Abstract

In information technology, a user interface is “everything designed into an information device with which a human being may interact.” [1] The user interface already had permeated into our daily life routines. People use user interfaces in many different fields such as government, public health, shopping and personal mobile applications, by their own purposes of needs.

The goal of this project is to create an Interactive Visual Entity Resolution Interface to use in public health research area as part of privacy preserving interactive record linkage (PPIRL) research. Entity Resolution is a process of finding and matching datasets that belong to the same entity. The specific interface that this project will create will be used in public health area to manage personal (cliental) information. The interface saves and retrieves the data received from the database and performs entity resolution by letting the user decide whether to link the record or not.

As a programmer, one of the biggest challenge of designing a user interface is making the interface in user’s perspective. Not everyone wants to learn to manage the data using a program. Therefore, the interface needs to be relatively easy to learn and use. Designing a good user interface is a difficult process for a programmer because, a good interface that addresses less difficulties to the user will have a complex and nicely designed program inside the interface. To design a nice and easy interface, iteration of detailed plan and task by task programming was needed.

1. Introduction

As the volume and velocity of data explodes in so-called Big Data phenomenon, Entity Resolution is becoming a greater challenge. Many institutions and companies use Big Data to categorize and arrange the data that matters to them. People access, observe, recognize, change, and save the data to reduce the time and cost that takes to retrieve and reuse the data.

However, ambiguous records can always exist. Entity Resolution is a process to reduce these ambiguous datasets by linking and grouping. By reducing the ambiguous records, the volume of the datasets will be reduced, and if volume gets reduced, the time consumed to retrieve data will also be reduced. In this project, the area of interest will be narrowed down to public health area. Institutions related to public health usually handles personal records and the ambiguous records are mostly records with similar or missing information.

The purpose of this interface is to mostly perform the Entity Resolution and decrease the work of the user. However, generally speaking, computers, which can be referred to programs or algorithms, by themselves, cannot perform the tasks of entity resolution effectively. Therefore, the Entity Resolution Interface in this project will interact with the user to manually resolve complex entities that cannot be resolved automatically by computer.

An interface should facilitate user’s job for a task. Therefore, a good interface need to minimize the user’s complicated jobs and thinking process. The interface will be created in this project would need to make the user feel comfortable during the record matching process. By successfully making the interface, public health institutions such as hospitals should benefit from using this interface by efficiently managing the client data.

The interface will be displayed on the web and the data will be imported from csv files. Therefore, this project requires some basic knowledge of database and network programming (WebSocket API), in addition to web programming languages such as HTML/CSS, Javascript, DOM, Javascript Object Notation (JSON), Javascript library (JQuery) and Javascript data visualization library (d3.js). The output file will also be made in csv file.

The next part of this paper will talk about the Basic Interface Design. Section 2 will talk about the interface design. Section 3 will discuss about the user’s side and the output. Section 4 will conclude this paper.

2. Interface Design

One of the most important part of programming an interface is design. The design of an interface can make the user feel either the interface is useful or not useful. Therefore, designing a useful and easy interface was one of the main goal of this project.

The interface first needs a text box that has some information about the interface. So we added a text inside a text box using d3.js library. The interface will have a lot of texts. Therefore, several abbreviations is necessary and the description of the abbreviations are placed in this text box.

Entity Resolution can be separated into three different processes: Deduplication, Record Linkage, and Reference Matching. [2] This section will mainly related with the Deduplication and Record Linkage process. Deduplication is the task of “clustering the records or mentions that corresponds to the same entity.” [2] The interface will first import the data from csv files, and save them into javascript array files that are formatted as Javascript Object Notation (JSON) format. Then the records are separated into rows and paired or clustered into sets of three or four similar data.

The imported data files were in csv files: “cluster.csv” and “uncertain\_db.csv.” The “cluster.csv” file has reference information of the real dataset: cluster id, database number, record number and number that indicates that there are n number of similar datasets that will be clustered. The “uncertain\_db.csv” has the personal information of the records: first name, last name, race, gender, registration number, date of birth, and record number.

Once the database files are imported into javascript, the data files are saved into JSON formatted javascript arrays. The cluster database is used to find the records that will be clustered by references. While clustering the records, the centroid is randomly selected among the clustered data. Then the centroid is sent to a function called “getRow()” to pick out the information need to be displayed. After the centroid is selected, node records are selected by reference gradually. Then the node records are also sent to “getRow()” function.

The dataset for this project contains personal data: first name, last name, date of birth, registration number, gender, and ethnicity. Privacy protection is also a big challenge of this project because the data handled in this project are related with identification. The interface need to conceal most of the information. If two records have exactly same information, the interface hides all of the credentials to protect them. However, if two records contain similar information but slightly different, then the interface shows only parts of the credentials to determine the differences of two or more records that can belong to same entity. Furthermore, if the user cannot determine the identity of the two records by the given parts of the credentials, the interface provides few more information to the user to identify the two records.

After the csv file is saved into JSON format, the data gets sent to encode functions to hide the credentials. The program makes comparison of the rows first. After the centroid is selected, the node rows are compared to the centroid. Few different encoding functions needs to be made because the information for each column of the row was different. The “stringEncode()” function encodes first name and last name of clients. The “binaryEncode()” function encodes gender and race of the clients. The “dobEncode()” function encodes the date of birth of the clients and the “idEncode()” function encodes registration number of the clients.

The encoding functions does the second step of the Entity Resolution: Record Linkage. Record Linkage is a process of matching the data after deduplication and storing them to another location. [2] In this project, after encoding functions compare the node row to the centroid row, they unite into one row in order to be displayed.

The encoding functions compare two rows character by character, and if a character are identical, then the character gets replaced into “-” in the new row. If the two records have only one letter difference, the one character that has difference gets replaced into “D” in the new row. If two or more characters are different, the whole word gets replaced into “Diff.” If two letters are transposed, the two letters gets replaced into “TX.” The abbreviations are briefly explained in the instruction text box as explained in section 2.

The interface displays the data in table format because records in public health are easily readable in table format. The JSON data, encoded and united into one row, are pushed into another JSON data with database and record numbers and cluster id. The data is accessed one by one using a for-loop. Then the data is displayed into the table using d3.js library by using append method.

After the data gets encoded, the encode functions will lead to decode functions. This was the most challenging part of the project. It was not easy to change objects that are already displayed on the web. We resolved this problem by adding html id to every single object in the table cell. Then we used the method of getElementByID to change the object in the table cell. In that way we were able to open the information that were encoded.

1. User’s Side and Output

The last process of Entity Resolution, Reference Matching, is a task of cleaning up the records. [2] Reference matching in this project is to decide linking of the records. This is the part that the user gets involved. The interface has a few buttons: yes, no, and unknown. The user will examine the similarity of the two records, centroid and node, and decide if the records belong to the same entity. If the user think that the information provided belong to the same person, the user will click yes, and if the user think that the records belong to two different person, then the user will click no. If there are not enough information to decide, the user will click unknown.

Linking the records can have several ways to display the status. In this project we added one more column in the original data file. The linked status will start as unknown from the beginning. By clicking the button provided in the interface, the program will change the linked status in the database. However, we thought that saving the output data file every time a button is clicked would take too much memory space and time. Therefore, we decided to make the program have the dataset until another button is clicked.

We decided to make a save button to output the data when all processes were finished. Using javascript, the data we were able to output the data in same csv format as the input file.

4. Conclusion

Living in the world that demands more technology, we need to constantly develop a new technology and supplement the old ones. In this paper, we presented an Entity Resolution interface that can be used in Public Health area. The interface manages similar records that exist in Public Health institutions and match them to let the user decide whether the records belong or do not belong to the same entity. Our goal was to build this interface as a part of the PPIRL research, and we concluded that we could clearly build a working interface in this project.

# [1] [Margaret Rouse](http://www.techtarget.com/contributor/Margaret-Rouse), “User Interface (UI),” Tech Target, April 2005, Website, http://searchsoa.techtarget.com/definition/user-interface

# [2] Benjamin Bengfor, “[Entity Resolution for Big Data](http://www.datacommunitydc.org/blog/2013/08/entity-resolution-for-big-data),” Data Community DC, August 15, 2013, Website, http://www.datacommunitydc.org/blog/2013/08/entity-resolution-for-big-data/