Bikal Basnet

[Email address]

Abstract

This report entails the entity matching system for SkyGeek. The approach will equally be suitable in case where similarity matching will be most appropriate

Entity Matching for Skygeek

Proof of Concept

Contents

[Introduction: 2](#_Toc405293045)

[Problem Scope: 2](#_Toc405293046)

[Objective: 3](#_Toc405293047)

[Data size: 3](#_Toc405293048)

[Analysis: 3](#_Toc405293049)

[1. Levhenstein 3](#_Toc405293050)

[2. Jacquard Similarity: 3](#_Toc405293051)

[3. JaroWrinkler: 4](#_Toc405293052)

[4. LongestCommonSubSequence: 4](#_Toc405293053)

[Results: 4](#_Toc405293054)

[Conclusion: 5](#_Toc405293055)

[Limitation 5](#_Toc405293056)

# Introduction:

This report focusses on Entity Matching for individual SkyGeek customers and proposes the best algorithm for the Entity Matching, based upon the analysis performed on four data scenario cases, using four different algorithms.

The Entity matching is done using the Customers data from the SEOM database.

# Problem Scope:

We have customers with mistyped spelling, missing middle name for their Full name. We want to identify similar or mistyped customer names to the customer names present in the database, as exactly as possible.

For the particular case, we have analyzed four typical data scenarios and have benchmarked against the algorithms

Scenario 1. Aruce Smit

Scenario 2. Aruce Smith

Scenario 3: Bruce Bahadur Smith

Scenario 4: Arude Smith

All of the scenario names, are names mutated from “Bruce Smith” from the Customers database.

# Objective:

The main objective of the research is to match the mutated names with the most similar name in the customer database.

For the particular case we want to match, all of them to Bruce Smith as exactly as possible (i.e. Expected result).

# Data size:

For the research, user names from “Customer Table of SEOM database” were used.

# Analysis:

For the analysis four algorithms were used.

1. Levhenstein:

It measures the difference between tow sequences of strings, where the distance between two strings is defined by the minimum number of single character edits (insertion, deletion or substitution) that has to be performed to change a word into another word.

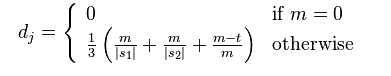
1. Jacquard Similarity:

It measures the similarity between two strings using statistical approach i.e. number of common characters to total number of characters contained in both set of words i.e.



1. JaroWrinkler:

It was developed in 1990 by Winkler and uses complex mathematical formulae to calculate the string similarity.



Where s1 is word 1

S2 is word 2

M is the number of matching characters

And t is the number or transpositions

1. LongestCommonSubSequence:

It measures similarity by finding the longest subsequence of characters t common to both words.

# Results:

All four algorithms were analyzed against the data scenarios, for the most appropriate match and the result set has been presented in excel sheet attached herewith the document.

# Conclusion:

Based on the analysis, Levhenstein was observed to perform significantly better for all the four data scenario cases and is recommended.

# Limitation

However this should not be used as basis to concluded that, Levhenstein is suited in all the scenarios.

Depending on the data scenarios, that is to be compared and across different scenarios, different algorithms may perform better.

Should data scenario change, it is highly advised to run benchmarking across all four algorithms to find the most appropriate ones

# Reproducibility:

MDs database is a product from Microsoft shipped with SQL server that provides enhancesd analytical capabilities, table based integration and other additional support for Microsoft products.

SSIs also uses the capabilities from it to perform SSIS operations.

REF:

1. http://social.technet.microsoft.com/wiki/contents/articles/5648.whats-new-in-master-data-services-mds-in-sql-server-2012.aspx

<http://web.archive.org/web/20070626030519/http://blogs.technet.com/patricg/archive/2007/09/19/microsoft-master-data-management.aspx>

STEP 1 : Install MDs database

Search for master data services confiduration manager via search .

Click on Database Confiduarion and click create database.

Now you are ready to use the Mds database faciiliteis as fuzzy grouping etc.

STEP 2 : Now execute the following scripts

1. fn\_LevenshteinValue
2. fn\_ScalarMinValue
3. spa\_LevenshteinCalculator