CS838 Project Stage 4 Report

Fangzhou Mu, Nafisah Islam, Meera George

I. Datasets

We combined samples of *tracks.csv* and *songs.csv*. The samples, *track_sample.csv* and *songs_sample.csv*, contain ***53935*** and ***100000*** tuples, respectively. ***32000*** tuple pairs survived after blocking and were stored in *pairs_passed.csv*. The matcher developed in Stage 3 was applied to the candidate pairs. The set of matches is stored in *matches.csv*. A final dataset, *merged_data.csv*, was created by merging matched tuples.

All datasets are available at Github Link.

II. Data merging

We conducted the following three steps.

(1) The matcher was applied to the candidate pairs and a set of matches was obtained.

(2) With respect to each match, the IDs were used to locate associated attribute values (i.e. *song_title, year, artists*) in both tables.

(3) The following rules were used when we created the final table.

(a) For *song_title* and *artists*, we selected ***the longer string*** from the corresponding two tuples, assuming that it is more comprehensive.
(b) For *year*, we selected ***the smaller value*** from the corresponding two tuples, assuming that it is more likely to be when the song was composed.
(c) When an attribute value is missing in one of the two matched tuples, we put the only value available (from the other tuple) in the final table.
(d) In case that an attribute value is missing in both tables, we left the value in the final table blank.
(e) Unique attributes in either table were simply carried over to the new table.

III. Statistics

The schema of the final table is ***Songs-Tracks****[movie_title,year,episode,song_title,artists]*. We may combine this table with *movies.csv* in Stage 5 to obtain another table for data analysis. The table contains ***7280*** tuples. Some sample tuples in the final dataset are shown below.

movie_title	**year**	**episode**	**Song_title**	**artists**
one tree hill	2007	running to stand still (#5.10)	only fooling myself	kate voegele
t in the park	2015	weekend highlights: part 4 (#1.64)	what became of the likely lads	the libertines
beavis and butt-head	1994	date bait (#4.18)	god of emptiness (from the album covenant)	morbid angel
la virgen de los sicarios	2000	NaN	el santo cachon	romualdo brito+los embajadores vallenatos with robinson damian

IV. Code

Code for data preprocessing, blocking and matcher development was completed in Stage 3. Code for data merging, *Merging.ipynb*, is available at Data Merging Link.

```
import pandas as pd
import os
import re

songs = pd.read_csv('dataset/songs_sample.csv')
tracks = pd.read_csv('dataset/tracks_sample.csv')
matchIDPairs = pd.read_csv('dataset/matches.csv')

# filtering the matched tuples from both dataset
matchedTracks = tracks[tracks['id'].isin(list(matchIDPairs['ltable_id']))]
matchedSongs = songs[songs['id'].isin(list(matchIDPairs['rtable_id']))]
```

```
import math

#Schema of the merged table
E = pd.DataFrame(columns = ['movie_title','year','episode','song_title','artists'])

for index, row in matchIDPairs.iterrows():
    left_entry = matchedTracks[matchedTracks['id']==row['ltable_id']]
    right_entry = matchedSongs[matchedSongs['id']==row['rtable_id']]

    assert(len(left_entry)==1)
    assert(len(right_entry)==1)

    track_id = int(left_entry['id'].item())
    song_id = int(right_entry['id'].item())

    if(math.isnan(left_entry['year'].item())):
        left = 0
    else:
        left = int(left_entry['year'].item())

    if(math.isnan(right_entry['year'].item())):
        right = 0
    else:
        right = int(right_entry['year'].item())

    if left >= right and left != 0:
        year = left
    else:
        year = right

    #for song title, larger length value is chosen if two value doesn't have exact string match
    left = str(left_entry['song_title'].item())
    right = str(right_entry['song_title'].item())

    if len(left) >= len(right):
        song_title = left
    else:
        song_title = right

    #for artist, larger length value is chosen if two value doesn't have exact string match
    left = str(left_entry['artists'].item())
    right = str(right_entry['artists'].item())

    if len(left) >= len(right):
        artists = left
    else:
        artists = right

    #since movie and episode are unique attributes in the left table, keeping the value as it is
    movie_title = str(left_entry['movie_title'].item())
    episode = str(left_entry['episode'].item())

    if episode == 'NaN':
        episode = ''

    #creating an entry for table E with all values
    entry = pd.Series([track_id, song_id, movie_title, year, episode, song_title, artists], index=
['track_id','song_id','movie_title','year','episode','song_title','artists'])

    #appending the merged value to table E
    E = E.append(entry, ignore_index=True)
```

```
#Writing the table E to file
E.to_csv('merged_data.csv',sep=',',index=False)
```