JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Lexicon.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/lexicon/dumpXML.html)
* [Next Class](http://docs.google.com/simplenlg/lexicon/LexiconInterface.html)
* [Frames](http://docs.google.com/index.html?simplenlg/lexicon/Lexicon.html)
* [No Frames](http://docs.google.com/Lexicon.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.3dy6vkm)|
* [Method](#id.4d34og8)

simplenlg.lexicon

## Class Lexicon

* java.lang.Object
  + simplenlg.lexicon.Lexicon

Direct Known Subclasses: [MultipleLexicon](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html), [NIHDBLexicon](http://docs.google.com/simplenlg/lexicon/NIHDBLexicon.html), [XMLLexicon](http://docs.google.com/simplenlg/lexicon/XMLLexicon.html)

* public abstract class Lexicon  
  extends java.lang.Object
* This is the generic abstract class for a Lexicon. In simplenlg V4, a Lexicon is a collection of [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) objects; it does not do any morphological processing (as was the case in simplenlg V3). Information about WordElement can be obtained from a database ( [NIHDBLexicon](http://docs.google.com/simplenlg/lexicon/NIHDBLexicon.html)) or from an XML file ( [XMLLexicon](http://docs.google.com/simplenlg/lexicon/XMLLexicon.html)). Simplenlg V4 comes with a default (XML) lexicon, which is retrieved by the getDefaultLexicon method. There are several ways of retrieving words. If in doubt, use lookupWord. More control is available from the getXXXX methods, which allow words to retrieved in several ways
  1. baseform and [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html); for example "university" and Noun
  2. just baseform; for example, "university"
  3. ID string (if this is supported by the underlying DB or XML file); for example "E0063257" is the ID for "university" in the NIH Specialist lexicon
  4. variant; this looks for a word given a form of the word which may be inflected (eg, "universities") or a spelling variant (eg, "color" for "colour"). Acronyms are not considered to be variants (eg, "UK" and "United Kingdom" are regarded as different words).  
     *Note:* variant lookup is not guaranteed, this is a feature which hopefully will develop over time
  5. variant and [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html); for example "universities" and Noun
* For each type of lookup, there are three methods
  + getWords: get all matching [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) in the Lexicon. For example, getWords("dog") would return a List of two WordElement, one for the noun "dog" and one for the verb "dog". If there are no matching entries in the lexicon, this method returns an empty collection
  + getWord: get a single matching [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) in the Lexicon. For example, getWord("dog") would a for either the noun "dog" or the verb "dog" (unpredictable). If there are no matching entries in the lexicon, this method will create a default WordElement based on the information specified.
  + hasWord: returns true if the Lexicon contains at least one matching WordElement

Author: Albert Gatt (simplenlg v3 lexicon), Ehud Reiter (simplenlg v4 lexicon)

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**Lexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#Lexicon())() |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [**close**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#close())() close the lexicon (if necessary) if lexicon does not need to be closed, this does nothing |
| * + static [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) | * + [**getDefaultLexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getDefaultLexicon())() returns the default built-in lexicon |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**getWord**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWord(java.lang.String))(java.lang.String baseForm) get a WordElement which has the specified base form (of any category) |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**getWord**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWord(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) get a WordElement which has the specified base form and category |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**getWordByID**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordByID(java.lang.String))(java.lang.String id) get a WordElement with the specified ID |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**getWordFromVariant**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordFromVariant(java.lang.String))(java.lang.String variant) returns a WordElement which has the specified inflected form and/or spelling variant that matches the specified variant, of any category. |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**getWordFromVariant**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String variant, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns a WordElement which has the specified inflected form and/or spelling variant that matches the specified variant, of the specified category |
| * + java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWords**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWords(java.lang.String))(java.lang.String baseForm) returns all Words which have the specified base form |
| * + abstract java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWords**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWords(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns all Words which have the specified base form and category |
| * + abstract java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWordsByID**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsByID(java.lang.String))(java.lang.String id) returns a List of WordElement which have this ID. |
| * + java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWordsFromVariant**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsFromVariant(java.lang.String))(java.lang.String variant) returns Words which have an inflected form and/or spelling variant that matches the specified variant, of any category. |
| * + abstract java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWordsFromVariant**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String variant, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns Words which have an inflected form and/or spelling variant that matches the specified variant, and are in the specified category. |
| * + boolean | * + [**hasWord**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWord(java.lang.String))(java.lang.String baseForm) return true if the lexicon contains a WordElement which has the specified base form (in any category) |
| * + boolean | * + [**hasWord**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWord(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) return true if the lexicon contains a WordElement which has the specified base form and category |
| * + boolean | * + [**hasWordByID**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWordByID(java.lang.String))(java.lang.String id) return true if the lexicon contains a WordElement which the specified ID |
| * + boolean | * + [**hasWordFromVariant**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWordFromVariant(java.lang.String))(java.lang.String variant) return true if the lexicon contains a WordElement which matches the specified variant form (in any category) |
| * + boolean | * + [**hasWordFromVariant**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWordFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String variant, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) return true if the lexicon contains a WordElement which matches the specified variant form and category |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**lookupWord**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#lookupWord(java.lang.String))(java.lang.String baseForm) General word lookup method, tries base form, variant, ID (in this order) Creates new word if can't find existing word |
| * + [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * + [**lookupWord**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#lookupWord(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) General word lookup method, tries base form, variant, ID (in this order) Creates new word if can't find existing word |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### Lexicon public Lexicon()

### Method Detail

#### getDefaultLexicon public static [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) getDefaultLexicon() returns the default built-in lexiconReturns:default lexicon

#### lookupWord public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) lookupWord(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) General word lookup method, tries base form, variant, ID (in this order) Creates new word if can't find existing wordParameters:baseForm - category - Returns:word

#### lookupWord public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) lookupWord(java.lang.String baseForm) General word lookup method, tries base form, variant, ID (in this order) Creates new word if can't find existing wordParameters:baseForm - Returns:word

#### getWords public abstract java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWords(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns all Words which have the specified base form and categoryParameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs")category - - syntactic category of word (ANY for unknown) Returns:collection of all matching Words (may be empty)

#### getWord public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) getWord(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) get a WordElement which has the specified base form and categoryParameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs")category - - syntactic category of word (ANY for unknown) Returns:if Lexicon contains such a WordElement, it is returned (the first match is returned if there are several matches). If the Lexicon does not contain such a WordElement, a new WordElement is created and returned

#### hasWord public boolean hasWord(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) return true if the lexicon contains a WordElement which has the specified base form and categoryParameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs")category - - syntactic category of word (ANY for unknown) Returns:true if Lexicon contains such a WordElement

#### getWords public java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWords(java.lang.String baseForm) returns all Words which have the specified base formParameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs") Returns:collection of all matching Words (may be empty)

#### getWord public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) getWord(java.lang.String baseForm) get a WordElement which has the specified base form (of any category)Parameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs") Returns:if Lexicon contains such a WordElement, it is returned (the first match is returned if there are several matches). If the Lexicon does not contain such a WordElement, a new WordElement is created and returned

#### hasWord public boolean hasWord(java.lang.String baseForm) return true if the lexicon contains a WordElement which has the specified base form (in any category)Parameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs") Returns:true if Lexicon contains such a WordElement

#### getWordsByID public abstract java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWordsByID(java.lang.String id) returns a List of WordElement which have this ID. IDs are lexicon-dependent, and should be unique. Therefore the list should contain either zero elements (if no such word exists) or one element (if the word is found)Parameters:id - - internal lexicon ID for a word Returns:either empty list (if no word with this ID exists) or list containing the matching word

#### getWordByID public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) getWordByID(java.lang.String id) get a WordElement with the specified IDParameters:id - internal lexicon ID for a word Returns:WordElement with this ID if found; otherwise a new WordElement is created with the ID as the base form

#### hasWordByID public boolean hasWordByID(java.lang.String id) return true if the lexicon contains a WordElement which the specified IDParameters:id - - internal lexicon ID for a word Returns:true if Lexicon contains such a WordElement

* + - getWordsFromVariant  
      public abstract java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWordsFromVariant(java.lang.String variant,  
       [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category)  
      returns Words which have an inflected form and/or spelling variant that matches the specified variant, and are in the specified category.  
      *Note:* the returned word list may not be complete, it depends on how it is implemented by the underlying lexiconParameters:variant - - base form, inflected form, or spelling variant of wordcategory - - syntactic category of word (ANY for unknown) Returns:list of all matching Words (empty list if no matching WordElement found)

#### getWordFromVariant public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) getWordFromVariant(java.lang.String variant, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns a WordElement which has the specified inflected form and/or spelling variant that matches the specified variant, of the specified categoryParameters:variant - - base form, inflected form, or spelling variant of wordcategory - - syntactic category of word (ANY for unknown) Returns:a matching WordElement (if found), otherwise a new word is created using thie variant as the base form

#### hasWordFromVariant public boolean hasWordFromVariant(java.lang.String variant, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) return true if the lexicon contains a WordElement which matches the specified variant form and categoryParameters:variant - - base form, inflected form, or spelling variant of wordcategory - - syntactic category of word (ANY for unknown) Returns:true if Lexicon contains such a WordElement

* + - getWordsFromVariant  
      public java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWordsFromVariant(java.lang.String variant)  
      returns Words which have an inflected form and/or spelling variant that matches the specified variant, of any category.  
      *Note:* the returned word list may not be complete, it depends on how it is implemented by the underlying lexiconParameters:variant - - base form, inflected form, or spelling variant of word Returns:list of all matching Words (empty list if no matching WordElement found)

#### getWordFromVariant public [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) getWordFromVariant(java.lang.String variant) returns a WordElement which has the specified inflected form and/or spelling variant that matches the specified variant, of any category.Parameters:variant - - base form, inflected form, or spelling variant of word Returns:a matching WordElement (if found), otherwise a new word is created using thie variant as the base form

#### hasWordFromVariant public boolean hasWordFromVariant(java.lang.String variant) return true if the lexicon contains a WordElement which matches the specified variant form (in any category)Parameters:variant - - base form, inflected form, or spelling variant of word Returns:true if Lexicon contains such a WordElement

#### close public void close() close the lexicon (if necessary) if lexicon does not need to be closed, this does nothing

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Lexicon.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/lexicon/dumpXML.html)
* [Next Class](http://docs.google.com/simplenlg/lexicon/LexiconInterface.html)
* [Frames](http://docs.google.com/index.html?simplenlg/lexicon/Lexicon.html)
* [No Frames](http://docs.google.com/Lexicon.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.3dy6vkm)|
* [Method](#id.4d34og8)