Match

Overview

The Match tool will look at the numbers generated by Score and compare them to a threshold value. Input entities compared by Score that meet or exceed the threshold will have their identities changed to the URI of the person in VIVO, so that when the data is finally pulled into VIVO the new data will be linked to existing data. In this way you can fetch publications for your existing researchers.Match.java takes a model generated by Score and renames matches, creates links, or removes literals based on the associated scores.

Match uses a *threshold* value, which is compared with the values for each individual produced by Score to determine if action should be taken. If the Score value equals or exceeds the threshold, then action is taken. Typically this action is a rename. A rename means that the matched input is given the same URI as the individual in VIVO matching it, which identifies the two as the same entity. This can then be merged into VIVO. In this way an existing individual in VIVO can have more data added to it. For example, a publication can have a previously missing publisher added to it.

Match config

Match - Uses the score-data produced by Score to evaluate pairs of records from the input and VIVO.

Records that match above a user-specified threshold are considered to be a matching set. Matching records from the input can be renamed to the matching VIVO URI, or can be linked to them via user-specified triples. Additionally, the input data can be sanitized to remove all rdf:type statements and all literal value statements, preserving only the node relationship triples.

Match Parameters

wordiness - (optional) sets the lowest level of log messages to be displayed to the console. The lower the log level, the more detailed the messages.

Possible Values:

- <Param name="wordiness">OFF</Param> - Results in no messages being displayed.
- <Param name="wordiness">ERROR</Param> - Results in only messages from the ERROR level to be displayed. Error messages detail when the tool has experienced an error preventing it from completing its task
- <Param name="wordiness">WARN</Param> - Results in only messages above and including WARN level messages to be displayed. Match does not produce any WARN level messages.
- <Param name="wordiness">INFO</Param> - (Default) Results in all messages above and including INFO level messages to be displayed. INFO level messages detail when the tool has started and ended and when it begins/ends a phase ('Finding matches' and 'Beginning Rename of matches') and how many matches have been found.
- <Param name="wordiness">DEBUG</Param> - Results in all messages above and including DEBUG level messages to be displayed. DEBUG level messages detail each matching input URI to its VIVO URI as they are processed. Additionally, it will display stacktrace information if an error occurs.
- <Param name="wordiness">ALL or TRACE</Param> - Results in all messages above and including TRACE level messages to be displayed, since trace is the lowest level it is the same as ALL in practice. TRACE level messages details every matching set as it is processed in each phase along with SPARQL queries and start and stop for their execution.

threshold - match records with a cumulative weighted-score over this value

Example:

- <Param name="threshold">0.5</Param> - match things whose total weighted-score is greater than 0.5

link - (optional) link the two matched entities together using two provided predicates

Example:

- <Param name="link">http://xmlns.com/foaf/0.1/made=http://xmlns.com/foaf/0.1/maker</Param> - links a foaf:Agent from the VIVO model to a foaf:Thing in the input model by adding linking triples (<vivo-agent> <foaf:made> <input-thing>) and (<input-thing> <faof:maker> <vivo-agent>) to the input

rename - (optional) rename the input node to the matching VIVO URI

Possible Value:

- (default) <Param name="rename">false</Param> - don't rename nodes
- <Param name="rename">true</Param> - rename nodes

clear-type-and-literals - (optional) sanitize the input data to remove all rdf:type statements and all literal value statements, preserving only the node relationship triples

Example:

- (default) <Param name="clear-type-and-literals">false</Param> - don't sanitize input data
- <Param name="clear-type-and-literals">true</Param> - sanitize input data

batch-size - (optional) number of records to process in batch - default 150 - lower this if getting StackOverflow or OutOfMemory Exceptions

Example:

- (default) <Param name="batch-size">150</Param> - process 150 records per batch
- <Param name="batch-size">50</Param> - process 50 records per batch

input-config - (optional - at least one of this and/or inputOverride) the configuration file that describes the input jena model. The parameters for this config file are described in the Models section below.

Example:

- <Param name="input-config">/absolute/path/to/file.conf.xml</Param> - An absolute path to a jena model config file on linux/unix/macosx operating systems
- <Param name="input-config">C:/absolute/path/to/file.conf.xml</Param> - An absolute path to a jena model config file on a windows operating system
- <Param name="input-config">relative/path/to/file.conf.xml</Param> - A path to a jena model config file that is relative to the folder the shell was in when this command was executed

inputOverride - (optional - at least one of this and/or input-config) specify the parameters for the jena model without a config file and/or override specific parameters from the given config file. The parameters that can be set/overridden are described in the Models section below.

Example:

- <Param name="inputOverride">paramName=valueToUse</Param>

score-config - (optional - at least one of this and/or scoreOverride) the configuration file that describes the score jena model. The parameters for this config file are described in the Models section below.

Example:

- <Param name="score-config">/absolute/path/to/file.conf.xml</Param> - An absolute path to a jena model config file on linux/unix/macosx operating systems
- <Param name="score-config">C:/absolute/path/to/file.conf.xml</Param> - An absolute path to a jena model config file on a windows operating system
- <Param name="score-config">relative/path/to/file.conf.xml</Param> - A path to a jena model config file that is relative to the folder the shell was in when this command was executed

scoreOverride - (optional - at least one of this and/or score-config) specify the parameters for the jena model without a config file and/or override specific parameters from the given config file. The parameters that can be set/overridden are described in the Models section below.

Example:

- <Param name="scoreOverride">paramName=valueToUse</Param>

output-config - (optional - will contain all related nodes for matches) the configuration file that describes the output jena model. The parameters for this config file are described in the Models section below.

Example:

- <Param name="output-config">/absolute/path/to/file.conf.xml</Param> - An absolute path to a jena model config file on linux/unix/macosx operating systems
- <Param name="output-config">C:/absolute/path/to/file.conf.xml</Param> - An absolute path to a jena model config file on a windows operating system
- <Param name="output-config">relative/path/to/file.conf.xml</Param> - A path to a jena model config file that is relative to the folder the shell was in when this command was executed

outputOverride - (optional - will contain all related nodes for matches) specify the parameters for the jena model without a config file and/or override specific parameters from the given config file. The parameters that can be set/overridden are described in the Models section below.

Example:

- <Param name="outputOverride">paramName=valueToUse</Param>

Arguments

Short Option	Long Option	Parameter Value Map	Description	Required
i	inputJena-config	CONFIG_FILE	inputJena JENA configuration filename	true
I	inputOverride	override the JENA_PARAM of inputJena jena model config using VALUE	false	
o	output-config	CONFIG_FILE	outputConfig JENA configuration filename	true
V	vivoOverride	override the JENA_PARAM of vivoJena jena model config using VALUE	false	
s	score-config	CONFIG_FILE	score data JENA configuration filename	true

S	scoreOverride	override the JENA_PARAM of score jena model config using VALUE	false	
t	threshold	THRESHOLD	match records with a score over THRESHOLD	true
l	link	link the two matched entities together using INPUT_TO_VIVO_PREDICATE and INPUT_TO_VIVO_PREDICATE	false	
r	rename		rename or remove the matched entity from scoring	false
c	clear-type-and-literals		clear all rdf:type and literal values out of the nodes matched	false

Usage

```
//from the env file
Match="java $OPTS -Dprocess-task=Match org.vivoweb.harvester.score.Match"

//from the script file
SCOREINPUT="-i $H2MODEL -ImodelName=$MODELNAME -IdbUrl=$MODELDBURL -IcheckEmpty=$CHECKEMPTY"
SCOREDATA="-s $H2MODEL -SmodelName=$SCOREDATANAME -SdbUrl=$SCOREDATADBURL -ScheckEmpty=$CHECKEMPTY"
MATCHOUTPUT="-o $H2MODEL -OmodelName=$MATCHEDNAME -OdbUrl=$MATCHEDDBURL -OcheckEmpty=$CHECKEMPTY"
MATCHTHRESHOLD = 1.0

$Match $SCOREINPUT $SCOREDATA $MATCHOUTPUT -t $MATCHTHRESHOLD -r -c
```

This call in the scripts will rename entries in `$SCOREINPUT` according to `$SCOREDATA` which is weighted above `$MATCHTHRESHOLD` while clearing literals and types and sending a copy of the matched pieces to `$MATCHOUTPUT`.
Scripts like pubmed which is interested in only matches will use the match output.

Methods

match

Find all nodes in the given namepsace matching on the given predicates

1. Create a querystring on the scoredata to get the results with scores greater or equal to the threshold.
2. Execute the query with the input URI in sInput and the vivo URI in sVivo
3. For each solution to the query
    a. Create string of the sInput
    b. Create String of the sVIVO
    c. Pair up the sInput URI and the sVivo URI in a uriMatchMap
4. Return the the uriMatchMap

rename

Rename the resource set as the key to the value matched

1. For each entry in a <string,string> map
    a. Get the resource related to the old URI
    b. Get the new URI from the map.
    c. Rename the old resource to the new resource

link

Link matched scoreResources to vivoResources using given linking predicates

1. Create vivoToInput property
2. Create inputToVivo property
3. For each uri in the match set
    a. Get the resource for the input URI
    b. Get the resource for the vivo URI
    c. Add the inputToVivo property as applied to the resources
    d. Add the vivoToInput property as applied to the resources

Sparql

The match class runs a sparql query on the score data. This can help access the score data for other purposes.

```
?sInput = Input URI
?sVivo  = Vivo URI

PREFIX scoreValue: <http://vivoweb.org/harvester/scoreValue/>
SELECT DISTINCT ?sVivo ?sInput (sum(?weightValue) AS ?sum)
WHERE {
  ?s scoreValue:InputRes ?sInput .
  ?s scoreValue:VivoRes ?sVivo .
  ?s scoreValue:hasScoreValue ?value .
  ?value scoreValue:WeightedScore ?weightValue .
}
GROUP BY ?sVivo ?sInput
HAVING (?sum >= threshold )
ORDER BY ?sInput
```