.

To solve the identified problems, the following research questions will be addressed:

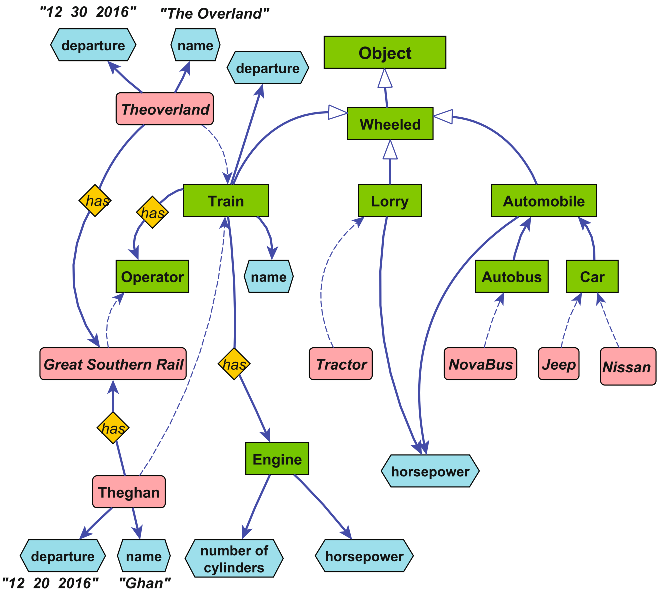

Fig. 2. Ontology B

1. **Big data has varieties and masses of data, how can we use it in ontology matching?**
2. **How can high-quality mappings be extracted without introducing ontological mismatches?**
3. **How can alignment inconsistency, ambiguity and inaccuracy in ontology matching be addressed?**

3.2 Contributions

The expected contributions of the research include:

- a novel instance-based technique that uses mining association rules to compute similarities between instances of different ontologies.
- the development of an effective aggregation and extraction technique which will incorporate user knowledge and a dynamic weighted method to combine the alignments resulting from the individual matchers.
- an effective filtering (alignment improvement) technique to identify the inconsistent and inaccurate mappings when matching Big Data.

4 Research Methodology and Approach

The research is divided into three stages. The first stage will investigate and develop algorithms that will incorporate data instances, identify missing type information, and partition the ontologies. In the second stage, a technique to aggregate and extract alignments will be developed. An alignment improvement technique will be developed in the third stage. Overall, the problems and questions identified in Sect. 3 will be addressed in these stages.

4.1 Research Stage 1

To begin with, we will adopt and extend the hierarchical clustering method of Typifier [11], which makes use of data instances to recover implicit subtypes. Python NLTK[1] will be used to reduce each form of a term in the data to some standardised form that can be easily recognised. Python NLTK would perform activities such as normalisation of date and number formats, term extraction, tokenisation, lemmatisation and stop word elimination on concept or label names to reduce their dissimilarities as well as generate pseudo schema attributes used by the Typifier algorithm. The extended type info. identifier will handle arbitrary formats (such as string vs. actual date format using data instances) to help address the variety problem in big data by inferring fine grain type information from data instances in order to bring the schema to the same level of granularity. The type info. identifier will then be applied on real world data such as OAEI[2], DBpedia[3] and Freebase[4] to identify any additional information in order to avoid class mismatch. Furthermore, we will empirically select an effective clustering algorithm that clusters instances in addition to schemas and adopt it as our ontology partitioner by evaluating their validity and runtime performance. The preliminary results of this evaluation is shown in Sect. 5.

In addition, the parallel workflow strategy will be adopted to design a matcher. The matcher will mainly incorporate existing instance-based, name-based, language-based, graph-based, taxonomy-based and model-based techniques (see [5]). The matcher will consists of the following:

Structural matcher compares the structure of entities in the data
Label matcher compares the similarity between strings (text)
Instance matcher computes the similarity between data instances
Semantic matcher uses external resource and logic to identify schemas and instances that are semantically related

The matcher architecture is illustrated in Fig. 3. This matcher is similar to that of PRIOR+ with the difference being an integration of a semantic and instance-based matcher. The semantic matcher will employ external background knowledge and will be integrated with existing logic-based techniques [19] while the instance-based matcher will employ link-key extraction method. The matcher will then be tested on the partitioned ontologies. The output of this matcher will be a series of alignments (correspondences).

[1] https://pypi.python.org/pypi/nltk.

[2] http://oaei.ontologymatching.org OAEI is an annual ontology matching competition that provides authoritative test and evaluations of ontology matching techniques or algorithms.

[3] http://www.dbpedia.org.

[4] http://www.freebase.com/.

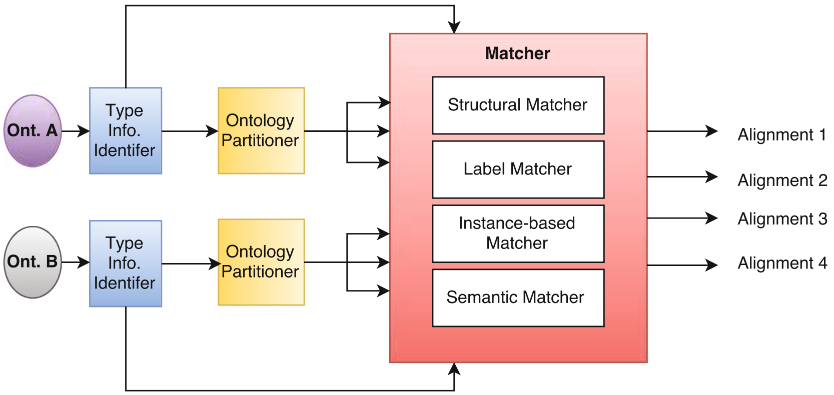

Fig. 3. Proposed architecture for stage 1

4.2 Research Stage 2

The second stage of this research will investigate and develop techniques that will help in producing quality mappings. An aggregation and alignment extraction technique will be developed by integrating user knowledge (similar to ALCOMO[5]) with weighting [5] techniques. This technique will enable us to select the best correspondences produced by the individual matchers. The aggregation and alignment extraction technique will be implemented on the set of alignments produced by the matcher. The results will be evaluated using standard measures such as precision, recall and F-measure by comparison with OAEI gold standard.

4.3 Research Stage 3

This stage will investigate reducing ambiguity to address alignment inconsistency and inaccuracy in ontology matching. An alignment improvement technique will be developed by adopting and extending existing approaches such as [15] using SAT solvers [6] and some reasoning techniques [13]. This will ensure consistency and accuracy in ontology matching by reducing ambiguity of data instances and helping to identify sets of exact and non-exact correspondences. The alignment improvement technique will be evaluated on the result of stage 2.

5 Preliminary Results

Matching and validating large ontologies is very difficult. However, when the ontologies are partitioned, it becomes easier for the single modules to be matched. Relatively, partitioning Big Data makes it easier to be used in an ontology matching process. In order to obtain an effective ontology partitioner, an ontology

[5] http://web.informatik.uni-mannheim.de/alcomo/.

partitioning test was carried on the OAEI-IM'10[6] ontology datasets using Scikit clustering[7] and Python. The Person1 and Person2 datasets consist of 2000 entities and 9000 RDF triples each. The attributes included in the ontology are: name, surname, street number, address, suburb, postcode, state, date of birth, age, phone number and social security number. The ontologies contain 4 types, namely Person, Suburb, State and Address, each with distinct attributes. With this simple schema, it is expected that the clustering should identify 4 clusters, one for each type.

The selected clustering algorithms are the K-Means, DBSCAN, Mean shift, Affinity Propagation and Birch. After running these algorithms on the ontologies, the results showed that only K-means and Birch (see Table 1) identified the desired clusters. This is because some of these clustering methods are unable to handle high dimensional data. In addition, Mean Shift and DBSCAN are density-based clustering methods that require a (nearly) continuous density function, hence, cannot yield any useful results in discrete scenarios. Affinity Propagation has high time and memory complexity, hence, is limited to clustering small sized datasets. Although Birch produced the desired clusters, the identification of its threshold parameter is complex and data specific; therefore, will be inappropriate to use for partitioning large datasets. K-Means also require a threshold parameter to be identified. However, the parameter selection for K-Means is relatively simple and it is expected that, the structure of the ontologies and the information from the missing type identifier will enable a reasonable estimate of this parameter. We will suggest K-Means as good for partitioning simple datasets but experiments need to be performed on complex datasets.

Currently, we are performing a similar evaluation on hierarchical clustering methods. The idea is to hierarchically cluster the instances to infer hidden structure in the ontology which can help to further partition a large-grain ontology into finer distinctions that may help find better matches and even guide the future evolution of the ontology.

6 Evaluation Plan

To evaluate our matching algorithms, we will adopt the benchmark tests from OAEI ontology matching campaign 2016. We will begin the evaluation by checking if the datasets have the different type of heterogeneity[8] (e.g., class heterogeneity, attribute-type heterogeneity) and if not, we will introduce specific heterogeneity using the framework in [2]. This framework permits the user to upload the source ontology and input heterogeneities between the source ontology and target ontology based on his/her knowledge and actions. We will then follow the evaluation criteria of OAEI, calculating the precision, recall and f-measure of each test case. The results of our algorithms will be compared with that of other OAEI participants. Finally, our matching system will be submitted

[6] http://oaei.ontologymatching.org/2010/im/.

[7] http://scikit-learn.org/stable/modules/clustering.

[8] Heterogeneities such as RDF, EXCEL and DB have been left aside.

Table 1. Ontology clustering results

Clustering algorithm	Validity (Number of clusters)	Runtime performance(s)
K-Means	4	1.17[a]
DBSCAN	0	0.15
Mean Shift	1	0.4
Affinity Propagation	92	2.09
Birch	4	1.33

[a] This is for multiple runs using inertia to select best result (0.117 per run on average)

to the OAEI'18 workshop/conference for its performance to be compared with top-ranked systems that will be participating in OAEI'18 campaign.

7 Conclusion

This research will result in the development of algorithms that will improve upon the existing state of the art in ontology matching by using techniques from different fields such as information retrieval and graph matching. The novelty lies in the individual matchers as well as the aggregation and alignment extraction method. For instance, to deal with class mismatch, a technique to infer fine grain type information from data instances will be incorporated into the matching process to bring the schema to the same level of granularity. A dynamic weighted sum in addition to user knowledge will then be applied to select the final mappings which will be improved using SAT solvers to eliminate ambiguity and inconsistencies.

Acknowledgement. This research is supported by the Data to Decisions Cooperative Research Centre (D2D CRC). The thesis is supervised by Prof. Markus Stumptner and Dr. Wolfgang Mayer.

References

1. Agrawal, R., Ailamaki, A., Bernstein, P.A., Brewer, E.A., Carey, M.J., Chaudhuri, S., Doan, A., Florescu, D., Franklin, M.J., Garcia-Molina, H., Gehrke, J., Gruenwald, L., Haas, L.M., Halevy, A.Y., Hellerstein, J.M., Ioannidis, Y.E., Korth, H.F., Kossmann, D., Madden, S., Magoulas, R., Ooi, B.C., O'Reilly, T., Ramakrishnan, R., Sarawagi, S., Stonebraker, M., Szalay, A.S., Weikum, G.: The Claremont report on database research. ACM Sigmod Rec. **37**(3), 9–19 (2008)
2. Chowdhury, N.A., Dou, D.: Evaluating ontology matchers using arbitrary ontologies and human generated heterogeneities. In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7566, pp. 664–681. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33615-7_15

3. Do, H.H., Rahm, E.: COMA: a system for flexible combination of schema matching approaches. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002), Hong Kong, China, pp. 610–621. VLDB Endowment (2002)
4. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In: Proceedings of the 11th International Conference on World Wide Web (WWW 2002), pp. 662–673. ACM, New York (2002)
5. Euzenat, J., Shvaiko, P., et al.: Ontology Matching, 2nd edn. Springer, Heidelberg (2013)
6. Gomes, C.P., Kautz, H., Sabharwal, A., Selman, B.: Satisfiability solvers. Foundations of Artificial Intelligence, Chap. 2, vol. 3, pp. 89–134. Elsevier (2008)
7. Hu, W., Chen, J., Zhang, H., Qu, Y.: How matchable are four thousand ontologies on the semantic web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., Leenheer, P., Pan, J. (eds.) ESWC 2011. LNCS, vol. 6643, pp. 290–304. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21034-1_20
8. Hu, W., Qu, Y.: Falcon-AO: a practical ontology matching system. Web Seman. Sci. Serv. Agents World Wide Web **6**(3), 237–239 (2008)
9. Knoblock, C.A., Szekely, P.: Exploiting semantics for Big Data integration. AI Mag. **36**(1), 25–38 (2015)
10. Li, J., Wang, Z., Zhang, X., Tang, J.: Large scale instance matching via multiple indexes and candidate selection. Knowl.-Based Syst. **50**, 112–120 (2013)
11. Ma, Y., Tran, T., Bicer, V.: Typifier: inferring the type semantics of structured data. In: Proceedings of the 29th International Conference on Data Engineering (ICDE 2013), Brisbane, Australia, pp. 206–217. IEEE (2013)
12. Mao, M., Peng, Y., Spring, M.: An adaptive ontology mapping approach with neural network based constraint satisfaction. Web Seman. Sci. Serv. Agents World Wide Web **8**(1), 14–25 (2010)
13. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Reasoning support for mapping revision. J. Logic Comput. **19**(5), 807 (2008)
14. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: a literature review. Expert Syst. Appl. **42**(2), 949–971 (2015)
15. Pührer, J., Heymans, S., Eiter, T.: Dealing with inconsistency when combining ontologies and rules using DL-programs. In: Aroyo, L., Antoniou, G., Hyvönen, E., Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 183–197. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13486-9_13
16. Schneider, T., Hashemi, A., Bennett, M., Brady, M., Casanave, C., Graves, H., Gruninger, M., Guarino, N., Levenchuk, A., Lucier, E., Obrst, L., Ray, S., Sriram, R.D., Vizedom, A., West, M., Whetzel, T., Yim, P.: Ontology for big systems: the ontology summit 2012 communiqué. Appl. Ontology **7**(3), 357–371 (2012)
17. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013)
18. Wang, P., Zhou, Y., Xu, B.: Matching large ontologies based on reduction anchors. In: IJCAI, Barcelona, pp. 2343–2348 (2011)
19. Zhang, W., Zhao, H., Mei, H.: A propositional logic-based method for verification of feature models. In: Davies, J., Schulte, W., Barnett, M. (eds.) ICFEM 2004. LNCS, vol. 3308, pp. 115–130. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30482-1_16