How To...
enable
semantic
search/search
for synonyms
in TREX

Version 1.00 – May 2005

Applicable Releases:
SAP NetWeaver '04
1 Scenario

Many customers would like to have a semantic search that allows to search not only for a term itself but also for synonyms. For example, if you search for ‘aspirin’, you would like to obtain a search result that also contains all documents dealing with ‘acetylsalicylic acid’.

This behavior can be configured in TREX, and your search result in KMC can be enriched by the synonym search. This guide describes this configuration in TREX.

Please note: There is no KMC iView that supports this scenario additionally with dedicated navigation. This means that you cannot choose from KMC which synonyms should be taken into account.

2 Introduction

The additional functionality of using synonyms as search terms in TREX is realized using topic maps. These topic maps are generated by an administrator and cannot be changed dynamically. Instead they have to be generated again each time you change them. TREX provides a python script to achieve this. Topic maps can be used within a specific index or for a TREX installation in general. TREX provides this function in a python extension that has to be activated and configured.
3 The Step-By-Step Solution

The following procedure contains 2 main steps:

- Creating a topic map
- Making the Python extension provided by TREX use it

Prerequisite: You have installed TREX with python support (ActivePython is part of the standard installation)

3.1 Create a topic map

1. Create a directory for your scenario.
   <TREX_DIR>/topicmaps/example/

2. In the directory
   <TREX_DIR>/topicmaps/example/, create the file example.txt. The content is a list of synonyms separated by semicolons.
3. Create a topic map file from example.txt. This step is necessary each time you change example.txt.

Go to the directory `<TREX_DIR>` and open a command prompt. Always execute TREXSettings first to set the environment variables dynamically.

Enter the following command:

```
\extensions\xtm\txt2xtm.py --input=<TREX_DIR>/topicmaps/example/example.txt --output=<TREX_DIR>/topicmaps/example/example.xtm
```

This creates the file `example.xtm` in your directory.

Now create `example.db` from this file, using the following command:

```
<TREX_DIR>/extensions/xtm/xtm2db.py <TREX_DIR>/topicmaps/example/example.xtm <TREX_DIR>/topicmaps/example/example.db
```
4. Create the file topicmap.ini in your directory. Create the following content in the file:

```
[topicmap_general]
tmcfile=example.db
stripZeroweight=1
AssocDefaultweight=1.0
RoleDefaultweight=1.0

# associations for semantic search: default is synonyms
[semanticsearch_assocs]
at-synset=1

# association synonym with role synonym
[semanticsearch_at-synset]
rt-synonym=1
```

5. Modify the file `<TREX_DIR>\TREXTopicmaps.ini` by entering the new topic map:

```
[topicmap_example]
Directory = topicmap/example
```

6. Activate the TREX extensions: Edit the file `<TREX_DIR>\TREXExtensions.ini`. Go to the section [extensionhandlers] and uncomment trexxpy.
7. Modify `<TREX_DIR>/extensions.py` by changing the following sections:
   - Topic Map extension
   - Semantic Search

Changes you have to carry out are marked in bold in the screenshots:

8. Restart TREX Indexserver

   If you have started TREX as a Windows service, restart the service. You can also start and stop TREX servers individually. See the documentation at [http://help.sap.com Knowledge Management – Administration Guide – Technical Operation in Detail – Starting and Stopping TREX.](http://help.sap.com)
### 3.2 Test your semantic search example

This test scenario is based on python scripts. If you would like to test using KMC, it will also work. The steps are listed in a generic way, followed by a description of the python scripts you might use.

1. Create documents containing in the body or title the terms that you have used in your topic map.
   Store them in a directory on your TREX server, for example, `test_semantic_search`

2. Create an index.
   Open a command prompt in `<TREX_DIR>` and use the python script `createIndex.py`:
   ```
   $ python CreateIndex.py semantic_test
   ```

3. Index the documents you have prepared for testing.
   Remain in `<TREX_DIR>` and use `indexDirectory.py`:
   ```
   $ python indexDirectory.py --directory=<directory>
   ```
   ```
   $ python indexDirectory.py --directory=C:/usr/sap/trex_03/test_semantic_search semantic_test
   ```
4. Search for a term that you have used in your topic map.

Use `search.py --query=<searchterm> <index>`

Search.py --query=cocktail semantic_test

The search results also contain documents containing appetizer, since it was given as a synonym in example.db

If problems occur, check
`<TREX_DIR>/trace/PythonExtensions.log`. 
4 Appendix

If you want to activate the semantic search only for a certain index, copy it to the index directory
<TREX_DIR>\index\semantic_test