Stage 4 Report

April 14, 2017

1 Stage 4: Combining Two Tables

Trang Ho, Thomas Ngo, Qinyuan Sun

1.1 Pipeline

We have two tables AOM and WHED with schema as following: - WHED(**a_id**, a_name, a_city, a_prov, a_country, a_web) - AOM(**person_id**, a_name, a_city, a_prov, a_country, a_email_server)

The AOM table contains information on affiliation name, city, province/state, country, and email server provided by conference attendants. Consequently, information on the AOM table may be incomplete or inaccurate. For example, conference attendants might provide affiliation name at the school level (e.g. Wisconsin School of Business) instead of at the university level (e.g. University of Wisconsin - Madison)

The WHED table contains standard information on name, city, province/state, country, website domain of affliations. Hence, we try to map each individual in AOM to an affiliation in WHED and keep the affliation information in WHED table when merging.

After stage 3, we have applied the entity matching to WHED and AOM tables to obtain a list of matching tuples for individuals and affliations in the US only. This is stored in matched_tuples.csv. To merge the two tables, we use the information in WHED as the anchor for affiliation. Hence, we keep the columns in WHED and remove all columns related to affiliation in AOM and obtain the table with the following schema: - MergedTable(**person_id**, a_id, a_name, a_city, a_prov, a_country, a_web)

1.2 Statistics of Merged Table

```
In [17]: import py_entitymatching as em
         MergedTable = em.read_csv_metadata('merged_tuples.csv', key = 'person_id')
         print("Number of tuples:", MergedTable.shape[0])
         print("Number of columns:", MergedTable.shape[1])
```

```
WARNING:py_entitymatching.io.parsers:Metadata file is not present in the given path; proceeding to read
```

```
Number of tuples: 3230
Number of columns: 7
```

```
In [12]: MergedTable.head(n = 4)
```

```
Out[12]:    a_id  person_id                          a_name       a_city    a_prov  \
         0    26       6378  abilene christian university      abilene     texas
         1    26      33444  abilene christian university      abilene     texas
         2   110       4676            adelphi university  garden city  new york
         3   110       8429            adelphi university  garden city  new york

                a_country                    a_web
         0  united states      http://www.acu.edu
         1  united states      http://www.acu.edu
         2  united states  http://www.adelphi.edu
         3  united states  http://www.adelphi.edu
```

1.3 Code for Merging

```
In [ ]: import py_entitymatching as em
        df = em.read_csv_metadata(’matched_tuples.csv’, key = ’_id’)
        # aom = em.read_csv_metadata(path_to_csv_dir + ’_aom.csv’, key = ’person_id’)
        # whed = em.read_csv_metadata(path_to_csv_dir + ’_whed.csv’, key = ’a_id’)
        # df.head()

        #use rename_col() to rename columns
        #use drop_cols() to drop merged colums
        # modify df to get the final results
        drop_list = [’rtable_a_name’,’rtable_a_city’,’rtable_a_prov’,’rtable_a_country’,’rtable_a_email
        df = em.drop_cols(df, drop_list)

        df = em.rename_col(df,’ltable_a_id’,’a_id’)
        df = em.rename_col(df,’ltable_a_name’,’a_name’)
        df = em.rename_col(df,’ltable_a_city’,’a_city’)
        df = em.rename_col(df,’ltable_a_prov’,’a_prov’)
        df = em.rename_col(df,’ltable_a_country’,’a_country’)
        df = em.rename_col(df,’ltable_a_web’,’a_web’)
        df = em.rename_col(df,’rtable_person_id’,’person_id’)


        # only one tuple in WHED should be matched to one tuple in aom.
        df_new = df.drop_duplicates(subset=[’person_id’], keep = False)
        em.set_key(df_new,’person_id’)
        df_new = em.drop_cols(df_new,’_id’)
        df_new.head(n = 5)
        df_new.to_csv(’merged_tuples.csv’, index=False)

In [ ]:
```