Week 5: Comparing Strings

Nate Bachmeier

TIM-7241:Constructive Research Design

May 23, 2021

Northcentral University

# Comparing Strings

Micheal Jordan, M. Jordan, and M.J. are three names for the same person. Humans can quickly examine these variations and conclude this fact. Meanwhile, computers need specialized fuzzy string comparison algorithms to make the same deductions. These entity name comparison algorithms focus on either name variations or misspellings. Gong, Wang, and Oard (2009) propose a hybrid solution that addresses both aspects (see Figure 1).

Figure 1: High-level Process

Their process imports and tokenizes names into “names” and “name formats.” Next, they use a directed graph to represent all permutations. Weighted edges between the nodes quantify the similarity to the original text. This configuration enables Dijkstra’s shortest path algorithm to find the right combination for fuzzy comparisons. Finally, the comparison determines if the match candidates are equivalent (Boolean value).

## Success Criterion

The process’s core use-case determines if two names are similar, regardless of spelling or variations (e.g., initials). Therefore, Gong, Wang, and Oard (2009) must maximize the solution’s recall and precision by minimizing false positives or negatives.

## Measuring Correctness

Previous researcher publications chose F-measurement as the

## Measuring Efficiency

## Test Cases