Australian
National
University

COMP3430 / COMP8430
Data wrangling

Lab 4: Comparison for Record Linkage

Objectives of this lab

- Today’s lab is the second in a series of five labs during which we will gradually build a complete record linkage system.

- We will be working with different comparison functions and learn how they work and why they are important in the RL process.

- Completion of the comparison module in the program.

Outline of this lab

- Learn how different comparison functions work
- Implement different comparison functions
- Evaluate different comparison functions
- Summary

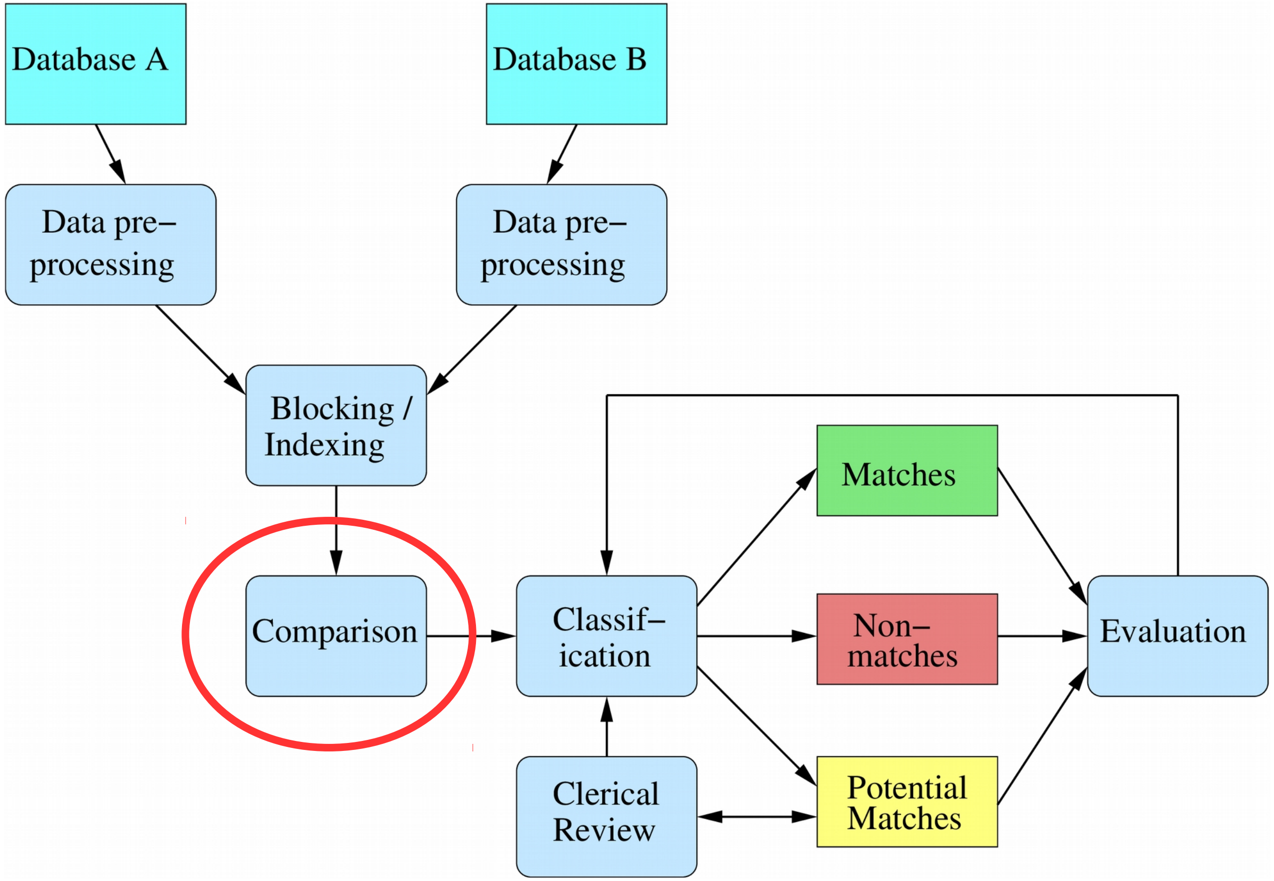
Preliminaries

- **Before you begin, aim to review lectures 15 and 16 if you have not already viewed them.**

- Go back over the work from lab 3 and remind yourself what we were doing and how the overall program is structured.

- You can download the blocking module with sample solutions in week 6 and use with your RL program if you find difficulties implementing the required two blocking functions.

What is Comparison?



- This week we focus on the next step in the linkage process, record pair comparison.
- The basic idea of a comparison function is to provide a numerical measurement of how similar two attribute values are.
- Why do you think comparison functions are important in the RL process?

How to compare values

- Before we begin let us see how Jaro comparison function works. The basic idea of this string comparison technique is outlined in lecture 16.

- See if you can compute the Jaro similarities between following value pairs.
  - jones / johnson
  - michelle / michael
  - shackleford / shackelford

- Also, see if you can compute the Jaccard and Dice similarities, and Bag distance for the above values.

Jaro Similarity

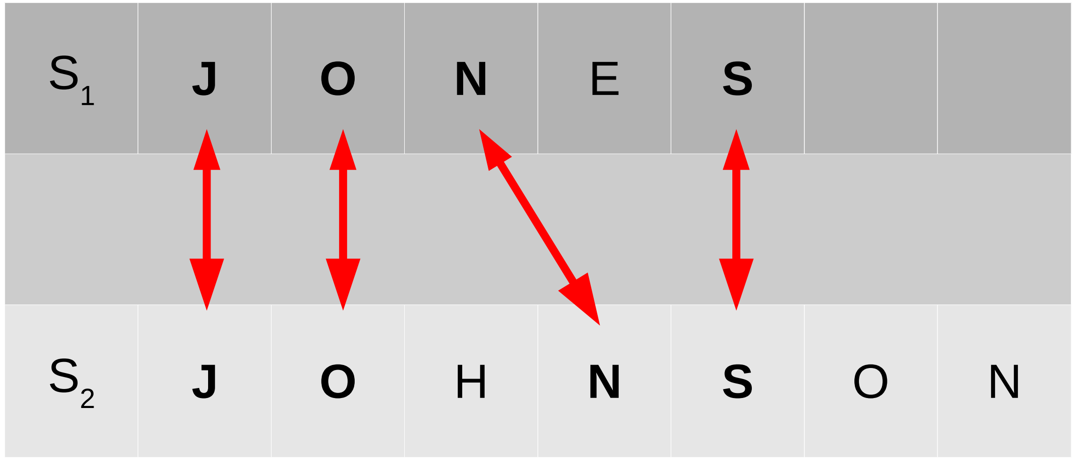
- Basic idea of the Jaro similarity between two strings S_1 and S_2
  - Count **c**, the number of agreeing (common) characters within half the length of the longer string
    - Search range = $(\max(|S_1|_{,} |S_2|) / 2) - 1$
  - Count **t**, the number of transposed characters (‘pe’ versus ‘ep’) in the set of common strings

$$\mathrm{Sim}_{\mathrm{Jaro}}(S_{1,} S_2) = 1/3 \text{ x } \Big(\mathbf{c}/|S_1| + \mathbf{c}/|S_2| + (\mathbf{c}\text{-}\mathbf{t})/\mathbf{c}\Big)$$

Jaro Similarity

$$\mathrm{Sim}_{\mathrm{Jaro}}(S_{1,}\ S_2) = 1/3 \times \left(\mathbf{c}/|S_1| + \mathbf{c}/|S_2| + (\mathbf{c}\text{-}\mathbf{t})/\mathbf{c}\right)$$

jones / johnson



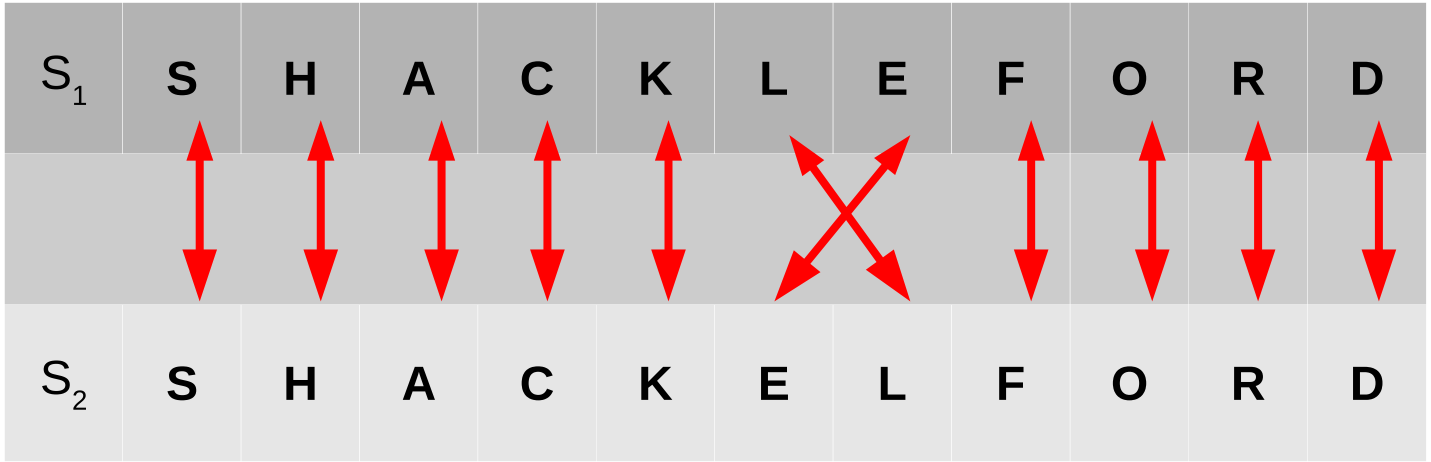
S_1	**J**	**O**	**N**	E	**S**		
S_2	**J**	**O**	H	**N**	**S**	O	N

c = 4
t = 0

SimJaro(S_1, S_2) = 1/3 x (4/5 + 4/7 + 4-0/4)

SimJaro(S_1, S_2) = 0.7905

shackleford / shackelford



S_1	**S**	**H**	**A**	**C**	**K**	**L**	**E**	**F**	**O**	**R**	**D**
S_2	**S**	**H**	**A**	**C**	**K**	**E**	**L**	**F**	**O**	**R**	**D**

c = 11
t = 1

SimJaro(S_1, S_2) = 1/3 x (11/11 + 11/11 + 11-1/11)

SimJaro(S_1, S_2) = 0.9697

Implement different comparison functions

- Now start looking at **comparison.py** and explore how the comparison functions work (inputs, return values, etc.).
- We have already provided two comparison functions, **exact_comp** and **jaro_comp.**

- Run the RL program using these comparison functions and see what the output looks like and how it performs**.**

- Now try to implement the other comparison functions as required in the lab tutorial document.

Questions to consider

- How does each comparison function compares to the others on different types of data (names, dates, postcodes, etc.)?

- Are they all equally fast or slow? See if you can measure the runtime of each comparison function with an example data set.

- Are there some problems / errors that none of these string comparison techniques can deal with?

- Extra tasks – see if you can implement the Winkler modification to the Jaro string comparison technique, as well as the edit distance function.

Summary

- In this lab we implemented different comparison functions and learnt how they can be used in the RL program.
- Make sure to complete any unfinished work in this module before you come to the next lab.
- In the next lab we will be looking at how different classification techniques work and how they can be used in the RL program.