Visualization Tools for Results of Entity Resolution

Cheng Chen¹, Mahmood Mohammed¹, and John R. Talburt¹
¹ Information Science Department, University of Arkansas at Little Rock, Little Rock, AR, USA

Abstract - This paper introduces methods for visualizing the results of Entity Resolution processes. They allow users to visualize the results from any resolution process. These tools will also help users to compare results from different rules-set in the process of Entity Resolution in Entity Identity Information Management. This will facilitate finding false positive and false negative errors. These methods have been applied to the results produced by OYSTER, an open source entity resolution system.

Keywords: Entity Resolution, Entity Identity Information Management, Visualization Tools, Information Visualization

1 Background

Entity Identity Information Management (EIIM) is a component of entity identity management (EIM) that utilizes data structures, data integration, and entity resolution (ER) methods and algorithms. EIIM aims at maintain entity identity integrity. Entity identity integrity requires that each entity in the domain should have one and only one representation in the system, which is called an identity. [1] Figure 1 shows a high-level view of EIIM components and processes.

![Figure 1: EIIM Components and Interactions.](image)

Figure 1 makes it clear that steps 4, 5, 7, and 9 are related to the resolution process. It is apparent that ER is the core process for EIIM. Entity resolution is the process of determining whether two records in an information system are referring to the same object or to different objects. The term entity describes the real-world object, a person, place, thing, etc. A reference is a collection of attribute values for a specific entity. According to the requirements of entity identity integrity in EIIM, the fundamental law of ER is that two entity references should be linked if and only if they are equivalent [2].

2 Problem Statement

In EIIM, the primary goal is to achieve and maintain entity identity integrity. Entity identity integrity is essential a one-to-one correspondence between the entity identity structures (EIS) in the information system and the real-world entities in the domain of interest. In practical applications, EIIM managers are responsible for evaluating the degree of entity identity integrity existing between EIS and real-world entities. For large amounts of data, it is hard for EIIM managers to perform this task by simply browsing and evaluating EIS at random.

In addition, for certain EIIM tools such as OYSTER, the EIS are written to storage as XML documents that can be accessed and updated in a future runs [2]. This requires an EIIM manager to understand XML in order look through and evaluate the EIS. Programming skills may also be needed in order to measure the degree of identity integrity. Even for the managers who have a solid foundation and experience in programming and data analysis, it is difficult to find the false negative errors if the references for a single identity are spread across several EIS.

Moreover, EIIM is a cyclical process not a one-time process. Consequently, EIS are not produced once, but are constantly improved based on better understandings of the data and the matching logic that compares the references. The matching logic is usually implemented as one or more matching rules [3]. Matching rules are the primary determinate in maintaining entity identity integrity in the ER process. Another problem occurs when EIIM managers change results in an attempt to improve the level of entity identity integrity. Isolating and observing the differences in the results produced by two different rule sets can be difficult.

There are numerous techniques that can solve part of the above two problems, but few use information visualization. Information visualization directly addresses the requirements of human perception to help users analyze complex relationships. Humans have the ability to recognize the spatial configuration of elements in a picture and notice the relationships between elements quickly. This highly developed visual ability allows people to grasp the content of a picture much faster than they can scan and understand text [4]. Information visualization methods could save EIIM manager’s time, and help them to make quick and accurate...
decisions, thus saving money for corporations or organizations. Moreover, even for people who do not know programming at all, information visualization tools provide them with the tools for information and data analysis in EIIM.

3 Information Visualization Methods

There are two information visualization methods to completely address the above two problems respectively: Treemap and Graph. Treemap methods solve the problem of visualizing EIS. EIIM managers can elucidate differences between EIS from different cycles of EIIM through Graph methods.

3.1 Treemap Methods

Treemapping is a visualization method for presenting hierarchical data by using nested rectangles. It maps hierarchical information to a rectangular 2-D display in a space-filling manner [5, 6], which is shown in Figure 2.

![A treemap](image)

Figure 2: A treemap

Treemaps have three significant features:

1) Each box on the chart may be contained in another box, hence the hierarchical view.

2) The size is usually determined by the relative size of a parameter in comparison to the full size of the chart (i.e. the 'bigger' the value of X, the bigger it is on the chart).

3) The color shows another dimension in the parameters, like a movement in time [7].

Even though EIS do not have a strong hierarchical relationship, the second and third features of treemaps show huge advantages for visualization of EIS. In EIIM, combining two references or EIS that are not equivalent is a false positive error. In contrast, failing to combine two references or EIS that are equivalent is a false negative error [1]. It is easy to infer that the EIS with bigger size are prone to have false positive errors while the EIS with smaller size are comparatively easy to get false negative errors. So it is a very important feature of treemaps that they can show the size of EIS by drawing a collection of rectangular bounding boxes whose sizes are entirely dependent on the number of references of EIS.

In addition to rectangle sizes, the color of each rectangle in treemaps can represent other features of EIS. For example, cohesion of EIS can be represented by the color of each rectangle in treemaps. In computer programming, cohesion refers to the degree to which the elements of a module belong together [8]. Similarly, the cohesion of EIS represents the degree to which the references comprising one EIS have similar values. Colors are set to vary due to the cohesion of each EIS, so EIIM managers can see the EIS with low cohesion and high cohesion immediately from color.

3.2 Graph

Graph visualization is based on the mathematical theory of networks and graph theory. A graph consists of vertices, also called nodes, and of edges (the connecting lines between the nodes). Directed graph is a graph, or set of nodes connected by edges, where the edges have a direction associated with them [9].

Due to the direction edges, directed graph is suitable to illustrate differences between EIS produced by ER processes with different rule sets. For example, two EIS A and B are produced in first EIIM cycle and then in second EIIM cycle EIS A and B are resolved as equivalent. Therefore, they integrated into a single EIS C. This difference can be revealed as a directed graph in Figure 3(a). The other case where EIS A in first EIIM cycle is resolved to two EIS, B and C, in second EIIM cycle is shown in Figure 3(b). For cases where EIS do not have any changes within several EIIM cycles, those changeless EIS are ignored most of the time because EIIM managers often focus more on differences than agreements.

The graph methods can also be used to compare ER results with a truth set. In Figure 3(a), if EIS C is the truth, the EIS A and B shows that the ER process made a false negative error. In Figure 3(b), if EIS B and C are the truth, EIS A illustrates a false positive error in this ER process.

![Directed graph](image)

Figure 3: Directed graph
4 Implementation with OYSTER

OYSTER (Open sYSTem Entity Resolution) is an ER system developed by the Center for Advanced Research in Entity Resolution and Information Quality (ERIQ) at the University of Arkansas at Little Rock. OYSTER provides access to a variety of entity resolution algorithms as well as support for EIIM and persistent entity identifiers [10].

Two information visualization methods, Treemap and Graph, have been applied to the EIS produced by OYSTER. In OYSTER, EIS are written to storage as XML documents, and each EIS is enclosed in an <Identity> element of an XML document <root>. OYSTER assigns each EIS a unique 16-character identifier known as an OYSTER ID [10].

4.1 Treemap Identity Viewer Prototype

The interface of Treemap Identity Viewer Prototype is shown in Figure 4. This prototype demo can be found online at https://sourceforge.net/p/treemapidentity/. This prototype implements treemap methods from JTreeMap API [7].

In this prototype, treemaps of EIS with OYSTER ID inside rectangles accordingly is shown in the upper part of the interface. When a user clicks a certain EIS rectangle in treemaps, a window showing the raw data from references of this EIS will pop up. The colors inside the treemap are determined by the cohesion of EIS. The bigger cohesion in EIS, the lighter the treemaps, and the darker rectangles show the smaller cohesion. The cohesion of EIS in the prototype is calculated by the formula

\[
\text{cohesion} = \sum_{i \in \Gamma} c(A_i)
\]

Where \( \Gamma \) is attribute domain, \( A_i \) represents certain attribute, and

\[
c(A_i) = \frac{\text{pairs of equal records}}{\text{total pairs}}
\]

Pairs here refer to pairs of records independent of order.

For example, one EIS has two attributes, first name and last name, and four records. The first names of four records are “Jim”, “Jim”, “Jim”, and “James”, so

\[
c(A_1) = \frac{3}{6} = 0.5
\]

The last names of the records are “Jones” “Jones” “Jolson” “Jove”, so

\[
c(A_2) = \frac{1}{6} = 0.17
\]

Therefore, the cohesion of this EIS is 0.5+0.17=0.67.

Additionally, the search field and the slider for the size filter are both located in the middle of the panel. The search
function employs Apache Lucene 4.1.0 [11] to scan all references from the input XML file to build indices. This allows the user to find EIS whose references contain the query words. Furthermore, users can limit the size of EIS using the size filter. The treemaps view will adjust according to the size setting.

4.2 Identity Cross Comparator Prototype

Figure 5 shows the interface of the Graph Identity Cross Comparator Prototype. The demo can be downloaded from the link http://sourceforge.net/p/graphicc/. The technology implementation of the graph in this prototype is JGraphX Version 1.11.0.0 [12].

The prototype’s interface is divided into two parts: graph and table. The table part located on the lower half of interface shows OYSTER ID and size of respective EIS from two ER processes. When the user clicks any OYSTER ID from the left table, that row will be selected and changed to green in color. At the same time, the graph part in the upper half of interface will show the graph of this EIS as green in color and one or more EIS which share the same references with the selected EIS from the other ER processes as orange in color. The selection from the right table also works in the same way as the left table.

There is the “Function” menu item in the menu bar of the interface. There are two functions under this menu item. The first function can hide the equal EIS from both left and right tables. This function was added because changes are the main point for this analysis. The equal EIS here refers to the EIS which are constant in two ER processes. The equal EIS not only occupies table space, but will waste the users’ time when analyzing. This hidden function helps with users’ analysis. The other function is calculating Talburt-Wang Similarity Index (TWI). The TWI of similarity is defined as

\[
TWI = \frac{\sqrt{|A| \times |B|}}{|V|}
\]

Where A and B are two partitions of a set S and V is the set of overlaps between A and B [2].

5 Conclusion

This paper discusses using information visualization methods to help EIIM manager to evaluate entity identity integrity. The treemap and graph information visualization methods are presented to help to solve the following questions.

\checkmark How to visualize the results from Entity Resolution through raw data of output from ER?

\checkmark How to compare results from different rules-set in the process of ER in EIIM to find the false positive and false negative errors?

Application of these two methods to the results produced by OYSTER shows that information visualization methods do make sense on analyzing the ER results, and improving entity identity integrity in EIIM. And the visualization methods display huge potential to provide convenience for EIIM
manages after being tested by few researchers in University of Arkansas at Little Rock.

6 Future Work

For the two information visualization methods, Treemap and Graph, there are prototyped demos now. More trials and testing with humans will be executed in the future to determine the effectiveness of these tools.

On the other hand, more functions will be added to make the tools more robust. For example, the formula to calculate the cohesion needs to be studied in more depth for Treemap Identity Viewer, as well as additional calculation functions for calculating the similarity of EIS from different EIIM cycles. Additionally, zoom-in and zoom-out functions will be added to Treemap view to make the program more user-friendly. Moreover, compatibility to different systems, devices and the big data is another challenge for the tools.

7 Acknowledgment

The research discussed in this paper was funded in part by a grant from the University of Arkansas for Medical Sciences and Arkansas Department of Education.

8 References


