Goma Documentation

Release 0.2 [4 - Beta]

sonntagsgesicht, based on a fork of Deutsche Postbank [pbrisk

Wednesday, 18 September 2019

Contents

CHAPTER 1

Introduction

1.1 Python library *goma*

goma - generic object mapping algorithm

A library for easy implementing complex mappings procedures.

1.2 Development Version

The latest development version can be installed directly from GitHub:

```
$ pip install --upgrade git+https://github.com/sonntagsgesicht/goma.git
```

1.3 Contributions

Issues and Pull Requests are always welcome.

1.4 License

CHAPTER 2

Tutorial

2.1 First setup basic objects

For a simple example setup imports and generate a nested list of match details.

```
>>> from goma.exactmatch import ExactMatch
>>> detail_list = [["Property1","Value1_2"],["Property2","Value2_2"],["Property3",
↪"Value3_2"]]
```

And define a mapping list as instruction for the mapping algorithm

```
>>> mapping_list = list()
>>> mapping_list.append(["Property1","Property2","Property3","Target"])
>>> mapping_list.append(["Value1_1","Value2_1","Value3_1","Target1"])
>>> mapping_list.append(["Value1_2","Value2_2","Value3_2","Target2"])
```

Finally generate an `ExactMatch` instance and run the matching algorithm

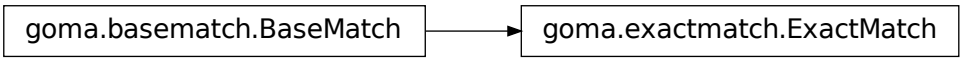
```
>>> exact_match = ExactMatch()
>>> exact_match.match(detail_list, mapping_list)
'Target2'
```

CHAPTER 3

API Documentation

3.1 ExactMatch

ExactMatch



class goma.exactmatch.**ExactMatch**
Bases: goma.basematch.BaseMatch

match(*match_details*, *mapping_list*)

matching based on exact entries of all columns of the mapping list

Parameters

- **match_details** (*list*) – holds the information based on which the mapping should be conducted, a row entry is structured as [‘Detail’, ‘Value’]
- **mapping_list** (*list*) – holds the mapping information, the first row describes the properties on which mapping should be conducted and a column named target
- **start_col** (*int*) – starting column for the matching algorithms of the mapping_list

The exact match uses a given mapping_list, e.g

Property1	Property2	Property3	Target
Value1_1	Value2_1	Value3_1	Target1
Value1_2	Value2_2	Value3_2	Target2

Given the above tables, the matching searches row by row, if all criteria in the matching list of a given list of details (match_details) are met.

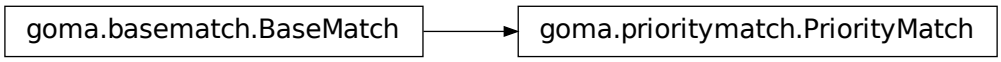
For a given list of match_details, e.g.

Property1	Value1_2
Property2	Value2_2
Property3	Value3_2

the match returns in a Target 2. If one matching criteria is not met, the match returns None.

3.2 PriorityMatch

PriorityMatch	Priority Match class



class goma.prioritymatch.**PriorityMatch**

Bases: goma.basematch.BaseMatch

Priority Match class

match(*match_details*, *mapping_list*, *start_col=0*)

matching based on prioritizing entries in the right column of the mapping list

Parameters

- **match_details** (*list*) – holds the information based on which the mapping should be conducted, a row entry is structured as [Detail ,Value]
- **mapping_list** (*list*) – hold the mapping information, the first row describes the properties on which mapping should be conducted and a column named target
- **start_col** (*int*) – starting column for the matching algorithms of the mapping_list

Returns target value which has been matched

Return type string

The priority match uses a given mapping_list, e.g

Property1	Property2	Property3	Target
Value1_1		Value3_1	Target1
Value1_1		Value3_2	Target2
	Value2_1	Value3_3	Target3
	Value2_1	Value3_4	Target4

Given the above tables, the matching searches column by column, starting on the left hand side (highest priority columns), for the respective mapping list.

For a given list of match_details, e.g.

Property1	Value1_2
Property2	Value2_1
Property3	Value3_3

the match returns in a first step a 2x4 matrix

	Value2_1	Value3_3	Target3
	Value2_1	Value3_4	Target4

As a second step, the empty columns are removed and the exact match algorithm is applied to the remaining properties to find the target, which for the given example yields Target3.

If for column no matching is found it continues recursively at the next column and searches for the respective property, i.e Property3.

CHAPTER 4

Releases

These changes are listed in decreasing version number order.

4.1 Release 0.2

Release date was Wednesday, 18 September 2019

4.2 Release 0.1

Release date was July 7th, 2017

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

g

Index