Interactive Visualization of Large Ontology Matching Results

Yiting Li, Cosmin Stroe, and Isabel F. Cruz

ADVIS Lab, Department of Computer Science, University of Illinois at Chicago, USA
yiting.star@gmail.com, cstroe@gmail.com, isabelcfcruz@gmail.com

Abstract. We add to the widely used AgreementMaker system the capability to visualize the results of matching large ontologies with a user interface that supports navigation and search of the ontologies. The interface also supports user intervention when using a feedback loop strategy where users validate candidate mappings that have been computed automatically by matching algorithms. The interface further displays properties of the concepts to facilitate the decision process.

1 Introduction

An ontology provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary. An increasing number of organizations are using ontologies to organize their knowledge. However, different ontologies exist for the same knowledge domain. To address this issue, *ontology matching* is needed, which is the process of finding the relationships, called *mappings*, between concepts (classes or properties) of two different ontologies, the *source* and the *target* ontologies [1]. Ontology matching can be performed automatically, manually, or semi-automatically.

A variety of algorithms, which we call *matchers*, have been developed for matching. For example algorithms that are based on string similarity of the class labels or on the structure of the ontologies. Advanced ontology matching systems, such as AgreementMaker, use combinations of a large variety of algorithms [2, 3]. In this paper, we do not focus on any particular matching algorithm, but rather on visualizing the results of the ontology matching process so as to enable the involvement of users with the objective of obtaining better results. The quality of a matching algorithm or of a combination of matching algorithms is measured in terms of precision, recall, and F-measure, by comparing the obtained mappings with the mappings that belong to the *gold standard* or *reference alignment*. The OAEI (Ontology Alignment Evaluation Initiative)[1] makes reference alignments available for a variety of their tracks, which is a great asset for the ontology matching community.

The purpose of our work is twofold. First, we want an interactive visualization method for large ontologies. Second, we want to support visual analytics in a semi-automatic ontology matching process. We define some of these terms next. *Semi-automatic ontology matching* integrates automatic and manual methods. Those mappings that are believed to be incorrect are presented to users for

[1] http://oaei.ontologymatching.org/

validation [4, 5]. The workflow consists of a loop where the outcome of the validation step is fed back into the ontology matching process. *Visual analytics* is the science of analytical reasoning supported by interactive visual interfaces [6]. In ontology matching, visual analytics can help users validate the mappings [4].

Our focus is on ontology matching visualization, not on ontology visualization, that is, we want to support the visualization of complex relationships between source and target ontology structures. Ontology matching visualization is further complicated when matching large ontologies. For example, the display of an ontology as a tree structure using the JTree class can be very helpful for small and medium size ontologies, but is less helpful for large ontologies because of the amount of scrolling needed to locate the different mappings. To better compare and analyze the matching results, a visual representation that can scale to large or very large ontologies is needed. In this paper, we investigate interactive visualizations that use pie charts, which naturally scale to any ontology size.

Our paper is organized as follows. In Section 2, we cover related ontology matching visualization approaches and in particular those that are intended for large ontologies, be they based on graphs, treemaps, or pie charts. We also cover interactive approaches based on matrices that support visual analytics. In Section 3, we describe our visualization technique, starting with the design criteria. We then describe the pie chart visualization and the comparative visualization of matching algorithms, as well as the user interface organization. In Section 4, we describe the use of our interactive interface that supports the validation of mappings in a feedback loop setting and its integration with AgreementMaker. Finally, in Section 5 we draw conclusions and point to future work.

2 Related Work

A recent survey of visualization methods for ontology matching [7] establishes a list of requirements for those systems to support user involvement. However, the functionality of the systems that are covered fall short of those requirements. Therefore, there is the urgent need to develop ontology matching visualization approaches that scale to large and very large ontologies. In what follows, we describe briefly relevant interactive visualization methods.

2.1 Cluster Visualization

The cluster representation [8] shows both detailed and general information of matching results and provides in addition a JTree visualization. Users can select the level at which they want to cluster the results. For the visualization of each ontology, this approach uses a spring-embedded graph drawing algorithm. This method is constrained by its computation complexity, which is $O(n^2 \times s)$ where n is the number of concepts in the ontology and s the number of iterations. Other drawbacks of the approach are that only the results of a single matching algorithm can be visualized. The concepts of each ontology are color coded so as to show whether they have been mapped and the level of similarity found with classes of the other ontology.

Another graph drawing representation that was developed for the AgreementMakerLight system [9], which extends AgreementMaker [2] to very large ontologies, provides a single visualization that also uses a spring-embedded technique where both ontologies and the mappings between classes are displayed. However, it displays few mappings at a time [10]. This technique does not allow to compare the results of more than one matching algorithm at once.

2.2 Treemap and Pie Chart Visualizations

The PROMPT+COGZ tool supports multiple visualizations, including one based on TreeMaps and another one that displays pie charts [11]. TreeMaps have the advantage that they can be used to visualize large amounts of data, but fit in a small area. Forcefully, details cannot be provided for large ontologies. Some details are provided by a pie chart view with information for each branch of the ontology, such as the number of candidate mappings, mapped classes, and classes that are not mapped. We note, however, that for the display of candidate mappings, the tool falls back on a JTree-like visualization, which uses a fish-eye view lens to allow for the display of larger ontologies. Clearly this is overall an advanced visualization tool. However, it does not seem to be able to show concurrent displays of more than one matching algorithm at a time. Together with the approach we present in this paper, this is the only other tool that supports pie charts with the difference that our pie charts drive the navigation across all levels of the ontologies, while their navigation appears instead to be driven by the TreeMap visualization.

2.3 Matrix Visualization

A matrix visualization where the classes of both ontologies are placed along the X and Y axes provides a more comprehensive view of the matching process as compared with the aforementioned methods because it allows for the whole mapping space to be visualized with equal detail. We know of two such visualizations: the one provided by iMerge [12] and the one provided by AgreementMaker [4]. Both systems support multiple visualizations, including a traditional JTree-like visualization for each ontology with connections between the two ontologies showing the mappings. Like the systems already mentioned, these two systems do not scale to very large ontologies, however AgreementMaker has the distinct capability of allowing for the comparison of different matching algorithms side by side and simultaneous navigation across the various similarity matrices. The color intensity supported by AgreementMaker, which depicts the matching confidence score for each mapping, adds an extra dimension to the visualization without adding extra space. Figure 1 shows the AgreementMaker visual interface for matrix visualization, which is called *Visual Analytics Panel* because it is used to support the visual analytics process. The top toolbar controls the matching process. The overall panel highlights a vector of points for the same mapping (the signature vector). Each matrix is associated with a matching algorithm [4]. AlignmentVis is an interactive user interface that supports matrices among other visualizations, using a multi-linked view paradigm [13]. However, it is not currently targeted to very large ontologies.

In conclusion, none of the above approaches provides both for scalability and for an interactive meaningful display that supports visual analytics for large or very large ontologies.

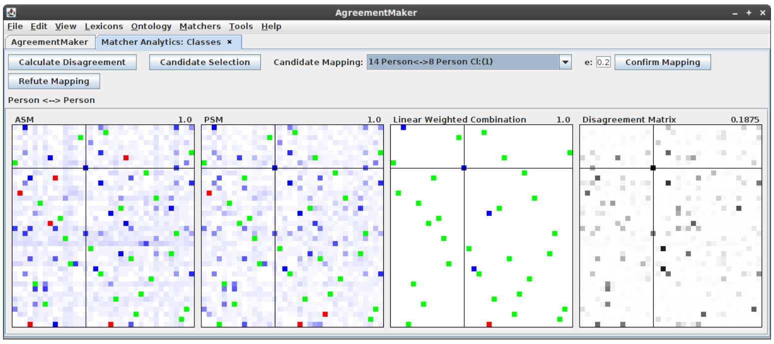


Fig. 1: The Visual Analytics Panel of AgreementMaker [4].

3 Visualization

3.1 Design Criteria

We have made the following choices and present their rationale:

– We allow for ontology navigation, exploration, and searching.
– We do not display all the mappings at once. If we did so, it would be difficult to find a visualization whose size does not depend on the number of mappings or on the size of the ontologies involved.
– We want to focus on the mappings one level at a time and aggregate the results for the children of the nodes at that level. As long as navigation and searching functions are available, users can easily locate any single ontology node in the whole structure and see the mappings they need.
– We choose a visualization based on pie charts. The reason of this choice is that no matter how large the data size is, the pie chart requires always the same modest area. Given this, we can display the results of more than one matching algorithm at a time.
– When matching two classes, the most valuable information is the confidence score found by the matching algorithm (between 0% and 100%). The visualization can give priority to those mappings that maximize the confidence score between two nodes.
– We want to make apparent the differences between matching algorithms.
– We enable user feedback acquisition.

3.2 Pie Chart Visualization

We start by describing how we visualize the information in terms of pie charts. For each visualization there are two pie charts, one that corresponds to a concept or node in the source ontology graph, S, called the *current node*, and a pie chart corresponding to a concept or node in the target ontology graph, T. We show the percentage of their children whose confidence score falls in a particular range. Figure 2 shows those ranges, namely 81%-100%, 61%-80%, 41%-60%, and $\leq$40%. For example, 41% of the children nodes of S have confidence scores in the range

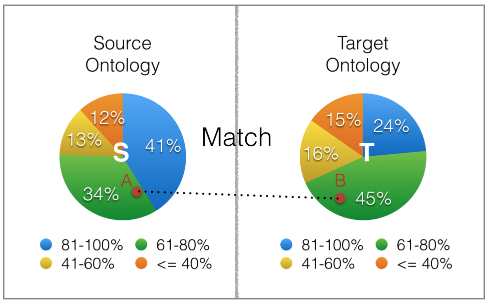


Fig. 2: User interface design that displays matching results.

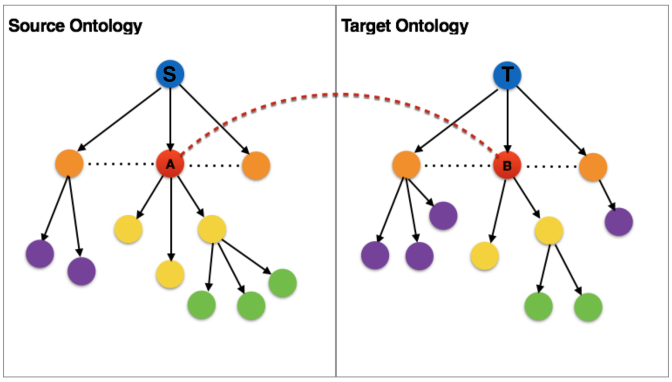


Fig. 3: Two ontology subgraphs showing a mapping between nodes A and B.

81%-100%. Figure 2 also highlights a mapping between two nodes, A that is a child of S and B that is a child of T.

Figure 3 shows schematically subgraphs of both ontologies with roots S and T, their children, among which there are subclasses A and B, the siblings of A and B, and their children (and grandchildren). To enable navigation along the ontologies, users should be able to traverse the ontology *vertically* from a node to its children but also *horizontally* from a node to its siblings.

The next question we address from the interface viewpoint is how to combine both types of navigation. We provide a list that represents the children of a node and a tree view that represents the siblings of the current node. The main panel of the user interface for the initial version of the prototype is shown in Figure 4. In the center area there are the two pie charts previously discussed.

Immediately left of the pie charts there is a list. When users click on a pie chart slice, the list contains the ontology nodes with confidence score within the corresponding range, sorted by the confidence score. Clicking on a node in the list leads to an update of the pie charts, as that node becomes the current node.

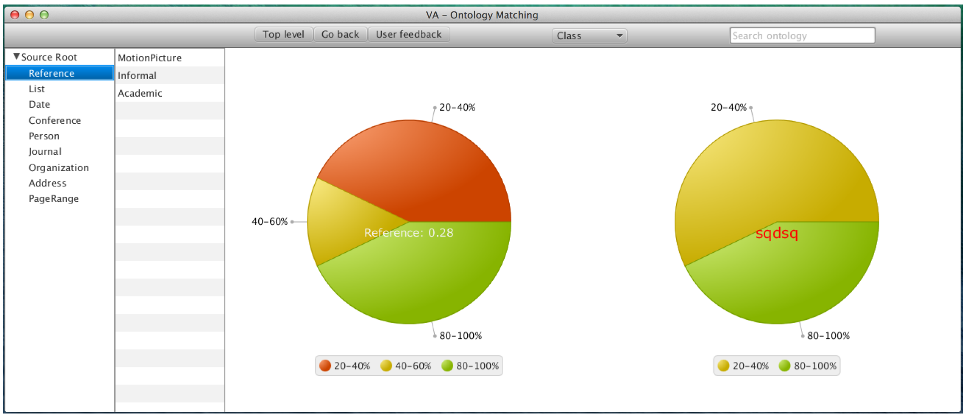


Fig. 4: Main panel, where *Reference* and *sqdsq* are concepts in the OAEI Benchmark Track.

The leftmost part of the interface contains the tree view. It shows all the siblings of the current node. When clicking on a node in the tree view, both the pie charts and the lists are updated, reflecting the change of the current node. On the top right there is a search box. Upon entering the name of the node in the search box, the left pie chart displays that source node and the right pie chart displays the target node that matches the source node. Accordingly, the tree and list view on the left are updated as well. In addition, for an easier navigation we provide additional functions such as "go to the top level", "go to the previous level", and "switch between class and property".

3.3 Algorithm Comparison

To compare the results of two matching algorithms, we need to visualize their results at the same time. We have therefore upgraded our user interface panel to load multiple results as shown in Figure 5. We are making full use of the containers in JavaFX to manage the visual elements within the available space. We use two tile panels to display two ontology pairs. The upper panel is the main panel (colored green) and the lower one is the sub panel (colored yellow). The tree view shows the siblings of the source node in the main panel and the list view shows the children of that source node. Figure 6 displays the schematic representation of the user interface, including the flow panel. The flow panel shows the algorithms we have loaded into the application. In this example, the main panel shows the results of the algorithm we loaded using the second button of the flow panel. When selected, the second button is colored green, so as to provide a color match with the main panel.

Users can choose any algorithm for the main panel or for the sub panel. The difference between the two panels is that all lists get updated according to the changes to the main panel.

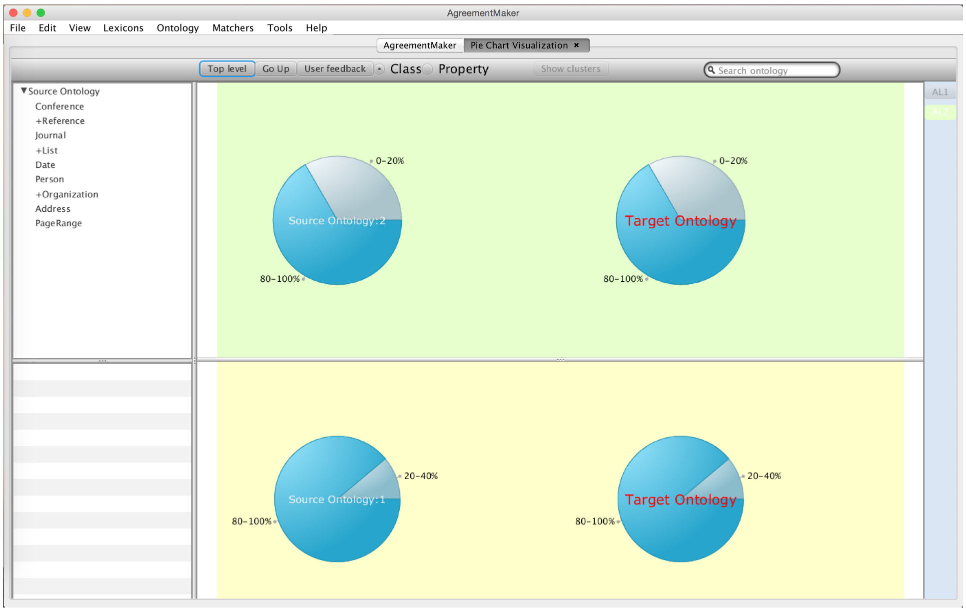


Fig. 5: Upgraded user interface to display multiple algorithm results.

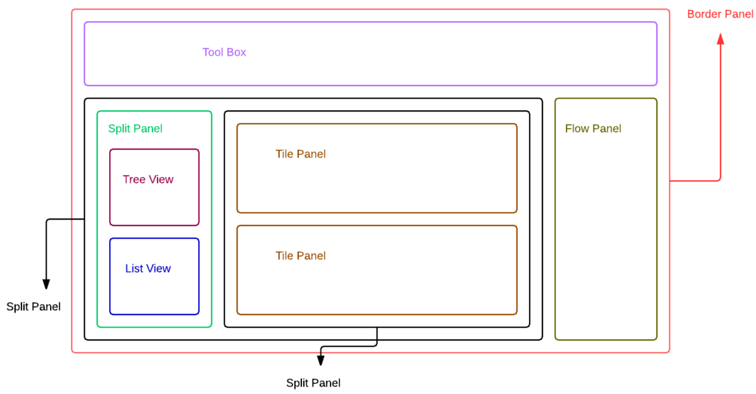


Fig. 6: Schematic representation of the upgraded user interface.

When the users click on a slice of the main pie chart, all features including all pie charts and lists are updated. To change the matching algorithm for the main panel, users only have to select another algorithm from the flow panel.

4 Interactive Ontology Matching

We list here our objectives for an interactive mechanism for matching ontologies that can assist users in a semi-automatic ontology matching process, where users provide feedback:

- Show candidate mappings for validation to the users; candidate mappings are determined automatically using quality measures [4, 14, 15].
- Register the validation choices made by the users.
- Allow for class and property navigation to assist users in their validation decisions.
- Allow users to search for a specific class and navigate through the classes.
- Support the creation of new mappings that are missed by the automatic process.

4.1 Interactive Workflow

Because automatic matching methods do not always provide complete or correct mappings, the combination of user validation with the automatic methods can lead to better results than the automatic methods alone.

The interactive workflow is shown in Figure 7. It shows a "user feedback loop" (UFL) strategy [14, 15] integrated with the visual analytics (VA) approach, in that the results provided by the user are fed back into the matching process and the user is helped by the interactive user interface [4]. In the figure, the visual

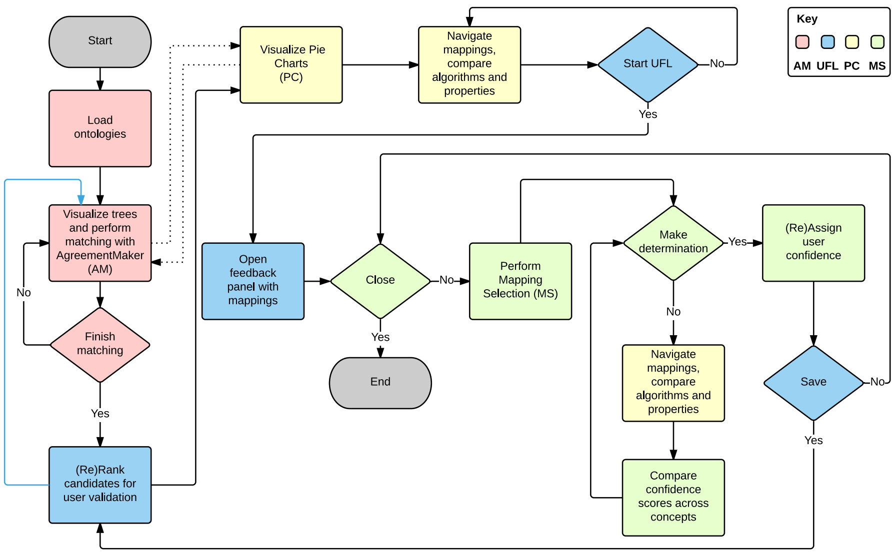
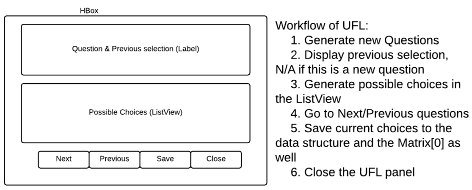


Fig. 7: Workflow of the interactive process.

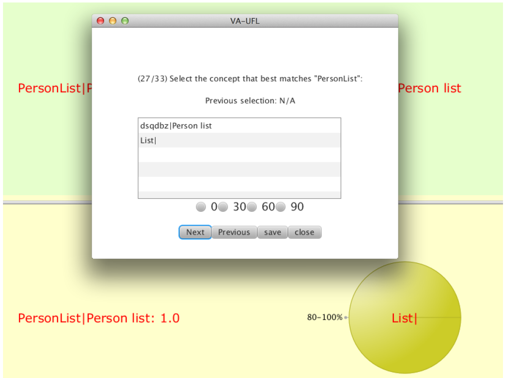
interfaces provided by AgreementMaker and the new interface communicate, so as to allow for complementary views. The user interface that is used to perform the mapping selection is an important component, which we describe next.

4.2 User Interface for Mapping Selection

The user interface for the mapping selection must display one or more mappings to be validated by the users. When it shows more than one mapping, users are asked to choose among them. Navigation using the main user interface (Figure 5) provides the confidence scores to assist users in making their selection. Figure 8a shows the schematic representation of the interface and Figure 8b shows a snapshot of its implementation.



(a) Schematic representation.



(b) Mapping selection, where users input their level of confidence (0-30-60-90).

Fig. 8: User interface.

4.3 Property Comparison

Automatic algorithms match classes according to a variety of lexical, syntactic, and structural criteria [2]. In addition, they may use other criteria, which can be incorporated into the automatic algorithms or visualized and presented to the users. For example, the properties associated with the classes can be considered [16]. In our user feedback loop strategy when allowing users to choose among mappings, we present the properties associated with the classes. Our interface displays both the confidence scores and floating panels that display the properties of each of the classes, as shown in Figure 9. We note that we only display once the panel associated with the source concept, *ConferenceEvent*.

4.4 Integration with AgreementMaker

Using the AgreementMaker system, the source and target ontologies are visualized side by side using a tree paradigm as shown in Figure 10. The control panel

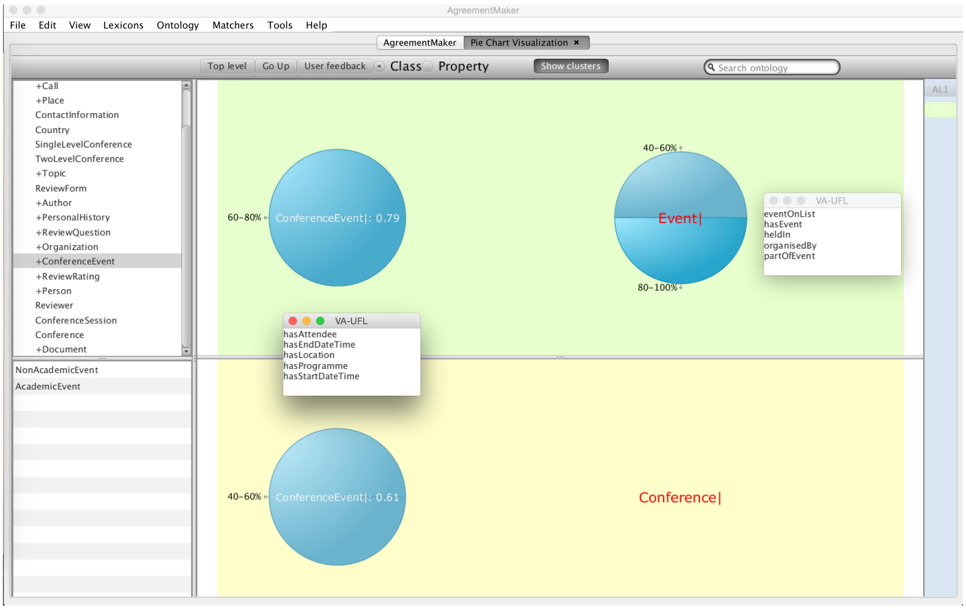


Fig. 9: Property clustering.

(at the bottom of the interface) allows for users to run a variety of matchers. In this example, two automatic matchers have been activated (the Parametric String Matcher and the Vector-based Multi-word Matcher) in addition to manual matching. The picture depicts the display of the two ontologies and the mappings obtained in this way.

Every set of mappings in AgreementMaker is represented by the MatchingTask class. The MatchingTask class contains the following elements: a matcher, its associated parameters (e.g., the confidence score threshold), and the mappings produced by the execution of the matcher. To open our visualization system, users can select the Pie Chart Visualization tab of the drop down menu, as shown in Figure 10. The key point in the integration of the pie chart visualization with AgreementMaker is that we pass all the MatchingTask instances from AgreementMaker to the pie chart visualization. After selecting the Pie Chart Visualization tab, another tab shows up and the pie charts will be initialized automatically (see Figure 5). Users can easily switch between the tree and pie chart visualizations by clicking on the available tabs at the top of the display.

5 Conclusions and Future Work

We devised a visualization tool for large ontology matching that integrates seamlessly with the widely used AgreementMaker system. It supports advanced navigation, interaction, and analysis and decision making features. In particular, for navigation, our tool supports the visual representation of the source and target ontologies and mappings between classes in those ontologies. It allows for the

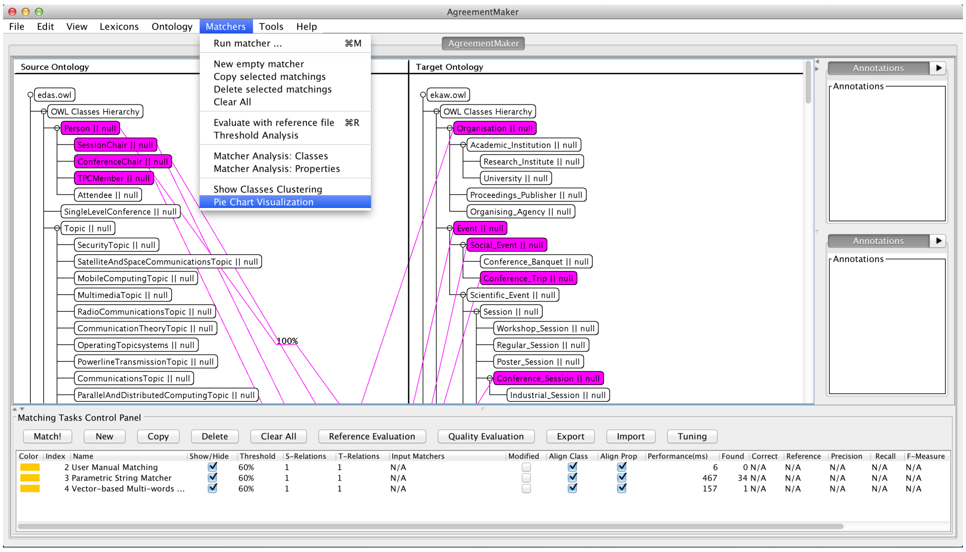


Fig. 10: Integration with AgreementMaker.

detailed access to pairs of classes to match, while providing ready access to other parts of the ontologies. Our tool displays confidence scores between classes and provides an overview of the confidence scores for the children of those classes.

For interaction, our tool supports user-driven navigation of classes and properties, the ability to search for a specific class, and to traverse the ontologies vertically (children of a class) and horizontally (siblings of a class).

Finally, for analysis and decision making, our tool displays several possible mappings to the users, so that they can choose among them, as part of a user feedback loop strategy that combines automatic with manual matching methods.

Clearly, there are several directions for future work. The first one is that we would like to extend the comparison of matching algorithms to more than two at a time. It may be the case that no single visualization strategy works separately, especially for very large ontologies. AgreementMaker already provides several different strategies [4, 2, 13]. Experiments would be needed to determine the usability and effectiveness of the different strategies when used separately or in coordination with one another.

Acknowledgments

This research was partially supported by NSF Awards IIS-1143926, IIS-1213013, and CCF-1331800. We would like to thank one of the anonymous reviewers, whose questions helped improve the final version of the paper.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Heidelberg (DE) (2007)
2. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB **2**(2) (2009) 1586–1589
3. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: ISWC International Workshop on Ontology Matching (OM). Volume 551 of CEUR Workshop Proceedings. (2009) 49–60
4. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: IEEE International Conference on Data Engineering (ICDE). (2012) 1321–1324
5. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively Learning Ontology Matching via User Interaction. In: International Semantic Web Conference (ISWC). Volume 5823 of Lecture Notes in Computer Science., Springer (2009) 585–600
6. Bertini, E., Lalanne, D.: Investigating and Reflecting on the Integration of Automatic Data Analysis and Visualization in Knowledge Discovery. SIGKDD Explorations Newsletter **11**(2) (May 2010) 9–18
7. Ivanova, V., Lambrix, P.: User Involvement for Large-Scale Ontology Alignment. In: International Workshop on Visualizations and User Interfaces for Knowledge Engineering and Linked Data Analytics. (2014)
8. Lanzenberger, M., Sampson, J.: AlViz - A Tool for Visual Ontology Alignment. In: Conference on Information Visualization (IV), IEEE Computer Society (2006) 430–440
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight ontology matching system. In: International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), Springer (2013) 527–541
10. Pesquita, C., Faria, D., Santos, E., Neefs, J.M., Couto, F.M.: Towards Visualizing the Alignment of Large Biomedical Ontologies. In: Data Integration in the Life Sciences (DILS). (2014)
11. Falconer, S.M., Storey, M.A.D.: A Cognitive Support Framework for Ontology Mapping. In: International Semantic Web Conference/Asian Semantic Web Conference (ISWC/ASWC). Volume 4825 of Lecture Notes in Computer Science., Springer (2007) 114–127
12. El Jerroudi, Z., Ziegler, J.: iMERGE: Interactive Ontology Merging (poster). In: International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW). (2008)
13. Aurisano, J., Nanavaty, A., Cruz, I.F.: Visual Analytics for Ontology Matching Using Multi-Linked Views. In: ISWC International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data (Voila!). (2015)
14. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-As-You-Go Multi-User Feedback Model for Ontology Matching. In: International Conference on Knowledge Engineering and Knowledge Management (EKAW). Springer (2014) 80–96
15. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Quality-Based Model for Effective and Robust Multi-User Pay-As-You-Go Ontology Matching. In: Semantic Web Journal. IOS Press (2015)
16. Hu, W., Qu, Y.: Falcon-AO: A Practical Ontology Matching System. Web Semantics: Science, Services and Agents on the World Wide Web **6**(3) (2008) 237–239