cTAKES 3.0 - Dictionary Lookup

{scrollbar} 65%Contents of this Page2 Menu cTAKES 3.0 to Include

Overview of Dictionary Lookup

The dictionary lookup annotator finds the entries from one or more dictionaries that match the document text in some way. Within this annotator, these matches are called lookup hits.

The dictionary lookup annotator is very customizable. It can look for matches where the words in the dictionary entries appear in the same order as the words in the document text, or it can look for permutations of the words from the dictionary. Moreover, it can look just for exact matches of the words, or it can also look for matches to the canonical forms of the words.

Searches for a lookup hit are limited to within windows, where the window type is defined in the LookupDescriptorFile. A window can be the words that fall within the same Sentence, the same Chunk, the same LookupWindowAnnotation or any other annotation. See the clinical documents pipeline project for an example of an analysis engine (LookupWindowAnnotator.xml) that creates LookupWindowAnnotations.

Implementation of Dictionary Lookup

Starting with version 1.3, cTAKES includes UMLS (SNOMED CT and RxNorm) dictionaries. To use those dictionaries, you must have a UMLS username and password, and an internet connection (to verify your UMLS username and password). If you do not have a UMLS username and /or are not interested in those dictionaries, you can build your own or use the small sample dictionaries (see below).

The behavior of the dictionary lookup annotator is controlled by the parameters and resources defined in the analysis engine descriptor, and by the contents of the resource called the LookupDescriptorFile.

For example, if the analysis engine descriptor DictionaryLookup.xml contains a resource named LookupDescriptorFile with value lookup /LookupDesc.xml, then the parameter settings and resources named within DictionaryLookup.xml, together with the values within lookup /LookupDesc.xml will control the actions of the dictionary lookup annotator.

The lookupInitializer and lookupConsumer classes are specified within the LookupDescriptorFile. The algorithm used for looking up the terms is defined by the lookupInitializer, which creates the lookup hits. The lookupConsumer adds annotations to the CAS for some or all of the lookup hits.

An example of adding only some of the lookup hits to the CAS is if you have a dictionary of RxNorm terms with their RxNorm codes, and a dictionary of terms from the OrangeBook, and want to create annotations for those terms that are in the OrangeBook that also have an RxNorm code.

This can be done using class org.apache.ctakes.dictionary.lookup.ae.FirstTokenPermLookupInitializerImpl as the lookupInitializer, and using class OrangeBookFilterConsumerImpl as the lookupConsumer, provided you have the RxNorm dictionary, and you configure the LookupDescriptorFile resource to use your RxNorm dictionary.

Dictionary entries need to have been tokenized the way the pipeline tokenizes the document text. For example, the lookup algorithm will not find a lookup hit if a dictionary entry is "ear, skin" but the document text contains the same text ("ear, skin") and the pipeline has tokenized that text as the three tokens "ear" "," "skin". To find a lookup hit for the three tokens, the dictionary entry should be tokenized, with a space before the comma: "ear , skin".
Editing dictionary lookup AE descriptors in Eclipse
The analysis engine descriptors for this annotator use elements of type configurableDataResourceSpecifier. These cannot be modified from the Parameters or Resources tabs of the Component Descriptor Editor (at least not in UIMA 2.2). To view these values or edit them, use the Sources tab or open the descriptor with a text editor.

To determine the LookupDescriptorFile for an analysis engine, open the analysis engine descriptor (e.g. DictionaryLookupannotator.xml) and note the URL for the LookupDescriptorFile resource (e.g. lookup/LookupDesc.xml).

A LookupDescriptorFile such as lookup/LookupDesc.xml, found in resources/, defines the dictionary(s) used, and the classes that interact with the dictionary(s). The implementation tag identifies the type of dictionary: Lucene index (luceneImpl), database (jdbcImpl), or delimited flat file (csvImpl). See class org.apache.ctakes.dictionary.lookup.ae.LookupParseUtilities.java for implementation details.

To better understand the dictionary lookup annotator code you could start by reading the Javadoc API for the classes DictionaryLookupAnnotator. java and FirstTokenPermutationImpl.java.'

DictionaryLookupAnnotatorUMLS.xml

This uses the bundled UMLS (SNOMED CT and RxNorm) dictionaries. Before using this analysis engine descriptor, update the UMLSUser and UMLSPW parameters within this descriptor with your UMLS username and password. You will need to have an active connection to the internet so your UMLS username and password can be verified.

DictionaryLookupAnnotator.xml

This uses the small sample dictionaries. This annotator can be run out-of-the-box without modifying any parameters, but annotates a very limited set of terms such as carcinoma, aspirin, knee, and pain.

DictionaryLookupannotatorCSV.xml

This is an example of how to use a dictionary contained in a delimited file rather than a database or a Lucene index. This is only recommended for small dictionaries.

DictionaryLookupannotatorDB.xml

This is a skeleton of how you could use a dictionary contained in a database that can be accessed via a JDBC driver instead of using a Lucene index or flat file.

Sample dictionaries

This project includes two sample dictionaries that are used by default:

(1) a sample database (a Lucene index) containing a few drug names

(2) a sample database (using 2 Lucene indexes) containing a few anatomical sites, procedures, and disorders/diseases

These can be used to verify your cTAKES install and to give a small flavor of what cTAKES can do, and unlike the bundled UMLS dictionaries, do not require a UMLS username or an internet connection.

The programs used to create these Lucene indexes are scripts/java/org/apache/ctakes/dictionary/lookup/tools /CreateLuceneIndexForExampleDrugs.java and scripts/java/org/apache/ctakes/dictionary/lookup/tools /CreateLuceneIndexForSnomedLikeSample.java

To view the contents of a Lucene index, you could use a tool such as Luke.

Creating your own dictionaries

To create a dictionary yourself, you could download a copy of the UMLS Metathesaurus and build upon the program mentioned above to create a Lucene index of the desired vocabulary.

Alternatively, you could create a Lucene index from a pipe-delimited file by using a different program (scripts/java/org/apache/ctakes/dictionary /lookup/tools/CreateLuceneIndexFromDelimitedFile.java) in that same package.