Introduction

The Term Matching tool is an application that identifies matches between a set of source terms and a specific terminology. For each source term, the top potential matches are provided so that the user can determine which is most applicable.

How it works

The application relies upon a running instance of the West Coast Informatics Terminology Server. This instance must contain the desired version of the target terminology to match against. The user must also create a text file containing the source terms that are to be matched. At this point, the application is ready to be run.

Once launched, the application will go through each line of the input file and use it to query the terminology server for the best match. The terminology server will return the top matches as determined by its analysis. The source term and the top matches are added to an output Excel file for the user to be able to review.

For each potential match, a score is provided to indicate how closely the source term matches the potential match. The score may be between 0 and 1 with 1 being a perfect match. The first result is always the top score, with following results having either the same or lower scores.

Inputs

When running the application, the user must specify several attributes. All attributes can be pre-defined in the application such that the user need not provide their values per launch of the application. This is done via the pom.xml file's properties. Alternatively, the values can be defined during execution by using Java variables while launching the application via command line e.g.-Dversion=latest -DmaxCount=2. See Example-B below.

There are not any default values provided in the pom.xml file.

- **run.config**: This is the file that is used to specify the terminology server instance to query against. An example value for this attribute is: "C:\projects\termmatcher\config.properties".

- **terminology**: This is the target terminology name which is to be queried. The pair comprised of this attribute and the related *version* attribute must exist in the instance specified in the *run.config*. An example value for this attribute is "RxNorm".

- **version**: This is the version of the terminology which is to be queried. The pair comprised of this attribute and the related *terminology* attribute must exist in the instance specified in the *run.config*. An example value for this attribute is "12032018" or "latest".

- **maxCount**: This is used to specify the maximum number of matches to return per source term. If maxCount is blank or zero, all potential matches will be returned. The number of results returned will not exceed the number specified *except* for the case where there are more potential matches with identical scores than requested via this attribute. This is illustrated in Example-B below. An example value for this attribute is “3”.

- **searchTermsFilepath:** This is the path and name of the file containing the list of each source term to match against. The file format is that each line contains exactly one description. If the description contains multiple words, they must be listed on the same line (where a line is ended with a line delimiter). An example value for this attribute is “inputFile.txt” whose contents are:

  ```
  Acetaminophen
  Midol
  Diclofenac
  ```

- **userName**: This is a valid user that has viewer credentials in the term server. Note in some instances, the user must be logged into the server for the application to complete successfully.

- **userPassword**: This is the password associated with the *userName*.

Output

The results will be saved to an Excel file. The Excel file will be created in the same directory from where the application is launched. The naming convention for the Excel file is: matcherOutput-<terminology>-<timestamp>.xls. For example, if the file was launched against RxNorm on Jan 1st at 5:24 pm, the output will be named: matcherOutput-RxNorm-Jan 01 17-24.xls

The output file’s first line consists of the column header. The column headers are:

- Search Term
- Concept Id
- Concept Description
- Score

For every search term encountered, a line will be added with the search term printed in the first column.

This will be followed with one line per potential match with the potential match’s concept Id, description, and score populating the 2nd, 3rd and 4th column respectively. Therefore, there will not be any content in the first column of a potential match.

Once all potential matches for the current search term have been added, the next search term is processed. See the examples below for example output.

Examples

Example-A

The below example will launch the application using only those parameters specified in the pom file. In this case, maxCount is 7, and the input file has a search term defined (wheat).

Call

```
> mvn install -Pterm-match
```

Output File Contents

```
Search Term   Concept Id   Concept Description                                   Score
wheat
              852311       wheat rust extract                                    0.08286375
              1012140      wheat bran allergenic extract                         0.08286375
              HT1044       Urofolli TO wheat                                     0.08286375
              HT1045       wheat TO Zuclopen                                     0.08286375
              852314       wheat rust extract 50 MG/ML Injectable Solution       0.07697423
              1012143      wheat bran allergenic extract 50 MG/ML Injectable Solution   0.07697423
              1012166      wheat bran allergenic extract 100 MG/ML Injectable Solution  0.07697423
```

Example-B

In this example, the application will override the input file location and the max count specified in the pom file. Note although maxCount is set to 3, there are four matches identified that have an identical score, and so all are included in the output.

Call

```
> mvn install -DsearchTermsFilepath=snomedSourceFile.txt -DmaxCount=3 -Pterm-match
```

Output File Contents

```
Search Term   Concept Id   Concept Description                                   Score
wheat
              852311       wheat rust extract                                    0.08286375
              1012140      wheat bran allergenic extract                         0.08286375
              HT1044       Urofolli TO wheat                                     0.08286375
              HT1045       wheat TO Zuclopen                                     0.08286375
```