Assignment 3: Duplicate Detection

Deadline: 3.7.2018
Presentation: 5.7.2018[1]

[1]The 800 Pound Gorilla in the Corner: Data Integration
Technische Universität Berlin

Abstract. *The goal of this assignment is to detect duplicate tuples in a dataset, which is called duplicate detection.*
Notice that:

- *The short questions usually require short answers too.*
- *One submission per each group is enough.*
- *You are free to use whatever tools/programming languages you intend to.*
- *Put your names and group number in the beginning of your final PDF report.*
- *Submit a compressed file that contains (1) the PDF report and (2) your source code files.*

1. Setup

1.1. Dataset

We have a relational address dataset with the following schema:

$$S(RecID, FirstName, MiddleName, LastName, Address,\\ City, State, ZIP, POBox, POCityStateZip, SSN, DOB).$$

The dataset has been attached to this document.

1.2. Evaluation

To evaluate your duplicate detection task, you need to first generate the right output file. The output file must be a CSV with two columns that specify IDs of duplicate tuples. For example, output.csv is a possible output CSV file:

```
tuple_id_1, tuple_id_2
A917320, A928999
A957153, A989217
```

Based on this output file, we decide that tuples *A917320* and *A928999* are duplicate. We also detect tuples *A957153* and *A989217* as duplicates.

The actual pairs of duplicate tuples, which is the ground truth, is not available for you. However, you can always send your output file to our web service and get feedback to evaluate your duplicate detection process. To communicate to the web service, you can either write your own script or use the attached *web_client.py*. This piece of code simply sends your output file to the web service and gets the results. Below is a possible outcome:

```
$ python web_client.py
Duplicate Detection Results:
Precision = 1.0
Recall = 8.08979674386e-06
F1 = 1.61794625991e-05
```

To use the *web_client.py* script, notice that:

- You might have to install *Python 2.7* and the *requests* module.
- The script assumes that the name of your output file is *output.csv* and it exists in the same directory.
- The web service automatically ignores the *reflexive relation* (x is duplicate of x) and considers the *symmetric relation* (if x is duplicate of y, then y is also duplicate of x). However, it does not consider the *transitive relation* (if x is duplicate of y and y is duplicate of z, then x is duplicate of z). So, your algorithm should take care of the transitive relation.

2. Tasks

The main task is to detect all the duplicate tuples that refer to the same real world entities. Table 1 shows two possible duplicate tuples. As you can see, while most of the values in the tuples are the same, some values are different because of representation problems (such as ”ARKANSAS” and ”Ar”), typos (such as ”AGAUS” and ”AUGAS”), or missing values (such as ”1979” and ””).

Table 1. Example of duplicate tuples.

RecID	FirstName	MiddleName	LastName	Address	City	State	ZIP	POBox	POCityStateZip	SSN	DOB
A917320	BRIDGETTE	A	AGAUS	0054 Chicken Dr	DECATUR	ARKARNSAS	72722			198952594	1979
A928999	BRIDGETTE	A	AUGAS	20054 ychicken dr	Decatur	Ar	72m722			199852594	

2.1. Task 1: Brute-Force Duplicate Detection

Here, we want to compare all the possible pairs of tuples with each other.

1. Provide an algorithm. Specify the input, output, similarity function, and time complexity.
2. Implement the algorithm and report the precision, recall, F1, and runtime.
3. What is the upsides and downsides of this method?

2.2. Task 2: Partition-Based Duplicate Detection

Here, we want to reduce the time complexity of brute-force approach by dividing dataset into partitions. Thus, instead of comparing all the possible pairs of tuples in the whole dataset, we need to compare tuples just inside of their partition.

1. Provide an algorithm. Specify the input, output, similarity function, and time complexity.
2. Implement the algorithm and report the precision, recall, F1, and runtime.
3. What is the upsides and downsides of this method?